

Liitokset



JYVÄSKYLÄN
AMMATTIKORKEAKOULU

Osaaminen kilpailukyvyksi

Liitokset - haut useaan tauluun

- Tavallisin liitos on valintaliitos
 - ehtona =, !=, <, <=, >=, >
 - yhtäläisyysliitos (=) yleisin (vrt. Inner join)
 - taulut liitetään toisiinsa yleensä avaimilla (perus- ja viiteavain)
- Esim.
`SELECT nimi, tilausnumero`
`FROM asiakas INNER JOIN tilaus`
`ON asiakas.asiakastunnus = tilaus.asiakastunnus;`
- Kvalifiointi: taulun nimi sarakenimen eteen
 - pakollista, jos saman niminen sarake kahdessa taulussa

Perinteinen liitos

- Liitosehto WHERE-lauseessa
- Esim. hae yrityksen nimi, ko. yrityksen henkilöiden nimet ja palkat yli 2500 tienaavista:
 - **SELECT** yritys.Yrtun, Nimi, -- Hae yrityksen tunnus, nimi
 - Sukunimi, Etunimi, Palkka -- ja henkilön nimi + palkka
 - **FROM** yritys, henkilo -- näistä tauluista
 - **WHERE** yritys.yrtun = henkilo.yrtun -- Liitosehto!
 - **AND** Palkka > 7500 -- Hakuehto palkalle
 - **ORDER BY** Nimi -- Lajittelu

Uusi liitossyntaksi (suositeltavampi vaihtoehto)

- Liitosehto FROM-lauseessa
- Hae yrityksen nimi, ko. yrityksen henkilöiden nimet ja palkat yli 2500 tienaavista:
 - **SELECT** yritys.Yrtun, Nimi, -- Hae yrityksen tunnus, nimi
 - Sukunimi, Etunimi, Palkka -- henkilön suku- ja etunimi + palkka
 - **FROM** yritys -- YRITYS -taulusta, joka
 - **INNER JOIN** henkilo -- liitetään HENKILO-tauluun
 - **ON** yritys.Yrtun = henkilo.Yrtun -- Liitosehto
 - **WHERE** Palkka > 7500 -- Hakuehto palkalle
 - **ORDER BY** Nimi; -- Lajittelu
- Standardin mukaan voidaan myös käyttää NATURAL JOIN -liitosta, jos liitossarakkeet ovat samannimiset

Monen taulun liitos

- Kun liität kolme tai useampia tauluja toisiinsa, on syytä olla huolellinen JOIN-määrittelyn kanssa
- Eräs ratkaisu on käyttää sellaista työkalua taulujen liittämiseksi, jolla SQL-käsky saadaan edelleen muokattavaksi, esim. MS Access:
 - SELECT OPINTOJAKSO.OpintojaksonNimi,
OPISK_OJAKSO.Kehitysehdotukset,
OPISKELIJA.OpiskNimi
 - FROM OPISKELIJA INNER JOIN (OPINTOJAKSO
INNER JOIN OPISK_OJAKSO ON
OPINTOJAKSO.OJtunnus =
OPISK_OJAKSO.OJtunnus) ON
OPISKELIJA.OpiskNro = OPISK_OJAKSO.OpiskNro

Liitokset: ulkoliitos (outer join)

- Outer Join ottaa mukaan myös ne rivit, jotka eivät täytä join-ehtoa; esim.:
 - haetaan myös ne asiakkaat, joille ei löydy tilauksia:

```
SELECT AsiakasNimi, tilausnumero  
FROM asiakas  
LEFT OUTER JOIN tilaus  
ON asiakas.asiakastunnus = tilaus.asiakastunnus;
```
 - vanha syntaksi:

```
SELECT AsiakasNimi, tilausnumero  
FROM asiakas, tilaus  
WHERE asiakas.asiakastunnus (+) =  
tilaus.asiakastunnus;
```
- OUTER-sana voidaan jättää pois SQL-standardissa

