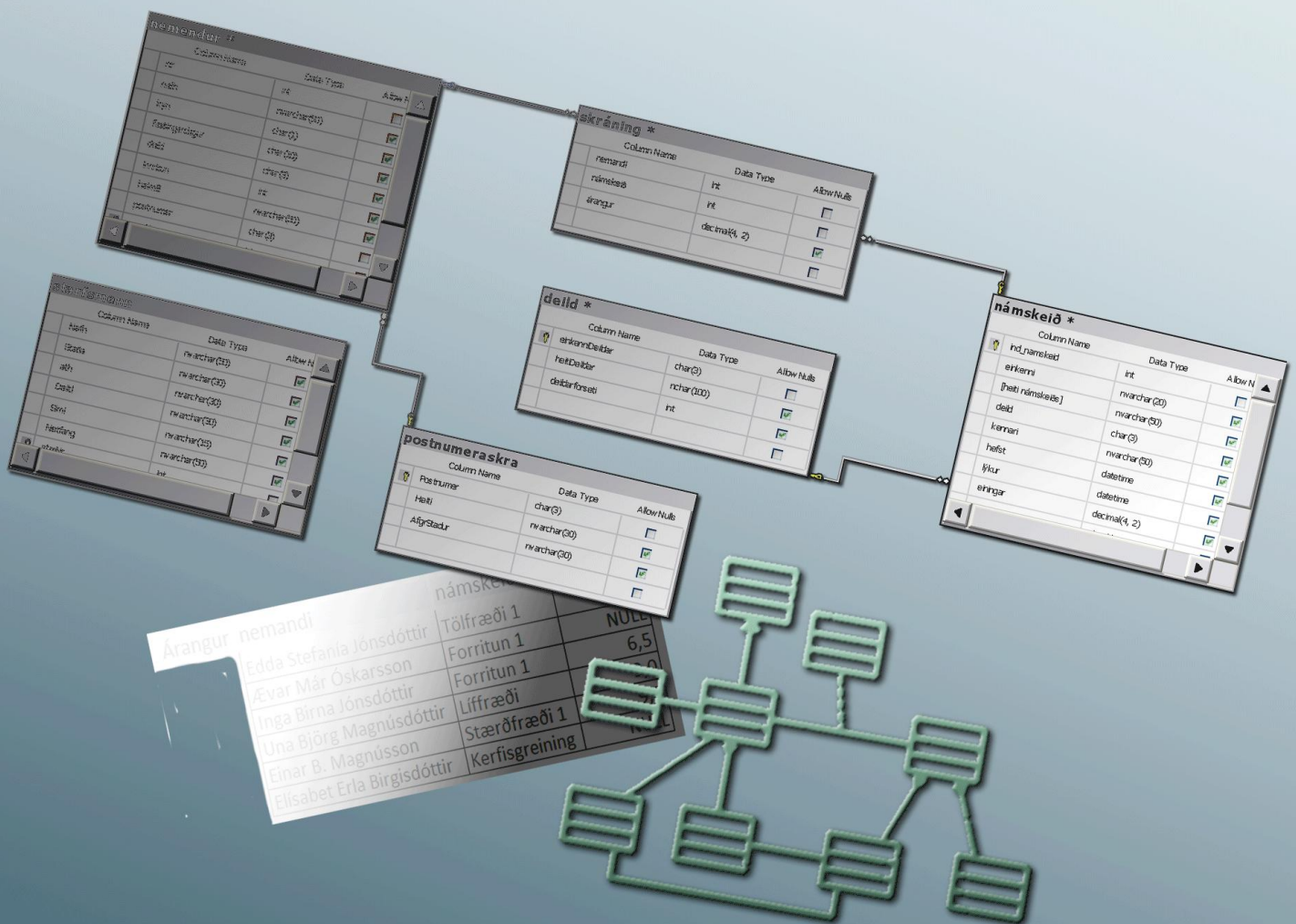


Gagnasafnarninn

Samantekt um gagnasafnsfræði



Jón Freyr Jóhannsson

Gagnasafnarinn - Samantekt um gagnsafnsfræði

ISBN 978-9979-9811-1-4

© 2008, Jón Freyr Jóhannsson

Fyrsta útgáfa - 2008

Hönnun og umbrot: Jón Freyr Jóhannsson

Rit þetta má eigi afrita með neinum hætti
sem ljósmyndun, prentun, ljósritun eða á
annan sambærilegan hátt, að hluta eða í heild,
án skriflegs leyfis höfundar

Útgefandi er Jón Freyr Jóhannsson, Hafnarfirði

ISBN 978-9979-9811-1-4



9789979981114

Efnisyfirlit

Efnisyfirlit.....	3
1 Inngangur að gagnasafnsfræði	7
1.1 Mikilvæg hugtök í kaflanum	7
1.2 Gögn á tölvutæku formi.....	7
1.3 Hvað er gagnasafnsfræði	7
1.4 Gagnasafn og gagnagrunnur.....	8
1.5 Gagnasafnskerfi	8
1.6 Kostir gagnasafnskerfa.....	8
1.7 Fyrirspurnarmál.....	9
1.8 Annað til fróðleiks og umhugsunar.....	10
1.9 Saga gagnasafnsfræðinnar	10
1.9.1 Álitamál	10
1.9.2 Lagalegar hliðar	11
1.9.3 Öryggismál.....	11
2 Töflulíkanið - <i>relational model</i>	13
2.1 Mikilvæg hugtök í kaflanum	13
2.2 Um gögn.....	13
2.3 Um töflur	16
2.4 Skilyrði venslatöflu	17
2.5 Uppbygging töflu	19
2.6 Reglur Codd's	20
2.7 Meira um hugtök	22
2.8 Annað til fróðleiks og umhugsunar.....	22
2.8.1 Skemu.....	22
2.8.2 Annar fróðleikur	23
3 Lyklar og önnur skilyrði.....	25
3.1 Mikilvæg hugtök í kaflanum	25
3.2 Skilyrði.....	25
3.3 Lyklar	26
3.3.1 Samsettur lykill	28
3.3.2 Mögulegir lyklar.....	28
3.3.3 Frumlykill	29
3.3.4 Aukalyklar	29
3.3.5 Hvað ef við finnum engan mögulegan lykil?	30
3.3.6 Aðkomulyklar	30
3.4 Skilyrði lykla	31
3.4.1 Skilyrði frumlykils.....	31

3.4.2 Aðkomulyklaskilyrði.....	31
3.5 Dálkaskilyrði.....	33
3.6 Annað til fróðleiks og umhugsunar.....	34
4 Kaflinn um gildið sem ekki er til	35
4.1 Mikilvæg hugtök í kaflanum	35
4.2 Hvað er NULL?.....	35
4.3 Nokkrar einfaldar reglur við meðhöndlun á NULL.....	37
4.3.1 Regla 1: Þekkja til NULL.....	37
4.3.2 Regla 2: Ekki bera NULL saman við einhver gildi, ákveddu hvort velja á eða hafna NULL.....	37
4.3.3 Regla 3: Skilja hvernig <i>safnaðgerðir</i> (e. <i>aggregates</i>) taka tillit til NULL	37
4.3.4 Regla 4: Taka tillit til NULL gilda í samanburði (skilyrðisbundinni leit).....	38
4.3.5 Regla 5: Þekktu hvernig reiknaðgerðir virka með NULL.....	38
4.3.6 Regla 6: GROUP BY skilar NULL færslum.....	38
4.3.7 Regla 6: Hafðu í huga þörf fyrir <i>skyldleikagreiningu</i> (e. <i>normalisation</i>)	38
4.3.8 Regla 7: Röktafla með þremur rökgildum.....	39
4.4 Annað til fróðleiks og umhugsunar.....	39
5 Vísar	41
5.1 Mikilvæg hugtök í kaflanum	41
5.2 Vísar (e. <i>index</i>).....	41
5.3 Uppbygging vísa	43
5.3.1 Raðaður/klasaður vísir.....	44
5.3.2 Óklasaður vísir.....	44
5.3.3 Tætifallsvísir	45
5.3.4 Þéttur vísir.....	46
5.3.5 Gisinn vísir.....	46
5.4 Tré (e. <i>tree</i>)	47
5.4.1 B-tré og B+ tré.....	47
5.5 Val á fyrirkomulagi vísa og taflna	48
5.5.1 Aðferð við val á vísu	48
5.6 Annað til fróðleiks og umhugsunar.....	50
6 Töflualgebra.....	51
6.1 Mikilvæg hugtök í kaflanum	51
6.2 Grunnaðgerðir töflualgebru.....	51
6.2.1 Grunnaðgerðir.....	52
6.2.2 Aðgerðir sem leiða má frá grunnaðgerðum	52
6.2.3 Virkjar og segðir	52
6.3 Færsluval (σ)	53
6.3.1 Formleg framsetning á færsluvali.....	53
6.3.2 Rökvirkjar í færsluvali.....	53
6.3.3 Dæmi um færsluval	54
6.4 Dálkaval.....	55
6.4.1 Dæmi um dálkaval	55

6.5 Mengjamargfeldi / krossfeldi.....	56
6.5.1 Mengjamargfeldi og færsluval	57
6.5.2 Dæmi um mengjamargfeldi	57
6.6 Sammengi	58
6.6.1 Dæmi um sammengi.....	58
6.6.2 Union all	60
6.6.3 Sammengishæfni	60
6.7 Mengjamismunur	61
6.7.1 Dæmi um mengjamismun	61
6.8 Sniðmengi	62
6.8.1 Dæmi um sniðmengi	63
6.9 Töflutenging.....	64
6.9.1 Vinstri ytri töflutenging	65
6.9.2 Hægri ytri töflutenging.....	66
6.9.3 Innri töflutenging.....	67
6.10 Töfludeilding.....	68
6.11 Færslur í töflum dæma	69
6.12 Yfirlit yfir helstu aðgerðir töflualgebru	72
6.13 Annað til fróðleiks og umhugsunar	74
7 Fallákveður - functional dependancies.....	75
7.1 Mikilvæg hugtök í kaflanum	75
7.2 Fallákveður og lyklar	75
7.3 Fallákveður í gagnasafnsfræði og stærðfræði	76
7.4 Táknmál fallákveða.....	76
7.5 Dæmi um fallákveður.....	77
7.6 Fallákveður á myndrænan hátt	78
7.7 Formleg skilgreining á fallákveðum.....	79
7.8 Hvernig finnum við fallákveður?.....	79
7.9 Frumreglur Armstrongs.....	80
7.10 Annað til fróðleiks og umhugsunar	80
8 Töfluhönnun	81
8.1 Mikilvæg hugtök í kaflanum	81
8.2 Fínhönnun gagnagrunna.....	81
8.3 Velformuð tafla	82
8.3.1 Dæmi um töflu í vanda.....	82
8.4 Staðalform	83
8.5 Fyrsta staðalform (1NF)	83
8.5.1 Dæmi um töflu á fyrsta staðalformi (1NF).....	84
8.6 Annað staðalform (2NF)	85
8.7 Þriðja staðalform (3NF).....	86
8.7.1 Setjum á þriðja staðalform (3NF).....	86
8.8 Samantekt um hvað þarf að gera milli staðalforma.....	86

8.9 Annað til fróðleiks og umhugsunar	87
8.9.1 Önnur staðalform.....	87
9 Hreyfingar - <i>transactions</i>	89
9.1 Mikilvæg hugtök í kaflanum	89
9.2 Hvað er hreyfing?	89
9.3 Eiginleikar hreyfinga - ACID	90
9.3.1 Eining - Atomicity.....	91
9.3.2 Heilleiki - Consistency	91
9.3.3 Einangrun - Isolation	91
9.3.4 Ending - Durability	91
9.4 Atburðaskrá - transaction log	92
9.4.1 Uppbygging venjulegrar atburðarskrár.....	92
9.5 Aðrar leiðir til útfærslu hreyfingastjórnunar.....	92
9.6 Annað til fróðleiks og umhugsunar	92
A. Stutt yfirlit yfir SQL skipanir	93
B. Hugtök íslensk-ensk.....	97
C. Hugtök ensk-íslensk.....	101
D. Atriðisorðasafn - Index	105
E. Heimildir.....	107

1 Inngangur að gagnasafnsfræði

1.1 Mikilvæg hugtök í kaflanum

bestun fyrirspurna (e. *query optimisation*)
einsmóta aðgangsstýringar og stjórnun (e. *uniform data administration*)
endurheimt við kerfishrun (e. *recovery from crashes*)
fyrirspurnarmál (e. *query language*)
gagnagrunnur (sjá gagnasafn og gagnasafnskerfi)
gagnaheilindi (e. *data integrity*)
gagnahuld (e. *data independence*)
gagnasafn (e. *database*)
gagnasafnskerfi (e. *database management system*)
minni þróunarkostnaður
rekstraraðili gagnasafnskerfis (e. *DBA, database administrator*)
samtíða aðgangur/vinnslur (e. *concurrent access*)
skilvirkur aðgangur (e. *efficient access*)
SQL (e. *structured query language*)
töflugagnagrunnur (e. *relational database management system*)
töflugagnasafn (e. *relational database*)

1.2 Gögn á tölvutæku formi

Öll þekkjum við hversu þægilegra er að geyma margskonar gögn á tölvutæku formi heldur en skrifuð á bréfnifsi eða falið inni í bókum sem ekki eru við höndina. Það eru hins vegar ekki öll gögn þannig að þau passi vel til nota á tölvutækan hátt. Margt af gögnum sem við viljum geyma henta hins vegar vel þessu formi og má þar nefna mataruppskriftir (eitthvað sem allir þekkja), símanúmer, netföng og heimilisföng, ljósmyndir og svo mætti lengi telja.

Þessi gögn er síðan misauðvelt að vinna með á tölvutæku formi, margar mataruppskriftir eru skannaðar (á myndformi) og erfitt að breyta þeim og vonlaus að leita (nema með sérstökum flóknari aðferðum sem skrá og greina myndir). Það er t.d. ekki auðvelt að finna uppskriftir sem hafa bæði sýrðan rjóma og avocado öðru vísi en að vita sjálfur fyrirfram hvar þessar uppskriftir eru.

1.3 Hvað er gagnasafnsfræði

Gagnasafnsfræði fjallar um skilgreiningu og skipulagningu gagna á markvissan hátt og þær fræðilegu forsendur sem að baki liggja.

Það er mikill munur á því að nota vel skipulögð gagnasöfn og kerfi og síðan á öðrum aðferðum sem helst hafa verið notaðar við geymslu og ekki síst við lestur, uppflettingu og endurheimt gagna.

1.4 Gagnasafn og gagnagrunnur

Gagnagrunnur (e. *database*) er skipulagt **kerfi** til að geyma tölvuskráð gögn um tiltekið efni, skipa gögnunum niður eftir innbyrðis venslum, vinna úr þeim og sækja þau.

Gagnagrunnur er í fræðilegum skilningi skilgreining á aðferðum þar sem umbúnaður og meðferð gagna skiptir meginmáli en ekki gögnin sjálf. Gagnagrunnur getur jafnvel verið tómur af gögnum.

Margir nota orðið gagnagrunnur í sömu merkingu og gagnasafn en í innan gagnasafnsfræðinnar er greint á milli merkingar þeirra.

Gagnasafn (einni kallað e. *database*) er samsafn **gagna um tiltekið efni**, skipulagt í samræmi við tiltekna hugtakaskipan þar sem lýst er einkennum gagnanna og venslum milli samsvarandi eininda þeirra. Orðið gagnasafn getur einnig verið notað yfir kerfi sem ekki geyma gögn á tölvutæku formi s.s. bókasafn eða spjaldskrá.

Í daglegu tali, þ.e. utan fræðiheims gagnasafnsfræðinnar, hefur orðið gagnagrunnur fengið mun almennari notkun. Sú notkun tekur til alls konar safna og einnig gagnasöfnunar og miðlunar, jafnvel hvort sem um tölvutæk gagnasöfn er að ræða eða ekki. Hér skulum við hins vegar nota fræðilegu skilgreininguna.

1.5 Gagnasafnskerfi

Til að flækja málin enn frekar er notað hugtakið **gagnagrunnskerfi** til að tákna það umhverfi, rekstrar og stjórnunarlega, sem gerir kleift að skipuleggja **gagnagrunna** og nýta **gagnasöfnin**.

Enska heitið *database*, skammstafað **db**, er notað yfir bæði gagnasafn og gagnagrunn, en enska heitið *database management system*, skammstafað **DBMS**, er notað yfir gagnasafnskerfi. Frá þessari málnotkun víkja margir, jafnvel innan tölvunarfræðinnar.

Gagnasafnskerfi er hugbúnaður, mjög flókinn í flestu tilfellum, sem er hannaður til að stýra gagnagrunni og geyma gagnasöfn. Í því felst m.a. að stjórna fyrirkomulagi mikils magns af gögnum, stýra aðgangi, koma í veg fyrir að árekstrar verði í notkun, t.d. þegar fleiri en einn vilja breyta eða eyða tilteknum færslum, stýra afritun, skráningu atburðaskrár, log, og svo mætti lengi telja.

Gagnasafnskerfi eru til í nokkrum meginflokkum en sá algengasti eru svokallaðar **töflu-gagnagrunnar** (e. *relational database management system*, **RDBMS**). Aðrar gerðir gagnagrunnskerfa eru ekki viðfangsefni hér¹.

1.6 Kostir gagnasafnskerfa

Gagnasafnskerfi hafa marga kosti umfram það fyrirkomulag að hafa gögn í stökum skrár eða skráarsöfnum sem ekki tilheyra gagnasafnskerfi. Meðal þeirra má nefna:

- **Gagnahuld** (e. *data independence*)². Hönnunarlega séð er talin kostur að fela fyrir öðrum hlutum kerfa hvernig tiltekin atriði eru útfærð eða geymd. Annars er hættu á því að forritun annarra hluta kerfis taki of mikið mið af því og geri því allar breytingar (til hagræðis) erfiðari og hættari við villum.

¹ Sjá má tilvísanir í aðrar gerðir gagnagrunnskerfa t.d. í: Database Management System. 9.12.2007. en.wikipedia.org/wiki/Database_management_system.

² Sjá m.a. Data independence. 9.12.2007. http://en.wikipedia.org/wiki/Data_independence,

- **Skilvirkur aðgangur** (*e. efficient access*). Með skilvirkum aðgangi er átt við að margir aðilar geti á öruggan og hraðvirkan hátt haft aðgengi að gögnum óháð notkun annarra á sömu gögnum.
- **Minni þróunarkostnaður**. Gagnagrunnskerfi hafa innbyggðar öflugar aðferðir við utanumhald gagna, skráningu þeirra, lestur og uppfærslu, sem annars þyrfti að útfæra með ærnum tilkostnaði.
- **Gagnaheilindi** (*e. data integrity*)³. Tryggja þarf að gögn séu rétt og samhengi þeirra tryggt. Með því er meðal annars átt við að gögn séu á því formi sem skilgreiningar þeirra segja til um. Einnig er átt við að tilvísanir milli gagna haldist réttar þó breytingar verði á einum stað. Oft er talað um það að gagnasafnið fari úr einni réttri stöðu í aðra, ekki sé skilið við hálfkláraðar uppfærslur eða skráningar.
- **Öryggi**. Gagnasafnskerfi hafa flest mjög öflugan öryggisþátt innbyggðan og minnka því þörf fyrir sérstök öryggiskerfi.
- **Einsmóta aðgangsstýringar og stjórnun** (*e. uniform data administration*). Á einum stað er stýrt aðgangi að gögnum, ekki er þörf á því að gera það á mörgum stöðum með misskilvirkum hætti, t.d. í sérstökum aðgangsstýringum, netaðgangsstýringum o.s.frv.
- **Samtíða aðgangur/vinnslur** (*e. concurrent access*)⁴. Tryggt er að margir geti unnið í gagnasafninu í einu án þess að hætta sé að árekstrum eða að röngum gögnum sé veitt til notenda þó margir vinni á sömu gögnum á sama tíma. Einnig er stefnt að því að ekki sé hætta á því að mismunandi útgáfur gagna séu í notkun.
- **Endurheimt við kerfishrun** (*e. recovery from crashes*). **Atburðarskráning** (*e. log*) tryggir að hægt sé að byggja upp gagnasafn að nýju við kerfishrun, einnig er tryggt að annað hvort er færslu lokið fullkomlega eða þá að hún er alls ekki framkvæmd og þannig komið í veg fyrir að gagnaheilindi tapist.
- **Bestun fyrirspurna** (*e. query optimisation*)⁵. Flest gagnagrunnskerfi hafa innbyggð leið til að gera fyrirspurnir sem ekki eru sérlega vel uppbyggðar skilvirkar með því að beita á þær reglum töflualgebru og einfalda fyrirspurn og tryggja að hún sé ávallt sem best miðað við uppbyggingu gagna.

Flest þessi hugtök verða skýrð beint eða óbeint í öðrum köflum.

1.7 Fyrirspurnarmál

Til að vinna með gögn í gagnasafni eru notuð fyrirspurnarmál og er **SQL** (*e. structured query language*) þeirra algengast í dag. Fyrirspurnarmál eru eins og forritunarmál í þeim skilningi að þau hafa formlega skilgreiningu og málfræði, eins og forritunarmál gera almennt. Uppbygging þeirra, þ.e. hvernig skipanir eru settar fram eru þó sjaldnast líkar hefðbundnum forritunarmálum (sjá nánar um það þegar fjallað er um stefjur og föll).

Sérstaklega verður fjallað um nokkrar af fyrirspurnum og skipunum SQL í sérstökum köflum. Athuga ber að þó staðlar séu til um SQL⁶, fylgja framleiðendur gagnagrunnskerfa þeim misvel. Sú umfjöllun sem í þessu riti er miðar við útfærslu í MS SQL Server 2005⁷ nema annað sé tekið fram.

³ Sjá m.a. Data integrity, 9.12.2007. http://en.wikipedia.org/wiki/Data_integrity

⁴ Sjá m.a. Concurrency control, 9.12.2007. http://en.wikipedia.org/wiki/Concurrent_access

⁵ Sjá m.a. Query optimization, 9.12.2007. http://en.wikipedia.org/wiki/Query_optimisation

⁶ Ágætis einfölduð samantekt á JCC's SQL Standards Page, 19.12.2007. <http://www.jcc.com/sql.htm>

⁷ MS SQL Server 2005 Home, 12.12. 2007. <http://www.microsoft.com/sql/default.msp>

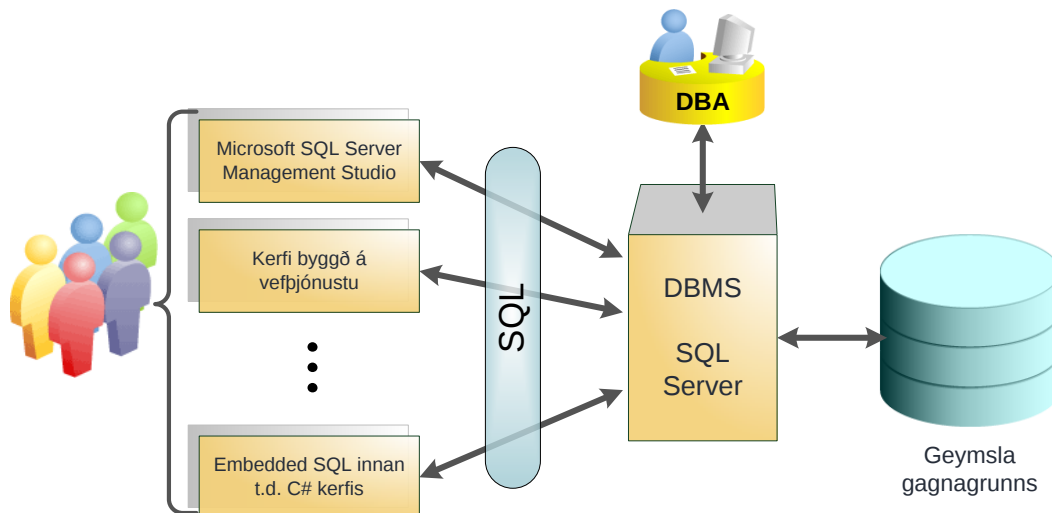
1.8 Annað til fróðleiks og umhugsunar

Notkun gagnasafnskerfa, gagnagrunna og gagnasafna er margþætt og þeir sem að þeirri vinnu koma hafa ólík sjónarhorn og ólík verkfæri til að vinna með.

Almennir notandi gagnanna, sem gagnasafnið hefur notað, hefur oftast en ekki mjög óbeinan aðgang að gögnunum, þ.e. hann hefur aðgang að gögnunum í gegnum forrit sem fela gagnagrunnsaðganginn í sér á þann hátt að notandi verður ekkert endilega var þess hvað að baki býr. Sumir notendur geta nýtt sér beint SQL fyrirspurnir til að vinna gögnin og gera það þá ýmist gegnum gagnasafnskerfið (t.d. MS SQL Server) eða jafnvel gegnum Excel sem býður nú alveg ágætis aðgang að því að sækja gögn í gagnasöfn.

Kerfisgerðarfólk er flest að vinna í umhverfi gagnasafnsins (s.s. Microsoft SQL Management Studio) eða öðrum verkfærum sem eiga auðvelt með að hafa samskipti við gagnagrunna (s.s. forritunarumhverfi MS Studio .NET). Þessir aðilar eru undantekningarlítið að nýta sér SQL fyrirspurnarmálið.

Rekstraraðilar gagnasafnskerfisins (e. *DBA, database administrator*) eru þeir sem vinna í gagnasafnskerfinu og auk þess að vinna þar með SQL hafa aðgang að ýmsu öðru svo sem stillingu öryggisatriða og öðru er varðar að tryggja góðan og skilvirkan rekstur gagnagrunnsins.



1.9 Saga gagnasafnsfræðinnar

Ekki er ætlunin að gera hér í þessum kafla grein fyrir sögu gagnasafnsfræðinnar þó áhugaverð sé en bendi hér að staði til að byrja á.

Benda má á samantekt á sögu gagnasafnsfræðinnar t.d. á Wikipedia⁸

Sjá einnig „The Rise of Relational Databases“⁹

1.9.1 Álitamál

Ýmis álitamál má hafa í huga við notkun gagnagrunna og má þar nefna höfundarrétti, persónuvernd og aðrar lagalegar hliðar. Einnig þarf að huga vel að öryggismálum. Þessu riti er ekki ætlað að taka sérstaklega á þessum þáttum en benda má á nokkrar heimildir sem áhugavert gæti verið að skoða

⁸ Database History. 9.12.2007. en.wikipedia.org/wiki/Database#History

⁹ The Rise of Relational Databases. 9.12.2007. nap.edu/readingroom/books/far/ch6.html

1.9.2 Lagalegar hliðar

Linn, Ann. 2000. *History of Database Protection, Legal Issues of Concern to the Scientific Community*¹⁰.

Persónuvernd¹¹ fjallar um ýmislegt er varðar söfnun upplýsinga um einstaklinga.

1.9.3 Öryggismál

Nokkrar greinar er að finna á svæðinu „MS SQL Server Security“¹²

Einnig er greinar að finna um öryggismál á svæði Bitpipe.com¹³

Að auki er auðvitað sjálfsagt að leita með Google.

¹⁰ Linn, Ann. 2000. *History of Database Protection, Legal Issues of Concern to the Scientific Community*. 12.12.2007. globalchange.gov/policies/linn.html

¹¹ Persónuvernd. 12.12.2007. <http://www.personuvernd.is/>

¹² MS SQL Server Security. 9.12.2007. database.ittoolbox.com/topics/t.asp?t=371&p=452&h1=371&h2=452

¹³ Database Security. 9.12.2007. bitpipe.com/rlist/term/Database-Security.html

2 Töflulíkanið

Þegar við tölum um **töflu** skv. skilningi gagnasafnsfræðinnar þá erum við með **sérstaka skilgreiningu** í huga. Tafla þarf þannig að uppfylla ýmis skilyrði til að teljast **tafla** í gagnasafnsfræði. Orðið **töflulíkan** (*e. relational model*) notum við yfir skilgreiningar og fræði sem liggja að baki **töflugagnasafni**.

2.1 Mikilvæg hugtök í kaflanum

dálkur eða **eigindi** (*e. attribute, column*)
færsla (*e. row, record, tuple*)
færslufjöldi töflu (*e. cardinality*)
gagnategund (sjá tag)
gildi (*e. value*)
gráða töflu (*e. degree, arity*)
grunnskilgreiningarmengi (*e. type*) skilgreint á mismundandi hátt eftir umhverfi
lýsigögn (*e. metadata*), gögn um gögn
raunlægt skema (*e. physical schema*)
reglur Codd's
röklegt skema (*e. conceptual/logical schema*)
skilgreiningarmengi (*e. domain, field specification*)
skilyrði venslatöflu (*e. definition of relation*)
sýn (*e. view*)
tafla (*e. relation*)
tag (*e. type*), sjá einnig skilgreiningarmengi
töflulíkan (*e. relational model*)
töfluskema (*e. relation schema, table schema*)
vensl (sjá tafla)
venslatafla (sjá tafla)
vísir (*e. index*)

2.2 Um gögn

Gögn er hægt að geyma á mörgu formi og það er hægt að túlka gögn á margvíslegan hátt. Segjum sem svo að við höfum **gildið** '8970517' í gagnasafninu okkar. Hvað táknar þetta gildi? Getur það verið símanúmer, vörunúmer, upphæð, mannfjöldi í tiltekinni borg eða eitthvað allt annað?

Það er talað um það að **gögn** verði að **upplýsingum** þegar við höfum upplýsingar um það hvernig túlka eigi gögnin. Ef gildið '8970517' er tilgreint sem vörunúmer er búið að eyða (mestu) af mögulegri mistúlkun á gildinu. Það að það vilji svo til að þetta gildi sé með sömu töluröð og símanúmer tiltekins einstaklings er bara skemmtileg tilviljun.

Ef við skilgreinum tiltekin gögn sem (íslensk) símanúmer þá höfum við takmarkað hvaða gildi eru möguleg. Gildin eru, með fáum undantekningum, sjö tölustafir. Númerið 87654321 er ekki gilt sem símanúmer, það er átta stafa og er því utan **skilgreiningarmengis** (*e. domain, field specification*). Við tölum um að gildi séu innan skilgreiningarmengis ef þau eru hluti af leyfilegum gildum og af réttu **tagi** (*e. type*), réttri gagnategund. Með gagnategund er t.d. átt við að stafir (texti) eru af öðru tagi en tölur (tölustafir), þá er einnig gerður greinarmunur á heiltölum og svo rauntölum.

Dæmi um takmarkað skilgreiningarmengi er það sem kallað er kynkóði í þjóðskrá¹⁴ þar er um að ræða mjög fá leyfileg gildi þó þau séu nú fleiri en flesta grunar. Kynkóðar eru nefnilega fjórir!

Kyn	Merking
1	Karlmaður 18 ára og eldri
2	Kona 18 ára og eldri
3	Drengur 17 ára og yngri
4	Stúlka 17 ára og yngri

Öll önnur gildi en þessi eru óleyfileg, þau eru utan skilgreiningarmengis.

Flest gagnasöfn (þetta á einnig við um forritunarmál) hafa skilgreind svokölluð **grunn-skilgreiningarmengi** (**tag**, *e. type*) sem oftast eru að minnsta kosti tölur, texti og dagsetningar. Hægt er að skilgreina þau nánar t.d. með því að tilgreina hvort tölur eru heiltölur, tugabrot eða aðrar tölutegundir. Einnig er oftast hægt að takmarka hvað texti getur verið langur og á hvaða formi dagsetning er. Þessar skilgreiningar geta verið mismunandi eftir því hvaða gagnasafnskerfi er um að ræða. Til viðbótar þessum grunnskilgreiningum getum við skilgreint okkar eigin tög t.d. símanúmer, kynkóða o.s.frv.

Þegar verið er að lýsa gögnum er oft talað um **lýsigögn** (*e. metadata*) sem táknar þá **gögn um gögn**. Flestir nota enska orðið.

¹⁴ Táknað þjóðskrár 3.12.2007. <http://www.thjodskra.is/taknmal/>

Tökum sem dæmi lýsigögn um töflu sem tilgreinir nemendur og upplýsingar um þá. Hér er fyrst töfluskemað yfir töfluna sem heitir Nemendur:

Nemendur										
nemandanúmer	nafn nemanda	kyn	fæðingardagur	deild	innritun	heimili	póstnúmer	einingum lokið	meðaleinkunn	í námi

Skoðum nú hvernig lýsigögn gætu verið skilgreind.

