

SQL Joins

SQL – Liitokset (JOINS)

- Joskus haettavaa tietoa ei löydy vain yhdestä taulusta, vain se on jaettu kahteen tai useimpiin tauluihin. Tässä tapauksessa on pakko yhdistä tietoa eri tauluista sillä tavalla, että on mahdollista laatia kyselyä, josta saadaan tarvittavaa tietoa.
- Ilman liitosehtoa kyselyn lopputulokseen tulee kaikki mahdolliset kombinaatiot taulujen riveistä. Esimerkki: `SELECT * FROM regions, countries`; Huoma, että regions taululla on 5 rivejä ja countries taululla on 25 rivejä. `SELECT`:in lopputuloksessa on $5 \times 25 = 125$ rivejä.
- Joten tavallisesti käytetään liitosehtoja jollain tavalla:
`SELECT * FROM regions, countries WHERE regions.region_id = countries.region_id`;
- Aika usein liitosehto on yhteinen arvo molemmista tauluista.

SQL – Liitokset (JOINS)

- Vaikka on mahdollista yhdistää taulua vain kirjoittamalla taulujen nimeä pilkun eroteltuna FROM osaan, tavallisesti tähän tarkoitukseen käytetään JOIN –syntaksia, joka antaa enemmän mahdollisuuksia hallita sitä, miten taulut yhdistetään.
- Esimerkki:
`SELECT * FROM regions JOIN countries ON regions.region_id = countries.region_id;`
- Vaihtoehtoa:
`SELECT * FROM regions AS r JOIN countries AS c ON r.region_id = c.region_id;`
`SELECT * FROM regions r JOIN countries c ON r.region_id = c.region_id;`

Kyselyn vaiheet

JOIN –määreellä ”**yhdistetään**” kaksi (tai useampia) taulua yhdeksi hakua varten. ”Yhdistäminen” kestää vain kyselyn ajan.

Tapoja yhdistää on seitsemän, joista neljällä on oma komentonsa SQL-kielessä.

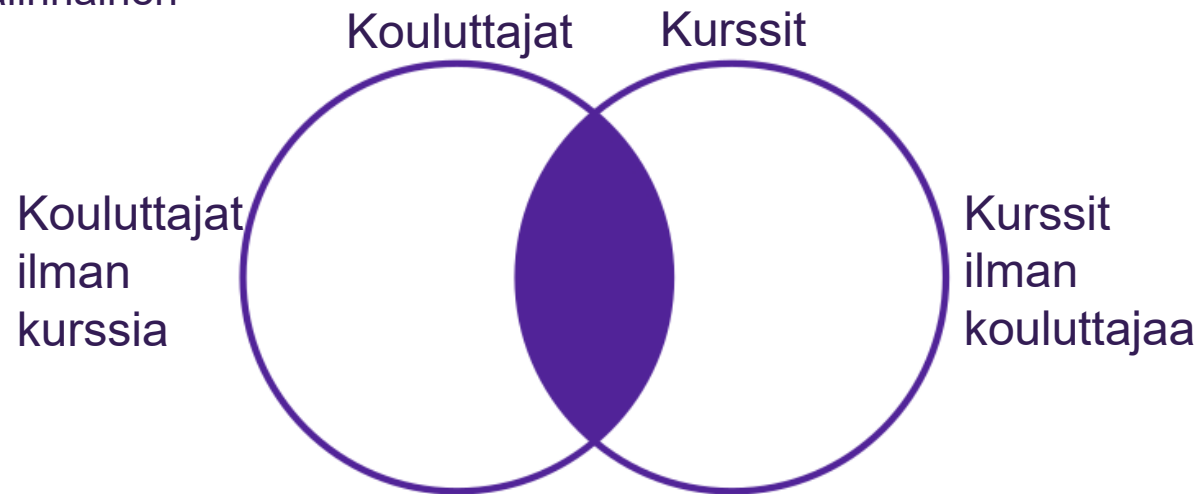
| | | |
|----------------------------------|------------------------------------|---|
| INNER JOIN | Sisäliitos | Tietueet, joilla oman sama avain yhdistetään |
| | Ulkoliitos | Sisäliitoksen tietueiden lisäksi, kaikki tietueet yhdestä tai molemmista tauluista yhdistetään. Tästä tyypistä on 3 liitosta. |
| LEFT JOIN / LEFT OUTER JOIN | Vasenliitos / Vasen ulkoliitos | Sisäliitoksen tietueiden lisäksi, kaikki vasemmat (ensimmäisen) taulun tietueet yhdistetään |
| RIGHT JOIN / RIGHT OUTER JOIN | Oikealiitos / Oikea ulkoliitos | Sisäliitoksen tietueiden lisäksi, kaikki oikean (toisen) taulun tietueet yhdistetään |
| FULL JOIN / FULL OUTER JOIN | Täysi liitos / Täysi ulkoliitos | Kaikki tietueet molemmista tauluista yhdistetään |

