

SQL - Tietokannan ylläpito

- n SQL sisältää operaatiot tietokannan sisällön muodostamiseen ja ylläpitoon:
- n insert - uusien rivien vienti tauluun
- n delete - rivien poisto
- n update - rivien muutos

1

SQL - Tietokannan ylläpito

- n Insert lauseella on kaksi muotoa:
- n insert into taulu [(sarakenimet)] values (arvot)
 - n tällä muodolla lisätään yksi rivi ja arvot annetaan vakioina tai vakioihin perustuvina lausekkeina
- n insert into taulu [(sarakenimet)] kysely
 - n tällä muodolla kyselyn tulosrivit lisätään tauluun

2

SQL - Tietokannan ylläpito

```
CREATE TABLE kurssi (  
  koodi numeric(8) NOT NULL ,  
  nimi varchar(40) NOT NULL ,  
  opintoviikot numeric(5,1) NOT NULL ,  
  luennoija varchar(12),  
  PRIMARY KEY (koodi ),  
  FOREIGN KEY (luennoija) REFERENCES  
  opettaja)  
  
insert into kurssi values  
(1234,'Tietokantojen perusteet',2,'HLAINE');  
n lisää tauluun kokonaisen rivin
```

3

SQL - Tietokannan ylläpito

- n Jos luennoijaa ei tiedetä, voidaan lisäys tehdä seuraavasti:

```
insert into kurssi values  
(1234,'Tietokantojen perusteet',2,NULL); tai  
  
insert into kurssi (koodi, nimi, opintoviikot)  
values (1234,'Tietokantojen perusteet',2);  
  
n sarakeluetteloä käytetään siis silloin kun annetaan  
vain osa sarakkeista – käytöllä voi myös varautua  
taulun rakenteen muutoksiin (sarakeluetteloä ei  
toistaiseksi voi käyttää Trainer-harjoituksissa)
```

4

SQL - Tietokannan ylläpito

- n Jos luennoijan tunnusta ei tiedetä, voitaisiin lisäys tehdä seuraavasti (kaikki kyselyn tulosrivit lisätään):

```
insert into kurssi  
select (1234, 'Tietokantojen perusteet', 2, opetunnus)  
from opettaja  
where nimi='Laine Harri';  
  
n Tämä toimii odotetusti, jos kannassa on vain yksi tämän  
niminen opettaja, muuten lisäys kaatuu avaimen  
yksikäsitteisyysvirheeseen, sillä lisäys epäonnistuu, jos se  
rikkoo eheysehtoja
```

5

SQL - Tietokannan ylläpito

- n Kurssille 'Ohjelmoinnin perusteet' ilmoittautuneiden opiskelijoiden kopiointi kurssin 'Java-ohjelmointi' vastaaviin ryhmiin:

```
CREATE TABLE ilmoittautuminen (  
  kurssikoodi numeric(8) not null,  
  ryhmänro numeric(2) not null,  
  opisknro numeric(5) NOT NULL ,  
  ilm_ aika date NOT NULL ,  
  PRIMARY KEY (opisknro, kurssikoodi) ,  
  FOREIGN KEY (kurssikoodi, ryhmänro) REFERENCES  
  harjoitusryhma on delete cascade,  
  FOREIGN KEY (opisknro) REFERENCES opiskelija )
```

6

SQL - Tietokannan ylläpito

Oracle-SQL:llä

```
Insert into ilmoittautuminen  
select java.kurssikoodi, ryhmänro, opisknro, sysdate  
from kurssi java, kurssi ohpe, ilmoittautuminen  
where java.nimi='Java-ohjelmointi' and  
ohpe.nimi='Ohjelmoinnin perusteet' and  
ohpe.koodi=ilmoittautuminen.kurssikoodi;
```

- Luodaan uusia rivejä ilmoittautuminen tauluun: ryhmä- ja opiskelijanumero otetaan Ohjelmoinnin perusteet kurssin liittyviltä ilmoittautumisriveiltä, kurssikoodiksi otetaan java-ohjelmoinnin kurssikoodi ja päiväkseksi nykyhetki.

7

SQL - Tietokannan ylläpito

- Rivien muutokset (update)

```
update taulu  
set sarake1=lauseke1 [, sarake2=lauseke2, ...]  
[where kohteen rajausehdot]
```

- samalla kertaa voi muuttaa useiden sarakkeiden sisältöä,
- muutetaan kaikki where-ehdon täyttävät rivit
- jos ehto puuttuu muutetaan kaikki taulun rivit

8

SQL - Tietokannan ylläpito

- Muutetaan kurssin Java-ohjelmointi opintoviikkomäärä opintopisteiksi kertomalla määrä kahdella

```
update kurssi  
set opintoviikot=opintoviikot*2  
where nimi='Java ohjelmointi';
```

- Muutos epäonnistuu, jos se rikkoo eheysehtoja.

