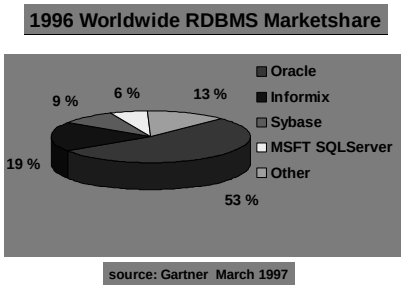
SQL

- SQL:llä voidaan...
  - määritellä ja muokata tietokantaa ja sen käyttöoikeuksia
  - virittää tietokannan talletusrakenteita
  - hakea tietoa tietokannasta
    - näytölle tai tiedostoon
    - sovellusohjelman käyttöön
  - tehdä päivityksiä tietokantaan (muuttaa dataa)
    - vuorovaikutteisesti
    - sovellusohjelman kautta

SQL -ohjelmistojen markkinaosuuksia



SQL - historiaa

- System R (IBM:n tutkimuslaboratorio) 1972-1973
- SEQUEL = Structured English QUEry Language
- Myöhemmin SQL
- Kaupallisia toteutuksia n. 1980 ->

SQL - standardointi

- ANSI 1986 {noin 40 sivua}
- ISO 1987
- ISO 1989 { + noin 20 sivua}
- ISO/ANSI SQL-92 {noin 1000 sivua}
- SQL-3 (työnimi) yhä tekeillä, pilkottu osiin
  - SQL/CLI (call level interface) -1996
  - SQL/PSM (persistent stored modules) - 1997

SQL-tietokanta

- SQL-tietokanta muodostuu yhden tai useamman kaavion (schema) määrittelemistä tauluista (table)
- Kullakin kaaviolla on omistaja, joka omistaa myös kaavion määrittelemät taulut. Taulu muodostuu riveistä (row)
- Taulu vastaa relaatiomallin relaatiota, mutta
  - sallii etenkin kyselyiden tuloksissa samanlaisen rivin toistumisen (duplikaatit) {monijoukko, multiset}

SQL- määrittelykielenä

- Käyttäjät
- Kaaviot
- Perustaulut (base table)
  - määritellään
  - rivit fyysisesti olemassa
- Johdetut taulut (derived table)
  - tästä myöhemmin
- DML Data Manipulation Language (tiedon) käsittelykieli DDL Data Definition Language (tiedon) määrittelykieli SQL:ssä molemmat!

SQL-kielestä

- SQL-kieli on avainsanat, taulu- käyttäjä ja sarakenimet voi kirjoittaa joko suur- tai pienaakkosina
  eli “select merkki” ≡ “SELECT MerKKI”
- Tietokannassa olevan datan suhteen kieli on kuitenkin herkkä kirjainmuodolle eli
  “Merkki=‘Ford’ ” ≠ “Merkki=‘FORD’ “

SQL-määrittelykielenä: käyttäjät

- create user *username* identified by *password*
- Lauseessa voi olla lisänä tkhj-kohtaisia lisämääreitä
- Käyttö edellyttää tietokannanhoitajan (database administrator, dba) oikeuksia.
- Tietokannanhoitaja on eräs käyttäjärooli. Käyttäjille voidaan myöntää rooleja
  - grant *role* to *user* -lauseella
- alter user *username* identified by *password*
  - *salasanan vaihto, sallittu kaikille*

SQL-määrittelykielenä: käyttäjät

- Käyttäjistä pääsee helposti eroon komennolla
- drop *username* [cascade]
  - hakasulkeet osoittavan valinnaisen osan
  - jos cascade on mukana käyttäjän omistamien taulujen olemassaolo ei estä käyttäjän eliminointia vaan ne hävitetään samalla

SQL-määrittelykielenä: käyttäjät

- Käyttäjällä on kaikki oikeudet omistamiinsa tauluihin. Hän voi luovuttaa oikeuksia myös muille komennolla
  grant *operaatiot* on *kohde* to
  {*username* | *role* | public }
  operatiot::= all, select, insert, update, delete, …
  kohde::= *taulu*, *johdettu taulu*, *sarakejoukko*,

  Luvat voi myös perua:
  revoke *operaatiot* on *kohde* from
  {*username* | *role* | public }

SQL-määrittelykielenä: kaaviot

- create *schema* authorization *username*
  - nimeää kaavion ja kytkee sen omistajaan
  - ei tarjolla kaikissa tkhj:ssä
- Standardin mukaan onnistuisi myös
- drop *schema* [cascade],
- joka hävittäisi koko kaavion määrittelemän osan tietokantaa