Tafla	Dálkur	Tag	Stærð	Uppruni	Skilgreiningarmengi
Nemendur	nemandanúmer	tala	6	skráð	einkvæm heiltala milli 1 og 999999
Nemendur	kyn	tala	1	skráð	kóði skv. þjóðskrá {1,2,3,4}
Nemendur	fæðingardagur	texti	8	skráð	texti með tölum eingöngu á forminu ddmmaaáá
Nemendur	deild	texti	3	skráð	2-3 stafa kóði, verður að vera að finna í töflunni deild. Dæmi 'TD', 'TVD', 'KLD'
Nemendur	innritun	texti	3	skráð	3 stafa kóði þar sem fyrsti stafur táknar á hvaða önn innritun er {h, v, s} og svo tveir öftustu tölustafir í ártali. Dæmi 'S08', 'H09'
Nemendur	heimili	texti	50	skráð	heimilisfang, getur verið annað en í þjóðskrá, á að tilgreina núverandi búsetu
Nemendur	póstnúmer	tala	3	skráð	Þriggja stafa póstnúmer sem verður að vera að finna í póstnúmeraskrá
Nemendur	einingum lokið	tala	3	reiknað	Reiknað gildi frá töflunni Árangur fyrir þennan tiltekna nemanda. Möguleg gildi eru frá 0 til 999
Nemendur	meðaleinkunn	tala	2,2	reiknað	Reiknað gildi frá töflunni Árangur fyrir þennan tiltekna nemanda. Möguleg gildi eru frá 0,00 til 10,00. Tala skal vera reiknuð með 2 aukastöfum
Nemendur	í námi	texti		skráð	Möguleg gildi eru; 'Í námi', 'Í námsleyfi', 'Hættur', 'Brautskráður'

Athugið að í þessari töflu er bara talað um *almenn grunntög*, texta og tölur, en það er síðan okkar að velja nákvæmari skilgreiningu sem hentar þessum lýsigögnum og sem gagnasafnskerfið býður upp á.

Í flestum gagnasafnskerfum höfum við leið til að skilgreina gagnaskilgreiningarmengin sem hér eru tiltekin og það er gert á mismunandi hátt eftir eðli hvers og eins. Sum reyna á það að fletta upp leyfilegum gildum í öðrum töflum, önnur getum við tilgreint beint í skilgreiningu viðkomandi töflu.

2.3 Um töflur

Tafla (e. *relation*) er einnig kölluð *vensl* eða *venslatafla*. Enska heitið *relation* er ekki eins lýsandi fyrir okkur, *tafla* vísar mun betur til þess sem við þekkjum. Við þurfum samt að gæta að því að hér erum við að tala um sérstaka skilgreiningu á töflu.

Tafla er tvívíð, þ.e. í henni eru línur (færslur) og dálkar:

		dálkar		
færslur	abcde	1245788	****	1212
	dseret	125	+++	1212
	gertrud	77477	##**	1312
	jk	9552	&	1212
	trest	74	%%	1112

Mynd 2-1 Uppbygging töflu

Gildin í hverri færslu hafa eitthvað sambengi, þau eru *túlkuð í einhverju eigin sambengi* og þurfa ekki að vera túlkuð í neinu sérstöku sambengi við aðrar færslur. Dæmi má taka að færsla í gengisskráningu tiltekur hvert gengi er á tiltekinni mynt á tilteknum tíma, sú færsla er alveg óháð gengi annarra mynta á þessum tíma eða öðrum. Annað dæmi gæti verið um skráningu árangurs nemanda á tilteknu prófi, sú færsla er án sambengis við aðrar, en hefur það sambengi að tengjast tilteknum nemanda og tilteknu prófi eða námsmati.

Fyrir hvern og einn dálk skilgreint hvers konar gildi er þar að finna, öll gildi í tilteknum dálki eru af einu tagi, tilheyra einu skilgreiningarmengi.

Dæmi um þetta er í dálki sem skilgreindur er sem einkunn (nemanda) er aðeins að finna gildi af tilgreindu tagi. Ef *skilgreiningarmengi* einkunna er {A,B,C,D,E,F,X} þá er einungis að finna slík gildi, þar eiga að sjálfsögðu ekki heima óskyldir hlutir eins og símanúmer eða heimilisföng.

Þegar við tölum venjulega um töflu, t.d. í Excel, þá gæti það verið eitthvað líkt þessu:

Gengtemp.xls [Compatibility Mode] - Microsoft Excel

Home

Insert

Page Layout

Formulas

Data

Review

View

Add-Ins

N24

1

A

B

C

D

E

F

G

H

I

J

K

			Mynt	Kaup	Sala				Mynt	Kaupgengi	Sölugengi
2		Austurrískur schillingur	ATS	5.65540	5.70040		Bandaríkiadóllar	USD	63.45	63.77	
3		Belg. franki	BEF	1.92830	1.94070		Sterlingspund	GBP	117.79	118.39	
4		Kanadískur dóllari	CAD	47.65000	47.95000		Kanadískur dóllari	CAD	50.60	50.90	
5		Sv. franki	CHF	47.18000	47.44000		Dönsk króna	DKK	10.88	10.94	
6		Þýskt mark	DEM	39.81830	40.05030		Norsk króna	NOK	9.64	9.70	
7		Dönsk kr.	DKK	10.43900	10.49900		Sænsk króna	SEK	8.90	8.96	
8		Spánskur peseti	ESP	0.46870	0.47270		Svissneskur franki	CHF	51.88	52.20	
9		Finnskt mark	FIM	13.07800	13.15600		Japanskt jen	JPY	0.60	0.60	
10		Franskur franki	FRF	11.86200	11.93200		SDR	XDR	95.25	95.83	
11		Sterlingspund	GBP	116.57000	117.19000		Euro	EUR	80.93	81.41	
12		Írskt pund	IEP	100.06700	100.77700						
13		Ítölsk líra	ITL	0.04034	0.04068						
14		Japanskt jen	JPY	0.50807	0.51242						
15		Hollenskt gyllini	NLG	35.30750	35.57750						
16		Norsk kr.	NOK	9.42600	9.48000						
17		Portúgalskur escudo	PTE	0.38870	0.39210						
18		Sænsk kr.	SEK	8.97300	9.02700						
19		Bandaríkiadóllar	USD	70.86000	71.24000						
20		SDR	XDR	93.84170	94.50770						
21		ECU	XEU	78.60980	79.20380						

Hér eru tilgreindir helstu gjaldmiðlar sem notaðir eru í gengisskráningu

Til vinstri eru gjaldmiðlar sem ekki eru lengur skráðir en hægra megin er sú tafla sem notuð er í dag

1

2

Mynd 2-2 Excel-tafla með mismunandi gögnum í hverjum dálki

	mynt	kaup	sala	dagur
1	USD	63.45000	63.77000	20050208
2	GBP	117.79000	118.39000	20050208
3	CAD	50.60000	50.90000	20050208
4	DKK	10.88000	10.94000	20050208
5	NOK	9.64000	9.70000	20050208
6	SEK	8.90000	8.96000	20050208
7	CHF	51.88000	52.20000	20050208
8	JPY	0.60000	0.60000	20050208
9	XDR	95.25000	95.83000	20050208
10	EUR	80.93000	81.41000	20050208

Mynd 2-3 Excel-tafla sem gæti verið venslatafla

Excel-taflan hér við hliðina er hins vegar mun *líklegri* til að standast skilgreiningu á töflu, hver dálkur hefur bara eina gerð gagna (af sama tagi) ②, ef undan er skilið heiti dálks efst ①. Vandinn er sá að Excel gerir okkur ekki mögulegt að skilyrða innslátt eða skráningu þannig að það standist skilgreiningu á töflu (*venslatöflu*).

Þegar við tölum um töflur héðan í frá verður það í merkingunni venslatafla, þ.e. tafla sem stenst skilgreiningar venslatöflu.

2.4 Skilyrði venslatöflu

Til að tafla geti verið venslatafla þarf hún að uppfylla eftirfarandi skilyrði

- **Hefur einkvæmt nafn**, engin önnur tafla heitir sama nafni. Tafla sem er í öðru gagnasafni getur heitið sama nafni en nafn gagnasafns er notað sem forskeyti og gerir töflunöfnin þannig einkvæm. Þetta er svipað og með breytuheiti þegar nöfn klasa eða pakka koma fyrir framan breytuheitið og gera nöfnin því einkvæm þó breyta með sama heiti sé til á öðrum stað
- **Dálkar innihalda gildi af einu tilteknu tagi, einu skilgreiningarmengi**. Öll gildi verða að vera af sama tagi og vera túlkuð á sama hátt. Ef dálkur er skilgreindur sem einkenni myntar (í gengisskráningu) skal þar bara vera að finna slík einkenni, ekki skal þá hafa þar einkenni skuldabréfa eða vísitölur því þá er verið að túlka innihaldið á ólíkan hátt.
- **Reitir í töflu hafa einfalt gildi**. Þetta þýðir það að ef skilgreindur er dálkurinn símanúmer þá er þar að finna í mesta lagi eitt símanúmer, dálkurinn sölugengi inniheldur aðeins eina tölu. Athugið að þó dálkurinn nafn hafi einn reit þar sem kemur fyrir „Jón Freyr Jóhannsson“ þá er þar um eitt gildi að ræða, þetta er einn strengur og þó við túlkum hugsanlega bil þannig að það greini milli gilda þá er

Þetta einn strengur og brýtur þess vegna ekki þetta skilyrði. Það sama á við ef við notum einhver samsett tög, þau eru skilgreind sem eitt gildi.

- **Hver og einn dálkur hefur einkvæmt nafn.** Þetta er samskonar skilyrði og við setjum eigindum í klösum, við verðum að geta vísað til dálks án hættu á misskilningi. Gagnasafnskerfin sem við notum leyfa okkur ekki að breyta út frá þessu svo við erum í góðum málum
- **Röð dálka skiptir ekki máli.** Við getum bætt inn nýjum dálkum og felld burtu og það á ekki að hafa nein áhrif. Það er því ekki um það að ræða að vísa til númera eða bókstafa eins og við gerum í Excel.
- **Röð færslna skiptir ekki máli.** Það er gagnasafnkerfið sem sér um að geyma færslurnar, það kemur okkur ekki við. Við getum síðan ákveðið í fyrirspurnum okkar hvernig við viljum að **úttakið** raðist
- **Engar tvær færslur eru eins.** Færslurnar eru sem sagt einkvæmar. Oftast skilgreinum við dálk eða dálka sem mynda **einkvæman lykil** sem tryggir þetta skilyrði, en það á ekki að vera nauðsynlegt að skilgreina slíka lykila til að vera með einkvæmar færslur. Hins vegar gera mörg gagnagrunnskerfi ráð fyrir því að innihald færslna geti verið eins þar sem færslueinkenni bætist við og gerir færsluna einkvæma, en það er þá í raun í samræmi við skilyrðið en við höfum ekki fulla stjórn á þessari viðbót.

Dæmi um töflu sem uppfyllir þessi skilyrði:

Nemendur	nafn	kennitala	heimilisfang	deild	innritun	sími	netfang
	Edda Stefania Jónsdóttir	2906925539	Mávahrauni 111	LD	H12	6976666	edda@haskoli.net
	Ævar Már Óskarsson	1907835579	Merkurgötu 222b	VD	H05	8977766	aevarmar@yahoo.com
	Inga Birna Jónsdóttir	1908816659	Löngumýri 142	VD	H07	8943434	ingabirna@skoli.org
	Una Björg Magnúsdóttir	2301908899	Mávahrauni 111	LD	H10	8989977	unab@simnet.is
	Einar B. Magnússon	1601862229	Pinghólsbraut 160	TD	H07		einarb@simnet.is
	Elisabet Erla Birgisdóttir	0502015559	Löngumýri 142	TD	H21		elisabet@haskoli.net

Mynd 2-4 Dæmi um töflu sem uppfyllir skilyrði venslatöflu

Dæmi um töflu sem **ekki** uppfyllir þessi skilyrði:

Nemendur	nafn	kennitala	heimilisfang	deild	innritun	sími	netfang	sími
	Edda Stefania Jónsdóttir	2906925539	Mávahrauni 111	LD	H12	6976666	edda@haskoli.net	
	Ivar Meow	Útlendingur	Merkurgötu 222b	VD	Hættur	8977766	aevarmar@yahoo.com	+1 555 1613
	Inga Birna Jónsdóttir	1908816659	Löngumýri 142	VD	H07	8943434	ingabirna@skoli.org	
				TVD		4811188		
	Una Björg Magnúsdóttir	2301908899	Mávahrauni 111	LD	H10	8989977	unab@simnet.is og unabjorg@haskoli.net	
	Einar B. Magnússon	1601862229	Pinghólsbraut 160	TD	H07		einarb@simnet.is	
	Elisabet Erla Birgisdóttir	0502015559	Löngumýri 142	TD	H21		elisabet@haskoli.net	

Mynd 2-5 Dæmi um töflu sem uppfyllir ekki skilyrði venslatöflu

Það sem ekki er uppfyllt af skilyrðum í töflunni hér að framan er:

- Tveir dálkar hafa sama nafn: sími
- Önnur færslan hefur gildi sem er (hugsanlega) utan skilgreiningarmengis í dálknum innritun
- Fjórða færslan virðist vera háð röð annarra (ef hún á að tilheyra Ingu Birnu)
- Fimmta færslan hefur tvö gildi í dálknum netfang

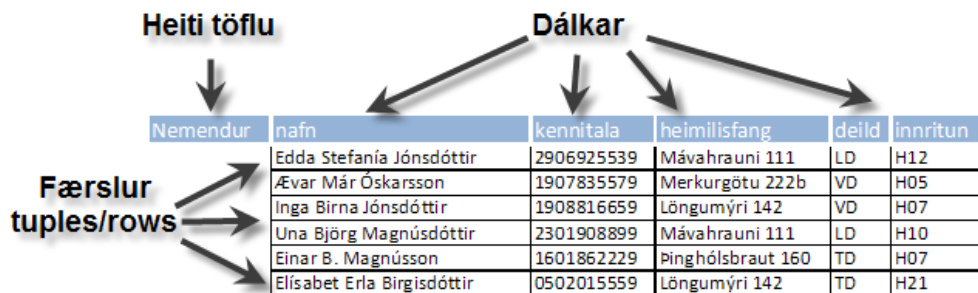
Til að sýna mun á skilyrðum sem venslatöfluskilgreining segir til um og Excel þá er hér helsti munur:

	Tafla		Excel	
Línur	Færslur	Verður að vera einkvæm	Línur	þarf ekki að vera einkvæm
Dálkar	Dálkar	Aðeins eitt tag. Aðeins eitt gildi	Dálkar	Innihaldið getur verið af ýmsu tagi. Samhengi eða lyklun ekki bundin dálkum

Mynd 2-6 Munur á innihaldi töflu og Excel-skjali

2.5 Uppbygging töflu

Uppbygging töflu er á þann hátt að hún hefur nafn og síðan tilgreinda dálka, sem uppfylla skilyrði um venslatöflur. Færslurnar eða línurnar hafa síðan innihaldið, þ.e. þar eru gögnin. Flest gagnasafnskerfi geyma í sitthvoru lagi skilgreiningu á töflu og dálkum og svo gögnin, innihaldið eða færslurnar.



Mynd 2-7 Uppbygging töflu

Til að lýsa innihaldi töflu þ.e.a.s. tilgreina hvaða dálkar eru til staðar og skilgreining þeirra, þá er talað um að setja fram **töfluskema** (*e. relation schema, table schema*). Töfluskema er sett fram þannig að tilgreint er hvað taflan heitir, síðan eru dálkarnir taldir upp og yfirleitt er einnig tilgreint tag og/eða skilgreiningarmengi hvers dálks.

Ein aðferð sem notuð er við að setja fram töfluskema er að gera það á textaformi, t.d.:

```
Nemendur(nafn, heimilisfang, deild, innritun, sími, netfang)
```

Með tagskilgreiningum (skilgreiningarmengi) væri þetta t.d.:

```
Nemandi(nafn: char(50),
        heimilisfang: char(50),
        deild: char(3) [mengi gildra deildarkóða er að finna í ...],
        innritun: char(3) skilgreint sem önn og ár [[svh][00-99]],
        sími: char(15),
        netfang: char(50))
```

Töfluskema er oft sett fram á eftirfarandi hátt, þar sem í fyrsta reitnum er nafn töflunnar og síðan koma dálkarnir (með eða án tags eftir því sem þurfa þykir):

Nemendur	nafn	heimilisfang	deild	innritun	sími	netfang
----------	------	--------------	-------	----------	------	---------

2.6 Reglur Codds

Edgar Codd hefur verið kallaður faðir gagnasafnsfræðinnar, þ.e. á sviði töflugagnagrunna. Hann setti fram **reglur** sem gagnagrunnskerfi þarf að uppfylla til að geta talist töflu- eða venslagagnagrunnur. Reglurnar eru 13, en það er sérstakt að það er alltaf talað um 12 reglur (hann númerar þær frá 0-12, eins og sönnum tölvumanni sæmir!).

Reglurnar setti Codd fram til að reyna að koma í veg fyrir að framleiðendur gagnagrunnskerfa færu „út og suður“. Reglur Codds eru hins vegar svo stífar að fæst kerfi geta uppfyllt þær að fullu, þau kerfi sem byggja á „hreinu“ SQL ná ekki einu sinni að uppfylla þær allar.

Við eigum eftir að rekast á Codd síðar þegar við fjöllum um **skyldleikagreiningu** (*e. normalisation*).

Hér eru **reglur Codd's**¹⁵:

The rules

Rule 0: The system must qualify as *relational*, as a *database*, and as a *management system*.

For a system to qualify as a relational database management system (RDBMS), that system must use its *relational* facilities (exclusively) to *manage* the *database*.

Rule 1: The *information rule*¹⁶:

All information in the database to be represented in one and only one way, namely by values in column positions within rows of tables.

Rule 2 : The *guaranteed access rule*:

All data must be accessible with no ambiguity. This rule is essentially a restatement of the fundamental requirement for primary keys. It says that every individual scalar value in the database must be logically addressable by specifying the name of the containing table, the name of the containing column and the primary key value of the containing row.

Rule 3: *Systematic treatment of null values*¹⁷:

The DBMS must allow each field to remain null (or empty). Specifically, it must support a representation of "missing information and inapplicable information" that is systematic, distinct from all regular values (for example, "distinct from zero or any other number," in the case of numeric values), and independent of data type. It is also implied that such representations must be manipulated by the DBMS in a systematic way.

Rule 4: *Active online catalog based on the relational model*:

The system must support an online, inline, relational catalog that is accessible to authorized users by means of their regular query language. That is, users must be able to access the database's structure (catalog) using the same query language that they use to access the database's data.

¹⁵ Codd's rules (1.12.2007)

¹⁶ Táknað í raun að tafla þarf að uppfylla skilyrðin sem áður eru nefnd í kaflanum

¹⁷ Sérstakur kafli er um meðhöndlun á gildisleysunni NULL: **Kaflinn um gildið sem ekki er til**

Rule 5: The *comprehensive data sublanguage rule*:

The system must support at least one relational language that

- (a) Has a linear syntax
- (b) Can be used both interactively and within application programs
- (c) Supports data definition operations (including view definitions), data manipulation operations (update as well as retrieval), security and integrity constraints, and transaction management operations (begin, commit, and rollback).

Rule 6: The *view updating rule*:

All views that are theoretically updatable must be updatable by the system.

Rule 7: *High-level insert, update, and delete*:

The system must support set-at-a-time *insert*, *update*, and *delete* operators.

Rule 8: *Physical data independence*:

Changes to the physical level (how the data is stored, whether in arrays or linked lists etc.) must not require a change to an application based on the structure.

Rule 9: *Logical data independence*:

Changes to the logical level (tables, columns, rows, and so on) must not require a change to an application based on the structure. Logical data independence is more difficult to achieve than physical data independence.

Rule 10: *Integrity independence*:

Integrity constraints must be specified separately from application programs and stored in the catalog. It must be possible to change such constraints as and when appropriate without unnecessarily affecting existing applications.

Rule 11: *Distribution independence*:

The distribution of portions of the database to various locations should be invisible to users of the database. Existing applications should continue to operate successfully :

- (a) when a distributed version of the DBMS is first introduced; and
- (b) when existing distributed data are redistributed around the system.

Rule 12: The *nonsubversion rule*:

If the system provides a low-level (record-at-a-time) interface, then that interface cannot be used to subvert the system, for example, bypassing a relational security or integrity constraint.

2.7 Meira um hugtök

Hér eru hugtök sem nauðsynlegt er að kunna skil á vegna skilgreininga á töflulíkaninu.

- **Gráða** töflu (*e. degree, arity*), fjöldi dálka
- **Færslufjöldi** töflu (*e. cardinality*)

Mörg hugtök hafa fleiri en eitt heiti og það getur verið ruglingslegt á köflum, einnig þarf að huga að mismunandi heitum á íslensku og á ensku. Hér að finna nokkur veigamestu hugtökin og hvernig þau eru notuð á íslensku og ensku.

Íslenskt heiti			Enskt heiti		
Formlegt	Einnig notað		Formlegt	Einnig notað	
Vensl	Venslatafla	Tafla	Relation	Table	File
Færsla			Tuple	Row	Record
Eigindi	Dálkur		Attribute	Column	Field

Mynd 2-8 Mismunandi notkun hugtaka á íslensku og ensku

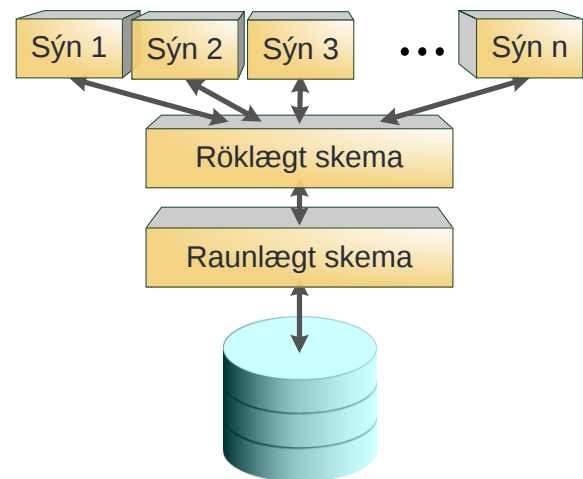
2.8 Annað til fróðleiks og umhugsunar

2.8.1 Skemu

Hér fyrr í kaflanum var talað um skemu, þá aðferð sem við notum til að sýna skilgreiningar á töflum. Málið er þó flóknara en svo að aðeins sé um eina tegund af skemum að ræða.

Af því sem notandinn og sá sem vinnur að gerð gagnasafnsins sjá af skemum eru tvær megintegundir og kallast þau **sýn** (*e. view*) og **röklægt skema** (*e. conceptual* eða *logical schema*). Sú tegund af skemum sem kynnt var hér fyrr í kaflanum er af gerðinni röklægt skema og er það sem algengast er að við notum við vinnu okkar við gagnagrunninn.

Sýn, einnig kallað sjónarhorn, er svokölluð **sýndartafla** (*e. virtual table*) þar sem búið er til sjónarhorn á eina eða fleiri töflur þar sem hægt er að takmarka og vinna frekar upplýsingar í þessa sýndartöflu. Það að kalla þetta sýndartöflu byggir á því ekki er um raunverulega töflu að ræða þar sem gögn eru geymd, heldur er hún búin til í hvert sinn sem hún er notuð. Að öðru leiti má (að mestu leiti) nota sýndartöflurnar sem hverja aðra töflu og allar vísanir í slíka er eins og um venjulega töflu sé að ræða. Í SQL hluta bókarinnar er frekar fjallað um skilgreiningu og notkun á sýn.



Raunlæg skemu (*e. physical schema*) lýsa því hins vegar hvernig gagnasafnskerfin skipuleggja töflur í skrár og hvernig **vísar** (*e. index*) eru notaðir. Þessi skemu eru gagnleg þeim sem vinna við stillingar á afköstum gagnasafnskerfa og öðrum slíkum sérfræðingum en það verður ekki fjallað frekar um þau hér.

2.8.2 Annar fróðleikur

Skilgreining Codd's á töflugagnagrunninum¹⁸: The relational model for database management: version 2. 1990

Ýmsir fræðimenn á sviði gagnasafnsfræði hafa áhyggjur af „úrkyngun“ töflulíkansins og færa sterk rök fyrir því að t.d. SQL og reyndar önnur fyrirspurnarmál brjóti reglur töflulíkansins¹⁹.

¹⁸ Codd. 1990

¹⁹ Criticisms of SQL. 9.12.2007

3 Lyklar og önnur skilyrði

3.1 Mikilvæg hugtök í kaflanum

aðkomulykill (e. *foreign key*)
aðkomulyklaskilyrði (e. *referential integrity*)
aðkomutafla (e. *foreign table, child table*)
aukalykill (e. *alternate keys*)
eigin skilgreind gagnatög (e. *UDT, user defined data types*)
barnafærslur (e. *child row*)
breyting gagna (e. *update rules*)
dálkur eða *eigindi* (e. *attribute, column*)
einkvæmur lykill (e. *unique key*)
eyðing gagna (e. *delete rules*)
fallákveður (e. *functional dependencies*)
foreldrafærslur (e. *parent row*)
frumlykill (e. *primary key*)
gagnaheilindi, gagnaheilleiki (e. *data or entity integrity*)
gildisbil (e. *range*)
gildisleysan NULL (e. *NULL*)
innsetning gagna (e. *insert rules*)
keðjuverkun (e. *cascade*)
kerfisgreining (e. *systema analysis and design*)
lykill (e. *key*)
lykiltafla (e. *key table, parent table*)
lyklaskilyrði (e. *key constraints*)
mögulegur lykill (e. *candidate key*)
núllstilling (e. *nullify*)
samanburðarmengi (e. *check constraint*)
samsettur lykill (e. *composite key*)
sjálfgefin gildi (e. *default value*)
skilyrði (e. *constraint*)
skilyrði skilgreiningarmengis (e. *domain constraints*)
skyldleikagreining (e. *normalisation*)

3.2 Skilyrði

Talað er um *skilyrði* (e. *constraint*) þá er átt við takmarkanir sem við setjum útfærslu á töflum, færslum og dálkum, innihaldi þeirra og samhengi, innan taflna og milli þeirra. Með notkun þessara skilyrða má verjast því að verið sé að vista óleyfileg eða óheppileg gögn í gagnasafninu. Oft er talað um skilyrði sem leið til að tryggja *gagnaheilindi* (e. *data integrity*). Einnig er verið að tryggja að tiltekið innbyrðis samhengi gagna haldist.

Sum að þeim skilyrðum sem sett eru færslum og innihaldi þeirra tengjast eðli og virkni forrita sem nota gagnasöfnin, svokölluð „business logic“ eða „business rules“. Það er hluti af viðfangsefnum *kerfisgreiningar* (e. *system analysis and design*) að finna og skjal-festa slíkar takmarkanir og reglur og verður ekki fjallað hér um þær aðferðir sem þar eru notaðar.

Um það er deilt hvort slík skilyrði eigi að staðsetja í gagnagrunninum en sú deila er ekki viðfangsefnið hér. Mestu máli skiptir hér að gera sér grein fyrir þeim möguleikum sem bjóðast til að skilgreina skilyrðin.

Almenn skilyrði venslatöflu takmarka það hvernig hægt er að setja upp töflur í gagn-safni. Þau skilyrði eru forsenda þess að hægt sé að beita þeim aðgerðum sem gagnasöfn bjóða upp á.

Til viðbótar við þessi skilyrði fyrir venslatöflu getum við skilgreint önnur skilyrði fyrir tiltekna töflur og dálka í gagnasafni. Athuga þarf að gagnasafnskerfi bjóða mismunandi aðferðir til að skilgreina þessi skilyrði og sum þeirra bjóða einungis hluta þeirra skilyrða sem nefnd verða hér.

Fjallað verður hér um helstu skilyrði sem sett eru töflum, fyrst fjallað um lykla og síðan skilyrði sem sett eru lykllum ásamt öðrum skilyrðum taflna, dálka og færslna.

3.3 Lyklar

Í gagnasafnsfræði notum við orðið *lykil* (e. *key*) til að tákna það eða þau gildi sem ein-kenna, *einkvæmt* (e. *unique*), tiltekna færslu í töflu. Lykill er þannig að hann vísar á eina og aðeins eina færslu, það er ekki um fleiri færslur að ræða sem uppfylla skilyrði lykilsins, þ.e. hafa sama gildi lykils.

Þegar við skilgreinum einhvern einn eða fleiri dálka töflu sem lykil þá erum við í leiðinni að setja þá takmörkun eða skilyrði að einungis skuli vera til ein færsla sem hafi þetta gildi lykilsins. Þetta er eitt meginskilyrði sem við setjum færslum í töflum²⁰.

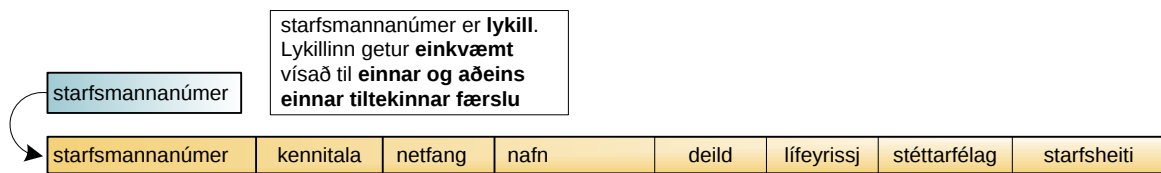
Við þekkjum ágætlega notkun slíkra lykla í daglegu lífi, við notum nöfn sem lykil í síma-skrá (er þó ekki alltaf einkvæmur²¹) og við flettum upp orðum í orðabók með orð sem lykil. Rík hefð er fyrir því á Íslandi að nota kennitölu sem lykil fyrir allt mögulegt, en við skulum venja okkur á að leita annarra leiða til að búa til lykla því yfirleitt er það hægt og gott að hafa milli mismunandi möguleika að velja. Kosturinn við notkun kennitölurnar er hins vegar sá að við höfum ákveðið að nota þær einkvæmt, þ.e. hver einasta kennitala vísar bara á einn einstakling og hver einstaklingur, hér á landi, á bara eina kennitölu.

Um lykil gildir það að hann verður að vera þannig að ekki sé nægilegt að nota hluta hans til að hann vísi einkvæmt á færslu, það verður að vera hann allur. Það er rétt að hafa þetta í huga þegar talað er hér síðar um *samsetta lykla* (e. *composite key*).

²⁰ Sum gagnasafnskerfi eru þannig útfærð að þau leyfa í raun að fleiri en ein færsla séu alveg eins og brjóta þannig helsta skilyrði töflu, en ef það er einhver huggun þá er það nú svo að færslur fá allar eitthvert ein-kvæmt einkenni sem gagnasafnskerfið úthlutar en það sjáum við sjaldnast. Með því móti er færslan í raun einkvæm ef þetta einkenni er tekið með.

²¹ Í símaskrá getur sama nafnið komið fyrir oftast en einu sinni og vísað á sinn hvorn einstakling, jafnvel geta verið alnafnar með sama heimilisfang, þannig að slíkar færslur eru ekki einkvæmar, vísa ekki endi-lega á þann einstakling sem við leitum eftir

Á skemamyndinni hér fyrir neðan er starfsmannanúmer lykill.



Í töflunni sem hér er á eftir er skrá yfir bifreiðar og í þessu tilviki er (eðlilega?) notast við fastnúmer bifreiðar sem lykil.

Bifreið	fastnr_bifr	skrán_dags	eink_framl	gerð	verð
	MU-143	20.12.2003	TOY	Corolla	NULL
	ME-956	2.3.2004	TOY	StreetCruser	1.450.000
	BÍ-200	14.5.1994	FOR	Xtra	1.212.000
	HO-001	26.7.1956	VOL	Amason	560.000
	GA-844	7.10.2004	FOR	Mondeo	2.074.100
	DÍ-777	19.12.1998	FIA	Major	250.000
	ÆR-002	2.3.2005	FOR	MiniParker	3.600.000

Lyklar eru ekki nauðsynlegir til þess að við getum unnið með töflur eða stakar færslur, það getum við án lyklanna. Lyklar segja okkur hins vegar hvað það er sem gerir færslur einkvæmar, hvað greinir eina færslu frá annarri. Það getur verið fleiri en einn dálkur sem gerir það og það getur einnig verið samsetning dálka sem gerir það (sbr. samsettur lykill hér á eftir) eða hreinlega að það þurfi alla færsluna í heild til að gera hana aðgreinanlega frá öðrum færslum (sjá einnig Hvað ef við finnum engan mögulegan lykil? hér á eftir).