SQL – Liitokset (JOINS)

SELECT ... x [INNER] JOIN y ...

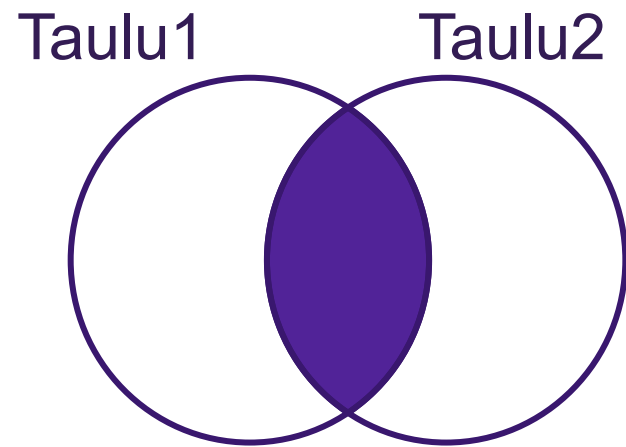
- INNER JOIN valitsee rivejä, joista löytyy liitosarvoa molemmille tauluille.

HUOM: INNER sana on valinnainen



```
SELECT kouluttajat.nimi, kurssit.nimi  
FROM kouluttajat INNER JOIN kurssit ON kouluttajat.kurssi_id = kurssit.kurssi_id;
```

INNER JOIN



```
SELECT *  
FROM Taulu1  
INNER JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain
```

INNER JOIN

INNER JOIN ON K2 = K3

Tietueet, joissa arvo K2- ja K3 -sarakkeissa on samaa kerätään yhteen tauluun.

Taulu1

| K1 | K2 |
|----|------|
| 1 | Null |
| 2 | B |
| 3 | C |
| 4 | Null |
| 5 | E |

Taulu2

| K3 | K4 |
|------|------|
| A | Null |
| B | Null |
| C | 8 |
| D | 9 |
| Null | 10 |