SQL-määrittelykielenä: Taulut

- create table *tablename* (
  column definition 1, ….,
  column definition n
  [, *constraint 1*, …])

sarakemäärittely ::=
column_name datatype [not null]
[default *value*] [*column constraint* …]

SQL-määrittelykielenä: Taulut - sarake

- *Datatype* ilmoittaa sarakkeen tietotyypin. Tietotyyppiin saattaa liittyä kokomääritys
- Merkkijonot
  - character [varying] [(*maksimipituus*)]
    - kiinteän (myös char) tai vaihtuvanpituinen (myös varchar) merkkijono,
    - maksimipituuden yläraja järjestelmäkohtaista
    - puuttuva maksimipituus = 1
    - nimi varchar(40),
    - hetu char(11),

SQL-määrittelykielenä: Taulut - sarake

- Binäärinen tieto
  - bit [varying] {(*maksimipituus)*}
    - binääristä tietoa, esim. kuvia
    - maksimipituuden yläraja järjestelmäkohtaista
    - joissain järjestelmissä on erotettu erittäin pitkät merkkijonot (large varchar, long) ja binääritiedot (large binary, blob) omiksi tietotyypeikseen
      - *esim Oraclessa varchar maksimi 2000 ja long 64G*
      - *ei voi välttämättä käyttää hakuehdoissa*

SQL-määrittelykielenä: Taulut - sarake

- Tarkkoja numeerisia tyyppejä
  - numeric [(*precision* [,*scale*])]
  - decimal [(*precision* [,*scale*])] = dec
  - integer = int (4 tavua)
  - smallint (2 tavua)
  - precision = kokonaispituus numeropaikkoina
    - » Oraclella max 44 digits
  - scale = desimaaliosan pituus
  - decimal (5,2) maksimiarvo: 999.99

SQL-määrittelykielenä: Taulut - sarake

- Likiarvoisia numeerisia tyyppejä:
  - float (8 tavua?),
  - real (4 tavua),
  - double precision (8 tavua, järjestelmäkohtainen),
  - eksponentti ja mantissa erikseen

SQL-määrittelykielenä: Taulut - sarake

- Aikoja
  - Date päiväys
  - Time kellonaika
  - Timestamp päiväys ja kellonaika (=Oraclella Date)
  - Interval aikaero
  - Aikoja voidaan verrata ja niillä voi laskea
    this_day date,
    this_day + 3 on kolmen päivän päästä

SQL-määrittelykielenä: Taulut - sarake

- Not null - määre kieltää tyhjäarvot
  - määrettä on käytettävä pääavainsarakkeiden yhteydessä
    - reknro varchar(8) not null,
- Default -määreellä voidaan määritellä oletusarvo, joka sarakkeeseen sijoitetaan, jos lisäysoperaation yhteydessä ei anneta sarakkeelle arvoa
  - oletusarvoisesti tyhjäarvo (null)

SQL-määrittelykielenä: Taulut

- Eheysehdot liittävät tauluun rajoitteita.
  - Primary key osoittaa pääavaimen,
  - Foreign key puolestaan viiteavaimen
    - taulu voi sisältää monta viiteavainta, mutta enintään yhden pääavaimen, joka voi koostua useasta sarakkeesta
    - yksisarakkeisten kohdalla rajoitteet voidaan esittää sarakekohtaisina, muuten taulukohtaisena

```
create table auto (
   reknro varchar(8) not null primary key,
   ….)
```

SQL-määrittelykielenä: Taulut

```
Create table harjoitusryhma (
    kurssitunnus varchar(10) not null,
    ryhmanro numeric(3) not null,
    …..,
    primary key (kurssitunnus, ryhmanro),
    foreign key (kurssitunnus) references kurssi(kurssitunnus)
)
```

SQL-määrittelykielenä: Taulut

- Viiteavain määrittelyyn voidaan liittää toimintasääntö operaation rikkoessa viite-eheyden

  foreing key (*sarakkeet*) references *taulu* [(*sarakkeet2*)]
  [on delete {restrict | cascade |nullify}]
  [on update {restrict | cascade |nullify}]

  viitteen kohde katoaa:
  restrict estää rikkovan operaation (oletus)
  cascade vyöryttää, poistaa tai muuttaa viittaavat rivit
  nullify tyhjentää viittaukset

SQL-kysely perusteet

Kyselyn yleisrakenne:

```
select tulostietomäärittely
   from taulukkeet
   [where valintaehdot]
   [group by ryhmitystekijät]
   [having ryhmärajoitteet]
   [order by järjestysperusta]
```

Kysely tuottaa nimettömän tulostaulun.