Lykill gefur okkur þann möguleika að ef við þekkjum gildið á lyklinum þá vitum við hvaða (stöku) færslu er um að ræða. Tæknileg útfærsla gagnagrunnskerfis gerir það að verkum að uppfletting eftir lyklum er oftast mun hraðvirkari en leit eftir öðrum dálkum. Tilvísanir í færslur, t.d. frá öðrum töflum, byggja því oft á þessum lyklum. Sjáum meira um þetta í kaflanum þar sem fjallað er um *fallákveður* (*e. functional dependencies*).

3.3.1 Samsettur lykill

Þegar ekki er mögulegt að finna einn dálk sem fullnægir kröfunni um einkvæman lykil. Við getum þá skilgreint **samsettan lykil** (*e. composite key*), þ.e. lykil sem samanstendur af fleiri en einum dálk sem saman gera lykilinn einkvæman.

Tökum sem dæmi námskeið sem heitir Fjármál fyrirtækja sem er kennt bæði í Tækni- og verkfræðideild og í Viðskiptadeild. Um er að ræða námskeið með mismunandi skilgreiningu og sitthvorn kennarann.

námskeið	deild	kennari	önn
Gagnasafnsfræði	TD	Jón Freyr Jóhannsson	V09
Fjármál fyrirtækja	TVD	Agnar Jón	H08
Fjármál fyrirtækja	VD	Sigurjón Valdimarsson	H08
Gagnaskipan	TD	Hallgrímur Arnalds	V09

Til að greina á milli þeirra er því nauðsynlegt að finna annað til að greina milli þeirra en heiti námskeiðs. Samsettur lykill af deild og námskeiðsheiti gæti leyst þennan vanda.

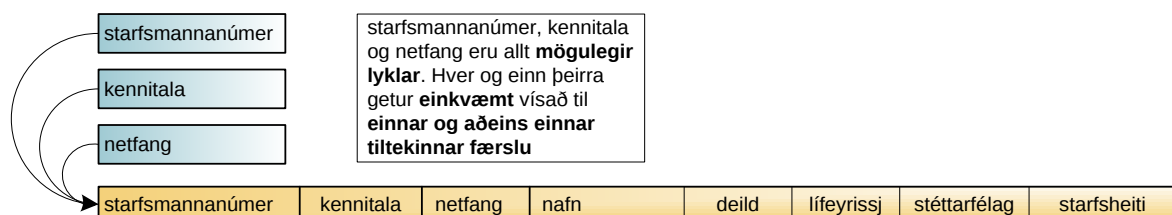
Lyklar eru undirstrikaðir hér í töflunni fyrir neðan.

<u>námskeið</u>	<u>deild</u>	kennari	önn
Gagnasafnsfræði	TD	Jón Freyr Jóhannsson	V09
Fjármál fyrirtækja	TVD	Agnar Jón	H08
Fjármál fyrirtækja	VD	Sigurjón Valdimarsson	H08
Gagnaskipan	TD	Hallgrímur Arnalds	V09

3.3.2 Mögulegir lyklar

Í töflu getur verið mögulegt að fleiri en einn dálkur eða samsetning dálka séu mögulegir lyklar, þ.e. margir dálkar eða samansafn dálka sem ákvarða einkvæmt færslu. Við þurfum því oft að velja milli **mögulegra lykla** (*e. candidate key*). Í skrá yfir netnotendur í tilteknu fyrirtæki gæti t.d. verið um það að ræða að kennitala starfsmanns, starfsmannanúmer eða netfang sé hvert um sig **einkvæmur lykill**. Þetta köllum við mögulega lykla, því hver um sig getur ákvarðað einstaklinginn (færslu) einkvæmt.

Mögulegir lyklar eru þeir **dálkar** (einn eða samsetning margra dálka) **sem ákvarða færslu einkvæmt**. Þeir eru minnsta mögulega samsetning dálka sem geta sameiginlega talist lykill. Þegar við tölum um mögulega lykla þá er átt við lykil sem er þannig að ef við tökum einhvern hluta hans burtu þá er hann ekki lengur einkvæmur, eða einfaldlega sagt: **lykill má ekki innihalda óþarfa!**



Mynd 3-1 Mögulegir lyklar töflu

Dæmi: Ef við gefum okkur það að engir alnafnar búi á sama stað, þá getum við sagt að nafn + heimili sé mögulegur lykill í símaskrá. Við vitum að nafn er ekki nóg (vegna alnafna) og ef við sleppum nafni þá erum við heldur ekki með einkvæman, mögulegan lykil.

3.3.3 Frumlykill

Tafla hefur skilgreindan einn einkvæman *frumlykil* (e. *primary key*)²². Við veljum einn af mögulegu lyklunum til að vera frumlykil.

kennitala	nafn	aldur
0123456789	Sveinn Baldursson	55
1122334455	Anna Pálsdóttir	43
0909094321	Elín Aradóttir	35
1705602222	Arnaldur Guðmundsson	18

Mynd 3-2 Frumlykill (kennitala) tilgreindur með undirstriki

Frumlykill er oft táknður á töfluskema með undirstriki (sbr. kennitala hér á myndinni). Formlega sagt er það frumlykillinn sem ákvarðar alla hina dálka töflunnar, fjöllum meira um það þegar við skoðum *töflualgebru*.

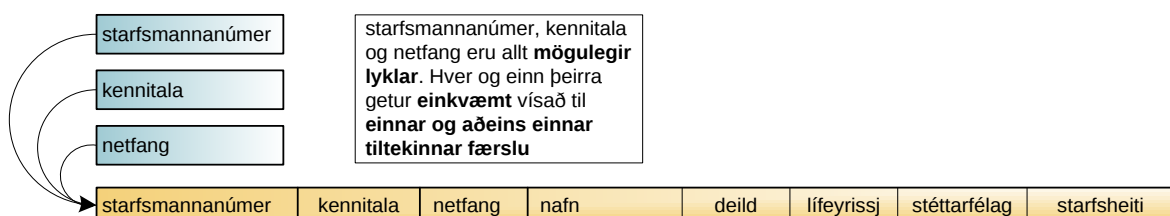
Það hvaða lykil við veljum sem frumlykil getur ráðist af nokkrum þáttum. Eitt sem þarf að hafa í huga er það að gagnasafnskerfi útfæra lykla þannig búinn er sjálfkrafa til *vísir* (e. *index*) úr lyklinum og er því yfirleitt mun hraðvirkara að leita eftir færslu byggt á gildi lykilsins. Fjallað verður betur um vísa í öðrum kafla.

Annað atriði sem getur ráðið vali frumlykils er það hvaða lykil hentar að nota fyrir vísun aðkomulykla frá öðrum töflum, aðkomulyklar geta einungis vísað í frumlykil töflu. Sjá nánar um aðkomulykla hér á eftir.

3.3.4 Aukalyklar

Aukalyklar (e. *alternate keys*) kallast þeir *mögulegu lyklar* sem *ekki voru valdir* sem frumlyklar. Mögulegir lyklar geta verið margir í hverri töflu. Þeir eru allir jafnrétt háir ef svo má segja en á einhverju stigi getum við ákveðið að velja einhvern þeirra sem frumlykil. Þeir sem eftir eru kallast aukalyklar en hefði hver og einn getað verið valinn sem frumlykill.

Ef starfmannanúmer væri valið sem frumlykill í töflunni sem myndin sýnir, þá væru kennitala og netfang aukalyklar töflunnar.



²² Þ.e.a.s. ef við skilgreinum lykla, við höfum alltaf þann möguleika að líta á færsluna í heild sem einkvæma og þannig uppfylla skilyrði fyrir töflu.

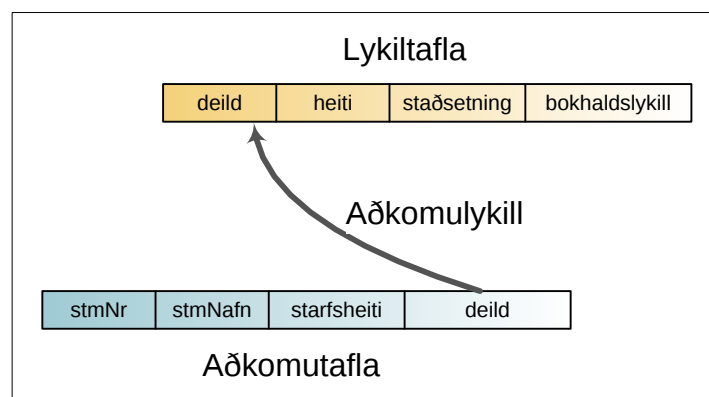
3.3.5 Hvað ef við finnum engan mögulegan lykil?

Það getur í raun ekki gerst að við finnum engan mögulegan lykil því samkvæmt skilgreiningu þá er hver færsla einkvæm og í versta falli þá eru allir dálkar samanlagt einkvæmur lykill ef slík staða kemur upp.

Ef gögnin sem við vinnum með eru þess eðlis að við höfum engan „eðlilegan“ einkvæman lykil, þ.e. færslan í eðli sínu ekki einkvæm, þá verðum við að búa til aukadálk, t.d. með hlaupandi númeri, til að gera færsluna einkvæma. Flest gagnasafnskerfi gera okkur þetta auðvelt ef þarf. Þessi aukadálkur er í eðli sínu einkvæmur og því nægjanlegur einn og sér til að vera frumlykill eða að minnsta kosti að gera færsluna í heild einkvæma.

3.3.6 Aðkomulyklar

Aðkomulyklar (*e. foreign key*) eru ekki eiginlegar lyklar innan töflunnar sem þeir eru í. Þeir vísa í lykila í **öðrum töflum** (stundum reyndar innan sömu töflu). Þeir eru notaði til að tengja saman færslur úr mismunandi töflum.



Mynd 3-3 Aðkomulykill í færslu í sömu eða annarri töflu

Aðkomulykill er einnig kallaður **tengilykill** eða **framandlykill**. Taflan sem aðkomulykill vísar til er kölluð **lykiltafla** (*e. key table*). Taflan sem aðkomulykillinn er staðsettur í (sbr. neðri taflan hér á myndinni að ofan) er kölluð **aðkomutafla** (*e. foreign table*) þegar verið er að ræða um samhengi þessara taflna byggt á aðkomulykli.

Aðkomulykill verður að vera af sömu lengd og úr sama skilgreiningarmengi og lykillinn sem vísað er til.

Lykiltafla er á ensku oft kölluð **parent**-tafla og aðkomutafla **child**-tafla, færslurnar samsvarandi kallaðar **foreldrafærslur** (*e. parent row*) og **barnafærslur** (*e. child row*).

3.4 Skilyrði lykla

3.4.1 Skilyrði frumlykils

Til að geta verið **frumlykill** (e. *primary key*) töflu þá þarf hann að uppfylla eftirfarandi skilyrði:

- hann verður að vera einkvæmur fyrir hverja einustu færslu töflunnar, þ.e. hann verður að uppfylla skilyrði þess að vera lykill (ætti líklega að vera augljóst)
- hann má ekki á neinum tíma innihalda **tómagildið** (*gildisleysuna* **NULL**), hvorki í heild sinni eða hluta. Ástæðan er sú að þá er hann ekki lengur einkvæmur, NULL telst ekki gildi og getur því ekki verið einkvæmt gildi jafnvel þó aðeins ein færsla innihaldi NULL. Sama skilyrði gildir í raun fyrir mögulega lykla því annars hefðu þeir ekki getað komið til greina sem frumlyklar.

Nánar er fjallað um gildisleysuna **NULL** í öðrum kafla.

3.4.2 Aðkomulyklaskilyrði

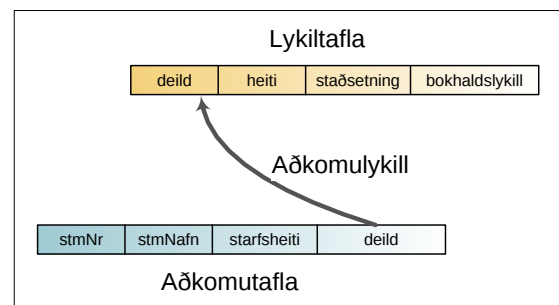
Aðkomulyklar eru í raun alls ekki lyklar heldur gildi sem vísa á lykilgildi annarrar færslu í sömu töflu eða annarri. Vegna þessa þá þurfa ekki sömu skilyrði að gilda og um lykla, þ.e. þeir mega innihalda NULL (sem táknar þá að engin vísun er til staðar eða þekkt).

Grundvallarskilyrði aðkomulykla kallast **aðkomulyklaskilyrði** (e. *referential integrity*).

Í því flest að gildið í aðkomulykli í aðkomutöflunni **verður að vera til** í lykiltöflunni.

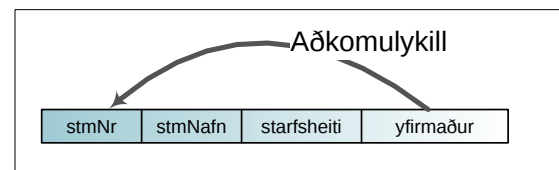
Ef svo er ekki má ekki vera þar annað gildi en **gildisleysan NULL**.

Aðkomulykill má sem sagt ekki vísa á gildi sem ekki er til í lykiltöflunni.



Eins og kom fram í skilgreiningu á aðkomulykli þá getur **aðkomulykill einungis vísað til frumlykils** í lykiltöflunni

Aðkomulykill getur vísað til lykils í eigin töflu.



Hafa þarf í huga að með aðkomulyklaskilyrði erum við að setja skilyrði sem þarf að uppfylla (oftast) í fleiri en einni töflu og það flækir samspil gagna augljóslega og við þurfum að taka tillit til þess við alla vinnu við gagnasafnið og skilgreiningu þess.

Til að tryggja **gagnaheilindi** (e. *data integrity*), þ.e. að gögn séu rétt í þeim skilningi að aðkomulyklaskilyrði séu varðveitt, eru ýmsar aðferðir.

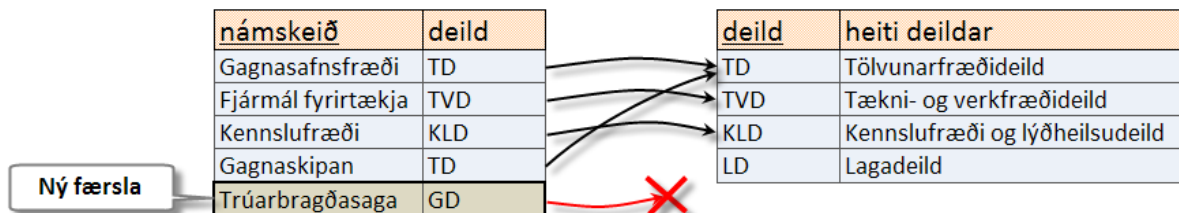
Viðfangsefni sem þarf að takast á við vegna aðkomulyklaskilyrða tengjast helst skráningu færslna, breytingu og eyðingu. Þessi viðfangsefni eru á ensku kölluð „Insert, Update and Delete Rules“.

Tökum hvert af þessum viðfangsefnum fyrir. Athuga þarf að útfærslur á aðkomulyklaskilyrðum eru útfærðar á sérstakan hátt í SQL sem verður fjallað um nánar í viðeigandi köflum SQL hluta bókarinnar.

Innsetning gagna (*e. insert rule*)

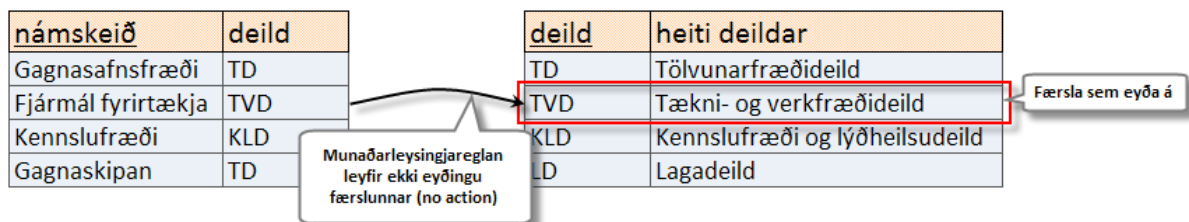
Einfaldasta viðfangsefnið er við innsetningu nýrrar færslu í aðkomutöflu. Tryggja þarf að aðkomulyklar **vísi á lykild** sem eru til í lykiltöflu (*e. dependent*) eða hafi NULL gildi ella (*e. nullify*).

Við getum reyndar sett strangara skilyrði þannig að aðkomulykill verði alltaf að hafa samsvarandi gildi í lykli lykiltöflunnar (*e. no effect*: í þeirri merkingu að ekkert sé gert ef ekki er til gildi). Einnig er hægt að setja það skilyrði að ef ekki sé til færsla í lykiltöflu með viðeigandi gildi þá sé búin til slík færsla (*e. automatic insert*).

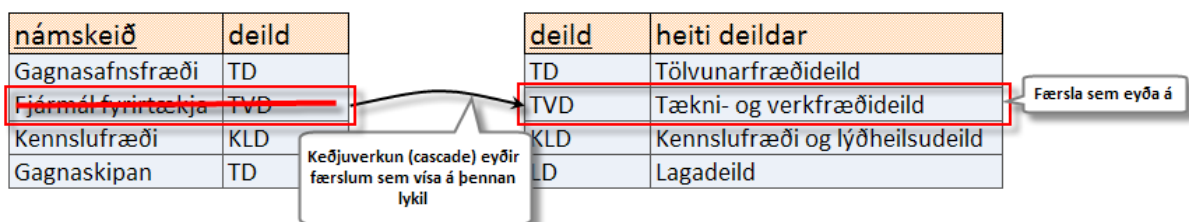
**Eyðing gagna** (*e. delete rule*)

Við eyðingu gagna þarf sérstaklega að horfa til þess að þegar færslu er eytt úr lykiltöflu hvort það séu einhverjir aðkomulyklar sem vísa á þessa færslu. Nokkur viðbrögð eru möguleg.

Hægt er að skilgreina reglu þar sem bannað er að foreldrafærslu sé eytt ef til eru barna-færslur sem vísa til hennar (*e. restrict*). Þetta er stundum kölluð **munaðarleysingjareglan** (*e. orphan rule*) þar sem slíkt er bannað.

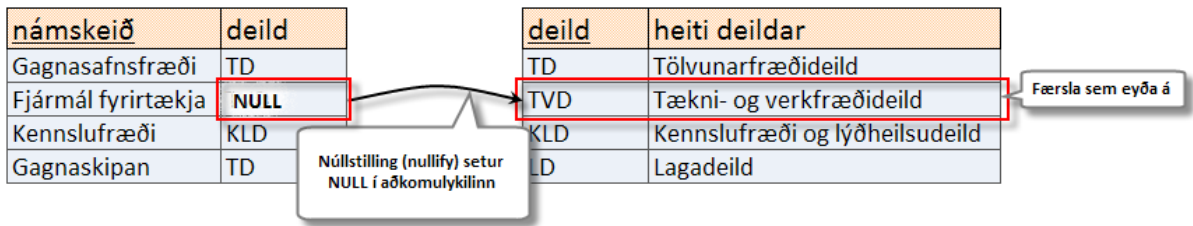


Keðjuverkun (*e. cascade*) er það kallað að ef foreldrafærslunni er eytt þá er öllum (barna-) færslum eytt sem vísa til hennar²³.



²³ Ég biðst afsökunar á því hve noturlega þetta hljómar með þessu orðfæri (höf.)

Núllstilling (*e. nullify*) er það kallað þegar leyft er að eyða foreldrafærslu en NULL sett í aðkomulykla þeirra færslna sem vísuðu til hennar (ef það er þá leyfilegt, ef slíkt er ekki leyft ræður sú regla).



Breyting gagna (*e. update rules*)

Breyting gagna þ.e. breytingu á lykiligildi í lykiltöflu (foreldratöflu) er hægt að setja samskonar skilyrði og við eyðingu, en við breytingu á gildi aðkomulykils í aðkomutöflu (barnatöflu) er hægt að setja skilyrði líkt þeim sem sett eru innsetningu.

3.5 Dálkaskilyrði

Hverjum dálki töflu fyrir sig er hægt að setja skilyrði og eru þau margþætt og verður hér reynt að setja þau fram á einhvern skipulegan hátt

Lykilskilyrði (*e. key constraint*)

Í meginatriðum er tvær leiðir til að setja dálkum lykilskilyrði. Skilgreina má dálk (eða samansafns dálka sbr. samsetta lykla sem **frumlykil** (*e. primary key*). Dálkurinn (eða dálkarnir) verða því að fylgja þeirri reglu lykla að vera einkvæmir. Þá er einnig hægt að skilgreina dálka sem einkvæma þó þeir séu ekki frumlyklar. Það er einfaldlega gert með því að skilgreina þá með lykilorðinu **einkvæmur** (*e. unique*). Í raun er verið að knýja fram skilyrði sem dálkurinn uppfyllir sem mögulegur lykill, þ.m.t. ætti það að vera þannig að hann megir ekki innihalda gildisleysuna NULL en það á ekki við í öllum gagnasafnskerfum.

Athuga þarf að við þessa skilgreiningu á einkvæmni verður til **vísir** (*e. index*) sem getur bæði verið kostur og galli. Kosturinn er að leit getur verið einfaldari eða hraðar, ókosturinn er sá að hver vísir í töflu kostar nokkuð í umsýslu.

Skilyrði skilgreiningarmengis (*e. domain constraints*)

Einföldustu skilyrði dálka eru þegar við skilgreinum að þau innihaldi einhver af grunn-tögum sem í boði eru (s.s. integer, decimal, char).

Dæmi um skilgreiningu töflu í SQL:

```
create table starfsmenn
(
    nafn char(50),
    heimili char(50),
    starfsmannanumer int,
    haed decimal (3,2),
    netfang char(50)
)
```

Ganga má lengra í að tilgreina skilgreiningarmengið með því að tilgreina lögleg gildi eða sjálfgefin gildi ef engin eru tiltekin.

```
kyn char(1) check (kyn IN ('f','m'))
skoli char(3) default 'HR'
```

Almennt getum við sagt að dálkaskilyrði eða skilyrði skilgreiningarmengis innihaldi eitt eða fleiri af eftirtöldum atriðum:

- **Gagnatag** (e. *data type*)²⁴. Ýmist er um að ræða grunntög en einnig bjóða sum gagnagrunnskerfi upp á **eigin skilgreind gagnatög** (e. *UDT, user defined data types*)
- **Stærð** (e. *length*)²⁵. Textagagnatög og einnig tölur eru af mismunandi lengd og nákvæmni.
- **Form dagsetninga** (e. *date format*)
- **Gildisbil** (e. *range*). Tilgreint er eitthvert bil gilda sem eru leyfileg
- **Samanburðarmengi** (e. *check constraint*)²⁶. Kannað er hvort skilyrði tilheyri einhverju skilgreindu mengi gilda, einnig má snúa skilyrðinu við þannig að gildið **megi ekki tilheyra** samanburðarmenginu. Formið á þessu skilyrði er oftast eins og sýnt er í dæminu hér að ofan (**check (kyn IN ('f','m'))**) en rétt er að kynna sér í hverju gagnasafnskerfi fyrir sig hvernig þetta er útfært.
- **Núllstilling** (e. *nullify*). Skilgreint er hvort dálkur má innihalda NULL. Þetta er gert með því að setja lykilorðin **NULL** eða **NOT NULL** til að segja til um hvort megi, sjálfgefið er yfirleitt að leyfa NULL. Hafa þarf þó í huga að ef dálkur er skilgreindur sem frumlykill (e. *primary key*) þá þarf ekki að tilgreina neitt því hann má alls ekki innihalda NULL
- **Sjálfgefin gildi** (e. *default value*). Tilgreint hvort nota eigi eitthvert sjálfgefið gildi ef ekkert annað er til staðar
- **Aðkomulykill** (e. *foreign key*). Við skilgreiningu á aðkomulykli er tiltekið hvað gera skuli við innsetningu, eyðingu eða breytingu gagna, sbr. undirkaflann um aðkomulyklaskilyrði hér að framan. Fyrir hvern og einn af þessum möguleikum er hægt að tilgreina þann valkost sem nota á, breyta eða banna breytingu eða þá að núllstilli.

3.6 Annað til fróðleiks og umhugsunar

Oft er á ritum og í skilgreiningum notaðar skammstafanirnar FK og PK til að tákna aðkomulykil (FK, foreign key) og frumlykil (PK, primary key).

Tafla	Starfsmenn
(PK) starfsmannanúmer	<<PK>>starfsmannanúmer
nafn	-nafn
(FK) deild	<<FK>>-deild

²⁴ Kannaðu hvaða grunngagnatög eru í því gagnagrunnskerfi sem þú vinnur í

²⁵ Útfærslur má finna í SQL handbókum

²⁶ Nánari lýsingu á möguleikum samanburðarmengisskilyrða, check, er að finna í SQL handbókum og einnig í SQL hluta bókarinnar

4 Kaflinn um gildið sem ekki er til

4.1 Mikilvæg hugtök í kaflanum

gildisleysan NULL (e. NULL)

skyldleikagreining (e. normalisation)

röktafla með þremur rökildum (e. three value logic table)

4.2 Hvað er NULL?

Í gagnasafnsfræði er notað gildi sem heitir NULL (borið fram sem „nöll“) til að tákna það **að ekki sé neitt gildi til staðar**. Þetta gildi hefur mjög sérstaka merkingu og hefur þó nokkur áhrif á hefðbundna notkun rökfræðitákna og rökfræðisetninga. Ef ekki er tekið tillit til þessa gildis er viðbúið að það komi óvæntar villur.

NULL er hvorki núll né nix²⁷. Í stærðfræði hefur talan núll sérstaka merkingu og í mengjafræði er talað um tóma mengið, í tölvumeðhöndlun á texta er talað um „tóma strenginn“ til að tákna „ekki neitt“ eða „“. Notkun NULL í gagnasafnsfræði **er ekki neitt af þessu**. **NULL táknar að ekkert gildi sé til staðar eða það sé ekki vitað**. NULL er gildisleysa.

NULL er notað í eftirfarandi tilvikum:

- Gildi vantar
- Gildi er óþekkt (á þessari stundu a.m.k.)
- Á ekki við að gildi sé til staðar
- Þegar gildi er ekki þekkt (ekki endilega vegna þess að það sé ekki til)
- Ekkert gildi á að vera til staðar

Eitt af því sem NULL gildið hjálpar okkur með er að finna eigindi sem vantar gildi og hugsanlega getum við unnið eitthvað með það og sett viðeigandi gildi, en hafa verður í huga að það á alls ekki alltaf við að breyta þurfi NULL gildi í eitthvað annað.

Segjum sem svo að við höfum töflu þar sem færð er í skráning nemenda á tiltekið námskeið ásamt einkunn þegar þar að kemur.

Dálkur er fyrir nafn námskeiðsins og annar þar sem fært er nafn nemandans. Gera má ráð fyrir því að í upphafi námskeiðs séu allir skráðir sem þar ætla að vera. Hvenær gildi fara inn í töfluna getur verið erfitt að segja.

²⁷ Orðasambandið *núll* og *nix*: Merkingin er ‘alls ekki neitt’. Orðasambandið hefur borist hingað til lands úr dönsku þar sem sagt er *nul* og *nix* á sama hátt og í íslensku. *Nix* í dönsku er tekið að láni úr þýsku *nichts* ‘ekkert’ eða lágbýsku *nichs* í sömu merkingu. Í þýsku er notað bæði *Null und nichts* og *Null und nichtig* í sömu merkingu og í íslensku.

Guðrún Kvaran. „Hvað merkir ‘nix’ í orðasambandinu ‘núll og nix’?“. *Vísindavefurinn* 23.5.2005. <http://visindavefur.hi.is/?id=5013>. (Skoðað 5.10.2007).

Tökum dæmi af eftirfarandi töflu

Árangur	nemandi	námskeið	einkunn
	Edda Stefanía Jónsdóttir	Tölfræði 1	8,5
	Ævar Már Óskarsson	Forritun 1	NULL
	Inga Birna Jónsdóttir	Forritun 1	6,5
	Una Björg Magnúsdóttir	Líffræði	9,0
	Einar B. Magnússon	Stærðfræði 1	9,5
	Elísabet Erla Birgisdóttir	Kerfisgreining	NULL

Í þessari töflu eru tvö NULL. Hvernig eigum við að túlka þau? Það að skilgreina þau sem töluna núll? Ef svo væri þá veri verið að segja að þessir nemendur hafi fengið þessa tilteknu einkunn í námskeiðinu og það er alls ekki það sama og að segja að ekki sé búið að gefa þeim einkunn. Kannski á Ævar eftir að taka sjúkrapróf í Forritun 1 og Elísabet sé þarna með skráningu í námskeið sem ekki er hafið. Eins og sjá má á þessu þarf að fara virkilega varlega í að leggja nokkra merkingu í NULL. NULL er eins og áður segir gildisleysa og í eðli sínu án frekari merkingar.

Athuga þarf einnig að fall eins og meðaltal (**AVG()**) taka ekki NULL með, ef hins vegar það stæði talan núll í stað NULL þá væru þau gildi tekin með í meðaltalinu og það er alls ekki það sem æskilegt er.

Á sama hátt virkar fallið að telja (**COUNT()**) þannig að NULL er sleppt í tölunni. **COUNT(einkunn)** í töflunni hér að ofan mundi því skila tölunni 4²⁸.

Þú getur með því að leita upp eigindin sem hafa NULL-gildið ef þú vilt t.d. finna hvar vantar gögn. En NULL gerir samt meira en að tákna að það vanti gögn eins og áður er sagt, það hefur líka þá meiningu að ekkert gildi eigi þar við.

Þeir sem koma nýir að notkun gagnasafna eiga oft í erfiðleikum með að meðhöndla þessa gildisleysu sem NULL er. Það þarf því að taka sérstakt tillit til þessa m.a. í samanagerð gilda og við tengingu taflna.

Í skilgreiningu á dálkum töflu þá getum við sett inn skilyrði þar sem við tiltökum hvort yfir höfuð megi nota NULL í þeim dálki eða ekki. Ekki þarf að taka slíkt fram í dálkum sem eru lyklar því þeir mega ekki innihalda NULL.

²⁸ Athuga þarf að ef notað er táknið *, sem táknar að allt eigi að vera með, þá skilaði COUNT(*) tölunni 6, en þá þyrfti reyndar að vera það skilyrði líka að aðeins væri verið að taka þennan dálk en aðrir væru ekki með. Sleppum því að velta okkur mikið upp úr því, það er mikilvægara að hafa í huga að það er í flesta staði varhugavert að nota „*“ í fyrirspurnum, slíkt getur haft ýmsa óæskilegar verkanir.

4.3 Nokkrar einfaldar reglur við meðhöndlun á NULL

4.3.1 Regla 1: Þekkja til NULL

Ekki misskilja NULL sem töluna núll eða sem tómastrenginn, ekki reyna heldur að túlka NULL sem eitthvað annað, það er einfaldlega óþekkt.

Fyrirspurnir í SQL meðhöndla gildisleysuna NULL á mismunandi hátt og sem betur fer er það oftast á þann hátt sem maður gæti haldið!

AND rökaðgerðin

NULL and [eitthvert gildi sem er FALSE] → FALSE

NULL and [eitthvert gildi sem er TRUE eða NULL] → NULL

OR rökaðgerðin

NULL or [eitthvert gildi sem er TRUE] → TRUE

NULL or [eitthvert gildi sem er FALSE eða NULL] → NULL

4.3.2 Regla 2: Ekki bera NULL saman við einhver gildi, ákveddu hvort velja á eða hafna NULL

Ekki bera gildi eiginda (gildi í dálki) saman við önnur gildi með hefðbundnum samanburðarvirkjum ef mögulegt er að NULL geti verið þar.

Í samanburði þar sem venjulega væri notað jafngildismerkið „=" þarf að nota **IS NULL** eða **IS NOT NULL**, annað er ekki leyfilegt, NULL ≠ NULL.

Dæmi um óleyfilegan samanburð:

```
SELECT pósthúmer from Bíll WHERE eigandi = NULL -- Má ekki!
SELECT pósthúmer from Bíll WHERE eigandi = 'NULL' -- Má ekki!
```

Dæmi um það að hafna NULL:

```
SELECT nemandi
FROM skráning
WHERE önn is not null
```

4.3.3 Regla 3: Skilja hvernig *safnaðgerðir* (e. *aggregates*) taka tillit til NULL

COUNT (dálkur), skilar fjölda gilda í tilteknum dálki, NULL ekki meðtalið nema ef notað er **COUNT (*)**

SUM (dálkur), skilar samtölu gilda í tilteknum dálki, NULL ekki meðtalið

AVG (dálkur), skilar meðaltali gilda í tilteknum dálki, NULL ekki meðtalið

MIN (dálkur), skilar minnsta gildi í tilteknum dálki, NULL ekki meðtalið, enda getur það hvorki verið minnsta né hæsta gildi þar sem það hefur ekkert gildi!