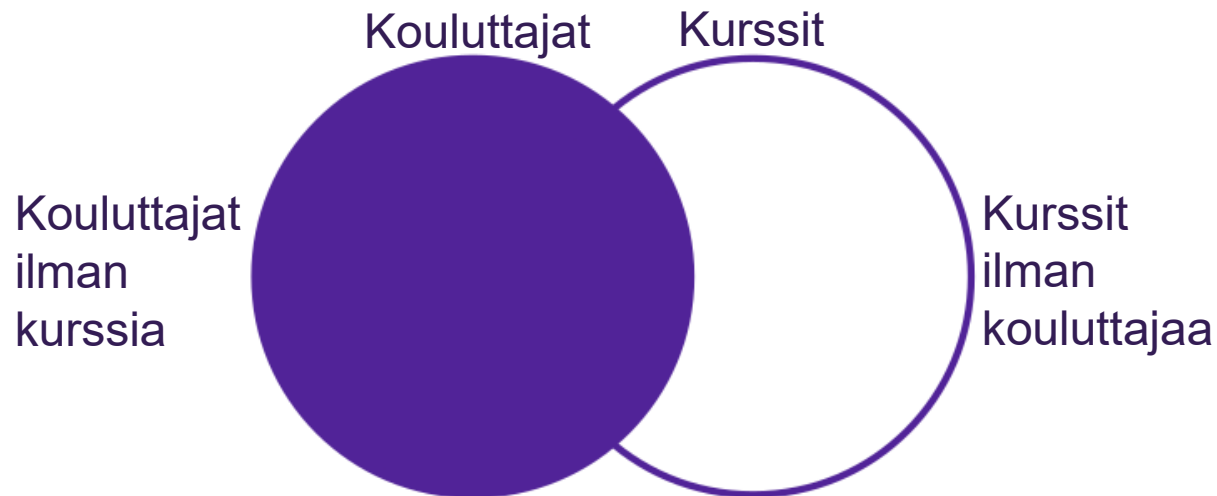


| K1 | K2 | K3 | K4 |
|----|----|----|------|
| 2 | B | B | Null |
| 3 | C | C | 8 |

SQL – Liitokset (JOINS)

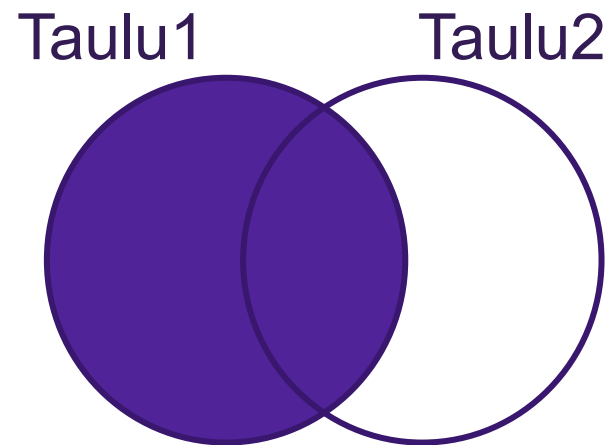
SELECT ... x LEFT JOIN y ...

- LEFT JOIN valitsee rivejä, joista löytyy liitosarvoa molemmille tauluille, mutta kaikki rivit vasemmasta olevasta taulusta pitää olla myös mukana tuloksissa liitosehdoista riippumatta.



```
SELECT kouluttajat.nimi, kurssit.nimi  
FROM kouluttajat LEFT JOIN kurssit ON kouluttajat.kurssi_id = kurssit.kurssi_id;
```


LEFT JOIN



```
SELECT *  
FROM Taulu1  
LEFT JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain
```

LEFT JOIN

LEFT OUTER JOIN

LEFT JOIN ON K2 = K3

**Kaikki vasemman/
ensimmäisen taulun
tietueet kerätään
yhteiseen tauluun.**

| K1 | K2 |
|----|------|
| 1 | Null |
| 2 | B |
| 3 | C |
| 4 | Null |
| 5 | E |

| K3 | K4 |
|------|------|
| A | Null |
| B | Null |
| C | 8 |
| D | 9 |
| Null | 10 |



| K1 | K2 | K3 | K4 |
|----|------|------|------|
| 1 | Null | Null | Null |
| 2 | B | B | Null |
| 3 | C | C | 8 |
| 4 | Null | Null | Null |
| 5 | E | Null | Null |

SQL – Liitokset (JOINS)

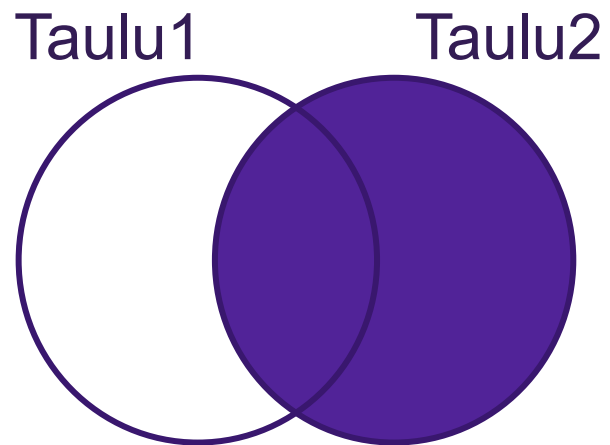
INNER JOIN vs LEFT JOIN

Esimerkkiä:

- `SELECT * FROM regions INNER JOIN countries ON regions.region_id = countries.region_id;`
- `SELECT * FROM regions LEFT JOIN countries ON regions.region_id = countries.region_id;`
- Mitä eroa on niiden välillä?

