

3. Taulujen määrittely ja muuttaminen

DDL: Taulujen luonti, muutos ja poisto

DML: taulujen tietojen ylläpito

Tapahtumien (transaktioiden) hallinta

Näkymät, synonyymit ja muut tietokantaobjektit



JYVÄSKYLÄN
AMMATTIKORKEAKOULU

Osaaminen kilpailukyvyksi

Taulujen perustaminen

- Taulun perustajalla täytyy olla valtuudet (GRANT) tietokantaan
- Kullekin taulun sarakkeelle annetaan nimi ja tietotyyppi (+ maksimipituus ja mahdolliset desimaalit)
- Tarvittaessa määritetään rajoitteita (NOT NULL, PRIMARY KEY, FOREIGN KEY, UNIQUE, DEFAULT ja CHECK)

SQL-esimerkki: taulun luonti

- Taulu luodaan CREATE TABLE –käskyllä:

```
CREATE TABLE YRITYS (  
  YritysID          smallint NOT NULL,  
  YritysNimi        varchar(80),  
  Toimialakoodi     char(10),  
  YrLisaysPvm       date DEFAULT Current_date,  
  CONSTRAINT pk_YritysID PRIMARY KEY (YritysID))
```

- Viite-eheys REFERENCES-määreellä, ks. Rajoitteet

```
CONSTRAINT fk_Toimialakoodi FOREIGN KEY (Toimialakoodi)  
REFERENCES TOIMIALA (Toimialakoodi)
```

Muutamia SQL-92-standardin tietotyyppejä

- CHAR [(pit.), kiinteä, oletuspit. 1]
- VARCHAR [(pit.), vaihtuvanmitt. merkkijono]
 - VARCHAR on sama kuin CHAR VARYING
- NUMERIC [(koko pituus [,desimaaliosassa])]
 - NUMERIC on sama kuin DECIMAL
- INTEGER [kokonaisluku, pit. tuotekoht.]
 - Voidaan lyhentää INT
- DATE [vuosi (0001-9999), kk ja päivä]
- TIME [(pit.), tunti, minuutti ja sekunti]
- BLOB [SQL-99, binary large object]

Ocelot-tietokantatuote sis. SQL2:n ja osan SQL3:n tietotyypeistä

Create Table

Table name and column descriptors. "OK" to move SQL statement to main screen.

Schema Name

Table Name

Checks

Column Name

Data Type

- ☐ BIT
- ☐ BIT VARYING
- ☐ BLOB
- ☐ BOOLEAN
- ☒ CHAR
- ☐ CHAR VARYING
- ☐ CLOB
- ☐ DATE
- ☐ DECIMAL
- ☐ DOUBLE PRECISION
- ☐ FLOAT
- ☐ INTEGER
- ☐ INTERVAL
- ☐ NUMERIC
- ☐ REAL
- ☐ REF
- ☐ SMALLINT
- ☐ TIME
- ☐ TIME WITH TIME ZONE
- ☐ TIMESTAMP
- ☐ TIMESTAMP WITH TIME ZONE

Description Length

Character Set

Collation

☐ Primary?

☐ Not Null?

Default

CREATE TABLE koe(koe CHAR)

SQL:2003 – muutokset aikaisempaan standardiin

- Uudet tietotyypit:
 - BIGINT (tarkkuus tuotekohtainen, kuitenkin aina $BIGINT \geq INTEGER \geq SMALLINT$)
 - MULTiset (voidaan luoda ARRAY-tyyppinen kokoelma)
- Kaksi tietotyyppiä poistettu:
 - BIT
 - BIT VARYING
- Sequence
- CREATE TABLE LIKE ...
- Ym.

Oraclen tietotyyppejä

- CHAR [255 tavua, kiinteäpituuksinen]
- VARCHAR2 [2 kt (4 kt Oracle 8:ssa)]
- NUMBER [numeerinen 1×10^{-130} – $9,99 \times 10^{125}$]
- LONG [2 Gt]
- RAW [255 tavua, binaarinen]
- LONG RAW [2 Gt, binaarinen]
- DATE [pvm ja aika, kiinteäpituuksinen]
- ROWID [rivin ID:n muuttujatyyppi]
- NCHAR ja NVARCHAR2: kansallisen kielen tietotyyppi
- CLOB: yksitavuinen, NCLOB: yksi- tai monitavuinen, 4 Gt
- BFILE: osoitin tallennettuun ulkoiseen bin.tiedostoon
- Oraclen oma (esim. SDO_GEOMETRY for Oracle Spatial)

SQL Serverin (Transact-SQL) tietotyyppejä

- `char [ ( n ) ]`
 - vakiomittainen merkkietieto, $n = 1 \dots 255$
- `varchar [ ( n ) ]`
 - vaihtuvamittainen merkkietieto, $n = 1 \dots 255$
- `binary [ ( n ) ]`, `varbinary [ ( n ) ]`
- `datetime`, `smalldatetime`
 - pvm-typit
- `text`, `image`
 - suuret tietomäärät (maks 2 Gtavua)
- `bit`
- `timestamp`
 - muuttuu riviä päivitettäessä
- Kokonaisluvut:
 - `int` (32 bitin)
 - `smallint` (16 bittiä)
 - `tiny` (0 ... 255)
- Desimaaliluvut:
 - `decimal [ ( p, [ , s ] ) ]` **tai** `numeric [ ( p, [ , s ] ) ]`
 - p = tarkkuus (1...38) , s = skaala, desimaaliosien määrä
 - `float [ ( n ) ]`
 - liukuluku, tarkkuus 15 numeroa
 - `real`
 - liukuluku, tarkkuus 7 numeroa
- Raha:
 - `money`, `smallmoney`

Tuotekohtaisista eroista: http://troels.arvin.dk/db/rdbms/#data_types

SQL-esimerkki: taulun muutos

- Taulu muutetaan ALTER TABLE –käskyllä, esim. sarakkeen lisäys (loppuun)
`ALTER TABLE yritys`
`ADD COLUMN maa char (20)`
- ALTER TABLE tuli osaksi SQL-standardia vasta SQL3:ssa
- Katso tuotekohtaisesti, voidaanko lisätä kerralla useita sarakkeita tai sarake johonkin väliin
- Sarakkeen muutos: `ALTER COLUMN` ja poisto: `DROP COLUMN`, ks. oma tuotteesi, esim.