MAX (dálkur), skilar hæsta gildi í tilteknum dálki

4.3.4 Regla 4: Taka tillit til NULL gilda í samanburði (skilyrðisbundinni leit)

Þegar við notum **WHERE** til að takmarka niðurstöður í **SELECT** setningu þá er mikilvægt að taka tillit til hvað gera á við NULL.

Eftirfarandi fyrirspurn skilar öllum með einkunn undir 5, en tekur NULL ekki með:

```
SELECT nemandi
FROM Árangur
WHERE einkunn < 5
```

Fyrirspurnin hér á eftir tekur hins vegar þá með sem hafa NULL (hvort það er skynsamlegt er auðvitað bara háð því til hvers fyrirspurnin er ætluð):

```
SELECT nemandi
FROM Árangur
WHERE einkunn < 5 or einkunn is null
```

4.3.5 Regla 5: Þekktu hvernig reikniaðgerðir virka með NULL

```
(100 x 3) + 10 = 310
(Null x 4) + 10 = Null
(20 x Null) + 100 = Null
(20 x 100) + Null = Null
1 + NULL = NULL
```

Að sjálfsgöðu má gera ráð fyrir enginn setji fyrirspurn svona upp, en ef við erum að vísa til gilda sem við vitum ekki hvort geta innihaldið NULL eða ekki þá er rétt að tryggja það á annan hátt í fyrirspurninni að búið sé að velja slík gildi frá. Sama á við um frádrátt og deilingu.

4.3.6 Regla 6: GROUP BY skilar NULL færslum

GROUP BY skilar gildum með NULL og hefur þau ýmist fremst eða aftast í niðurstöðum eftir því hvernig röðun er valin. Eftirfarandi fyrirspurn skilar öllum færslum þ.m.t. þeim sem hafa NULL, þó það sé verið að flokka saman eftir einkunnum.

```
SELECT einkunn, nemandi, námskeið
FROM Árangur
GROUP BY einkunn, nemandi, námskeið
```

4.3.7 Regla 6: Hafðu í huga þörf fyrir *skyldleikagreiningu* (*e. normalisation*)

NULL er áhugavert og ekkert athugavert þó að það komi fyrir í færslum, þegar það á við auðvitað. Ef hins vegar mikið er um NULL í tilteknum dálkum þá gæti það bent til þess að vinna þurfi eitthvað frekar með hönnun töflunnar og eru þá notaðar aðferðir *skyldleikagreiningar* (*e. normalisation*) sem fjallað er um í öðrum kafla. Það er ekkert sjálfgefið að slíkt leiði til einhverra breytinga en þetta er svona vísbending um að slíkt gæti þurft að gera.

4.3.8 Regla 7: Röktafla með þremur rökgildum

Gamla röktaflan okkar um gildin satt og ósatt verður að stækka þegar við vinnum með töflu í töflugagnagrunni. Hún hefur stækkað því við verðum að taka tillit NULL. Nýja taflan er á þennan veg:

Röktafla með þremur rökgildum			
p	q	p og q	p eða q
True	True	True	True
True	False	False	True
True	NULL	NULL	True
False	True	False	True
False	False	False	False
False	NULL	False	NULL
NULL	True	NULL	True
NULL	False	False	NULL
NULL	NULL	NULL	NULL

4.4 Annað til fróðleiks og umhugsunar

Sjá í Wikipedia um NULL, http://en.wikipedia.org/wiki/Null_%28SQL%29

5 Vísar

5.1 Mikilvæg hugtök í kaflanum

aukavísir eða *óklasaður vísir* (*e. secondary index, non-clustered index*)

B+ trjáa gagnaskipan (*e. B+ structure*)

bendir (*e. pointer*)

dýpt trés (*e. tree depth*)

fata (*e. bucket* eða *bin*)

færslustaðgreinir (*e. row locator*)

gisinn vísir (*e. sparse index*)

gráða trés (*e. order*)

hrúga (*e. heap*)

jafn leitartíma

klasaður vísir (*e. clustered index*)

klösuð tafla (*e. clustered table*)

lauf (*e. leaf node*)

lög (*e. level*)

óklasaður vísir (*e. non-clustered index*)

raðaður vísir (*e. ordered index*)

tætifall (*e. hash function*)

tætifallsvísir (*e. hash index*)

töfluskanna so. (*e. tablescan*)

vísatafla (*e. index table*)

vísauppsetning (*e. indexing*)

vísir (*e. index*)

þéttur vísir (*e. dense index*)

5.2 Vísar

Samkvæmt skilgreiningu á töflu í töflugagnasafni þá skiptir röð færslna ekki máli og ekki er hægt að ganga út frá því að þar séu gögn í einhverri tiltekinni röð, hvorki í þeirri röð sem færslurnar verða til eða annarri röð.

Til að finna færslur sem hafa einhverja tiltekna eiginleika þarf því oft að lesa alla töfluna til að vera viss um að finna þær sem leitað er að.

Þetta fyrirkomuleg virkar alveg því það er auðvitað hægt. Það hentar hins vegar illa að þurfa t.d. að lesa sig í gegnum alla símaskrána til að finna símanúmer þeirra sem eiga heima á tilteknu heimilisfangi, sérstaklega ef um væri að ræða símaskrá stórs svæðis.

Í mörgum bókum, kennslubókum og öðrum, er að finna atriðaorðasafn þar sem er að finna markverð orð og þar sem tilgreind er staðsetning þeirra innan bókarinnar og þannig er flýtt fyrir uppflettingu okkar á þessu tilteknu orði og það kemur í veg fyrir að

við þurfum að lesa alla bókina til að finna orðið. Í enskum bókum eru þessi atriðisorðasöfn kölluð *Index*, á íslensku er þetta kallað atriðisorðasafn.

Þegar við tölum um *vísa* (e. *index*) í gagnasafnsfræði þá erum við að tala um svipuð fyrirbæri. Við byggjum upp fljótlega leið til að finna tiltekin gildi án þess að þurfa að lesa alla viðkomandi töflu.

Gagnasafnskerfi gera þetta ekkert á ólíkan hátt og gert er í bókum. Á sérstökum stað er öllu því sem hægt er að leita að með *vísi* (e. *index*) raðað og síðan er númer sem vísar til þess hvar finna má atriðið í textanum (í bókum) eða færsluna (í gagnasafni).

Til að forðast misskilning þá þarf að taka það fram að í atriðisorðasafni bóka þá eru orð valin sem á að setja í slíkt safn en þegar í gagnasafni veljum við dálka sem slík *vísauppsetningin* (e. *indexing*) á að eiga við og þá er fyrirkomulagið þannig að öll gildi eru tekin með, meðan í bók þá velur höfundur það sem hann telur markvert að fari í slíkan lista. Samlíkinguna má ekki taka of bókstaflega þó hún geti hjálpað til við skilning á fyrirbærinu, útfærslan er til dæmis allt önnur.

8	23	Jón Freyr Jóhannsson	8980517	Kaplaskeiði 17
11	43	Hallgrímur Arnalds	8770666	Kóðavöllum 13
23	11	Halldór Halldórsson	6864545	Tegurteig 2 ½
43	8	Marta Kristín Lárusdóttir	5550123	Viðmótsbraut 1
68	68	Daníel Brandur	8880888	Vefvangi 3 C#

Vísar (e. *index*) geta verið á fleiri en einn dálk í hverri töflu og er það þá gert til að flýta fyrir mismunandi fyrirspurnum. Hér fyrir neðan er sýnt hvernig tveir vísar eru notaðir, til sitt hvors gagnsins.

Vísir á
númer

8	23	Jón Freyr Jóhannsson	8980517	Kaplaskeiði 17	
11	43	Hallgrímur Arnalds	8770666	Kóðavöllum 13	
23	11	Halldór Halldórsson	6864545	Tegurteig 2 ½	
43	8	Marta Kristín Lárusdóttir	5550123	Viðmótsbraut 1	
68	68	Daníel Brandur	8880888	Vefvangi 3 C#	

Vísir á nafn

Daníel
Halldór
Hallgrímur
Jón Freyr
Marta Kristín

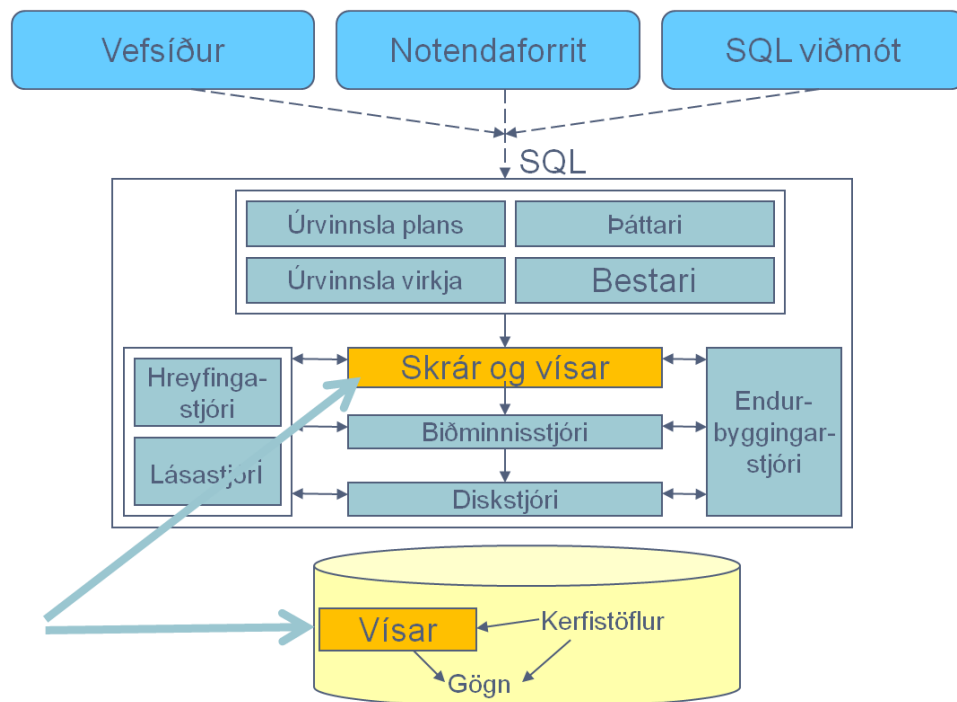
Nú gæti einhver með réttu haldið því fram að best væri að hafa vísi á hvern einasta dálk til að auðvelda leit að færslum. Málið er því miður ekki svo einfalt því vísar kosta pláss og þeir kosta líka tíma. Þegar færslu er bætt við gagnasafn er hún einfaldlega sett aftast (eða á einhver annan tiltölulega ófyrirsjáanlegan hátt), vísabygginguna þarf hins vegar oft að byggja upp á nýtt því nú er komið nýtt gildi sem setja þarf á sinn stað og því getur fylgt nokkuð umstang og töluverður vinnslutími. Þetta er m.a. ástæða þess að mörg stór gagnasöfn, þar sem mikið gagnamagn fer í gegnum á hverjum tíma, eru þannig að vísarnir eru ekki byggðir samtímis, oft á kvöldin eða utan álagstíma.

Ef engin *vísir* (e. *index*) er skilgreindur á tiltekna töflu þarf alltað að lesa hana frá upphafi til enda ef leitað er eftir tilteknum skilyrðum til að tryggja að engin færsla verði útundan.

Þetta köllum við að **töfluskanna** (e. *tablescan*). Margar töflur eru svo litlar að litlu skiptir hvort notaðir eru vísar til að lesa eftir eða þá að skanna töfluna í heild²⁹.

5.3 Uppbygging vísa

Vísar eru tæknileg útfærsla sem byggir á uppbyggingu taflna og skráa sem gagnasafnskerfið heldur utanum og við getum haft einhver áhrif á en ættum sjálfsagt að skipta okkur lítið af nema við höfum til þess mikla sérfræðipekkingu. Útfærsla vísa getur verið mjög mismunandi eftir því um hvaða gagnasafnskerfi er að ræða og einnig hvernig kerfið er sett upp.



30

Oftast er uppsetning vísa með þeim hætti að þeir eru settir í töflu þar sem annar dálkur færslunnar inniheldur gildi dálksins sem vísað er til og hinn dálkurinn inniheldur **bendi** (e. *pointer*) á sjálfa færsluna sem gildið tilheyrir.³¹

gildi (e. <i>value</i>)	vísir (e. <i>pointer</i>)
--------------------------	----------------------------

²⁹ Gagnagrunnskerfi lesa svokallaðar **síður** (e. *page*) í einu og oft er tafla svo lítil að hún kæmist í eina eða tvær slíkar síður en ef notaðir væru **vísar** (e. *index*) þyrfti fyrst að lesa síðu með vísinum og síðan síður (eða síður) með því sem leitað er að. Oft á tíðum er slíkt óhagkvæmt. Kerfi eins og SQL Server meta það hvort fljótlegra er að fara síðuskönnunarleiðina eða lesa eftir vísunum og taka þannig oft af okkur völdin ef svo má segja.

³⁰ Mynd Frá Birni Þór Jónssyni, með góðfúslegu leyfi

³¹ Athugið að frumlykill er alltaf gerður að vísu, sama á við ef notuð er skilgreiningin UNIQUE. Þetta á við um SQL server 2005, ekki víst að önnur kerfi geri þetta á sama hátt.

5.3.1 Raðaður/klasaður vísir

Vísataflan (*e. index table*) er oft röðuð eftir gildunum og því einfalt að leita í henni. Slík vísatafla er kölluð **raðaður vísir** (*e. ordered index*).

Klasaðir vísar (*e. clustered index*) virka þannig að færslum töflu er raðað í þeirri röð sem vísirinn er. Þetta eru einu tilvikin þar sem gera má ráð fyrir því að færslur gagnasafns séu í einhverri tiltekinni röð. Að sjálfsögðu getur aðeins einn klasaður verið í hverri í töflu því röðun færslanna er þá aðeins með einu móti. Tafla sem röðuð er á þennan hátt er kölluð **klösuð tafla** (*e. clustered table*).

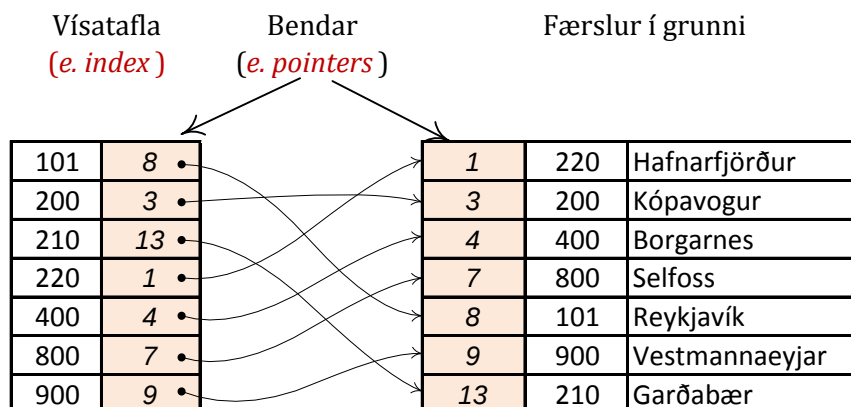
5.3.2 Óklasaður vísir

Tafla sem **ekki** er klösuð tafla hefur þá eiginleika sem töflum eru almennt settar þ.e. röð færslna skiptir ekki máli og þú getur ekki gefið þér neinar forsendur um röðun þeirra. Í flestum tilvikum eru óklasaðar töflur með færslur í **hrúgum** (*e. heap*). Hrúga er óröðuð gagnaskipan, slíkt hentar vel þegar skrá er öll lesin (ekki verið að leita í henni). **Hrúga** vísar til þess að engin röðun á sér stað, nýrri færslu er bætt við aftast.³²

Óklasaðir vísar (*e. nonclustered index*) hafa sjálfstæða gagnaskipan frá töflunni sem þeir tilheyra, þeir eru í raun sérstök tafla. Þessar **vísatöflur** (*e. index tables*) samanstanda af gildi vísisins og síðan **bendi** (*e. pointer*) sem vísar til færslunnar sem geymir innihaldið. Bendirinn í vísafærslunni sem bendir á færsluna með innihaldinu er kallaður **færslustaðgreinir** (*e. row locator*).

Ólíkt lyklum þá þurfa vísar ekki að vera einkvæmir, fleiri en ein færsla getur innihaldið gildi vísisins. Ef fleiri en einn vísir er í töflu þá getur bara einn verið klasaður, hinir óklösuðu vísarnir eru einnig kallaðir **aukavísar** (*e. secondary index*).

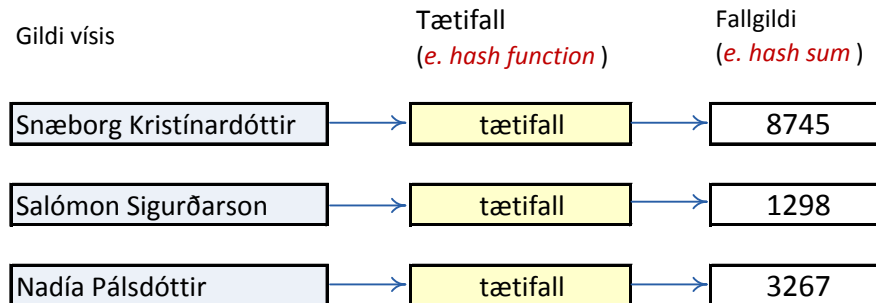
Öll umsýsla um vísa er á valdi gagnasafnskerfisins og okkar hlutverk eingöngu að skilgreina hvort þeir eigi að vera eða ekki.



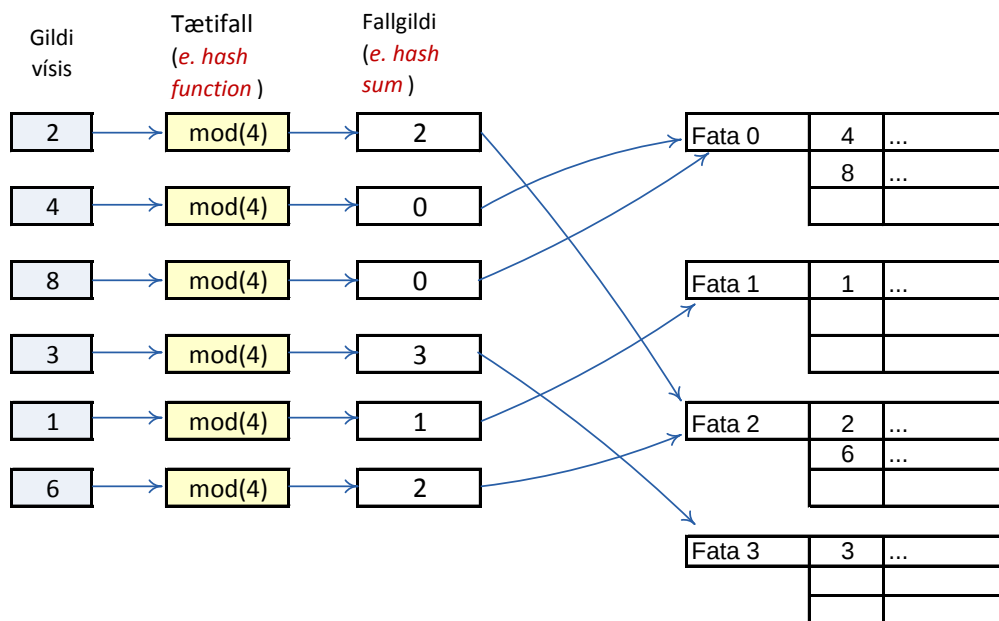
³² Til er gagnaskipan sem er þannig að hrúga er útfærð í tré þar sem um einhverja röðun er að ræða, t.d. þannig að efst í því tré sé hæsta gildið og hnútur neðan við annan hafi alltaf lægra gildi. Þetta getur verið á hinn veginn líka þannig að lægsta gildið sé efst í rótinni.

5.3.3 Tætifallsvísir

Önnur útfærsla á vísu er **tætifallsvísir** (*e. hash index*). Þá er notað eitthvert fall (tætifall, (hash-function), sem er þannig að það varpar gildi vísis í eitthvert annað gildi en niðurstöðugildin eru færri en þau upprunalegu. Á þennan hátt er búin til hálfgerð flokkun þar sem safnað er saman á einn stað þeim gildum/færslum sem hafa sama fallgildi.



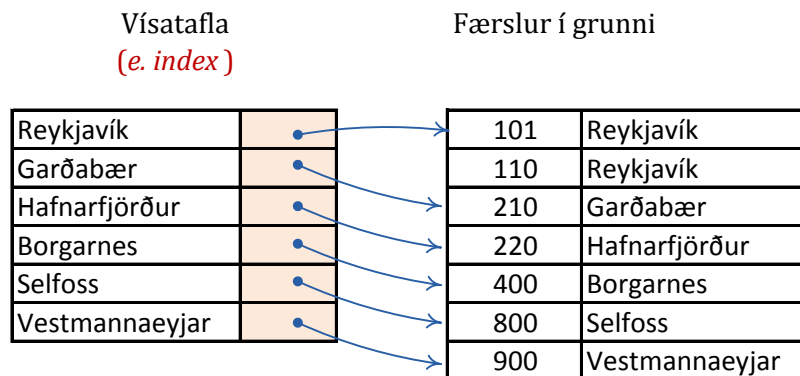
Fallgildið, niðurstaðan úr tæti fallinu, er síðan nýr vísir og vísar á eitthvert safn af færslum (oft kallað **fata** (*e. bucket* eða *bin*)). Í hverri fötu eru oftast fleiri en eitt gildi. Æskilegt er að tæti fallið tryggi góða dreifingu þannig að föturnar fyllist jafnt³³.



³³ Á Netinu má finna upplýsingar um föll til að nota, ágætt er að byrja á Wikipedia:
http://en.wikipedia.org/wiki/Hash_function

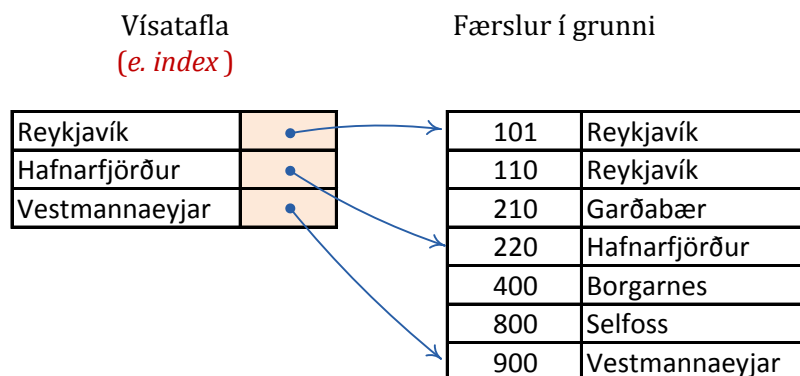
5.3.4 Þéttur vísir

Það er kallaður þéttur vísir ef öll gildi vísis koma fram í vísatöflunni *þéttur vísir* (*e. dense index*). Ef vísirinn er ekki með einkvæmt gildi þá geta fleiri en ein færsla tilheyrt hverju vísisgildi.



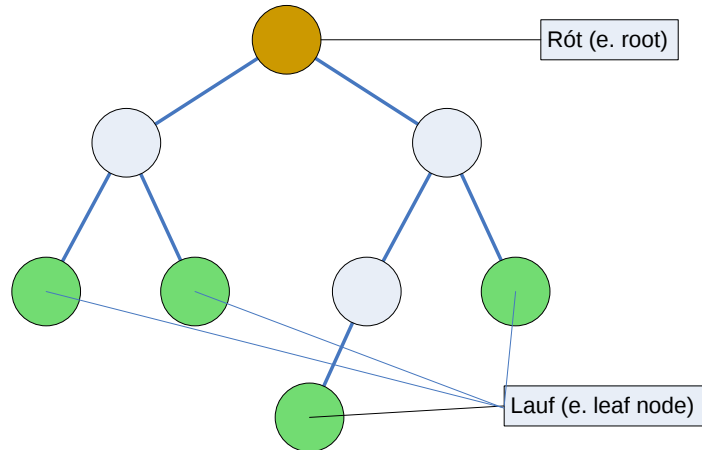
5.3.5 Gisinn vísir

Það er kallaður *gisinn vísir* (*e. sparse index*) ef **ekki** koma öll gildi vísis koma fram í vísatöflunni *þéttur vísir* (*e. dense index*). Ef vísirinn er ekki með einkvæmt gildi þá geta fleiri en ein færsla tilheyrt hverju vísisgildi og leita þarf þeirra gilda sem ekki koma fyrir í vísatöflunni „á milli“ annarra gilda. Þetta fyrirkomulag krefst þess að færslur séu raðaðar eftir gildum vísisins. Ástæða þess að notaður er gisinn vísir er sú að með því má minnka pláss sem nota þarf fyrir vísu og einnig þarf ekki eins oft að bæta við, breyta og eyða úr vísatöflunni.



5.4 Tré

Tré eru notuð til að lágmarka leitartíma í gögn ásamt því að auka hraða í viðhaldi gagna. **Hnútar** (*e. node*) tengjast á **lögum** (*e. level*) frá **rót** (*e. root*) að **laufum** (*e. leaf node*). **Gráða trés** (*e. order*) segir til um það hve margir hnútar geta verið „undir“ hverjum hnút. **Dýpt trés** (*e. tree depth*) (eða hæð) er það hve mörg lög eru í trénu eða nánar sagt hver er mesta fjarlægð frá rót trés að laufi. Dýpt trésins hér á myndinni er 3, ef aðeins er um að ræða rótina og ekkert annað þá er dýptin 0.

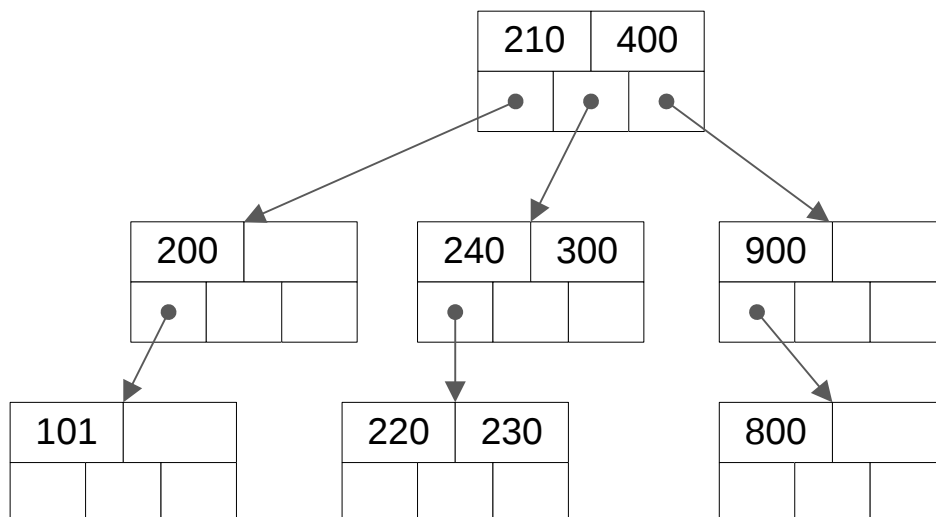


5.4.1 B-tré og B+ tré³⁴

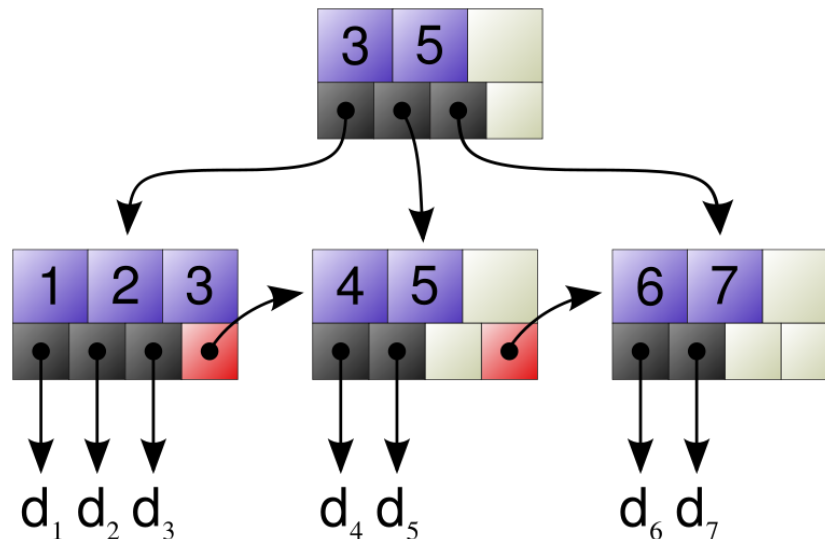
Mörg gagnasafnskerfi byggja á **B+ trjáa gagnaskipan** (*e. B+ structure*), þar á meðal MS SQL Server, og tryggir sú gagnaskipan að jafnaði hraðann og **jafnan leitartíma**.

Til er gagnaskipan sem heitir B-tré

Munur þá þeirri gagnaskipan og B+ tré er sú að gildin (*e. values*) / færslurnar (*e. rows*) eru í B+ tré einungi geymd í laufunum (þ.e. allt annað eru bara bendar eða vísar).



³⁴ B tré eru ýmist kölluð *binary* tré eða *balanced* tré, það er ekkert samkomulag um hvað B-ið þýðir.



5.5 Val á fyrirkomulagi vísa og taflna

Fyrirkomulag vísa og taflna ræðst af eðli taflna, hve mikið eru þær lesnar og hvernig, hve mikið bætist við þær og hver dreifing á lykildum eru (er t.d. nýjum færslum bætt við í fastri númeraröð eða eru gildi vísadálkanna slembin) og svo hvernig færslum er eytt úr töflunni.

5.5.1 Aðferð við val á vísunum

Skóða mikilvægustu fyrirspurnirnar í röð:

1. Í fyrirspurnum nýtast óklasaðir vísar best við val þar sem tiltölulega fáar færslur uppfylla fyrirspurn.
2. Skóða hvort (mun) betra keyrsluplan sé til með nýjum vísi
3. Sumir vísar nýtast fleiri en einni fyrirspurn
4. Vísar flýta fyrirspurnum
5. Vísar geta hægt á breytingum EÐA flýtt fyrir þeim
6. Gera vísi aðeins ef hann kemur að gagni
7. Stundum getur stærra innra minni komið í stað vísa

Viðmið fyrir lyklun og vísun skráa

- Lykla stórar skrár, ef skrár eru litlar getur verið einfaldara að lesa þær í heild
- Frumlykill er sjálfkrafa settur í vísaskrá
- Lykla dálka sem leitað er eftir, sem oft eru notaðir í WHERE skilyrðum
- Dálkar sem koma fyrir í ORDER BY og GROUP BY skipunum
- Þegar gildin eru a.m.k. fleiri en 100 en ekki ef færri en 30. Athugið að þetta eru auðvitað bara þumalputtareglur sem nota má og brjóta eftir þörfum!
- Grunnar geta takmarkað fjölda lyklaskráa
- Stilla lykla-/vísaskráanotkun í hóf vegna skráa sem mikið er breytt eða bætt við því endurraða þarf við breytingar
- Núllgildi eru ekki tekin með í lyklaskrár

Við val á formi þarf að hafa vel í huga hvernig notkunin á að vera. Taflan hér gefur vísbendingar um það hvað hafa þarf í huga miðað við mismunandi fyrirkomulag vísa og taflna.