Huoma, että RIGHT JOIN on samaa kuin LEFT JOIN, mutta toisinpain.

RIGHT JOIN



```
SELECT *  
FROM Taulu1  
RIGHT JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain
```

RIGHT JOIN

RIGHT OUTER JOIN

RIGHTJOIN ON K2 = K3

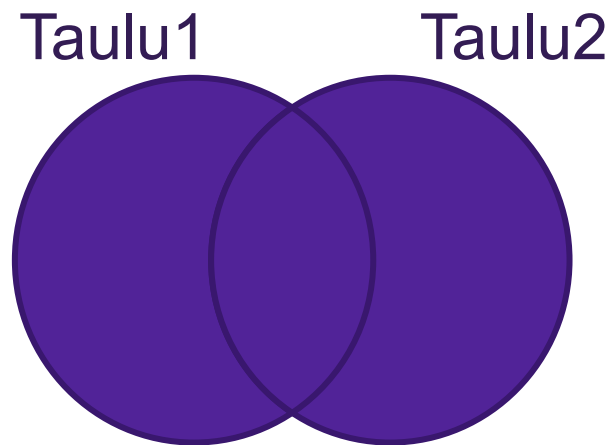
Kaikki oikean/ toisen taulun tietueet kerätään yhteiseen tauluun.

| K1 | K2 | | K3 | K4 |
|----|------|-------|------|------|
| 1 | Null | | A | Null |
| 2 | B | ————— | B | Null |
| 3 | C | ————— | C | 8 |
| 4 | Null | | D | 9 |
| 5 | E | | Null | 10 |



| K1 | K2 | K3 | K4 |
|------|------|------|------|
| 2 | B | B | Null |
| 3 | C | C | 8 |
| Null | Null | A | Null |
| Null | Null | D | 9 |
| Null | Null | Null | 10 |

FULL OUTER JOIN / FULL JOIN



```
SELECT *  
FROM Taulu1  
FULL OUTER JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain
```

FULL JOIN

FULL OUTER JOIN

FULL JOIN ON K2 = K3

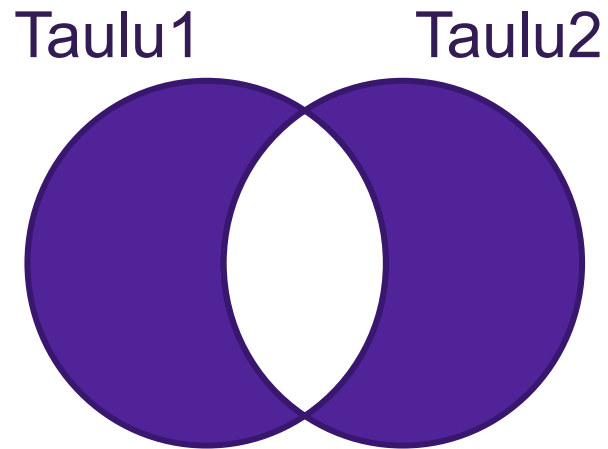
Kaikki tietueet
kerätään yhteiseen
tauluun.

| K1 | K2 | | K3 | K4 |
|----|------|--|------|------|
| 1 | Null | | A | Null |
| 2 | B | | B | Null |
| 3 | C | | C | 8 |
| 4 | Null | | D | 9 |
| 5 | E | | Null | 10 |



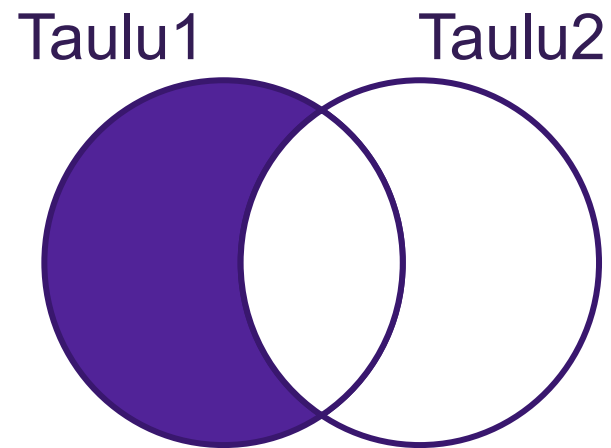
| K1 | K2 | K3 | K4 |
|------|------|------|------|
| 1 | Null | Null | Null |
| 2 | B | B | Null |
| 3 | C | C | 8 |
| 4 | Null | Null | Null |
| 5 | E | Null | Null |
| Null | Null | A | Null |
| Null | Null | D | 9 |
| Null | Null | Null | 10 |