Liitos itseensä

- Liitos itseensä: tarvitaan korrelaationimeä (alias)
- Esim. haetaan JAMK:n eri yksiköt (org.hierarkiasta):

```
SELECT a.nimi  
FROM asiakas a, asiakas b  
WHERE a.emoyritystunnus = b.asiakastunnus  
AND b.asiakastunnus = 'JAMK';
```
- Joissakin tuotteissa pitää käyttää AS-sidesanaa korrelaationimen edessä, joissakin pitää olla välilyönti
- Kun korrelaationimi on otettu käyttöön, ei alkuperäistä nimeä saa käyttää sarakkeen kvalifioinnissa (esim. edellisessä käskyssä `SELECT asiakas.nimi` antaisi virheilmoituksen)

Yhdiste (UNION)

- Kaksi taulua (tai useampia) yhdistetään samantyyppisen tiedon avulla
- Esim. opettajien ja opiskelijoiden puhelinnumerot:

```
SELECT OpeNimi AS Nimi, PuhNro, 'opettaja'  
FROM opettaja
```

```
UNION
```

```
SELECT OpiskNimi AS Nimi, PuhNro, 'opiskelija'  
FROM opiskelija  
ORDER BY 1;
```



Alikyselyt



JYVÄSKYLÄN
AMMATTIKORKEAKOULU

Osaaminen kilpailukyvyksi

Alikysely

- Voidaan käyttää SELECT, INSERT, UPDATE ja DELETE -komennoissa
- **Sisin kysely suoritetaan ensiksi**
- Voi palauttaa vain yhden sarakkeen ja sille yhden tai useamman arvon
 - vertailuoperaattori “=“: alikysely voi palauttaa vain yhden arvon
 - IN-operaattori tai vertailuoperaattori ANY tai ALL -ehdon kanssa sallii alikyselyn palauttavan useita arvoja
- Oltava aina suluissa ja haettavana saa olla vain yksi sarake
- Käyttää automaattisesti distinct-määrettä

Alikyselyt – esimerkki1: =

- Hae sen henkilön nimi, joka myy solmioita:

```
SELECT HenkNimi
```

```
FROM henkilo
```

```
WHERE HenkID =
```

```
(SELECT MyyjaID
```

```
FROM tuote
```

```
WHERE TuoteNimi = 'Solmio');
```

- Aika usein alikyselyssä etsitään arvoa koostefunktiolla (palauttaa aina yhden arvon)



Alikyselyt – esimerkki 2: IN

- Hae niiden tuotteiden nimet ja hinnat, joita myy laiteosastolla olevat henkilöt:

```
SELECT TuoteNimi, hinta
```

```
FROM tuote
```

```
WHERE myyjaID IN
```

```
(SELECT myyjaID
```

```
FROM myyja
```

```
WHERE osasto = 'laite');
```

- Myös NOT IN on käyttökelpoinen, kun halutaan sulkea pois alikyselyn palauttama tulosjoukko



Alikyselyt – esimerkki 3: ANY, ALL ja EXISTS

- Jos käytetään vertailuoperaattoria $>$ tai $<$, niin IN ei käy
SELECT nimi, hinta, myyja
FROM tuote
WHERE hinta $>$ ANY
(SELECT hinta
FROM tuote
WHERE myyja = 'Pekka');
-- hakuehto:
-- ne myyjät
-- jotka myyvät
-- kalliimpia tuotteita
-- kuin Pekka
- ANY: jos alikyselyn mikä tahansa arvo täyttää ehdon
- ALL: jos alikyselyn jokainen yksittäinen arvo täyttää ehdon
- EXISTS: tosi, jos alikysely palauttaa ainakin yhden rivin

Alikyselyt – esimerkki 4 (koostefunktion käyttö)

- Tyypillistä on hakea alikyselyssä jokin koostefunktion palauttama arvo ja verrata sitä toiseen joukkoon

SELECT nimi, hinta, myyja

FROM tuote

WHERE hinta >

(SELECT AVG(hinta)

FROM tuote

WHERE myyja = 'Pekka');

-- hakuehto:

-- ne myyjät

-- jotka myyvät **keskimäärin**

-- kalliimpia tuotteita

-- kuin Pekka