Gagnaskipulag

	Raðskrá (<i>e. sequential</i>)	Vísuð (<i>e. indexed</i>)	Tætiskrá (<i>e. hashed</i>)
Geymsluspláss	Ekkert til spillis	Ekkert til spillis vegna færslna en krefst rýmis fyrir vísatöflu	Getur þurft að taka frá rými til að mæta nýjum færslum
Lesið í röð eftir frumlykli	Mjög hraðvirkt	Miðlungshraðvirkt	Óhagkvæmt nema notaður sé tætivísir
Slembinn lestur eftir frumlykli	Óhagkvæmt	Miðlungshraðvirkt	Mjög hraðvirkt
Lestur eftir mörgum lyklum	Mögulegt en krefst þess að öll tafla sé lesin	Mjög hraðvirkt ef vísar eru fyrir alla lyklana	Ekki mögulegt nema notaður sé tætivísir
Eyða færslum	Pláss fer til spillis nema töflunni sé endurraðað	Ef kvik stýring er á plássi (e. dynamic allocation), krefst þess að vísar séu uppfærðir	Mjög auðvelt
Bæta við færslum	Krefst þess að töflu sé endurraðað	Ef kvik stýring er á plássi (e. dynamic allocation), krefst þess að vísar séu uppfærðir	Mjög auðvelt
Uppfæra færslur	Krefst þess oft að töflu sé endurraðað	Einfalt en krefst þess að vísar séu uppfærðir	Mjög auðvelt

5.6 Annað til fróðleiks og umhugsunar

Um tætiföll á Wikipedia: http://en.wikipedia.org/wiki/Hash_function

Sjá einnig ýmsan fróðleik á eftirfarandi slóðum:

B-tree. 2.1.2008. <http://en.wikipedia.org/wiki/B-tree>

B+ tree. 2.1.2008. http://en.wikipedia.org/wiki/B%2B_tree

Amiri, Khalil. More on indexing: B+ trees. 2.1.2008.
<http://www.doc.ic.ac.uk/~amiri/advDB/Btrees.pdf>

Estimating the Size of a Clustered Index. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms178085.aspx>

Estimating the Size of a Heap. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms189124.aspx>

Estimating the Size of a Nonclustered Index. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms190620.aspx>

Index Disk Space Example. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms191183.aspx>

Creating Indexes. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms190197.aspx>

General Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms191195.aspx>

Unique Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms187019.aspx>

Nonclustered Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms179325.aspx>

Clustered Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms190639.aspx>

6 Töflualgebra

6.1 Mikilvæg hugtök í kaflanum

aðgerðarstofn (e. *operands*)
aðgerðir (e. *operations*)
dálkaval (e. *projection*)
deiling (e. *division*)
færsluval (e. *selection*)
hlutmengi (e. *subset*)
hreiðraðar segðir (e. *nested expressions*)
hægri ytri töflutenging (e. *right outer join*)
innri töflutenging (e. *inner join*)
jafngildistenging (e. *equijoin*)
mengi (e. *set*)
mengjamargfeldi/krossfeldi (e. *cartesian product*)
mengjamismunur (e. *set difference*)
rökvirðjar (e. *logical operator*)
samanburðarvirðjar (e. *comparison operators*)
sammengi (e. *union*)
sammengishæf (e. *union-compatible*)
segð (e. *expression*)
sniðmengi (e. *intersection*)
stök mengis (e. *elements*)
sundurlæg mengi (e. *disjoint sets*)
takmörkun (e. *restriction*)
töflualgebra (e. *relational algebra*)
töflutenging (e. *join* eða *theta-join*)
valskilyrði (e. *selection predicate*)
velformað (e. *well-formed*)
venjuleg tenging (e. *natural join*)
vinstri ytri töflutenging (e. *left outer join*)
virðjar (e. *operations*)
ytri töflutenging (e. *outer join* eða *full outer join*)

6.2 Grunnaðgerðir töflualgebru

Töflur í töflugagnasafni eru skilgreind sem *mengi* (e. *set*) í skilningi hefðbundinnar algebru og strjállar stærðfræði (e. *discrete mathematics*). Færslurnar er litið á sem *stök mengisins* (e. *elements*). Meðhöndlun þessa mengja köllum við *töflualgebru* (e. *relational algebra*).

6.2.1 Grunnaðgerðir

Í töflualgebru er 5 **grunnaðgerðir**

- **Færsluval** (*e. selection*), táknað með σ (grískt sigma)
- **Dálkaval** (*e. projection*), táknað með π (grískt pí)
- **Mengjamargfeldi/krossfeldi** (*e. cartesian product*), táknað með $\times$
- **Sammengi** (*e. union*), táknað með $\cup$
- **Mengjamismunur** (*e. set difference*), táknað með $-$

6.2.2 Aðgerðir sem leiða má frá grunnaðgerðum

Til viðbótar við fyrrgreindar grunnaðgerðir má síðan leiða aðrar aðgerðir sem henta vel til algebrureikninga, þær sem við notum helst eru:

- **Töflutenging** (*e. join*), táknað með $\bowtie$
- **Sniðmengi** (*e. intersection*), táknað með $\cap$
- **Deiling** (*e. division*), táknað með $\div$

6.2.3 Virkjar og segðir

Í „venjulegri“ stærðfræði höfum við **aðgerðir** eða **virkja** (*e. operations*) fyrir tölur:

$+, -, *, /$ o.s.frv.

Í töflualgebru höfum við líka virkja³⁵ en **þær vinna á töflum** og **útkoman er ný tafla**. Niðurstaðan er tafla sem oftast er **ekki varanleg**, taflan varðveitist ekki eftir aðgerð nema sérstakar ráðstafanir séu gerðar til þess. Virkjar töflualgebru hafa ekki áhrif á upprunalegu töflurnar frekar en aðgerðir gera í venjulegri stærðfræði.

Við tölum um **segð** (*e. expression*) sem safn af tölum, breytum, virkjum og táknum til að hópa saman hluta segðarinnar. Segðir er síðan hægt að meta eða reikna og fá niðurstöðu. Segðirnar verða að lúta lögmálum málfræði sem um útreikningana eða matið gildir, talað er um að slíkt sé **velformað** (*e. well-formed*) og er þá t.d. átt við að $2+3$ sé velformuð segð en $*2+$ sé það ekki.

Í töflualgebru er **aðgerðarstofn** (*e. operands*) eru töflur sá hluti segðarinnar sem við köllum breytur. Niðurstöður eru einnig töflur og því má nota niðurstöðu einnar aðgerðar sem aðgerðarstofn annarrar aðgerðar. Segðir geta því verið **hreiðraðar segðir** (*e. nested expressions*) eins og við venjumst í stærðfræði eða í forritun þar sem föll eru inni í föllum. Þetta er kölluð **lokun** (*e. closure*)

Dæmi úr hefðbundinni forritun:

```
X = int ( avg ( listiA ) )
```

³⁵ Til að greina þessar aðgerðir frá sérstökum töfluaðgerðum er hér notað orðið **virkjar** í stað aðgerða

6.3 Færsluval

Mismunandi nöfn eru notuð fyrir þetta, s.s. val, **valskilyrði** (e. *selection predicate*), færsluval, **takmörkun** (e. *restriction*) eða SELECT.

Notum hugtakið **færsluval** (e. *selection*), táknað með σ , sigma.

σ

$\sigma_{\text{valskilyrði}}(R)$

- Velur færslur úr töflu R sem uppfylla **valskilyrði** eða **takmörkun**. Valskilyrði er sett sem lágvísir (e. Færsluval (e. selection) subscript) eftir σ -tákninu.
- Niðurstaða úr aðgerðinni er ný tafla sem inniheldur aðeins þær færslur sem uppfylla valskilyrði.
- Engar tvítekningar eru í niðurstöðutöflu³⁶
- Niðurstöðutafla getur verið notuð sem inntak í aðra aðgerð (virkja)

Þetta er í raun sama og í SQL er sagt með:

```
select * from R where <valskilyrði>
```

6.3.1 Formleg framsetning á færsluvali

Með formlegri framsetningu getum við skilgreint færsluval á eftirfarandi hátt:

$$\sigma_p(r) = \{t \mid t \in r \text{ og } p(t)\}$$

Þar sem p er valskilyrði sem er samsett af þáttum sem greina má í sundur með rökvirkjunum:

$\wedge$ (and), $\vee$ (or), $\neg$ (not)

6.3.2 Rökvirkjar í færsluvali

Við getum í töflualgebru notað hefðbundnu **rökvirkjana** (e. *logical operator*) **AND**, **OR**, **NOT** og einnig **samanburðarvirkjana** (e. *comparison operators*) $=$, $<$, $\leq$, $>$, $\geq$, $\neq$

Samanburður í færsluvali er með eftirfarandi hætti:

<dálkur> <samanburðarvirki> <dálkur>

eða

<dálkur> <samanburðarvirki> <fasti>

Dæmi:

$\sigma_{\text{árgangur} = '02'}(\text{nemendur})$

$\sigma_{(\text{undanfari} = 'T202') \text{ OR } (\text{undanfari} = 'T102')}(\text{námskeið})$

³⁶ Flest gagngrunnskerfi vinna þannig að þau **sleppa ekki tvíteknum** færslum nema sérstaklega sé um það beðið

6.3.3 Dæmi um færsluval³⁷

Veldu allar færslur þar sem verð vöru er meira en 500 IKR

```
select *
from [Stofnskrá vöru]
where verðIkr > 500
```

	vöruNr	heitiVöru	kaupMynt	upphæðiKaupmynt	dagsSíðustulnnkaupa	verðIkr
1	3	Hnakkur	SEK	2500.00	20050201	24300.00
2	4	Beisli	SEK	103.00	20050201	1001.16
3	7	CD-tómir 100 stk	GBP	4.00	20050201	505.72
4	9	Taumur - leður	NOK	62.00	20041201	669.60
5	10	Hjálmur	NOK	435.00	20041201	4698.00
6	11	Buxur í stærð 32	SEK	1000.00	20060208	9720.00
7	12	Buxur í stærð 30	SEK	970.00	20040901	9428.40
8	13	Buxur í stærð 28	SEK	1000.00	20040901	9720.00

Annað dæmi um færsluval

Veldu alla starfsmenn sem búa utan Reykjavíkur (póstnúmer > 155), táknað á töflu-algebru sem:

$\sigma_{\text{pnr} > 155}$ (Starfsmenn)

Í SQL er sagt:

```
select * from Starfsmenn where pnr > 155
```

Upprunataflan er:

Starfsmenn	nr.	nafn	staða	kyn	kennitala	deild	heimilisfang	pnr
	1	Sveinn Baldursson	lektor	2	121212-9999	B01	Kvos 1	101
	2	Baldur Sveinsson	doktor	2	111111-9999	C01	Áleiti 3	107
	3	Brynja Jónsdóttir	traktor	1	101010-9999	B02	Heykvísl 2	111
	12	Jón Brynjarsson	gjaldkeri	2	090909-9999	B02	Frábær 4	111
	5	Anna Pálsdóttir	prófessor	1	080808-9999	B02	Óshólmi 5	200
	35	Elín Aradóttir	rektor	1	011001-9999	A01	Marflöt 7	210

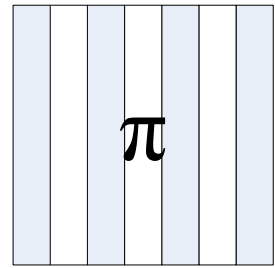
Niðurstaðan er:

5	Anna Pálsdóttir	prófessor	1	080808-9999	B02	Óshólmi 5	200
35	Elín Aradóttir	rektor	1	011001-9999	A01	Marflöt 7	210

³⁷ Skilgreining á töflum sem notaðar eru í dæmum kaflans og innihald þeirra er aftast í þessum kafla.

6.4 Dálkaval

Dálkaval (e. *projection*) gerir okkur kleift að velja tiltekna dálka töflu og nota aðeins gildi þeirra í aðgerðum. Dálkaval er táknað með π (grískt pí).



Dálkaval (e. projection)

Dálkaval $\Pi_{\text{dálkur-1}, \dots, \text{dálkur-n}} (R)$

- Vinnur með töfluna R og skilar niðurstöðutöflu þar sem einungis inniheldur dálkana sem tilgreindir eru (hinum sleppt), dálkarnir í niðurstöðutöflu heita sömu nöfnum og í upprunatöflunni.
- Fjarlægir tvíteknar færslur, þ.e. þær færslur sem eru eins eftir að dálkar hafa verið fjarlægðir³⁸

Þetta er í raun sama og í SQL er sagt með:

```
select dálkur-1, ..., dálkur-n from R
```

6.4.1 Dæmi um dálkaval

Veldu dálkana vöruNr, heitiVöru og verðiKr og þær færslur þar sem verðiKr er stærra en 500.

```
select vöruNr, heitiVöru, verðiKr
from [Stofnskrá vöru]
where verðiKr > 500
```

	vöruNr	heitiVöru	verðiKr
1	3	Hnakkur	24300.00
2	4	Beisli	1001.16
3	7	CD-tómur 100 stk	505.72
4	9	Taumur - leður	669.60
5	10	Hjálmur	4698.00
6	11	Buxur í stærð 32	9720.00
7	12	Buxur í stærð 30	9428.40
8	13	Buxur í stærð 28	9720.00

Annað dæmi um dálkaval

Veldu lista af starfsmönnum þar sem aðeins koma fram starfsmannanúmer, nafn, deild og póstnúmer, táknað á töflualgebru sem:

$\Pi_{\text{nr, nafn, deild, pnr}} (\text{Starfsmenn})$

Í SQL er sagt:

```
select nr, nafn, deild, pnr from Starfsmenn
```

³⁸ flest gagngrunnskerfi vinna þannig að þau **sleppa ekki tvíteknum** færslum nema sérstaklega sé um það beðið. Hægt er að nota `select distinct ...` í SQL Server

Upprunataflan er:

Starfsmenn	nr.	nafn	staða	kyn	kennitala	deild	heimilisfang	pnr
	1	Sveinn Baldursson	lektor	2	121212-9999	B01	Kvos 1	101
	2	Baldur Sveinsson	doktor	2	111111-9999	C01	Áleiti 3	107
	3	Brynja Jónsdóttir	traktor	1	101010-9999	B02	Heykvísl 2	111
	12	Jón Brynjarsson	gjaldkeri	2	090909-9999	B02	Frábær 4	111
	5	Anna Pálsdóttir	prófessor	1	080808-9999	B02	Óshólmi 5	200
	35	Elín Aradóttir	rektor	1	011001-9999	A01	Marflöt 7	210

Niðurstaðan er:

nr.	nafn	deild	pnr
1	Sveinn Baldursson	B01	101
2	Baldur Sveinsson	C01	107
3	Brynja Jónsdóttir	B02	111
12	Jón Brynjarsson	B02	111
5	Anna Pálsdóttir	B02	200
35	Elín Aradóttir	A01	210

6.5 Mengjamargfeldi / krossfeldi

Mengjamargfeldi / krossfeldi (*e. cartesian product*) gerir okkur kleift að tengja hverja einustu færslu í einni töflu við hverja einustu færslu í annarri. Krossfeldi er táknað með **X**.

P	Q	P x Q
a	1	a 1
b	2	a 2
	3	a 3
		b 1
		b 2
		b 3

Mengjamargfeldi P og Q, P X Q

- Skilgreinir töflu þar sem *sérhver færsla í P* er tengd við *sérhverja færslu í Q*.
- Töflur þurfa *ekki að vera sammengishæfar*
- **Fjöldi dálka** í niðurstöðutöflu er samanlagður fjöldi dálka í P og Q
- **Fjöldi færslna** í niðurstöðutöflu er margfeldi færslufjölda P og Q, ef Q hefur 20 færslur og P hefur 40, þá hefur P * Q 800 færslur!
- Engar færslur eru sameinaðar eða sleppt úr
- Getur verið **þungt/dýrt að reikna krossfeldi** stórra taflna
- Ef dálkar heita sama nafni í töflunum þarf að skeyta töflunafni fyrir framan dálkaheiti eða uppnefna á annan hátt³⁹

Formlega getum við skilgreint mengjamargfeldi sem:

$$A \times B = \{t \mid t \in A \text{ and } t \in B\}$$

Í SQL er sagt:

```
select * from A, B
eða
select * from A
cross join B
```

³⁹ Í SQL Server má nota „alias“ til þess að uppnefna

6.5.1 Mengjamargfeldi og færsluval

Algengt er að setja skilyrði sem takmarka færslufjöldann í notkun mengjamargfeldis, því þau geta orðið þung í keyrslu á stórum töflum. Algengast er að nota **töflutengingar** í stað mengjamargfeldis og færsluvals vegna þess að þar er búið að sameina þetta í eina aðgerð. Sjá nánar um töflutengingar hér síðar í kaflanum.

6.5.2 Dæmi um mengjamargfeldi

Venslum alla viðskiptavini í viðskiptamannatöflu við allar vörur í stofnskrá vöru.

Viðskiptamenn x Stofnskrá vöru

Í SQL er sagt:

```
select heiti, heitiVöru
from Viðskiptamenn
cross join [Stofnskrá vöru]
```

Niðurstaðan er hér til hliðar:⁴⁰

	heiti	heitiVöru
1	Jón Freyr Jóhannsson	Hnakkur
2	Edda Stefánía Jónsdóttir	Hnakkur
3	Inga Birna Jónsdóttir	Hnakkur
4	Elísabet Erla Birgisdóttir	Hnakkur
5	Birgir Ívar Pétursson	Hnakkur
6	Ævar Már Óskarsson	Hnakkur
7	Una Björg Magnúsdóttir	Hnakkur
8	Einar Búi Magnússon	Hnakkur
9	María Ólafsdóttir	Hnakkur
10	Rökkvi Toppsson	Hnakkur
11	Jón Freyr Jóhannsson	Beisli
12	Edda Stefánía Jónsdóttir	Beisli
13	Inga Birna Jónsdóttir	Beisli
14	Elísabet Erla Birgisdóttir	Beisli
15	Birgir Ívar Pétursson	Beisli

Annað dæmi um mengjamargfeldi

Veldu lista yfir alla kennara og tengdu þá við allar nemendur, táknað á töflualgebru sem:

$$(\Pi_{nr, nafn, deild, pnr} (Kennarar)) \times (\Pi_{nr, nafn} (Nemendur))$$

Í SQL er sagt: ⁴¹

```
select Kennarar.nr,
       Kennarar.nafn, pnr
from Kennarar
cross join Nemendur
```

Kennarar	nr.	nafn	deild	pnr
	1	Sveinn Baldursson	B01	101
	5	Anna Pálsdóttir	B02	200
	35	Elín Aradóttir	A01	210

Upprunatöflur eru hér til hliðar:

Nemendur	nr.	nafn
	1	Kári Hansson
	4	Edda Einarsdóttir

Niðurstaðan er hér fyrir neðan:

nr.	nafn	deild	pnr	nr.	nafn
1	Sveinn Baldursson	B01	101	1	Kári Hansson
1	Sveinn Baldursson	B01	101	4	Edda Einarsdóttir
5	Anna Pálsdóttir	B02	200	1	Kári Hansson
5	Anna Pálsdóttir	B02	200	4	Edda Einarsdóttir
35	Elín Aradóttir	A01	210	1	Kári Hansson
35	Elín Aradóttir	A01	210	4	Edda Einarsdóttir

⁴⁰ Upprunatöflur eru sýndar aftast í kafla

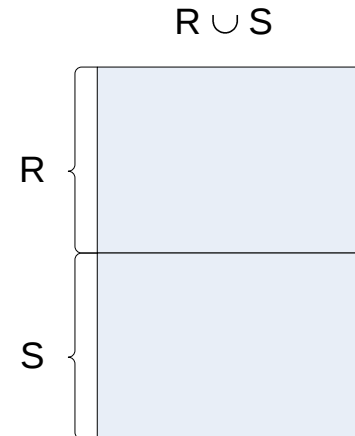
⁴¹ Athuga þarf að ef nöfn dálka eru sömu í báðum töflum þarf að uppnefna eða einkenna úr hvaða töflu dálkarnir koma

6.6 Sammengi

Sammengi (*e. union*) gera okkur kleift að velja allar færslur tiltekinna taflna. Sammengi er táknað með $\cup$ á milli taflnanna sem velja á saman.

Sammengi R og S, $R \cup S$

- **Sammengi** (*e. union*) tveggja taflna R og S skilgreinir töflu sem inniheldur allar færslur R, eða S eða bæði R og S.
- Tvíteknar færslur eru fjarlægðar.
- R og S verða að vera **sammengishæfar** (*e. union-compatible*)⁴².



Stærð niðurstöðutafla sammengis getur verið allt að:

$$\text{færslufjöldi}(R) + \text{færslufjöldi}(S)$$

Þetta á við ef mengin eru **sundurlæg mengi** (*e. disjoint sets*).

Og minnst getur niðurstöðutaflan verið:

$$\max(\text{færslufjöldi}(R), \text{færslufjöldi}(S))$$

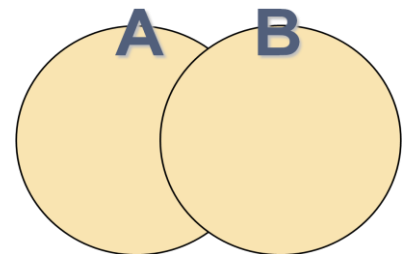
Það á við ef annað mengið er **hlutmengi** (*e. subset*) hins mengisins.

Formlega getum við skilgreint sammengi sem:

$$A \cup B = \{t \mid t \in A \text{ eða } t \in B\}$$

Í SQL er sagt:

```
select * from A
union
select * from B
```



Þetta skilar niðurstöðum beggja fyrirspurna **eftir að fjarlægðar hafa verið allar tvítekningar**.

6.6.1 Dæmi um sammengi

Veldu vöruNr og fjölda úr töflunni PöntunarLínur og vöruNr og magnALager úr töflunni Lagerfærslur, táknað á töflualgebru sem:

$$\begin{aligned} &\Pi_{\text{vöruNr}, \text{fjöldi}}(\text{PöntunarLínur}) \\ &\cup \\ &\Pi_{\text{vöruNr}, \text{magnALager}}(\text{Lagerfærslur}) \end{aligned}$$

⁴² Sjá hér á eftir um sammengishæfni

Í SQL er sagt:

```
select vöruNr, fjöldi from Pöntunarlínur
union
select vöruNr, magnALager from Lagerfærslur
```

Niðurstaðan er hér til hliðar:⁴³

	vöruNr	fjöldi
1	1	0
2	1	1
3	1	2
4	1	12
5	3	3
6	3	4
7	5	3
8	5	6
9	5	144
10	6	-3
11	6	2
12	6	3

Annað dæmi um sammengi

Veldu lista yfir öll póstnúmer þar sem við erum með annað hvort leiguhúsnæði eða eigin húsnæði táknað á töflualgebru sem:

$$\Pi_{\text{pnr}}(\text{Eiginhúsnæði}) \cup \Pi_{\text{pnr}}(\text{Leiguhúsnæði})$$

Í SQL er sagt:

```
select pnr from Eiginhúsnæði
union
select pnr from Leiguhúsnæði
```

Niðurstaðan er hér til hliðar:⁴⁴

pnr
103
107
111
210

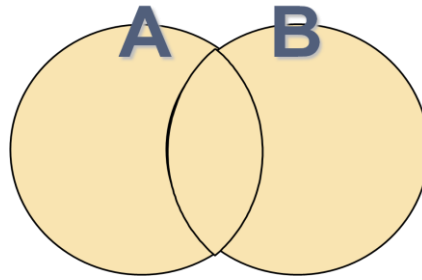
⁴³ Upprunatöflur eru sýndar aftast í kafla

⁴⁴ Upprunatöflur eru sýndar aftast í kafla

6.6.2 Union all

Með því að nota UNION ALL eru endurtekningar teknar með:

```
select * from A
union all
select * from B
```



Þetta skilar niðurstöðum beggja fyrirspurna *með öllum endurtekningu (á sömu pörum gilda)*.

Ef notað er UNION ALL á sömu töflur og í síðasta dæmi þá verður niðurstaðan þessi:

```
select vöruNr, fjöldi
  from Pöntunarlínur
union all
select vöruNr, magnALager
  from Lagerfærslur
```

	vöruNr	fjöldi
1	5	6
2	6	2
3	7	1
4	8	1
5	9	14
6	10	1
7	1	2
8	5	3
9	6	4
10	7	1
11	8	2
12	9	1
13	10	1

24	7	1
25	8	2
26	1	12
27	5	144
28	6	12

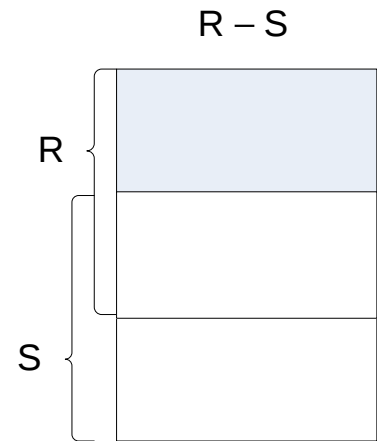
6.6.3 Sammengishæfni

Til að töflur séu *sammengishæfar* (e. *union-compatible*) þarf eftirfarandi að gilda:

- Sami fjöldi dálka í báðum töflum
- *Skilgreiningarmengi* (e. *domain*) dálkanna verður að vera það sama, í þeirri röð sem þeir eru, í báðum töflunum.
- Skilyrðin eiga við þá dálka sem eru valdir til að taka þátt í aðgerðinni (dálkavalið), hinar eiginlegu töflur geta haft aðra dálka.

6.7 Mengjamismunur

Mengjamismunur (e. *set difference*), gerir okkur kleift að velja aðeins þær færslur sem tilheyra einni töflu (tilteknu skilyrði) en sem eru ekki í tiltekinni annarri töflu. Það má einnig segja það sem svo að við getum valið færslur sem koma úr einni fyrirspurn að frádregnum þeim færslum sem koma úr annarri (þetta byggir auðvitað á því að skilgreina hvaða dálkar og gildi það eru sem skilja á milli). Mengjamismunur er táknaður með —

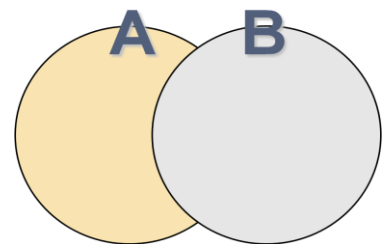


Mengjamismunur R og S, $R - S$

- Skilgreindir töflu þar sem eru allar færslur í R nema þær sem einnig tilheyra S
- $R \text{ og } S$ verða að vera *sammengishæfar*⁴⁵

Formlega getum við skilgreint mengjamismun sem:

$$A - B = \{t \mid t \in A \text{ og } t \notin B\}$$



Í SQL er sagt:

```
select * from A
except
select * from B
```

Þetta skilar þeim færslum fyrri fyrirspurnar sem **ekki koma fyrir** í þeirri seinni.

6.7.1 Dæmi um mengjamismun

```
select vöruNr from [Stofnskrá vöru]
except
select vöruNr from Lagerfærslur
```

	vöruNr
1	4
2	11
3	12
4	13
5	14

Niðurstaðan er hér til hliðar:⁴⁶

Annað dæmi um mengjamismun

Veldu lista yfir öll póstnúmer þar sem við erum með eigin húsnæði en þar sem ekki er leiguhúsnæði, táknað á töflualgebru sem:

$$\Pi_{\text{pnr}} (\text{Eiginhúsnæði}) - \Pi_{\text{pnr}} (\text{Leiguhúsnæði})$$

⁴⁵ Sjá skilgreiningu á Sammengishæfni hér framar

⁴⁶ Upprunatöflur eru sýndar aftast í kafla

Í SQL er sagt:

```
select pnr from Eiginhúsnæði
except
select pnr from Leiguhúsnæði
```

Niðurstaðan er hér til hliðar:

pnr
103
111

6.8 Sniðmengi

Sniðmengi (e. *intersection*), gerir okkur kleift að velja aðeins þær færslur sem tilheyra báðum mengjum, færslum sem tilheyra aðeins öðru menginu er sleppt. **Sniðmengi** er táknað með $\cap$.

Sniðmengi R og S, $R \cap S$

- Skilgreinir töflu þar sem eru allar færslur sem tilheyra bæði R og S
- R og S verða að vera sammengishæfar⁴⁷

Hægt er að tákna með sniðmengi með grunnaðgerðum:

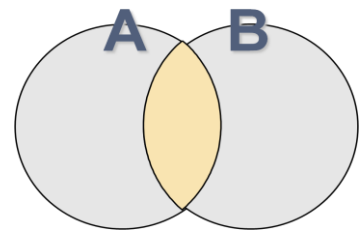
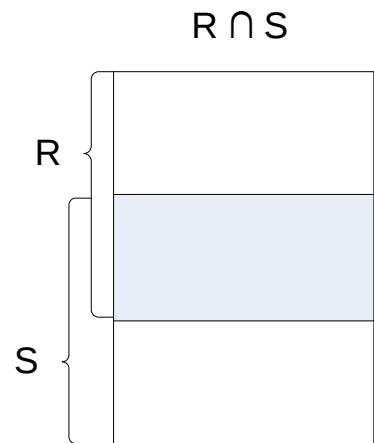
$$R \cap S = R - (R - S)$$

Formlega getum við skilgreint sniðmengi sem:

$$A \cap B = \{ t \mid t \in A \text{ og } t \in B \}$$

Í SQL er sagt:

```
select * from A
intersect
select * from B
```



Þetta skilar færslum sem eru sameiginlegar báðum fyrirspurnum, en sleppir þeim sem tilheyra aðeins annarri þeirra.

⁴⁷ Sjá skilgreiningu á Sniðmengishæfni hér framar

6.8.1 Dæmi um sniðmengi

Veldu þær vörur sem hafa sama vörunúmer og sama fjölda í töflunum Pöntunarlínur og Lagerfærslur, táknað á töflualgebru sem:

$$\Pi_{\text{vöruNr, fjöldi}}(\text{Pöntunarlínur}) \cap \Pi_{\text{vöruNr, magnALager}}(\text{Lagerfærslur})$$

Í SQL er sagt:

```
select vöruNr, fjöldi from Pöntunarlínur
intersect
select vöruNr, magnALager from Lagerfærslur
```

Niðurstaðan er hér til hliðar:

	vöruNr	fjöldi
1	1	2
2	5	3
3	5	6
4	6	2
5	7	1
6	8	1
7	8	2
8	9	1
9	9	14

Annað dæmi um sniðmengi

Veldu lista yfir öll póstnúmer þar sem er eigið húsnæði og einnig a.m.k. eitt leiguhúsnæði, táknað á töflualgebru sem:

$$\Pi_{\text{pnr}}(\text{Eiginhúsnæði}) \cap \Pi_{\text{pnr}}(\text{Leiguhúsnæði})$$

Í SQL er sagt:

```
select pnr from Eiginhúsnæði
intersect
select pnr from Leiguhúsnæði
```

Niðurstaðan er hér til hliðar:

pnr
107
210

6.9 Töflutenging

Töflutenging (*e. join*) gerir okkur kleift að tengja saman færslur úr fleiri en einni töflu og velja hvaða dálka við notum til að tengja saman með. Nota má töflutengingu til að velja saman færslur t.d. úr nemendatöflu (þar sem póstnúmer er að finna) við póstnúmer í póstnúmeratöflu og fá þannig fram bæði upplýsingar um nemendur og nafn póststöðvar. Töflutengingin fer því fram með því að tengja saman sameiginlega dálka, eða þá sem eru sambærilegir. Töflutenging er táknud með $\bowtie$.