Erikoistapaukset: Poissulkeva OUTER JOIN



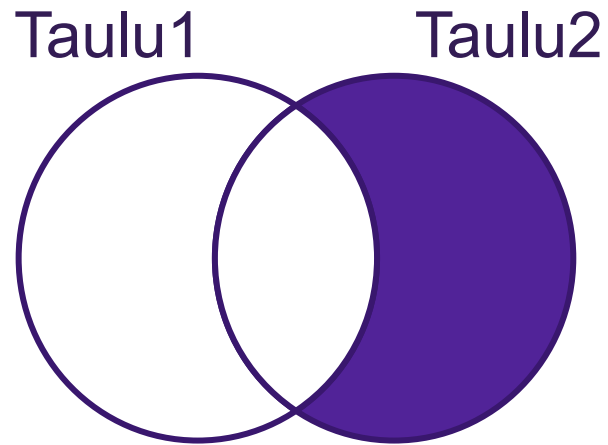
```
SELECT *  
FROM Taulu1  
OUTER JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain  
WHERE Taulu1.avain IS NULL OR  
Taulu2.avain IS NULL
```


Erikoistapaukset: Poissulkeva LEFT JOIN



```
SELECT *  
FROM Taulu1  
LEFT JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain  
WHERE Taulu2.avain IS NULL
```

Erikoistapaukset: poissulkeva RIGHT JOIN



```
SELECT *  
FROM Taulu1  
RIGHT JOIN Taulu2  
ON Taulu1.avain = Taulu2.avain  
WHERE Taulu2.avain IS NULL
```

SQL – Liitokset (JOINS)

- Liitosehtoa voi olla erilaisia, mutta kun kyseessä on saman kentän nimi molemmissa tauluissa, sitten ON sijaan voidaan käyttää USING(x).

Esimerkki:

- `SELECT * FROM regions INNER JOIN countries USING(region_id);`
- Toinen vaihtoehto on käyttää ... x NATURAL JOIN y ...(ilman liitosehtoa)

Esimerkki:

- `SELECT * FROM regions NATURAL JOIN countries;`

SQL – Liitokset (JOINS)

NATURAL JOIN on kätevä, mutta pitää olla varovainen sen käytössään

Miksi seuraavista lauseista ei tule samoja tuloksia?

- `SELECT * FROM employees JOIN departments USING(department_id);`
- `SELECT * FROM employees NATURAL JOIN departments;`

Mitä hyötyä on taulujen välisistä liitoksista?

- Erittäin tarpeellisia ja käyttökelpoisia.
- Yleensä taulut yhdistyvät ID-kentillä, joita ihmisen on vaikea lukea. Yhdistämällä useita tauluja yhteen saat ne ”**ihmislueuttavampaan**” muotoon.
 - Esim. Tuodaan taulusta1 henkilön etunimi ja sukunimi ja yhdistetään se tauluun2, josta saamme henkilön palkan.
- Lisäksi pystyt tekemään laajempaa analyysia ja koontia useista eri tauluista samaan aikaan. Tämä on kuitenkin hyvin haastavaa.
 - Esim. Tuodaan taulusta2 kaikkien miespuolisten ja naispuolisten henkilöiden lukumäärät ja yhdistetään ne taulusta2 laskettuun vertailulukuun

SQL – Liitokset (JOINS)

Tehtäviä:

- Write a query to find the addresses (depart_name, street_address, city, state_province, country_name) of all the departments.
- Write a query to find the names (first_name, last_name) of all employees and the department name they work for.
- Write a query to find the employee id, name (first_name, last_name) along with their manager_id, manager name (first_name, last_name). Be sure to show all employees even if they have not a manager.
- Write a query to get the **all** department names and number of employees in the department. Be sure to show all departments, even if they have no employees.