`ALTER TABLE yritys`

`DROP COLUMN maa CASCADE`

Eheysvaatimukset

- Tietokantaan kohdistuu kolmen tyyppisiä eheysvaatimuksia:
 - Avaineheys: taululla tulee olla perus- eli pääavain (PK)
 - Viite-eheys: viiteavaimelle tulee löytyä arvo viitattavan taulun pääavaimesta
 - Attribuuttieheys: attribuuttien tulee kuulua sallittuun joukkoon
- Määritetään nk. rajoitteella (ks. seur. kalvo)
- Palvelin tarkistaa eheysvaatimusten toteutumisen

Rajoitteet eli pakotteet (constraints)

- Määritellään CREATE TABLE tai ALTER TABLE -komennoissa
- Rajoitteita voidaan määritellä sarakkeille tai tauluille
- Rajoitteita ovat:
 - avaimet: PRIMARY KEY, FOREIGN KEY ja UNIQUE
 - oletusarvot ja raja-arvotarkistukset: DEFAULT ja CHECK
- esimerkkejä:

ALTER TABLE dept

ADD CONSTRAINT pk_deptno PRIMARY KEY (deptno),

ADD priority SMALLINT DEFAULT 1,

ADD CONSTRAINT chk_sukup CHECK (sukupuoli IN
(‘M’,’N’))

Primary Key, Unique ja Foreign key

- Primary Key (`CONSTRAINT pk_sarake PRIMARY KEY (sarakenimi1 [,sarakenimi2])`
 - Yksiköi taulun rivit
 - Sarakkeiden tulee olla Not Null -tyyppisiä
 - Käytetään hyväksi viite-eheyttä luotaessa
- Unique (`CONSTRAINT sarakenimi UNIQUE`)
 - Luo yksilöivän indeksin (oletus on ei klusteroitu)
 - Sarakkeet voivat sisältää Null-arvoja
 - Voi olla useita / taulu
- Foreign Key (`CONSTRAINT fk_sarake FOREIGN KEY (sarakenimi) REFERENCES vanhempi (perusvain) )`
 - Luo viite-eheyden taulujen välille
 - Voi sisältää useita sarakkeita

Viiteavaimen lisäys ja poistojen/päivitysten vyörytys

- Viite-eheys voidaan määritellä taulua luotaessa tai jälkeenpäin ALTER TABLE -käskyllä
- Esim. Puhelin-taulu, jonka vanhempi on Henkilo; jos henkilö poistetaan niin myös henkilön puhelimet poistetaan (ON DELETE CASCADE)

```
ALTER TABLE Puhelin  
  ADD CONSTRAINT fk_HenkiloID  
  FOREIGN KEY (HenkiloID)  
  REFERENCES Henkilo (HenkiloID)  
  ON DELETE CASCADE
```

- Viisi vaihtoehtoa: CASCADE, SET DEFAULT, SET NULL, NO ACTION, RESTRICT, joista kaksi viimeistä estävät poiston, jos vanhemmalla on lapsia
- Myös päivitykset voidaan vyöryttää: ON UPDATE CASCADE

Laskuri: automaattisesti kasvava sarake

- Taululle voidaan määritellä sarake, joka on automaattisesti kasvava numero
- Määrittelyssä annetaan alkuarvo ja lisäys
- Standardin mukaan laskuri luodaan joko IDENTITY-määreellä tai SEQUENCE-käskyllä (MySQL:ssä AUTO_INCREMENT-määreellä)
- Ominaisuus voidaan luoda myös CREATE TABLE, ALTER TABLE tai SELECT INTO –komennon yhteydessä
- Laskuri toimii usein perusavaimena

Laskurin luonti, 2 vaihtoehtoa

- Esimerkki 1:

```
CREATE TABLE tuote (  
  tuoteID INTEGER  
  GENERATED ALWAYS AS  
    IDENTITY  
  -- ja lisäksi esim.  
  START WITH 100  
  INCREMENT 1  
  MINVALUE 100  
  NO MAXVALUE  
  NO CYCLE,  
  ... )
```

- Esimerkki 2:

```
CREATE SEQUENCE  
  tuoteID AS INTEGER  
  START WITH 1  
  INCREMENT BY 1  
  MAXVALUE 100000  
  MINVALUE 1
```

SQL:2003 - GENERATED ALWAYS

- Esimerkki:

```
CREATE TABLE employees (  
  EMP_ID INTEGER,  
  SALARY DECIMAL(7,2),  
  BONUS DECIMAL(7,2),  
  TOTAL_COMP GENERATED ALWAYS AS (  
    SALARY + BONUS ),  
  HR_CLERK GENERATED ALWAYS AS (  
    CURRENT_USER )  
)
```

Taulun poisto

- Taulun poisto tapahtuu DROP TABLE -käskyllä, esim.

`DROP TABLE temp;`

- Tällöin vyörytetään poistot, jos CASCADE DELETE on lapsitauluun määritetty, muutoin poisto estetään (jos viite-eheys on asetettu)
- Samalla poistuu taulussa olevat tiedot (olihan siis backup tallessa tai ROLLBACK-käsky hanskassa?)