T		U		T $\bowtie$ U		
A	B	B	C	A	B	C
a	1	1	x	a	1	x
b	2	1	y	a	1	y
		3	z			

Töflutenging töflunnar T og töflunnar U, með skilyrðinu F , $T \bowtie_F U$:

- Skilgreindir töflu sem er krossfeldi T og U og þar sem færslur uppfylla valskilyrði F
- Valskilyrðið F er á forminu $T.a_i \theta U.b_i$ þar sem θ getur verið einn af samanburðarvirkjunum ($<, \leq, >, \geq, =, \neq$). θ er gríski bókstafurinn theta (þeta).
- Það er líka hægt að skrifa þetta á öðru formi, sem krossmargfeldi með færsluvali:

$$T \bowtie_F U = \sigma_F(T \times U)$$
- Ef skilyrðið F er einungis $=$ þá er um að ræða jafngildistengingu (*e. equijoin*)

Það eru til nokkrar útgáfur af töflutengingum⁴⁸, t.d. :

- **Töflutenging** (*e. join* eða *theta-join*) tengir saman færslur með gildum í sameiginlegum dálkum.
- **Jafngildistenging** (*e. equijoin*) er sértilvik töflutengingar, þ.e. þegar samanburðurinn er með $=$.
- **Venjuleg tenging** (*e. natural join*) sem er sértilvik jafngildistengingar.
- Töflutengingarnar theta-join, jafngildistenging og venjuleg tenging eru sameiginlegar kallaðar **innri töflutengingar** (*e. inner join*), virkar allar eins og sniðmengi.
- Innri töflutengingar sýna ekki færslur sem hafa NULL-gildi í sameiginlega dálkinum (samanburðardálkinum).
- **Ytri töflutenging** (*e. outer join*) tekur allar færslur með hvort þær hafa NULL-gildi í samanburðardálki eða ekki. Þrjár tegundir eru af ytri töflutengingu:
 - **Vinstri ytri töflutenging** (*e. left outer join*) tekur allar færslur úr fyrri töflu sem er nefnd (kölluð vinstri) og setur í niðurstöður, hvort sem það finnst samsvörun eða ekki í hægri töflunni, NULL-gildi sett til uppfyllingar
 - **Hægri ytri töflutenging** (*e. right outer join*) er spegilmynd vinstri ytri töflutengingar, sama og *left join* nema víxlað töflum miðað við *left join*
 - **Ytri töflutenging** (*e. [full] outer join*), tekin eru öll gildi bæði vinstra megin og hægri, NULL-gildi sett til uppfyllingar

Töflutenging:

- Jafngildir því að vera með færsluval með krossfeldi
- Þessi aðgerð gerir miklar kröfur til gagnagrunnskerfa og *röng* notkun töflutenginga getur haft veruleg áhrif á afköst gagnagrunna
- Flest gagnagrunnskerfi geta *bestað* fyrirspurnir þannig að valin er besta keyrsluleið, þetta atriði er því farið að skipta minna máli en áður

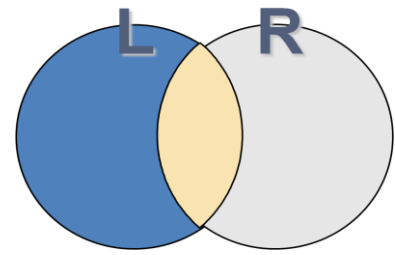
⁴⁸ sjá t.d. á: http://en.wikipedia.org/wiki/Relational_algebra

6.9.1 Vinstri ytri töflutenging

Allar færslur úr fyrri (vinstri) töflunni líka þær sem hafa ekki samsvarandi gildi í þeirri seinni (hægri)

Í SQL er sagt:

```
select L.vöruNr, magnALager
  from [Stofnskrá vöru] as L
left join
Lagerfærslur as R
  on R.vöruNr = L.vöruNr
```



Niðurstaðan er hér til hliðar:

	vöruNr	magnALager
1	1	12
2	1	2
3	1	0
4	2	NULL
5	3	3
6	3	4
7	4	NULL
8	5	144

Annað dæmi þar sem Z er niðurstöðutaflan úr töflutengingu:

```
Z: select X.D1, Y.D3
  from X
left join Y
  on X.D1 = Y.D2
```

Tafla X					Tafla Y		Tafla Z	
		D1			D2	D3		
		2			1	A	2	B
		3			2	B	3	C
		1			3	C	1	A
		5			4	D	5	NULL
		6			7	G	6	NULL
		15			11	K	15	O
		8			12	L	8	H
					13	M		
					14	N		
					15	O		
					21	U		

Annað dæmi um outer join

Í SQL er sagt:

```
SELECT heitiVöru as [Heiti vöru],
       sum(magnALager) as [Magn á lager]
FROM [Stofnskrá vöru] L
left outer join
Lagerfærslur R
  on L.vöruNr = R.vöruNr
GROUP BY heitiVöru
```

Niðurstaðan er hér til hliðar:

	Heiti vöru	Magn á lager
1	Beisli	NULL
2	Buxur í stærð 28	NULL
3	Buxur í stærð 30	NULL
4	Buxur í stærð 32	NULL
5	Camenbert	NULL
6	CD-tómur 100 stk	3
7	DVD-tómur 10 stk	3
8	Hjálmur	0
9	Hnakkur	7
10	Kók	14
11	Pepsi	11
12	Snickers 6pack	153
13	Taumur - leður	15

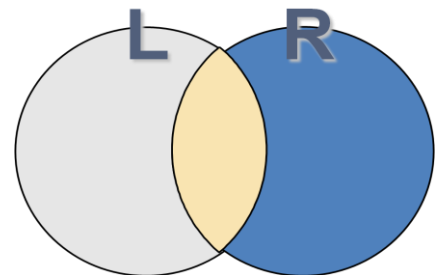
6.9.2 Hægri ytri töflutenging

Allar færslur úr seinni (hægri) töflunni líka þær sem hafa ekki samsvarandi gildi í þeirri seinni (vinstri).

Í SQL er sagt:

```
select L.vöruNr, magnALager
  from [Stofnskrá vöru] as L
right join
Lagerfærslur as R
  on R.vöruNr = L.vöruNr
```

Niðurstaðan er hér til hliðar:



	vöruNr	magnALager
1	1	12
2	5	144
3	6	12
4	7	2
5	8	0
6	9	0
7	10	-1
8	1	2
9	NULL	4
10	3	3

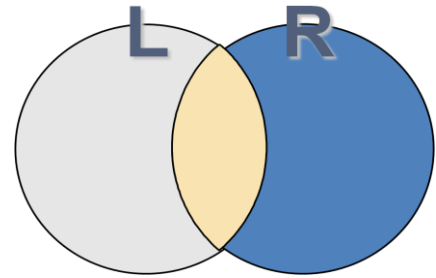
6.9.3 Innri töflutenging

Í innri töflutengingu er í raun verið að vinna með sniðmengi.

Hér er verið að sækja heiti vöru í stofnskrá svo hægt sé að birta nafn í stað númers sem er í lagerfærslu.

Í SQL er sagt:

```
select heitiVöru as [Heiti vöru],
       sum(magnALager) as [Magn á
       lager]
from [Stofnskrá vöru] S, Lagerfærslur L
     where S.vöruNr = L.vöruNr
group by heitiVöru
```



Hér er það upptalning á töflum í FROM hlutanum sem kemur í stað þess að setja INNER JOIN milli taflanna.

Niðurstaðan er hér til hliðar:

	Heiti vöru	Magn á lager
1	Taumur - leður	15
2	Snickers 6pack	153
3	Pepsi	11
4	Kók	14
5	Hnakkur	7
6	Hjálmur	0
7	DVD-tómir 10 stk	3
8	CD-tómir 100 stk	3

Annað dæmi um innri töflutengingu

```
Z: select X.D1, Y.D3
from X
     inner join Y
     on X.D1 = Y.D2
```

Tafla X		
	D1	
	2	
	3	
	1	
	5	
	6	
	15	
	8	

Tafla Y	
D2	D3
1	A
2	B
3	C
4	D
7	G
11	K
12	L
13	M
14	N
15	O
21	U

Tafla Z	
2	B
3	C
1	A
15	O
8	H

6.10 Töfludeiling

Deiling (*e. division*) gerir okkur kleift að tengja saman færslur þannig að birtar eru þær færslur úr vinstri töflunni (þeirri sem fyrst er nefnd) sem **venslast við allar** færslur hinnar töflunnar. Dæmi um slíkt væri að velja þá nemendur sem hafa tekið öll námskeið tiltekins kennara⁴⁹. Deiling er táknuð með $\div$.

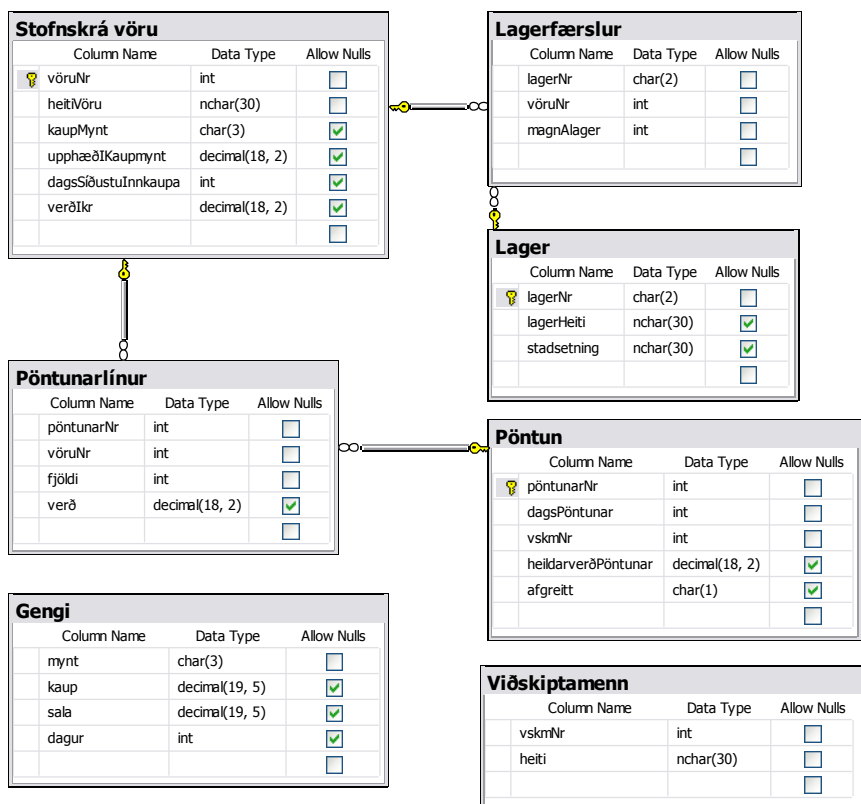
V		W	$V \div W$
A	B	B	A
a	1	1	a
a	2	2	b
b	1		
b	2		
c	1		

Töfludeiling er ekki framkvæmd með einni skipun eða setningu í SQL. Hún er framkvæmd með samsetningu fleiri en einnar setningar. Sjá dæmi á yfirlitinu yfir töflualgebraðgerðir hér á eftir. Ef það er ekki nóg skaltu leita á Internetinu.

⁴⁹ Einhvers staðar var þetta sagt: Suited to queries that include the phrase “for all”.

6.11 Færslur í töflum dæma

Töflulíkan vöru-, pantana- og birgðakerfis



Stofnskrá vöru

	vöruNr	heitiVöru	kaupMynt	upphæðIKaupmynt	dagsSíðustuInnkaupa...	verðIkr
1	1	Kók	USD	3.00	20050208	286.97
2	2	Camenbert	EUR	2.00	20050208	244.23
3	3	Hnakkur	SEK	2500.00	20050201	33187.50
4	4	Beisli	SEK	103.00	20050201	1367.33
5	5	Snickers 6pack	DKK	24.00	20050208	393.84
6	6	Pepsi	USD	2.93	20050201	271.61
7	7	CD-tómir 100 stk	GBP	4.00	20050201	697.56
8	8	DVD-tómir 10 stk	GBP	3.50	20041201	654.42
9	9	Taumur - leður	NOK	62.00	20041201	985.05
10	10	Hjálmur	NOK	435.00	20041201	6911.28
11	11	Buxur í stærð 32	SEK	1000.00	20060208	13440.00
12	12	Buxur í stærð 30	SEK	970.00	20040901	14061.12
13	13	Buxur í stærð 28	SEK	1000.00	20040901	14496.00
14	14	Úlpa	SEK	100.00	20050208	1344.00

Lager

	lagerNr	lagerHeiti	stadsetning
1	01	Aðallager	Fagrihvammur 2b
2	02	Hesthús	Kaplaskeiði 15
3	03	HR-lager	Ofanleiti

Viðskiptamenn

	vskmNr	heiti
1	1	Jón Freyr Jóhannsson
2	2	Edda Stefánía Jónsdóttir
3	3	Inga Birna Jónsdóttir
4	4	Elísabet Erla Birgisdóttir
5	5	Birgir Ivar Pétursson
6	6	Ævar Már Óskarsson
7	7	Una Björg Magnúsdóttir
8	8	Einar Búi Magnússon
9	9	María Ólafsdóttir
10	10	Rökkvi Toppsson

Lagerfærslur

	lagerNr	vöruNr	magnAlager
1	01	1	12
2	01	5	144
3	01	6	12
4	01	7	2
5	01	8	0
6	01	9	0
7	01	10	-1
8	02	1	2
9	01	3	3
10	02	5	6
11	02	6	2
12	02	7	0
13	02	8	2
14	02	9	14
15	02	10	1
16	03	1	0
17	02	3	4
18	03	5	3
19	03	6	-3
20	03	7	1
21	03	8	1
22	03	9	1
23	03	10	0

Pöntun

	pöntunarNr	dagsPöntun...	vskmNr	heildarverðPöntunar	afgreitt
1	1	20050210	1	1575.36	n
2	2	20050210	2	15064.78	n
3	3	20040218	2	7708.72	j
4	4	20040201	1	382.62	j
5	5	20040301	3	1575.36	j
6	6	20050101	2	362.14	j
7	7	20050201	3	14893.95	j
8	8	20050204	4	787.68	j
9	9	20050202	5	787.68	n
10	10	20050210	6	1880.81	j

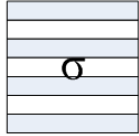
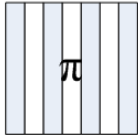
Pöntunarlínur

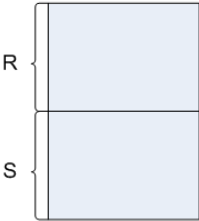
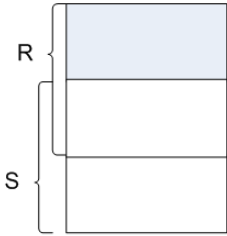
	pöntunarNr	vöruNr	fjöl...	verð
1	1	5	6	1575.36
2	2	6	2	362.14
3	2	7	1	465.04
4	2	8	1	436.28
5	2	9	14	9193.80
6	2	10	1	4607.52
7	3	1	2	382.62
8	8	5	3	787.68
9	3	6	4	724.28
10	3	7	1	465.04
11	3	8	2	872.56
12	3	9	1	656.70
13	3	10	1	4607.52
14	4	1	2	382.62
15	5	5	6	1575.36
16	6	6	2	362.14
17	7	7	1	465.04
18	7	8	1	436.28
19	7	9	14	9193.80
20	7	10	1	4607.52
21	7	1	1	191.31
22	9	5	3	787.68
23	10	6	3	543.21
24	10	7	1	465.04
25	10	8	2	872.56

6.12 Yfirlit yfir helstu aðgerðir töflualgebru

*töflutenging jafngildir $\sigma_F (t1 \times t2)$
þar sem F er $(t1.a = t2.b)$

x táknar niðurstöðutöflu
 $t, t1, t2$ tákna ótilgreindar töflur

Færsluval (e. selection) σ	$x = \sigma_{(skilyrði)} (t)$  Færsluval (e. selection)	<pre>select * into x from t where (skilyrði)</pre>
Dálkaval (e. projection) Π	$x = \Pi_{A,B,C,\dots,N} (t1)$  Dálkaval (e. projection)	<pre>select distinct A,B,\dots,N into x from t1</pre>
Krossfeldi (e. cartesian product) $\times$	$x = t1 \times t2$ P a b Q 1 2 3 P x Q a 1 a 2 a 3 b 1 b 2 b 3	<pre>select * from t1, t2</pre>

Sammengi (e. union) $\cup$ (stundum táknað með +)	$x = t1 \cup t2$ eða $x = t1 + t2$ $R \cup S$ 	<pre>select * into temp from t1; insert into temp (select * from t2); select distinct * into x from temp</pre>
Mengja- mismunur (e. set difference) $-$	$x = t1 - t2$ $R - S$ 	<pre>select * into x from t1 where not exists (select * from t2 where (t1.d1 = t2.d1) ... and (t1.dn = t2.dn))</pre>

Töflutenging* (e. join eða theta join) ⋈	$x = t1 \bowtie_{(t1.a = t2.b)} t2$ ef '=' þá jafngildistenging (<i>equijoin</i>), ef aðrir samanburðarvirkjar ($<, \leq, >, \geq, \neq$) þá <i>join</i>	<pre>select * from t1, t2 where t1.a = t2.b</pre>																							
Venjuleg tenging (e. natural join) ⋈	$x = t1 \ltimes t2$ Equijoin á alla dálka sem heita sama nafni T <table><tr><td>A</td><td>B</td></tr><tr><td>a</td><td>1</td></tr><tr><td>b</td><td>2</td></tr></table> U <table><tr><td>B</td><td>C</td></tr><tr><td>1</td><td>x</td></tr><tr><td>1</td><td>y</td></tr><tr><td>3</td><td>z</td></tr></table> T ⋈ U <table><tr><td>A</td><td>B</td><td>C</td></tr><tr><td>a</td><td>1</td><td>x</td></tr><tr><td>a</td><td>1</td><td>y</td></tr></table>	A	B	a	1	b	2	B	C	1	x	1	y	3	z	A	B	C	a	1	x	a	1	y	<pre>select t1.*, t2.* from t1, t2 where (t1.d₁ = t2.d₁) ... and (t1.d_n = t2.d_n)</pre> [[þar sem d1 ... dn tákna sameiginlega dálka t1 og t2]
A	B																								
a	1																								
b	2																								
B	C																								
1	x																								
1	y																								
3	z																								
A	B	C																							
a	1	x																							
a	1	y																							

Sniðmengi (e. inter-section) ∩	$x = t1 \cap t2$ R S R ∩ S	select * into x from t1 where exists (select * from t2 where (t1.d₁ = t2.d₁) ... and (t1.d_n = t2.d_n))																		
Deiling (e. division)	$x = t1 \div t2$ V W V ÷ W <table><tr><th>A</th><th>B</th></tr><tr><td>a</td><td>1</td></tr><tr><td>a</td><td>2</td></tr><tr><td>b</td><td>1</td></tr><tr><td>b</td><td>2</td></tr><tr><td>c</td><td>1</td></tr></table> <table><tr><th>B</th></tr><tr><td>1</td></tr><tr><td>2</td></tr></table> <table><tr><th>A</th></tr><tr><td>a</td></tr><tr><td>b</td></tr></table>	A	B	a	1	a	2	b	1	b	2	c	1	B	1	2	A	a	b	select distinct x.A from t1 as a where not exists (select * from t2 as y where not exists (select * from t1 as z where (z.A=x.A) and (z.B=y.B))
A	B																			
a	1																			
a	2																			
b	1																			
b	2																			
c	1																			
B																				
1																				
2																				
A																				
a																				
b																				

6.13 Annað til fróðleiks og umhugsunar

Allt efni sem finna má undir leitarorðunum relational algebra. Athugaðu einnig að leita að join.

7 Fallákveður

7.1 Mikilvæg hugtök í kaflanum

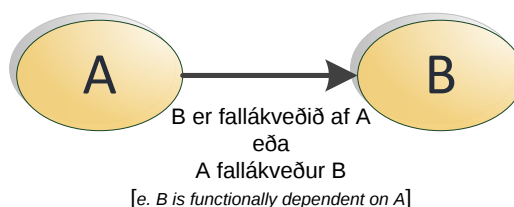
ekki-lyklar (e. *non-key attribute*)
fallákveða (e. *functional dependency*)
frumlykill (e. *primary key*)
frumreglur Armstrongs (e. *Armstrong's axioms*)
fullnægjandi [reglur] (e. *complete*)
gegnvirkni (e. *transitivity*)
hálfgegnvirkni (e. *pseudotransitive*)
mögulegur lykill (e. *candidate key*)
niðurbrot (e. *decomposition*)
réttar [reglur] (e. *sound*)
sammengi (e. *union*)
sjálfhverfni (e. *reflexivity*)
viðbót (e. *augmentation*)

7.2 Fallákveður og lyklar

Við hönnun á töflum (sem fjallað er um í næsta kafla) er margra leiða leitað til að hjálpa til við val á útfærslu. Ein öflugasta leiðin við töfluhönnun er kölluð skyldleikagreining, einnig kölluð normalisering (e. *normalisation*). Skyldleikagreining er byggð á greiningu á fallákveðum.

Þegar við hugsum um lykla (hvort sem er frumlykilinn eða mögulegu lyklana aðra) þá sjáum við fyrir okkur samband eins og það að kennitala geti vísað okkur á nafn, bílnúmer á bíl og eiganda, flugnúmer á brottfarar og áfangastað. Við getum sagt að lykillinn *fallákveði* (e. *functional dependency*) dálkana í færslunni

Flug	flugnúmer	dagur	brottfararstaður	áfangastaður
	FHE134	18.2.2008	Berlin Schoenefeld	Keflavík
	FHE153	18.2.2008	Keflavík	London Stansted
	FHE154	18.2.2008	London Stansted	Keflavík
	FHE276	18.2.2008	Barcelona	Keflavík
	FHE901	18.2.2008	Keflavík	Kaupmannahöfn
	FHE902	18.2.2008	Kaupmannahöfn	Keflavík
	FI204	18.2.2008	Keflavík	Kaupmannahöfn
	FI205	18.2.2008	Kaupmannahöfn	Keflavík
	FI213	18.2.2008	Kaupmannahöfn	Keflavík
	FI306	18.2.2008	Keflavík	Stokkhólmur Arlanda
	FI307	18.2.2008	Stokkhólmur Arlanda	Keflavík
	FI318	18.2.2008	Keflavík	Ósló Gardermoen
	FI319	18.2.2008	Ósló Gardermoen	Keflavík
	FI441	18.2.2008	Manchester	Keflavík
	FI450	18.2.2008	Keflavík	London Heathrow



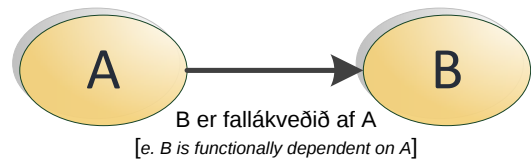
Í nemendatöflu gætum við verið með lykilinn kennitala og við gætum talað um að kennitalan **ákvarði** nafn, heimili, kyn og deild.

Nemendur				
kennitala	nafn nemanda	kyn	heimilisfang	deild

Við getum líka sagt að kennitala fallákveði nafn nemanda, heimili, kyn og deild,

eða

Nafn nemanda, heimili, kyn og deild eru fallákveðin af kennitölu



Um fallákveður **fallákveður** (*e. functional dependency*) og **mögulega lykla** (*e. candidate key*) má segja:

- **Gildi** (*e. value*) eins **dálks** ákvarðar gildi annars dálks, fallákveðin af dálknum.
- **Mögulegur lykill** (*e. candidate key*) Er einkvæmur. Einn mögulegra lykla er valinn sem **frumlykill** (*e. primary key*)
 - dæmi: bæði kennitala og greiðslukortsnúmer er í töflu og hvort um sig er mögulegur lykill
 - **Allir dálkar, sem ekki eru lykklar, eru fallákveðin** (*e. functionally dependent*) **af hvaða mögulega lykli sem er**
- **Ef lykklar „passa saman“ þá er um sömu færslu að ræða** (þetta er svona einfölduð leikskólaframsetning!)
- Fallákveður þurfa ekki að passa við **þá útfærslu sem við veljum**.
- Þeir dálkar sem ekki koma fyrir sem lykklar í fallákveðum eru kallaðir **ekki-lyklar** (*e. non-key attribute*).

7.3 Fallákveður í gagnasafnsfræði og stærðfræði

Fallákveður hafa sömu eiginleika og **föll** (*e. function*) í **stærðfræði**. Í stærðfræðinni er notuð framsetningin $f(x) = y$ og það er skilyrði þess að þetta sé fall að fyrir sama gildið af x komi sama gildið af y .

Ef við skilgreinum fallið

$$SQR(x)=y$$

sem fall sem setur x í annað veldi þá vitum við að $SQR(4)=16$.

Skilyrði þess að þarna sé um fall að ræða er að fyrir alltaf komi sama niðurstaðan, y -gildið, ef x gildið er það sama.

7.4 Táknmál fallákveða

Táknmálið sem við notum fyrir fallákveður er einfalt:

- fyrst teljum við upp **dálkinn eða dálkana** sem mynda (mögulegan) lykil hvernar fallákveðu.
- svo kemur ör ➔
- og síðan kemur **sá dálkur eða þeir dálkar sem eru fallákveðin**.

Ekki er óalgengt til að gera framsetninguna skýrari séu notaðir hornklofar utan um nöfn dálkanna:

[námskeið],[nemandi],[önn] → [einkunn]

en ég hef valið að hafa einfaldara formið á þessu:

nafn, fæðingardagur → heimilisfang

7.5 Dæmi um fallákveður

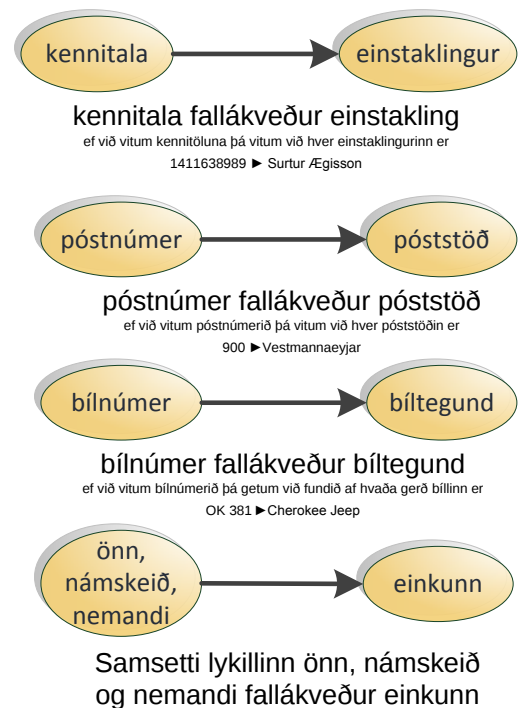
Þetta eru fallákveður:

kennitala → einstaklingur

póstnúmer → pósthöð

bílnúmer → bíltegund

námskeið, nemandi, önn → einkunn



Hvað með þessi?

götuheiti → póstnúmer

Til eru margar götur með sama nafni en í sitthvoru bæjarfélaginu, þetta er því ekki fallákveða

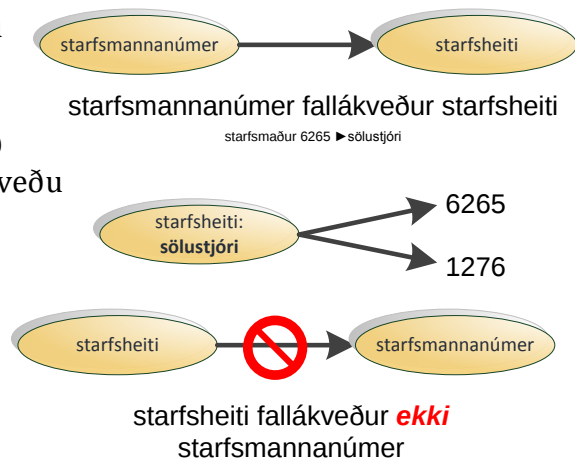
nafn → heimilisfang

Til eru alnafnar og því ekki um fallákveðu að ræða

nafn, fæðingardagur → heimilisfang

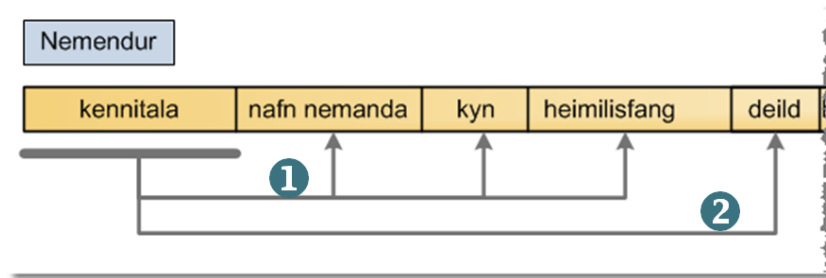
Til geta verið alnafnar sem búa á sama stað og fæddir sama dag og því ekki um fallákveðu að ræða

Sjá einnig myndir hér til hliðar um önnur dæmi.

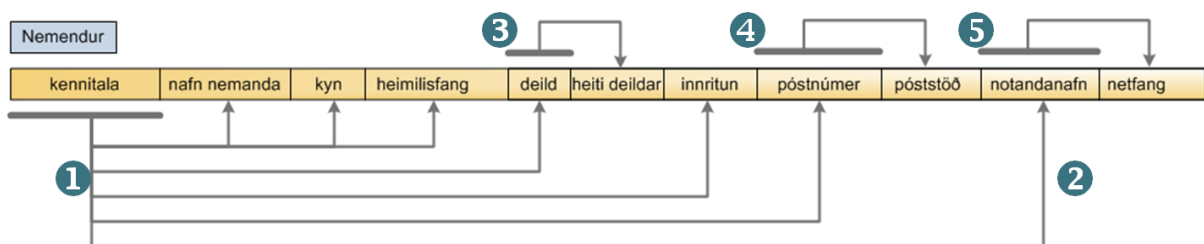


7.6 Fallákveður á myndrænan hátt

Gott getur við að setja upp fallákveður á myndrænan hátt til að fá betri yfirsýn þær og einnig þegar við erum, út frá fallákveðum, að finna lykla færslu.



- ❶ kennitala fallákveður nafn nemanda, kyn og heimilisfang (gegnum þjóðskrá)
kennitala → nafn nemanda, kyn, heimilisfang
- ❷ kennitala fallákveður deild (gert við innritun) og gert er ráð fyrir að nemandi geti einungis tilheyrt einni deild (forsenda sem við gefum okkur hér)
kennitala → deild



- ❶ kennitala → nafn nemanda, kyn, heimilisfang, póstnúmer, deild, innritun
- ❷ kennitala → notandanafn
(forsendur sem við gefum okkur, t.d. 'arnip07')
- ❸ deild → deildarnafn
(('TD' gefur okkur 'Tölvunarfræðideild')
- ❹ póstnúmer → póststöð
- ❺ notandanafn → netfang
(gefum okkur að reglan sé að netfang sé notendanafn + '@ru.is')

7.7 Formleg skilgreining á fallákveðum

Formleg skilgreining á fallákveðum er:

Fallákveðan $X \rightarrow Y$ gildir fyrir töfluna R , (X og Y eru hlutmengi R)
 ef fyrir sérhvert töflutilvik $r(R)$:
 fyrir allar færslur $t_1, t_2 \in r$,
 $t_1[X] = t_2[X] \Rightarrow t_1[Y] = t_2[Y]$

b.e. ef X -gildin eru eins þá eru Y -gildin eins

Á mannamáli væri þetta:

Kennitalan 1705625939 *gefur alltaf* 'Jón Freyr Jóhannsson', þ.e. *ef X -gildin eru eins þá eru Y -gildin eins*

Allar færslur töflunnar R verða að standast þessar kröfur

7.8 Hvernig finnum við fallákveður?

Oftast nær fáum við gefnar upp fallákveður, ekki alltaf með þeim orðum að þær heiti fallákveður heldur er talað um *samhengi gagna*. Við fáum t.d. gefið að út frá kennitölu getum við ákvarðað nafn, heimili, kyn o.s.frv. Gætum einnig fengið að út frá einkenni námskeiðs getum við fengið kennara, bækur, lýsingu námskeiðs o.s.frv.

Stundum þurfum við að rannsaka gögnin til að finna þetta samhengi en sjaldnast eru gögnin sjálf svo góð eða ítarleg að þau dugi til eitt sér.

Sem einhvern veggvísi um það hvernig finna má fallákveður eru hér nokkrir punktar til að hafa til hliðsjónar:

- Finnum bara út úr því!
- Reynsla og innsæi hjálpar okkur eitthvað
- Þurfum að skoða einstaka dálka (ekki-lykla, *e. non-key attribute*) og tengingu þeirra við aðra dálka
- Þurfum að hafa *mögulega lykla* (*e. candidate key*) í huga, þeir eiga það til að gleymast sem lyklar
- Höfum í huga (óformlega sagt):
 - „Ef við þekkjum A þá er einungis einn B sem passar við“
 - „Ef tvær færslur eru 'sammála' um A þá verða þær líka að vera sammála um B “
 - „Ef kennitalan er sú sama þá á lögheimilið að vera það sama““

7.9 Frumreglur Armstrongs

Frumreglur Armstrongs⁵⁰ (*e. Armstrong's axioms*) eru notaðar til að skilgreina og útfæra fallákveður.