SQL-kyselyt

```
select merkki, reknro
from auto
where vmalli=1996 and
        vari =‘punainen’ and merkki like ‘Fo%’
order by merkki, reknro
```

- Vuoden 1996 mallia olevien punaisten merkiltään ‘Fo’-alkuisten autojen merkki ja rekisterinumero merkin ja saman merkin sisällä rekisterinumeron mukaan järjestettynä

SQL-kyselyt

```
select merkki, reknro
from auto
where vmalli=1996 and
        vari =‘punainen’ and merkki like ‘Fo%’
order by merkki, reknro
```

‘projektio’

- Vuoden 1996 mallia olevien punaisten merkiltään ‘Fo’-alkuisten autojen merkki ja rekisterinumero merkin ja saman merkin sisällä rekisterinumeron mukaan järjestettynä

SQL-kyselyt

```
select merkki, reknro
from auto
where vmalli=1996 and
        vari ='punainen' and merkki like 'Fo%'
order by merkki, reknro
```

valinta

- Vuoden 1996 mallia olevien punaisten merkiltään 'Fo'-alkuisten autojen merkki ja rekisterinumero merkin ja saman merkin sisällä rekisterinumeron mukaan järjestettynä

SQL-kyselyt: Tulostietomäärittely

- Tulostietomäärittely määrittelee tulostaulun sarakkeiden nimet ja niiden sisällön
  - koostuu jonosta pilkulla erotettuja alkiomäärittelyjä
  - alkiomäärittely:
    lauseke [[AS] *uusinimi*]
  - lauseke määrittele sisällön ja oletusnimen. Se on yksinkertaisemmillaan sarakenimi tai vakio, mutta voi sisältää laskentaoperaatioita ja funktioita - alkion nimeksi tulee esim. Oraclessa oletusarvoisesti lauseke sellaisenaan ellei sitä nimetä uudelleen

SQL-kyselyt: Tulostietomäärittely

- Lausekkeessa, esim:
  - Pituus, — sarakenimi
  - Paino / Pituus *0.25 — aritmeettinen lauseke
  - Etunimi || ' ' || Sukunimi — merkkijonojen liimaus
  - 30 — numeerinen vakio
  - length(Sukunimi) — merkkijono funktio
  - substring(Sukunimi,1,1) — osamerkkijono
  - null — tyhjäarvo
  - ifNull(Sukunimi,'') — tyhjäarvon korvaus

SQL-kyselyt

- SQL:ssä on tarjolla muiden ohjelmointikielten tapaan joukko valmiita funktioita tuloksen muokkaukseen
  - merkkijonofunktiot (täyttö, liimaus, korvaus, muunnos, etsintä, osan eristys, jne)
  - numeeriset funktiot (pyöristys, katkaisu, itseisarvo,..)
  - päiväysfunktiot (päivän nimi, kuukauden nimi, kuukauden päästä, kuun viimeinen ,…)
  - yleiset funktiot (valinta arvon perusteella, tyhjäarvon korvaus, ..)
  - systeemifunktiot (käyttäjätunnus,nykyhetki, ...)
  - **ongelma: funtioiden nimet ja valikoima vaihtelevat järjestelmäkohtaisesti**

SQL-kyselyt - vaihtelua

- Oraclessa: (tulos suomeksi)

```
select
   'Tänään on '||  lower( rtrim(to_char(sysdate,'DAY'))) ||
   to_char(sysdate,' DD.') || ' '
   || lower(rtrim(to_char(sysdate,'MONTH'))) || 'ta'
from dual                (dual olkoon taulu, jossa on yksi rivi)
```

- Solidissa: (tulos englanniksi)

```
select
   'To-day is '|| dayName(curDate()) ||' '||
   convert_Char(DayOfMonth(curDate())) ||'th of '||
   MonthName(curDate())
from dual
```

SQL-kyselyt

- Tulostietomäärittely: * (tähti)
  - tulostauluun valitaan kaikki from osassa mainittujen 'taulukkeiden' sarakkeet
  - joissain järjestelmissä sallitaan myös muoto taulu.* jolloin mukaan tulevat kaikki tuon taulun sarakkeet