9

Oraclessa uusi arvo myös kyselyllä

- Oracle special:

```
update taulu  
set (sarakelista) = (alikyseley)  
where rajausehdot
```

- alikyseley on tyypillisesti kytketty alikyseley ja sen pitää tuottaa yksi rivi jokaista rajausehdon täyttävää riviä kohden, esim.

```
update henkilo  
set (palkka) = (select korotus from korotukset  
where korotus.hetu=  
henkilo.hetu)  
where hetu in (select hetu from korotukset)
```

10

SQL - Tietokannan ylläpito

- Rivien poisto (delete)

```
delete from taulu  
[where poistettavien rajausehdot];
```

- Poistetaan kaikki ehdon täyttävät rivit
- Jos ehto puuttuu, poistetaan kaikki rivit
- Poisto epäonnistuu, jos eheysehdot rikkoutuvat (ellei muuta ole määritetty)

11

SQL - Tietokannan ylläpito

- Poistetaan harjoitusryhmät, joihin ei ole ilmoittautuneita:

```
delete from harjoitusryhma  
where (kurssikoodi, ryhmänro) not in  
(select kurssikoodi, ryhmänro  
from ilmoittautuminen);
```

12

SQL - Tietokannan ylläpito

- Rivien siirtoa taulusta toiseen tarvitaan esimerkiksi siirrettäessä tietoja aktiivisesta taulusta historiatauluihin. Tämä suoritetaan kopiaamalla (lisäämällä) rivit kohdetauluun ja sen jälkeen poistamalla ne lähtötaulusta:

```
insert into ilmoihistoria
select * from ilmoittautumiset
where ilm_aika < date '1999-1-1';

delete from ilmoittautumiset
where ilm_aika < date '1999-1-1';
```

13

SQL - Tietokantatapahtuma (transaktio)

- Tietokantatapahtumalla tarkoitetaan yhtenä jakamattomana kokonaisuutena pidettävää tietokantaoperaatioiden joukkoa, esimerkiksi tilisiirto:

```
update tili set saldo=saldo-500
where tilinumero=123456;
update tili set saldo=saldo+500
where tilinumero=654321;
```

14

SQL - Tietokantatapahtuma (transaktio)

- Tkjhj takaa, että

 - tapahtuma suoritetaan kokonaan eikä vain osaa siitä (ei siis vain tililiittämistä)
 - Muut käyttäjät näkevät vain kokonaisen tapahtuman aiheuttamat muutokset (ulkopuolinen ei voi nähdä tilannetta, jossa tililtä 123456 on otettu 500, mutta tilille 654321 ei sitä ole vielä viety)
 - tapahtuman suorituksen aikana tehdyt muutokset kantaan on peruttavissa siihen asti kunnes tapahtumaan on sitouduttu
 - kun tapahtumaan on sitouduttu (se on valmis) muutokset jäävät pysyviksi ja näkyvät myös muille.

15

SQL - Tietokantatapahtuma (transaktio)

- Tapahtuma päätetään onnistuneesti komennolla `commit [work]`
- Tapahtuma voidaan päättää myös perumalla sen aikaansaamat muutokset komennolla `rollback [work]`
- Tilisiirtotapahtuma olisi kokonaisuudessaan siis

```
update tili set saldo=saldo-500
where tilinumero=123456;
update tili set saldo=saldo+500
where tilinumero=654321;
commit;
```

16

SQL - Tietokantatapahtuma (transaktio)

- Järjestelmät voidaan määritellä toimimaan `auto-commit` tilassa, jolloin jokaiseen ylläpito-operaatioon sitoudutaan välittömästi (tällöin tilisiirtoa ei voida koota transaktioksi)
- Normaalitilassa tapahtumia kuitenkin kootaan `commit`-operaatioiden avulla. Kahden `commit`:in välissä olevat operaatiot muodostavat tapahtuman.

17

SQL - Tietokantatapahtuma (transaktio)

- Yritetään poistaa tyhjät harjoitusryhmät, oletetaan, että ilmoittautumisten viiteavaimen liittyy `on delete cascade` -määre

```
commit;
select count(*) from ilmoittautumiset;
>> 3500 <<
delete from harjoitusryhma
where ryhmanro is not null;
select count(*) from ilmoittautumiset;
>>> 0 <<<
rollback;
select count(*) from ilmoittautumiset;
>> 3500 <<
```

Ei ole ihan oikein (oho!)

18