Mikilvægi frumreglna Armstrongs felast í því að þær eru sagðar:

- **Réttar [reglur]** (*e. sound*): Búa aðeins til fallákveður, ekkert sem ekki er fallákveða
- **Fullnægjandi [reglur]** (*e. complete*): Frumreglurnar búa til allar fallákveður mengis fallákveða

Grunnreglurnar

Sjálfhverfni (*e. reflexivity*):

- Ef $X \supseteq Y$, þá $X \rightarrow Y$
afskaplega einföld, táknað að:
kennitala, nafn $\rightarrow$ kennitala

Viðbót (*e. augmentation*):

- Ef $X \rightarrow Y$, þá $XZ \rightarrow YZ$, fyrir öll Z
kennitala $\rightarrow$ nafn, þá kennitala, heimili $\rightarrow$ nafn, heimili

Gegnvirkni (*e. transitivity*):

- Ef $X \rightarrow Y$ og $Y \rightarrow Z$, þá $X \rightarrow Z$
kennitala $\rightarrow$ póstnúmer og póstnúmer $\rightarrow$ staður,
þá kennitala $\rightarrow$ staður

Afleiddar reglur Armstrongs, þær má leiða út frá grunnreglunum þremur (til þæginda)

Sammengi (*e. union*):

- Ef $X \rightarrow Y$ og $X \rightarrow Z$ þá $X \rightarrow YZ$

Niðurbrot (*e. decomposition*):

- Ef $X \rightarrow YZ$ þá $X \rightarrow Y$ og $X \rightarrow Z$

Hálfgegnvirkni (*e. pseudotransitive*)

- Ef $X \rightarrow Y$ og $WY \rightarrow Z$ þá $WX \rightarrow Z$

7.10 Annað til fróðleiks og umhugsunar

Bendi á wikipedia: http://en.wikipedia.org/wiki/Functional_dependency

⁵⁰ William W. Armstrong í Dependency Structures of Data Base Relationships

8 Töfluhönnun

8.1 Mikilvæg hugtök í kaflanum

aðkomulykill (e. *foreign key*)
annað staðalform (e. *second normal form, 2NF*)
Boyce-Codd staðalform (e. *Boyce-Codd normal form, BCNF*)
breytingavandinn (e. *modification anomaly*)
eingildir dálkar (e. *no multivalued attributes*),
endurtekning gagna (e. *data duplication/data redundancy*)
eyðingarvandinn (e. *deletion anomaly*)
fallákveður (e. *functional dependencies*)
fimmta staðalform (e. *fifth normal form, 5NF*)
fjórða staðalform (e. *fourth normal form, 4NF*)
frumlykill (e. *primary key*)
fyrsta staðalform (e. *first normal form, 1NF*)
gegnavirkar fallákveður (e. *transitive dependencies*)
hlutalykilsfallákveða (e. *partial key dependencies*)
innsetningarvandinn (e. *insertion anomaly*)
lágmarksendurtekning gagna (e. *minimal data redundancy*)
normalisering (e. *normalisation*)
ósamsett eigindi (e. *atomic values*)
reglur Codds
skyldleikagreining (e. *normalisation*)
staðalform (e. *normal form*)
vandi (e. *anomaly*)
velformuð tafla (e. *well structured/well formed*)
priðja staðalform (e. *third normal form, 3NF*)

8.2 Fínhönnun gagnagrunna

Til að tryggja góða virkni gagnagrunna þarf að beita viðeigandi aðferðum við hönnun þeirra (e. *schema/table design*).

- Viðfangsefni þessarar fínhönnunar á gagnagrunnum eru m.a.:
- Koma í veg fyrir vandamál sem tengjast, nýskráningum og breytingum
- Niðurbrot á töflum þannig að þær verði minni og *velformaðar* (e. *well-structured*)

Ein af þeim aðferðum sem notaðar eru er kölluð *skyldleikagreining*, eða *normalisering* á slæmri íslensku (e. *normalisation*).

Skyldleikagreining byggir m.a. *fallákveðum* (e. *functional dependencies*) sem fjallað var um í síðasta kafla.

8.3 Velformuð tafla

Tafla kallast **velformuð** (*e. well structured/well formed*) ef hún hefur **lágmarksendurtekningu** (*e. minimal data redundancy*) og gerir kleift að bæta við færslum, eyða og uppfæra án þess að skerða gagnaheilindi, þ.e. taflan verði ekki vitlaus!

Markmiðið er að komast hjá frávikum frá réttum töflum. Frávikin köllum við **vanda** (*e. anomalies*) og þá skilgreinum við í þessum flokkum:

- **Innsetningarvandinn** (*e. insertion anomaly*) - ef við bætum við færslum verðum við að endurtaka upplýsingar
- **Eyðingarvandinn** (*e. deletion anomaly*) - ef við eyðum færslum þá getum við tapað upplýsingum sem við þurfum að nota seinna
- **Breytingavandinn** (*e. modification anomaly*) - ef við breytum upplýsingum í einni færslu þá gætum við þurft að breyta í fleiri færslum og/eða töflum

8.3.1 Dæmi um töflu í vanda

Emp_ID	Name	Dept_Name	Salary	Course_Title	Date_Completed
100	Margaret Simpson	Marketing	48,000	SPSS	6/19/200X
100	Margaret Simpson	Marketing	48,000	Surveys	10/7/200X
140	Alan Beeton	Accounting	52,000	Tax Acc	12/8/200X
110	Chris Lucero	Info Systems	43,000	SPSS	1/12/200X
110	Chris Lucero	Info Systems	43,000	C++	4/22/200X
190	Lorenzo Davis	Finance	55,000		
150	Susan Martin	Marketing	42,000	SPSS	6/19/200X
150	Susan Martin	Marketing	42,000	Java	8/12/200X

51

Er þetta tafla?

Já, færslur eru einkvæmar og engin marggildi.

Hver er frumlykill töflunnar?

Samsetti lykillinn (Emp_ID, Course_Title)

Vandi þessarar töflu felst í eftirfarandi:

- **Innsetning** – getum ekki bætt við starfsmanni nema hann taki námskeið
- **Eyðing** – ef við eyðum starfsmanni 140 þá töpum við upplýsingum um námskeiðið "Tax Acc"
- **Breyting** - ef við hækkum laun Margrétar (starfsmanns nr. 100) þá þurfum við að uppfæra margar færslur
- **Rót vandans => erum með tvær töflur í sömu töflu!**

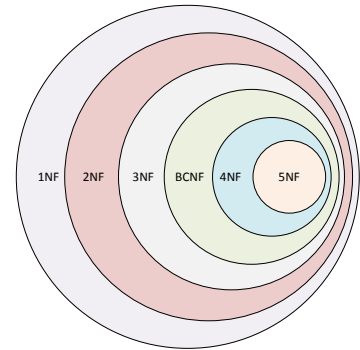
⁵¹ Enskar töflur í kaflanum er frá Connolly, Thomas og Begg, Carolyn. 2005, blár bakgrunnur og engin rammi

8.4 Staðalform

Edgar F. Codd, faðir töflugagnasafnsins, skilgreindi svokölluð **staðalform** (*e. normal form*) og voru þau í hans skilgreiningu þrjú, táknuð sem (1NF, 2NF, and 3NF)⁵². Hvert og eitt þessara formi er að takast á við vanda taflna sem ekki teljast velformaðar og til að losna við vandana þrjá sem fyrr er lýst.

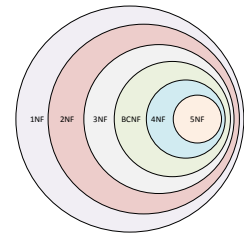
Skipulag staðalform er slíkt að þegar talað er um að tafla sé á öðru staðalformi (2NF) þá er falið í því að hún er einnig á *lægri staðalformum*, í þessu tilfelli 1NF. Þriðja staðalform felur því í sér að tafla er einnig á 1NF og 2NF.

Seinna voru skilgreind fleiri staðalaforms, 4NF, 5NF, 6NF, DKNF og BCNF (Boyce-Codd NF) en hér verður einungis farið í hin fyrstu þrjú⁵³.



8.5 Fyrsta staðalform (1NF)

Til að tafla teljist vera á fyrsta staðalformi þarf hún að uppfylla eftirfarandi skilyrði, sem eru í raun skilyrði þess að geta kallast tafla í skilningi gagnasafnsfræðinnar.



- Aðeins **eingildir dálkar**/reitir (*e. no multivalued attributes*), þetta er sama og við gerum sem skilyrði fyrir að eitthvað sé tafla.
- Hver einasti dálkur inniheldur **ósamsett eigindi** (*e. atomic values*).
- Engir tveir dálkar innihalda sömu upplýsingar.
- Allar færslur eru einkvæmar (með lykli eða samsetningu).
- Fallákveður eru innan töflu ekki milli taflna.
- **Allar töflur eru á fyrsta staðalformi (1NF).**
- ef þær eru það ekki þá uppfylla þær ekki skilyrði fyrir því að kallast **töflur** (*e. relation*)

Töflur á fyrsta staðalformi einfalda fyrirspurnir og röðun gagna, einnig er skalanleiki meiri. Hver færsla getur verið skilgreind einkvæmt sem er nauðsynlegt fyrir uppfærslur og eyðingu.

⁵² These normal forms have been summarized as requiring that all non-key attributes be dependent on "the key, the whole key and nothing but the key"

⁵³ Bendi á t.d. Wikipedia um Database normalization, http://en.wikipedia.org/wiki/Database_normalization

8.5.1 Dæmi um töflu á fyrsta staðalformi (1NF)

Þetta er ekki tafla, hún uppfyllir ekki skilyrði töflu skv. skilningi gagnasafnsfræðinnar:

Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address	Product_ID	Product_Description	Product_Finish	Unit_Price	Ordered_Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
					5	Writer's Desk	Cherry	325.00	2
					4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
					4	Entertainment Center	Natural Maple	650.00	3

Þegar búið er að koma töflunni á fyrsta staðalform og hún þá orðinn raunveruleg tafla lítur hún svo út:

Order_ID	Order_Date	Customer_ID	Customer_Name	Customer_Address	Product_ID	Product_Description	Product_Finish	Unit_Price	Ordered_Quantity
1006	10/24/2004	2	Value Furniture	Plano, TX	7	Dining Table	Natural Ash	800.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	5	Writer's Desk	Cherry	325.00	2
1006	10/24/2004	2	Value Furniture	Plano, TX	4	Entertainment Center	Natural Maple	650.00	1
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	11	4-Dr Dresser	Oak	500.00	4
1007	10/25/2004	6	Furniture Gallery	Boulder, CO	4	Entertainment Center	Natural Maple	650.00	3

Þetta er hugsanlega „subbuleg“ tafla, en hún er tafla á fyrsta staðalformi (1NF).

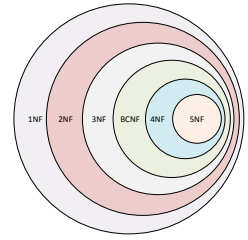
Vandi töflunnar er eftirfarandi er hins vegar til staðar og felst í :

- **Innsetning** - ef bætt er við vöru fyrir tiltekna pöntun þá þarf að endurtaka nafn og númer viðskiptamanns.
- **Eyðing** - ef við eyðum borðinu (Dining Table) úr pöntun 1006 þá missum við einnig verð og áferð.
- **Breyting** - ef við breytum verði vöru nr. 4 þá þarf að uppfæra í nokkrum færslum.

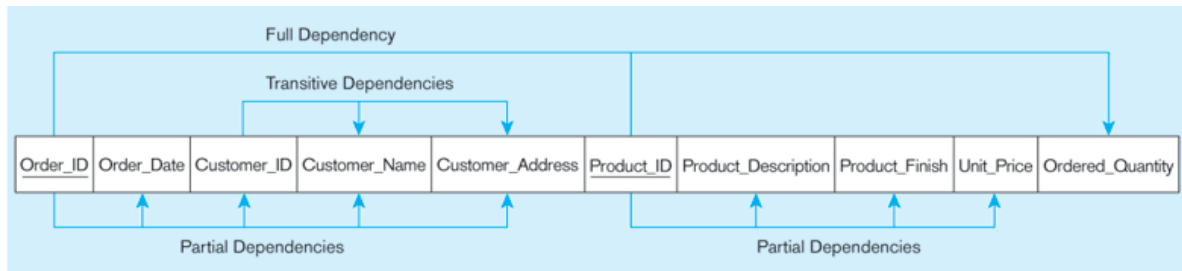
8.6 Annað staðalform (2NF)

Skilyrði þess að tafla sé á öðru staðalformi (2NF) eru:

- Er á fyrsta staðalformi (1NF)
- Sérhver dálkur utan lykils skal vera háður **öllum** lyklinum, ekki hluta hans (engar *hlutalykilsfallákveður* (e. *partial key dependencies*))



Eftirfarandi tafla er ekki á 2NF



Af myndinni má lesa eftirfarandi fallákveður:

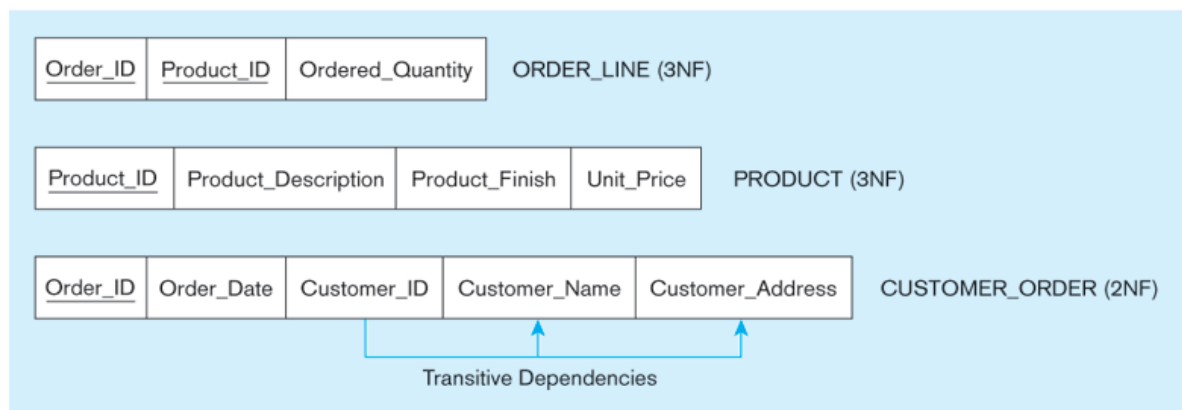
Order_ID → Order_Date, Customer_ID, Customer_Name, Customer_Address

Customer_ID → Customer_Name, Customer_Address

Product_ID → Product_Description, Product_Finish, Unit_Price

Order_ID, Product_ID → Ordered_Quantity

Ef þessi tafla er sett á 2NF verður hún svona:

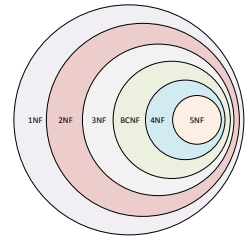


Hlutalykilsfallákveður hafa verið fjarlægðar, en enn eru *gegnvirkar fallákveður* (e. *transitive dependencies*)

8.7 Þriðja staðalform (3NF)

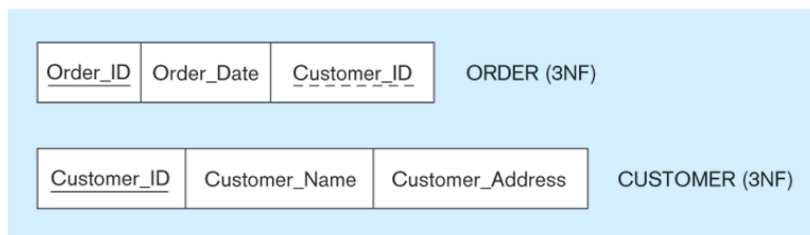
Skilyrði þess að tafla sé á þriðja staðalformi (3NF) eru:

- Tafla skal vera á 2NF (og þar með einnig á 1NF)
- Sérhver dálkur utan lykils skal einungis vera háður lyklinum, engar *gegnvirkar fallákveður* (e. *transitive dependencies*) leyfðar
- Fallákveður með gegnvirkni, utan lykils, eru settar í nýja töflu
- Fallákveðan verður *frumlykill* (e. *primary key*) í nýju töflunni og er áfram sem *aðkomulykill* (e. *foreign key*) í gömlu töflunni



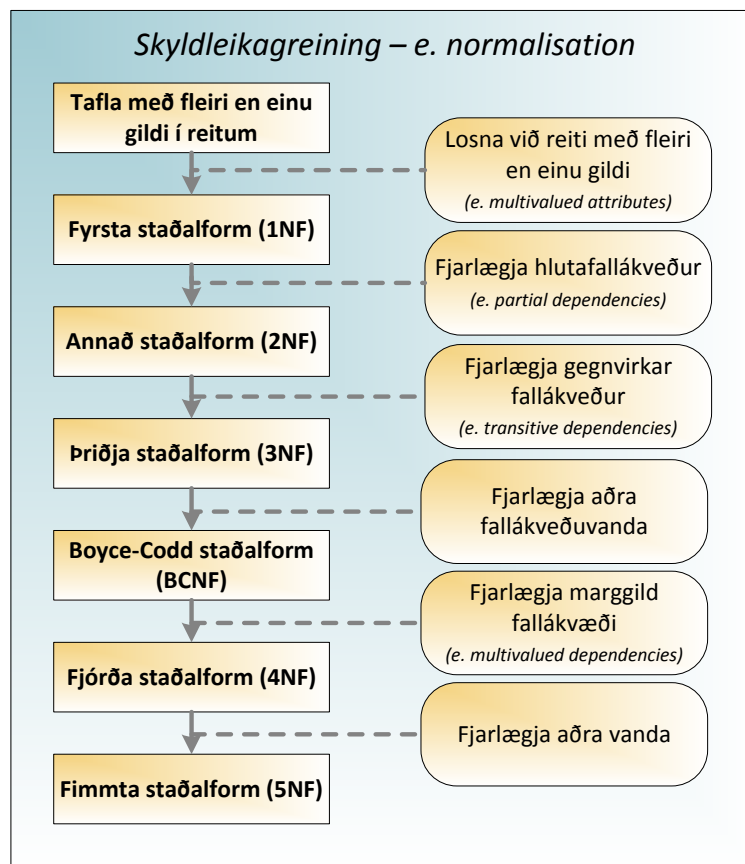
8.7.1 Setjum á þriðja staðalform (3NF)

Gegnvirkar fallákveður settar í nýja töflu:



8.8 Samantekt um hvað þarf að gera milli staðalforma

Myndin hér gerir grein fyrir þeim breytingum sem gera þarf á töflum milli normalforma.



8.9 Annað til fróðleiks og umhugsunar

8.9.1 Önnur staðalform

Um staðalform, m.a. önnur en fyrsta, annað og þriðja, má lesa m.a. á Wikipedia: Database normalization, http://en.wikipedia.org/wiki/Database_normalization

Flestallar bækur um gagnasafnsfræði verja einum eða fleiri köflum í *skyldleikagreiningu* (*e. normalisation*).

9 Hreyfingar

9.1 Mikilvæg hugtök í kaflanum

ACID

atburðarnúmer (*e. log sequence number, LSN*)

einangrun (*e. isolation*)

eining (*e. atomicity*)

ending (*e. durability*)

gagnaheilindi grunns (*e. database integrity*)

heilleiki (*e. consistency*)

hreyfing (*e. transaction, einnig LUW, e. logical unit of work*)

hreyfingarnúmer (*e. transaction ID number*)

hreyfingastjórnun (*e. transaction management*)

læsing (*e. lock*)

SAFA (Skrifa Atburðaskrá Fyrir Aðgerð) (*e. WAL, write-ahead-logging*)

skuggasíður (*e. shadow paging*)

tegund atburðar (*e. transaction log type*)

9.2 Hvað er hreyfing?

Hreyfing (*e. transaction*) í gagnagrunni er tiltekin aðgerð eða hópur aðgerða (skipana eða fyrirspurna) **sem mynda eina heild gagnvart bæði notanda og gagnagrunni**. Hver slík hreyfing er meðhöndluð sem ein heild á þann hátt að annaðhvort er hún öll framkvæmd eða alls ekki. Slíkur eiginleiki er mikilvægur því oft er um það að ræða að verið er að leita upplýsinga og uppfæra í fleiri en einu skrefi eða með fleiri en eini skipun.

Dæmi um hreyfingu sem þarf að tryggja á þennan hátt:

- flytja peninga milli reikninga og hreyfingin felst í að taka upphæð af einum reikningi annars vegar og hins vegar leggja upphæðin inn á annan reikning

Ef hreyfingin truflaðist (bilun, kerfishrun) á milli þessara tveggja þátta væri búið taka út af einum reikningi en ekki leggja inn á hinn og það er að sjálfsögðu óviðunandi staða.

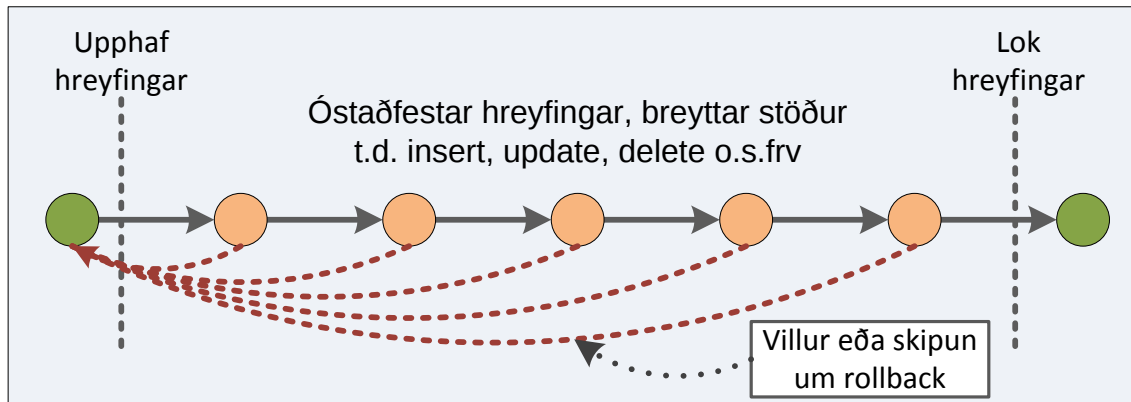
Ef ekki er tryggt að öll hreyfingin sé framkvæmd (t.d. vegna villu eða bilunar) er hætt á því að skilið verði við **gagnagrunninn í ólöglegu ástandi**. Talað er um að hreyfingar færi gagnagrunn úr einu löglegu ástandi til annars. Stjórnun hreyfinga tryggir einnig **gagnaheilindi grunnsins** (*e. database integrity*).

Sá eiginleiki gagnagrunna að geta sinnt mörgum notendum í einu gerir þá kröfu til gagnagrunnskerfa að þau tryggi að aðgerðir einstakra notenda hafi ekki áhrif á aðgerðir eða fyrirspurnir annarra.

Flóknar samsettar aðgerðir krefjast þess einnig af gagnagrunnskerfið að tryggt sé að aðgerðin geti framkvæmst öll án frávika. Dæmi um slíkt eru flóknar uppfærslur margra færslna og jafnvel í mörgum töflum.

Hreyfingastjórnun (*e. transaction management*) gagnagrunnskerfa snýst um að mæta þessum þörfum sem hér hafa verið nefndar.

Í stað orðsins **hreyfingar** (*e. transaction*) er stundum notað enska hugtakið **LUW** sem stendur fyrir *Logical Units of Work*.



Hefðbundin hreyfing í gagnagrunni er framkvæmd í eftirfarandi skrefum

1. Hreyfing hefst
2. Fyrirspurnir og skipanir eru framkvæmdar en breytingar vegna þeirra eru ekki enn sýnilegar
3. Niðurstöður gerðar sýnilegar (*e. commit*)

Tryggja þarf að hreyfingin sé framkvæmd öll (eða ekki) og að staða gagnagrunns sé lögleg.

9.3 Eiginleikar hreyfinga - ACID

Í gagnasafnsfræði er talað um fjóra grunneiginleika hreyfinga:

eining (*e. atomicity*)

- Hreyfing er „ein aðgerð“ – annað hvort framkvæmist öll hreyfingin eða ekkert af henni

heilleiki (*e. consistency*)

- Hver hreyfing færir grunninn úr einu löglegu ástandi í annað

einangrun (*e. isolation*)

- Hver hreyfing keyrir eins og hún væri ein í heiminum

ending (*e. durability*)

- Eftir að hreyfing er staðfest hverfa áhrif hennar ekki, þrátt fyrir kerfishrun

Þessir eiginleikar eru á ensku kallaðir **ACID** (eftir upphafsstöfum heita eiginleikanna **A**tomicity, **C**onsistency, **I**solation, **D**urability)

Að uppfylltum þessum eiginleikur er áreiðanleiki hreyfinga í gagnagrunni tryggður⁵⁴.

9.3.1 Eining - Atomicity

Með **einingu** (*e. atomicity*) er átt við að hreyfing sé **ein aðgerð**, annað hvort framkvæmist öll hreyfingin eða ekkert af henni.

Dæmi um hreyfingu sem krefst þessara eiginleika er bókun og greiðsla á flugmiða. Flugmiði á að vera greiddur og bókaður eða hvorki greiddur né bókaður. Óásættanlegt er að flugmiði sé greiddur án þess að bókun hafi átt sér stað og jafn óásættanlegt er að bókun verði án þess að greitt sé fyrir miðann. Ef slíkt kemur upp þarf að tryggja að hætt sé við hreyfingu og hún tekin öll tilbaka frekar en að henni sé einungis lokið að hluta.

Hætt við hreyfingar af þremur ástæðum

- Gagnagrunnur getur hætt við hreyfingu af innri ástæðum s.s. vegna brota á skilyrðum færslna og taflna eða lykla- eða skilgreiningarmengjaskilyrða
- Kerfið er sambandslaust eða hrynur í miðju kafi t.d. vegna bilunar eða tengivandamála
- Notandi (forrit) hættir við hreyfingu (*e. rollback*) vegna þess að einhver skilyrði eru ekki uppfyllt

9.3.2 Heilleiki - Consistency

Með **heilleika** (*e. consistency*) er átt við að hver hreyfing færir grunninn úr einu löglegu ástandi í annað. Með því er átt við að hreyfing getur ekki brotið reglur gagnagrunns, töfluskilyrði, skilyrði skilgreiningarmengja, lyklaskilyrði eða önnur skilyrði.

Hver hreyfing tekur við grunninum í löglegu ástandi og þarf að skilja við grunninn í löglegu ástandi

Hreyfingastjórnun tekur til ábyrgðar á heilleika gagna. Gagnagrunnskerfið sér um sum skilyrði s.s. skilgreind skilyrði aðkomulykla, frumlykla og skilgreiningarmengja.

Annars er ábyrgðin í höndum forritara, gagnagrunnskerfið hefur ekki möguleika á að gera meira en það sem skilgreint er í gagnagrunnskerfinu.

9.3.3 Einangrun - Isolation

Með **einangrun** (*e. isolation*) er átt við að hver hreyfing keyrir eins og hún væri ein í heiminum án áhrifa frá öðrum hreyfingum⁵⁵. Engar aðrar aðgerðir geta séð áhrif hreyfingarinnar fyrr en henni er að fullu lokið. Allar **millibilsaðgerðir** (*e. intermediate state*) eru ósýnilegar og þeim haldið til hliðar⁵⁶.

9.3.4 Ending - Durability

Með **endingu** (*e. durability*) er átt við að eftir að hreyfing er staðfest hverfa áhrif hennar ekki, þrátt fyrir kerfishrun. Hún er komin til að vera, hefur staðist öll töfluskilyrði, skilgreiningarmengjaskilyrði, lyklaskilyrði og önnur skilyrði sem um færslurnar gilda.

⁵⁴ Athuga þarf að í sumum tilvikum víkja gagnagrunnskerfi að einhverju leiti frá ACID eiginleikunum til að tryggja betri afköst.

⁵⁵ Frá þessari kröfu er oft víkið í gagnagrunnskerfum þar sem hún krefst þess í raun að öllum færslum sem unnið er með sé læst meðan hreyfing vinnur. Slíkt er frekar kostnaðarsamt (m.t.t. samvinnslu) og því notaðar ýmsar aðferðir til að mæta kröfu um einangrun að hluta.

⁵⁶ Vert er að skoða hugtakið raðanlegar áætlanir (*e. serializable schedule*) í samhengi við einangrun (*e. isolation*)

9.4 Atburðaskrá - transaction log

Atburðaskrá (e. transaction log) er helsta aðferðin til að tryggja að hreyfingar séu framkvæmdar réttar og til enda. Helsta aðferð sem er notuð er kölluð write-ahead-logging, skammstafað WAL, á íslensku má kalla þetta SAFA Skrifa Atburðaskrá Fyrir Aðgerð

Áður en aðgerð er framkvæmd er skráð hvaða aðgerð á að gera, hvaða gildi eru til fyrir og hvaða gildi á að setja í staðinn (ef um breytingu eða eyðingu er að ræða) Þegar aðgerð hefur verið framkvæmd er skráð að henni sé lokið. Ef aðgerð lýkur ekki á eðlilegan hátt er hægt að rekja sig tilbaka og koma grunni í upprunalegt horf

9.4.1 Uppbygging venjulegrar atburðarskrár

Atburðaskrá inniheldur vanalega þætti eins og:⁵⁷

- **Atburðarnúmer** (e. *log sequence number, LSN*): Einkvæmt númer færslu í atburðaskrá.
- **Fyrri atburðarnúmer** (e. *previous LSN*): Tenging við síðasta atburð, atburðaskráin verður því að eins konar **tengdum lista** (e. *linked list*).
- **Hreyfingarnúmer** (e. *transaction ID number*): Vísun í þá hreyfingu gagnagrunns sem veldur atburði.
- **Tegund atburðar** (e. *transaction log type*): Tilgreinir tegund færslu (atburðar).
- Aðrar upplýsingar um þær breytingar sem atburðinn hefur í för með sér.

Sjá nánar um atburðaskrá og tegundir atburða t.d. á Wikipedia.⁵⁸

9.5 Aðrar leiðir til útfærslu hreyfingastjórnunar

Til viðbótar við **SAFA** eða **WAL** (**Skrifa Atburðaskrá Fyrir Aðgerð** eða *write-ahead-logging*) má nefna aðferð eins og **skuggasíður** (e. *shadow paging*)⁵⁹. Hér verður ekki fjallað um þá aðferð en rétt er að kynna sér hana.

Bæði SAFA og skuggasíður byggja á útfærslum á **læsingum** (e. *lock*) sem einnig er rétt að kynna sér t.d. á Wikipedia⁶⁰

9.6 Annað til fróðleiks og umhugsunar

Fletta má upp með Google eftir 'Write-Ahead Transaction Log'.

Dæmi um vandamál við samsíða vinnslu:

http://coffee.kennesaw.edu/sql_java/concurrency.html

ARIES (Algorithms for Recovery and Isolation Exploiting Semantics) algóripminn við endurheimt gagnagrunna:

http://en.wikipedia.org/wiki/Algorithms_for_Recovery_and_Isolation_Exploiting_Semantics

⁵⁷ Sjá nánar t.d. á Wikipedia: http://en.wikipedia.org/wiki/Transaction_log

⁵⁸ Sjá um transaction log types á: http://en.wikipedia.org/wiki/Transaction_log

⁵⁹ Sjá Shadow Paging á: http://en.wikipedia.org/wiki/Shadow_paging

⁶⁰ Sjá Lock á http://en.wikipedia.org/wiki/Lock_%28computer_science%29

A. Stutt yfirlit yfir SQL skipanir

Hér er uppflettilisti yfir nokkrar helstu skipanir í SQL. Athugaðu að hér eru ekki sýndir allir möguleikar í hverri skipun. Notaðu hjálpina í SQL Server til að fá nánari upplýsingar, eða hjálpina í þínu gagnasafnskerfi ef það er annað en hér er tiltekið. Hafðu líka í huga að þetta er SQL málfræði *SQL Server 2005* og hún þarf ekki að vera sú sama og í öðrum útgáfum SQL Server frá Microsoft og ekki heldur annarra framleiðenda og ekki endilega eins og *SQL2003* staðallinn segir til um. Leitaðu því alltaf í hjálp kerfisins ef þú ert í vafa.