SQL-kyselyt

- Tarkentaminen
  - jos sarake esiintyy useassa kyselyn from-osan taulussa on siihen viitattaessa käytettävä tarkennetta.
  - Sarakeviittaus on muotoa
    [[[tietokanta.]kaavio.]taulu.]sarakenimi
  (jos kaaviota ei ole erikseen nimetty kaavio=taulun omistajan tunnus)

tktl.laine.auto.reknro:
kannan *tktl* käyttäjän *laine* taulun *auto* sarake *reknro*

SQL-kyselyt

- Tulostietoluettelon elementeille lasketaan normaalitapauksessa arvo jokaista valintaehdot täyttävää riviyhdistelmää kohden

```
select merkki
from auto
where vmalli=1996 and
        vari ='punainen' and merkki like 'Fo%'
        order by merkki
```

Jos taulussa auto olisi 100 punaista vuoden 1996 Fordia tulisi merkki 'Ford' tulostauluun 100 kertaa.

Toimii siis toisin kuin relaatioalgebran projektio

SQL-kyselyt

- Projektion kaltainen toistuvien arvojen karsinta saadaan aikaan liittämällä tulostietomäärittelyn alkuun avainsana **distinct**

```
select distinct merkki
from auto
where vmalli=1996 and
        vari ='punainen' and merkki like 'Fo%'
        order by merkki
```

Nyt Ford tulisi tulokseen vain kerran

SQL-kyselyt

- Normaalisti esimerkiksi raporttiin ei haluta samanlaisia, samaa asiaa ilmaisevia rivejä moneen kertaan. Jos tällaisia voisi tulla, on ne syytä karsia distinct-määreellä.
- Distinct-määrettä ei tarvita, jos esimerkiksi avaimen mukanaolo takaa rivien erilaisuuden
- Aina ei ole syytä karsia toistuvia arvoja
  - olkoon henkilö(…,palkka,…) taulu, josta halutaan laskea työntekijöiden keskipalkka. Tällöin pitää palkat hakea
  - select palkka from työntekijä, eikä
  - select distinct palkka from työntekijä.

SQL-kyselyt

- select palkka from työntekijä,
  - 9000, 9000,9000, 9000,12000,25000
  - keskiarvo: 12167
- select distinct palkka from työntekijä
  - 9000,12000,25000
  - keskiarvo: 15333

SQL-kyselyt

- From-osassa on oltava mukana ainakin ne taulut, joista halutaan tietoja tulokseen
- luettelossa 'tauluke' (table constructor, taulun kaltainen olio):
  - **taulu** (useissa SQL toteutuksissa vain tämä)
  - alikyselyn tulostaulu
  - liitostaulu = liitoksen tulostaulu

SQL-kyselyt

- Jos from-osassa on vain yksi taulu:
  - tulokseen mukaan ne taulun rivit, jotka täyttävät where- osassa annettavan valintaehdon.
  - *'Select * from taulu'* ilman valintaehtoa tuottaa koko taulun sisällön epämääräisessä järjestyksessä. (epämääräinen järjestys = ei taata että rivit tulevat aina samassa järjestyksessä)
  - jos where-osassa on ehtoja ne vaikuttavat kuten valintaehdot relaatioalgebassa

SQL-kyselyt

- Valintaehto on SQL:ssä monipuolinen.
- Vertailtavina sarakkeet, vakiot, näistä muodostetut lausekkeet, funktiot
- Vakion esitysmuoto:
  - merkkitieto ja päiväykset hipsuissa ('arvo')
    numeeriset ilman hipsuja (9999)
  - päiväyksen esitystapa riippuu järjestelmästä ja kieliasetuksista:
    - Solid: '1999-9-22', Oracle: asetuksilla valittavissa, esim: '22.9.1999' tai '22-SEP-1999' tai '1999-9-22' tai .....

SQL-kyselyt

- Vertailuoperaatoreina
  - =,<,>,<=,>=,<> (myös muodossa !=)
  - arvo in (joukko)
  - arvo between ala and ylä =
    arvo>= ala and arvo<=ylä
  - merkkijonoille: arvo like maski
    maskissa % on mikä tahansa merkkijono ja _ (alaviiva) mikä tahansa merkki
  - arvo is null, arvo is not null

SQL-kyselyt

- Reknro like 'OG%'
  - {kaikki OG alkuiset}
- Reknro like ' _ _ _ _23'
  - {kuusimerkkiset 23 loppuiset}
- merkki in ('Ford', 'Opel', 'Volvo') =
  - merkki='Ford' or merkki='Opel' or merkki='Volvo'
- Paikkakunta <'Helsinki'
  - aakkosvertailu (suhtautuminen paikallismerkistöön voi vaihdella)