Skipun (statement)	Málfræði (syntax)
AND / OR	<pre>SELECT nafn_dálks [, ...] FROM nafn_töflu WHERE skilyrði AND OR skilyrði</pre> er notað eins og hefðbundið er um þessar rökaðgerðir
ALTER TABLE (add column)	<pre>ALTER TABLE nafn_töflu ADD nafn_dálks gagnatag</pre> Notað til að breyta skilgreiningu á töflu sem til er fyrir, hér er verið að bæta við dálki
ALTER TABLE (drop column)	<pre>ALTER TABLE nafn_töflu DROP COLUMN nafn_dálks</pre> Notað til að breyta skilgreiningu á töflu sem til er fyrir, hér er verið að fella niður dálk
AS (alias/auknefni fyrir dálk)	<pre>SELECT nafn_dálks AS auknefni_dálks FROM nafn_töflu</pre> Notað til að gefa einhverju nýtt nafn. Notað annars vegar til að skilgreina "þægilegri" heiti á dálka eða töflur og hins vegar til að skilgreina Skiljanlegri eða betri heiti á dálka fyrir birtingu í niðurstöðu
AS (alias/auknefni fyrir töflu)	<pre>SELECT nafn_dálks FROM nafn_töflu AS auknefni_töflu</pre> Sjá hér á undan

BETWEEN	<pre>SELECT nafn_dálks [, ...] FROM nafn_töflu WHERE nafn_dálks BETWEEN gildi1 AND gildi2</pre> Notað til að einfalda samanburð, kemur í staðinn fyrir að segja: nafn_dálks >= gildi1 and nafn_dálks <= gildi2
CREATE DATABASE	<pre>CREATE DATABASE nafn_gagnasafns</pre> Notað til að stofna gagnagrunn, krefst DBA réttinda og skal ekki gerast nema að vel athuguðu máli
CREATE INDEX	<pre>CREATE [UNIQUE] INDEX nafn_vísis ON nafn_töflu nafn_sjónarhorns (nafn_dálks [ASC DESC] [, ...])</pre> Notað til að skilgreina vísi/index á tiltekinn dálk. Getur gert röðun/lestur hraðari ef rétt er að staðið
CREATE TABLE	<pre>CREATE TABLE nafn_töflu (nafn_dálks gagnatag [, ...])</pre> Notað til að skilgreina töflu. Oft hægt að gera í hefðbundnu um hverfi eins og <i>SQL Server Management Studio</i>
CREATE VIEW	<pre>CREATE VIEW nafn_sjónarhorns AS SELECT nafn_dálks [, ...] FROM nafn_töflu WHERE skilyrði</pre> Notað til að skilgreina sjónarhorn/view. Sjónarhorn virkar eins og tafla en getur spannað fleiri en eina töflu, verið með útreikninga sem ekki eru í upprunatöflunum og er einnig hægt að nota til að takmarka sýn notanda á undirliggjandi töflur
DELETE FROM	<pre>DELETE FROM nafn_töflu eða DELETE FROM nafn_töflu WHERE skilyrði</pre> Notað til að eyða gögnum úr töflu. Skilgreiningu á töflunni er ekki eytt, einungis innihaldinu. WHERE skilyrðið getur síðan takmarkað hverju er eytt. Ef eyða skal bæði innihaldi og skilgreiningu töflu má nota DROP TABLE
DROP DATABASE	<pre>DROP DATABASE nafn_gagnasafns [, ...]</pre> Notað til að eyða út heilum gagnagrunni. Notist með varúð, þarf DBA réttindi til að gera

DROP INDEX	DROP INDEX nafn_töflu.nafn_vísis [, ...] Notað til að eyða burtu vísi/index, gæti verið til að geta endurgert hann eða leiðrétt eða vegna þess að hann er ekki þarfur lengur eða jafnvel farin að draga úr afköstum
DROP TABLE	DROP TABLE nafn_töflu Notað til að eyða út töflu (skilgreiningu og gögnum) . Notist með varúð, þarf DBA réttindi til að gera
GROUP BY	SELECT nafn_dálks1 FROM nafn_töflu GROUP BY nafn_dálks1 segð Notað til að skilgreina flokka sem dregnir eru saman. T.d. ef flokkað væri eftir tegund bifreiða væri hver tegund fyrir tilgreind með síðan einhverjum uppsöfnuðum gildum (oft talað um <i>aggregate functions</i>) svo sem sum() , avg() o.fl.
HAVING	SELECT nafn_dálks1 FROM nafn_töflu GROUP BY nafn_dálks1 HAVING skilyrði Notað til að skilgreina afmarkanir eða skilyrði þeirra flokka sem GROUP BY velur. Virkar eins og WHERE gerir fyrir SELECT , nema að HAVING virkar á flokka en WHERE á stakar færslur
IN	SELECT nafn_dálks ... FROM nafn_töflu WHERE nafn_dálks IN undirfyrirspurn segð [, ...] Notað til að skilgreina "er í menginu" eða "tilheyrir". Virkar þannig að ef gildið sem er í nafn_dálks er að finna í undirfyrirspurn eða segð þá er gildi samanburðar "satt/true"
INSERT INTO	INSERT [INTO] nafn_töflu [(nafn_dálks1, nafn_dálks2 [, ...])] VALUES (gildil,gildi2 [, ...]) Notað til að setja færslur inn í töflur
LIKE	SELECT nafn_dálks [, ...] FROM nafn_töflu WHERE nafn_dálks LIKE mynstur Notað til að setja fram mynsturskilyrði (e. pattern matching) , t.d. nafn_nemanda like ('%hildur%') sem gæfi okkur hildur, gunnhildur o.s.frv.

ORDER BY	<pre>SELECT nafn_dálks [, ...] FROM nafn_töflu ORDER BY nafn_dálks [ASC DESC]</pre> Notað til að skilgreina hvernig niðurstöðum er raðað, hækkandi, lækkandi. Um er að ræða hefðbundna stafrófs- og töluröðun
SELECT	<pre>SELECT nafn_dálks [, ...] FROM nafn_töflu</pre> Velur tiltekna dálka úr töflu og birtir innihald þeirra
SELECT *	<pre>SELECT * FROM nafn_töflu</pre> Velur alla dálka úr töflu og birtir innihald þeirra, ekki er mælt með notkun *, frekar að telja upp dálkana
SELECT DISTINCT	<pre>SELECT DISTINCT nafn_dálks [, ...] FROM nafn_töflu</pre> Aðeins ein færsla með hverju einstöku gildi er birt
SELECT INTO	<pre>SELECT nafn_dálks [, ...] INTO nýtt_nafn_töflu FROM upprunalegt_nafn_töflu</pre> Velur tiltekna dálka og setur inn í nýja töflu, nýja taflan er þá skilgreind af þeim dálkum sem SELECT setningin skilar af sér, má nota til að afrita töflur
TRUNCATE TABLE	<pre>TRUNCATE TABLE nafn_töflu</pre> Eyðir bara gögnum úr töflunni, en ekki skilgreiningu á töflunni. Er því sambærilegt við DELETE, hins vegar verður ekki til atburðafærsla (log-transaction) fyrir hverja færslu sem eytt er, og því ekki hægt að bakka tilbaka með skipun eins og ROLLBACK
UPDATE	<pre>UPDATE nafn_töflu SET nafn_dálks=nýtt_gildi [,nafn_dálks=nýtt_gildi [, ...]] WHERE nafn_dálks=eitthvert_gildi</pre> Notað til að breyta gildi á tilteknum gildum í einni eða fleiri færslum töflu
WHERE	<pre>SELECT nafn_dálks [, ...] FROM nafn_töflu WHERE skilyrði</pre> Notað til að skilgreina afmarkanir eða skilyrði fyrirspurnar. Oft gert með samanburði þar sem skilyrt er að gildin sem SELECT setning skilar séu af tilteknum gildum. Einnig hægt að bera saman við undirfyrirspurn (annarri fyrirspurn sem er innan í þessari)

B. Hugtök íslensk-ensk

Íslenskt heiti	Enskt heiti
aðgerðarstofn	operands
aðgerðir	operations
aðkomulykill	foreign key
aðkomulyklaskilyrði	referential integrity
aðkomutafla	foreign table, child table
annað staðalform	second normal form, 2NF
atburðarnúmer	log sequence number, LSN
aukalykill	alternate keys
aukavísir	secondary index
B+ trjáa gagnaskipan	B+ structure
barnafærslur	child row
bendir	pointer
bestun fyrirspurna	query optimisation
Boyce-Codd staðalform	Boyce-Codd normal form, BCNF
breyting gagna	update rules
breytingavandinn	modification anomaly
dálkaval	projection
dálkur eða eigindi	attribute, column
dálkur eða eigindi	attribute, column
deiling	division
dýpt trés	tree depth
eigin skilgreind gagnatög	UDT, user defined data types
einangrun	isolation
eingildir dálkar	no multivalued attributes,
eining	atomicity
einkvæmur lykill	unique key
einsmóta aðgangsstýringar og stjórnun	uniform data administration
ekki-lyklar	non-key attribute
ending	durability
endurheimt við kerfishrun	recovery from crashes
endurtekning gagna	data duplication/data redundancy
eyðing gagna	delete rules
eyðingarvandinn	deletion anomaly
fallákveða	functional dependency
fata	bucket eða bin

fimmta staðalform	fifth normal form, 5NF
fjórða staðalform	fourth normal form, 4NF
foreldrafærslur	parent row
frumlykill	primary key
frumreglur Armstrongs	Armstrong's axioms
fullnægjandi [reglur]	complete
fyrirspurnarmál	query language
fyrsta staðalform	first normal form, 1NF
færsla	row, record, tuple
færslufjöldi töflu	cardinality
færslustaðgreinir	row locator
færsluval	selection
gagnaheilindi grunns	database integrity
gagnagrunnur	sjá gagnasafn og gagnasafnskerfi
gagnaheilindi	data integrity
gagnaheilindi, gagnaheilleiki	data or entity integrity
gagnahuld	data independence
gagnasafn	database
gagnasafnskerfi	database management system
gagnategund	sjá tag
gegnvirkar fallákveður	transitive dependencies
gegnvirkni	transitivity
gildi	value
gildisbil	range
gildisleysan NULL	NULL
gisinn vísir	sparse index
gráða trés	order
gráða töflu	degree, arity
grunnskilgreiningarmengi	type skilgreint á mismundandi hátt eftir umhverfi
hálfgegnvirkni	pseudotransitive
heilleiki	consistency
hlutalykilsfallákveða	partial key dependencies
hlutmengi	subset
hreiðraðar segðir	nested expressions
hreyfing	transaction, einnig LUW, logical unit of work
hreyfingarnúmer	transaction ID number
hreyfingastjórnun	transaction management
hrúga	heap
hægri ytri töflutenging	right outer join
innri töflutenging	inner join
innsetning gagna	insert rules
innsetningarvandinn	insertion anomaly
jafngildistenging	equijoin
keðjuverkun	cascade
kerfisgreining	systema analysis and design

klasaður vísir	clustered index
klösuð tafla	clustered table
lauf	leaf node
lágmarksendurtekning gagna	minimal data redundancy
lykill	key
lykiltafla	key table, parent table
lyklaskilyrði	key constraints
lýsigögn	metadata, gögn um gögn
læsing	lock
lög	level
mengi	set
mengjamargfeldi/krossfeldi	cartesian product
mengjamismunur	set difference
mögulegur lykill	candidate key
mögulegur lykill	candidate key
niðurbrot	decomposition
normalisering	normalisation
núllstilling	nullify
óklasaður vísir	non-clusterd index
óklasaður vísir	non-clustered index
ósamsett eigindi	atomic values
raðaður vísir	ordered index
raunlægt skema	physical schema
rekstraraðili gagnasafnskerfis	DBA, database administrator
réttar [reglur]	sound
röklegt skema	conceptual/logical schema
röktafla með þremur rökgildum	three value logic table
rökvirkjar	logical operator
SAFA	Skrifa Atburðaskrá Fyrir Aðgerð
samanburðarmengi	check constraint
samanburðarvirkjar	comparison operators
sammengi	union
sammengishæf	union-compatible
samsettur lykill	composite key
samtíða aðgangur/vinnslur	concurrent access
segð	expression
sjálfgefin gildi	default value
sjálfhverfni	reflexivity
skilgreiningarmengi	domain, field specification
skilvirkur aðgangur	efficient access
skilyrði	constraint
skilyrði skilgreiningarmengis	domain constraints
skilyrði venslatöflu	definition of relation
skuggasíður	shadow paging
skyldleikagreining	normalisation
sniðmengi	intersection

SQL	structured query language
staðalform	normal form
stök mengis	elements
sundurlæg mengi	disjoint sets
sýn	view
tafla	relation
tag	type, sjá einnig skilgreiningarmengi
takmörkun	restriction
tegund atburðar	transaction log type
tætifall	hash function
tætifallsvísir	hash index
töflualgebra	relational algebra
töflugagnagrunnur	relational database management system
töflugagnasafn	relational database
töflulíkan	relational model
töfluskanna so.	tablescan
töfluskema	relation schema, table schema
töflutenging	join eða theta-join
valskilyrði	selection predicate
vandi	anomaly
velformað	well-formed
velformuð tafla	well structured/well formed
venjuleg tenging	natural join
vensl	sjá tafla
venslatafla	sjá tafla
viðbót	augmentation
vinstri ytri töflutenging	left outer join
virkjar	operations
vísatafla	index table
vísauppsetning	indexing
vísir	index
ytri töflutenging	outer join eða full outer join
þéttur vísir	dense index
þriðja staðalform	third normal form, 3NF

C. Hugtök ensk-íslensk

Enskt heiti	Íslenskt heiti
alternate keys	aukalykill
anomaly	vandi
Armstrong's axioms	frumreglur Armstrongs
atomic values	ósamsett eigindi
atomicity	eining
attribute, column	dálkur eða eigindi
augmentation	viðbót
B+ structure	B+ trjáa gagnaskipan
Boyce-Codd normal form, BCNF	Boyce-Codd staðalform
bucket eða bin	fata
candidate key	mögulegur lykill
cardinality	færslufjöldi töflu
cartesian product	mengjamargfeldi/krossfeldi
cascade	keðjuverkun
check constraint	samanburðarmengi
child row	barnafærslur
clustered index	klasaður vísir
clustered table	klösuð tafla
comparison operators	samanburðarvirkjar
complete	fullnægjandi [reglur]
composite key	samsettur lykill
conceptual/logical schema	röklegt skema
concurrent access	samtíða aðgangur/vinnslur
consistency	heilleiki
constraint	skilyrði
data duplication/data redundancy	endurtekning gagna
data independence	gagnahuld
data integrity	gagnaheilindi
data or entity integrity	gagnaheilindi, gagnaheilleiki
database	gagnasafn
database integrity	gagnaheilindi grunns
database management system	gagnasafnskerfi
DBA, database administrator	rekstraraðili gagnasafnskerfis
decomposition	niðurbrot
default value	sjálfgefin gildi
definition of relation	skilyrði venslatöflu

degree, arity	gráða töflu
delete rules	eyðing gagna
deletion anomaly	eyðingarvandinn
dense index	þéttur vísir
disjoint sets	sundurlæg mengi
division	deiling
domain constraints	skilyrði skilgreiningarmengis
domain, field specification	skilgreiningarmengi
durability	ending
efficient access	skilvirkur aðgangur
elements	stök mengis
equijoin	jafngildistenging
expression	segð
first normal form, 1NF	fyrsta staðalform
foreign key	aðkomulykill
foreign table, child table	aðkomutafla
functional dependency	fallákveða
hash function	tætifall
hash index	tætifallsvísir
heap	hrúga
index	vísir
index table	vísatafla
indexing	vísauppsetning
inner join	innri töflutenging
insert rules	innsetning gagna
insertion anomaly	innsetningarvandinn
intersection	sniðmengi
isolation	einangrun
join eða theta-join	töflutenging
key	lykill
key constraints	lyklaskilyrði
key table, parent table	lykilafla
leaf node	lauf
left outer join	vinstri ytri töflutenging
level	lög
lock	læsing
log sequence number, LSN	atburðarnúmer
logical operator	rökvirkjar
metadata, gögn um gögn	lýsigögn
minimal data redundancy	lágmarksendurtekning gagna
modification anomaly	breytingavandinn
natural join	venjuleg tenging
nested expressions	hreiðraðar segðir
no multivalued attributes,	eingildir dálkar
non-clustered index	óklasaður vísir
non-key attribute	ekki-lyklar

normal form	staðalform
normalisation	skyldleikagreining
normalisation	normalisering
NULL	gildisleysan NULL
nullify	núllstilling
operands	aðgerðarstofn
operations	aðgerðir
operations	virkjar
order	gráða trés
ordered index	raðaður vísir
outer join eða full outer join	ytri töflutenging
parent row	foreldrafærslur
partial key dependencies	hlutalykilsfallákveða
physical schema	raunlægt skema
pointer	bendir
primary key	frumlykill
projection	dálkaval
pseudotransitive	hálfgegnavirkni
query language	fyrirspurnarmál
query optimisation	bestun fyrirspurna
range	gildisbil
recovery from crashes	endurheimt við kerfishrun
referential integrity	aðkomulyklaskilyrði
reflexivity	sjálfhverfni
relation	tafla
relation schema, table schema	töfluskema
relational algebra	töflualgebra
relational database	töflu-gagna-safn
relational database management system	töflugagnagrunnur
relational model	töflulíkan
restriction	takmörkun
right outer join	hægri ytri töflutenging
row locator	færslustaðgreinir
row, record, tuple	færsla
second normal form, 2NF	annað staðalform
secondary index	aukavísir
selection	færsluval
selection predicate	valskilyrði
set	mengi
set difference	mengjamismunur
shadow paging	skuggasíður
sjá gagnasafn og gagnasafnskerfi	gagnagrunnur
Skrifa Atburðaskrá Fyrir Aðgerð	SAFA
sound	réttar [reglur]
sparse index	gisinn vísir
structured query language	SQL

subset	hlutmengi
systema analysis and design	kerfisgreining
tablescan	töfluskanna so.
third normal form, 3NF	þriðja staðalform
fourth normal form, 4NF	fjórða staðalform
fifth normal form, 5NF	fimmta staðalform
three value logic table	röktafla með þremur rökgildum
transaction ID number	hreyfingarnúmer
transaction log type	tegund atburðar
transaction management	hreyfingastjórnun
transaction, einnig LUW, logical unit of work	hreyfing
transitive dependencies	gegnavirkar fallákveður
transitivity	gegnavirkni
tree depth	dýpt trés
type skilgreint á mismundandi hátt eftir umhverfi	grunnskilgreiningarmengi
type, sjá einnig skilgreiningarmengi	tag
UDT, user defined data types	eigin skilgreind gagnatög
uniform data administration	einsmóta aðgangsstýringar og stjórnun
union	sammengi
union-compatible	sammengishæf
unique key	einkvæmur lykill
update rules	breyting gagna
value	gildi
view	sýn
well structured/well formed	velformuð tafla
well-formed	velformað

D. Atriðisorðasafn - Index

ACID, 6, 91, 92, 93
 aðgerð, 5, 28, 53, 54, 55, 57, 59, 66, 74, 91, 92, 93, 94
 aðgerðarstofn, 53, 54
 aðkomulykill, 27, 31, 32, 33, 34, 35, 83, 88
 aðkomulyklaskilyrði, 27, 33, 36
 aðkomutafli, 27, 32, 33, 35
 annað staðalform, 83, 85, 87
 atburðarnúmer, 91, 94
 aukalykill, 27, 31
 aukavísir, 43, 46
 B+ tré, 4, 43, 49, 52, 107
 barnafærsla, 27, 32, 34
 bendir, 12, 43, 45, 46, 49
 bestun fyrirspurna, 9
 Boyce-Codd staðalform, 83
 breyting gagna, 27, 36
 breytingavandi, 83
 dálkaval, 5, 53, 57
 dálkur, 15, 18, 19, 20, 21, 27, 29, 30, 32, 35, 36, 38, 39, 40, 44, 45, 55, 57, 58, 59, 63, 66, 78, 83, 85, 87, 88, 95, 96, 98
 deiling, 40, 53
 dýpt trés, 43
 eigin skilgreind gagnatög, 27, 36
 eigindi, 15, 27, 37, 39, 83, 85
 einangrun, 91, 92, 93
 eingildir dálkar, 83, 85
 eining, 91, 92
 einkvæmur lykill, 20, 27, 30, 32
 einsmóta aðgangsstýringar og stjórnun, 9
 ekki-lykill, 77, 78
 ending, 91, 92
 endurheimt, 9, 94
 endurtekning gagna, 83
 eyðing gagna, 27, 34
 eyðingarvandi, 83
 færsla, 4, 10, 11, 15, 18, 20, 27, 28, 29, 30, 31, 32, 33, 34, 35, 40, 43, 44, 46, 47, 48, 50, 55, 56, 57, 58, 60, 63, 64, 66, 67, 68, 70, 78, 80, 81, 84, 85, 86, 92, 93, 94, 97, 98
 færslufjöldi töflu, 15
 færslustaðgreinir, 43, 46
 færsluval, 5, 53, 55, 56, 59, 66
 fallákveða, 5, 6, 27, 29, 77, 78, 79, 80, 81, 82, 83, 87, 88
 fallákveður, 5, 6, 27, 29, 77, 78, 79, 80, 81, 82, 83, 87, 88
 fimmta staðalform, 83
 fjórða staðalform, 83
 foreldrafærsla, 27, 32
 frumlykill, 4, 27, 31, 32, 33, 35, 36, 45, 77, 78, 83, 84, 88
 frumreglur Armstrongs, 77
 fullnægjandi [reglur], 77
 fyrirspurnarmál, 9, 11, 25
 fyrsta staðalform, 6, 83, 85, 86, 87
 gagnagrunnur, 3, 9, 10, 83, 92, 93, 94
 gagnaheilindi, 9, 11, 27, 33, 84, 91
 gagnaheilindi grunns, 91
 gagnaheilleiki, 27
 gagnahuld, 9
 gagnasafn, 9, 10, 11, 12, 16, 19, 28, 38, 44, 46, 96
 gagnasafnskerfi, 9, 10, 16, 21, 28, 31, 32, 36, 45, 49, 95
 gagnategund, 15, 16
 gegnvirkar fallákveður, 83, 87, 88
 gegnvirkni, 77, 88
 gildi, 4, 15, 16, 17, 18, 19, 20, 27, 28, 31, 33, 34, 35, 36, 37, 38, 39, 40, 44, 45, 46, 47, 48, 50, 57, 63, 66, 67, 68, 78, 94, 97, 98
 gildisbil, 27
 gildisleysan NULL, 22, 27, 33, 37, 38
 gisinn vísir, 43, 48
 gráða töflu, 15
 gráða trés, 43
 grunnskilgreiningarmengi, 15, 16
 hægri ytri töflutenging, 53
 hálfgegnvirkni, 77
 heilleiki, 91, 92
 hlutalykilsfallákveða, 83, 87
 hlutmengi, 53, 60, 81

hreiðruð segð, 53, 54
 hreyfing, 6, 91, 92, 93, 94
 hreyfingarnúmer, 91
 hreyfingastjórnun, 91
 hrúga, 43, 46
 innri töflutenging, 53, 66, 69
 innsetning gagna, 27
 innsetningarvandi, 83
 jafngildistenging, 53, 66
 keðjuverkun, 27
 kerfisgreining, 27
 klasaður vísir, 4, 43, 46
 klösuð tafla, 43, 46
 krossfeldi, 5, 53, 54, 58, 66
 læsing, 91
 lágmarksendurtekning gagna, 83
 lauf, 43, 49
 lykill, 3, 4, 5, 20, 27, 28, 29, 30, 31, 32, 33, 34, 35, 38, 77, 78, 81, 85, 87, 88
 lykiltlafla, 27, 32, 33, 34
 lyklaskilyrði, 27, 93
 lýsigögn, 15, 16, 17
 mengi, 21, 36, 53, 60, 64
 mengjamargfeldi, 5, 53, 58, 59
 mengjamismunur, 5, 53, 63
 minni þróunarkostnaður, 9
 mögulegur lykill, 27, 30, 31, 35, 77, 78
 niðurbrot, 77
 normalisering, 77, 83
 núllstilling, 27
 óklasæður vísir, 43, 50
 ósamsett eigindi, 83, 85
 raðaður vísir, 43, 46
 raunlægt skema, 15
 reglur Codd's, 83
 reglur Codd's, 15, 22
 rekstraraðili gagnasafnskerfis, 9
 réttar [reglur], 77
 röklegt skema, 15, 24
 röktafla með þremur rökgildum, 37
 rökvirki, 53
 SAFA, 91, 94
 samanburðarmengi, 27, 36
 samanburðarvirkjar, 53
 sammengi, 5, 53, 60, 61, 77
 sammengishæf, 53
 samsettur lykill, 27, 29
 samtíða aðgangur/vinnslur, 9
 segð, 53, 54, 97
 sjálfgefin gildi, 27, 35, 36
 sjálfhverfni, 77
 skilgreiningarmengi, 15, 16, 18, 19, 21, 32, 35
 skilvirkur aðgangur, 9
 skilyrði, 3, 15, 19, 20, 21, 27, 28, 31, 33, 34, 35, 36, 38, 59, 63, 66, 78, 85, 86, 93, 95, 96, 97, 98
 skilyrði skilgreiningarmengis, 27, 36
 skilyrði venslatöflu, 15, 28
 skuggasíður, 91, 94
 skyldleikagreining, 4, 22, 27, 37, 40, 77, 83, 89
 sniðmengi, 5, 53, 64, 65, 66, 69
 SQL, 6, 9, 11, 12, 13, 22, 24, 25, 33, 35, 36, 39, 45, 49, 55, 56, 57, 58, 59, 60, 61, 63, 64, 65, 67, 68, 69, 70, 95, 96, 107, 108, 109
 staðalform, 6, 83, 85, 86, 87, 88, 89
 sundurlæg mengi, 53, 60
 sýn, 6, 15, 18, 19, 22, 24, 32, 43, 45, 46, 54, 55, 83, 84, 85, 86, 87, 88, 96
 tæti fall, 43, 47, 52, 108
 tæti fallsvísir, 43, 47
 tag, 15, 16, 18, 19, 20, 21
 tegund atburðar, 91
 þéttur vísir, 43, 48
 þriðja staðalform, 6, 83, 88
 töflualgebra, 5, 11, 31, 53, 54, 55, 56, 57, 59, 60, 61, 63, 65, 74
 töflugagnagrunnur, 9, 10, 41
 töflulíkan, 15, 19, 24, 25
 töfluskema, 15, 17, 21, 31
 töflutenging, 5, 53, 59, 66, 67, 68, 69
 valskilyrði, 53, 55, 66
 vandi, 83
 velformað, 53, 54, 83, 84
 velformuð tafla, 83
 vensl, 10, 15, 18, 19
 viðbót, 20, 77
 virkja, 54, 55
 virkjar, 53, 54
 vísatafla, 43, 46
 vísauppsetning, 43
 vísir, 4, 9, 15, 18, 20, 24, 28, 31, 32, 33, 34, 35, 40, 43, 44, 45, 46, 47, 48, 49, 50, 51, 96, 97

E. Heimildir

3 Normal Forms Database Tutorial. 16.12.2007, <http://phlonx.com/resources/nf3/>

A Simple Guide to Five Normal Forms in Relational Database Theory. 15.12.2007.
<http://www.bkent.net/Doc/simple5.htm>

Amiri, Khalil. More on indexing: B+ trees. 2.1.2008.
<http://www.doc.ic.ac.uk/~amiri/advDB/Btrees.pdf>

ARIES (Algorithms for Recovery and Isolation Exploiting Semantics). 10.3.2008.
http://en.wikipedia.org/wiki/Algorithms_for_Recovery_and_Isolation_Exploiting_Semantics
Armstrong, William W. Dependency Structures of Data Base Relationships. 1974

B+ tree. 2.1.2008. http://en.wikipedia.org/wiki/B%2B_tree

B-tree. 2.1.2008. <http://en.wikipedia.org/wiki/B-tree>

Clustered Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms190639.aspx>

Codd, E. F. *The relational model for database management: version 2*. 1990. Sótt 9.12.2007.
<http://delivery.acm.org/10.1145/80000/77708/rmdb-codd.pdf?key1=77708&key2=9049127911&coll=ACM&dl=ACM&CFID=9087427&CFTOKEN=72328208>. Addison-Wesley Longman Publishing Co., Inc. Boston, MA,

Codd's 12 rules. 1.12.2007. en.wikipedia.org/wiki/Codd%27s_12_rules

Concurrency control. 9.12.2007. http://en.wikipedia.org/wiki/Concurrent_access

Connolly, Thomas og Begg, Carolyn. 2005. Database Systems: A Practical Approach to Design, Implementation and Management. 4ða útgáfa. Addison-Wesley

Creating Indexes. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms190197.aspx>

Criticisms of SQL. 9.12.2007. en.wikipedia.org/wiki/SQL#Criticisms_of_SQL

Data independence. 9.12.2007. http://en.wikipedia.org/wiki/Data_independence

Data integrity, 9.12.2007. http://en.wikipedia.org/wiki/Data_integrity

Database History. 9.12.2007. en.wikipedia.org/wiki/Database#History

Database Management System. 9.12.2007.
en.wikipedia.org/wiki/Database_management_system.

- Database normalization. 20.3.2008. http://en.wikipedia.org/wiki/Database_normalization
- Database Security. 9.12.2007. bitpipe.com/rlist/term/Database-Security.html
- Database Systems and SQL Server. 16.12.2007.
<http://www.raritanval.edu/departments/cis/full-time/Schwarz/sql/>
- Database transaction. 13.12.2007. http://en.wikipedia.org/wiki/Database_transaction
- Elmasri, Tamez og Navathe, Shamkant. 2007. *Fundamentals of Database Systems*. 5ta útgáfa. Pearson. Boston MA
- Estimating the Size of a Clustered Index. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms178085.aspx>
- Estimating the Size of a Heap. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms189124.aspx>
- Estimating the Size of a Nonclustered Index. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms190620.aspx>
- Functional dependency. 20.3.2008. http://en.wikipedia.org/wiki/Functional_dependency
- General Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms191195.aspx>
- Hagstofa Íslands. 3.12.2007. *Táknmál þjóðskrár*. <http://www.thjodskra.is/taknmal/>
- Hartley, Michael. *Glossary of Database Terms*. 13.12.2007. <http://www.dr-mikes-maths.com/database-glossary.html>
- Hash function (tætiföll). 1.2.2008. http://en.wikipedia.org/wiki/Hash_function
- Index Design Basics. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms179560.aspx>
- Index Disk Space Example. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms191183.aspx>
- Introduction to Data Modeling, 16,12,2007. <http://www.utexas.edu/its-archive/windows/database/datamodeling/>
- JCC's SQL Standards Page. 19.12.2007. <http://www.jcc.com/sql.htm>
- Linn, Ann. 2000. *History of Database Protection, Legal Issues of Concern to the Scientific Community*. 12.12.2007. globalchange.gov/policies/linn.html
- Lock. 10.3.2008. http://en.wikipedia.org/wiki/Lock_%28computer_science%29
- Mannino, Michael V. 2005. *Database Design, Application Development, and Administration*. 3ja útgáfa. McGraw Hill Higher Education, Boston, MA
- MS SQL Server 2005 Home. 12.12. 2007. <http://www.microsoft.com/sql/default.mspix>

- MS SQL Server Security. 9.12.2007.
database.ittoolbox.com/topics/t.asp?t=371&p=452&h1=371&h2=452
- Nonclustered Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms179325.aspx>
- Normalization Basics. 15.12.2007.
<http://databases.about.com/od/specificproducts/a/normalization.htm>
- Normalization, 15.12.2007. <http://www.troubleshooters.com/littstip/ltnorm.html>
- Persónuvernd. 12.12.2007. <http://www.personuvernd.is/>
- Query optimization. 9.12.2007. http://en.wikipedia.org/wiki/Query_optimisation
- Relational Algebra. 9.12.2007. http://en.wikipedia.org/wiki/Relational_algebra
- Rules of Data Normalization. 16.12.2007. <http://www.datamodel.org/NormalizationRules.html>
- Shadow Paging. 10.3.2008. http://en.wikipedia.org/wiki/Shadow_paging
- SQL Database Glossary. 13.12.2007.
<http://www.sqlstrings.com/Database-Glossary.htm>
- SQL Server 2005 Books Online (September 2007). Index Basics. 12.12.2007.
<http://msdn2.microsoft.com/en-US/library/ms190457.aspx>
- Susan Anderson-Freed. *B+ TREES*. 2.1.2008. <http://www-users.itlabs.umn.edu/classes/Spring-2006/csci4707/B+Trees.pdf>
- Taylor, Allan G. 2001. *SQL for Dummies*. 5ta útgáfa. Wiley Publishing, Inc. Indianapolis. Indiana
- Taylor, Allan G. 2003. *Database Development for Dummies*. IDG Books Worldwide, Inc. Foster City. California
- The Rise of Relational Databases. 9.12.2007. nap.edu/readingroom/books/far/ch6.html
- Transaction log. 10.3.2008. http://en.wikipedia.org/wiki/Transaction_log
- Types of Indexes. 2.1.2008. <http://msdn2.microsoft.com/en-us/library/ms175049.aspx>
- Umanath, Narayan S. og Scamell, Richard W. 2007. *Data Modeling and Database Design*. Thomson Course Technology, Boston MA
- Unique Index Design Guidelines. 2.1.2008. <http://technet.microsoft.com/en-us/library/ms187019.aspx>