

Håndtegnede joins

I de følgende opgaver skal I lave joins i hånden. I skal gå ud fra tabellerne, I kan se i arket "Academy tabeller.pdf".

Eksempelopgave:

Givet følgende query

```
SELECT teachers.first_name, teachers.last_name, courses.course_name
FROM teachers JOIN courses ON teachers.teacher_id = courses.teacher_id
WHERE teachers.first_name = 'Lars'
```

Hvordan ville en virtuel tabel, som er joinet mellem teachers og courses se ud (vi starter med at se bort fra SELECT og WHERE)?

Røde felter (de tre første kolonner fra venstre) kommer fra tabellen teachers og grønne felter (de tre sidste kolonner) kommer fra tabellen courses:

| teacher_id | first_name | last_name | course_id | course_name | teacher_id |
|------------|------------|-----------|-----------|-------------|------------|
| 1 | Lars | Pedersen | 1 | DAT2 | 1 |
| 1 | Lars | Pedersen | 3 | WEB | 1 |
| 2 | Mette | Jensen | 2 | SOFT | 2 |

Figur 1

Læg mærke til at rækkerne fra de to tabeller ikke er smidt tilfældigt sammen. I den virtuelle, joinede tabel, stemmer teacher_id fra tabellen teachers og teacher_id fra tabellen courses overens i hver række. Det er det, vi beder om, når vi skriver "ON teachers.teacher_id = courses.teacher_id" i vores query.

Når vi kobler WHERE på, får vi nedenstående:

| teacher_id | first_name | last_name | course_id | course_name | teacher_id |
|------------|------------|-----------|-----------|-------------|------------|
| 1 | Lars | Pedersen | 1 | DAT2 | 1 |
| 1 | Lars | Pedersen | 3 | WEB | 1 |

Figur 2

Den sidste række forsvandt, da den ikke levede op til klausulen "WHERE teachers.first_name = 'Lars'".

SELECT udvælger hvilke data, vi ønsker at se og så har vi vores endelige resultatsæt:

| first_name | last_name | course_name |
|------------|-----------|-------------|
| Lars | Pedersen | DAT2 |
| Lars | Pedersen | WEB |

Figur 3

I skal nu lave tilsvarende opgaver selv. For hver query skal I tegne den virtuelle, joinede tabel (figur 1) og det endelige resultatsæt (figur 3).

Opgave 1:

Vi vil gerne have navnene på de studerende, som har fået karakteren 7. Vi er ligeglade med hvilket fag, de har taget. Vi skal derfor bruge oplysninger fra tabellen students (first_name og last_name) og fra enrollments (grade).

```
SELECT students.first_name, students.last_name
FROM students JOIN enrollments ON students.student_id = enrollments.student_id
WHERE grade = '7'
```

Hvordan ville en virtuel tabel, som er joinet mellem students og enrollments se ud (vi starter med at se bort fra SELECT og WHERE)?

[illegible]

Og hvordan vil det endelige resultatsæt se ud, når vi kobler WHERE og SELECT på?

| first_name | last_name |
|------------|-----------|
| | |
| | |
| | |

Opgave 2

Vi vil gerne se hvilke karakterer, der er givet i hvilke fag, sorteret efter fag. Vi skal bruge data fra tabellen `courses` (`course_name`) og fra tabellen `enrollments` (`grade`)

```
SELECT course_name, grade
FROM courses JOIN enrollments ON courses.course_id = enrollments.course_id
ORDER BY course_name
```

Bemærk at der i ovenstående query ikke er angivet tabelnavn foran `course_name` og `grade`. Det er ikke nødvendigt, fordi kolonnerne kun findes i én af tabellerne, så der er ingen tvivl om hvad der menes. Men der er angivet tabelnavn foran `course_id`, fordi kolonnen findes i begge tabeller.

Hvordan ville en virtuel tabel, som er joinet mellem courses og enrollments se ud (vi starter med at se bort fra SELECT og ORDER BY)?

[illegible]

Og hvordan vil det endelig resultatsæt se ud, når vi kobler SELECT og ORDER BY på?

[illegible]

Opgave 3

Nu vil vi gerne have en oversigt over hvilke studerende, er har fået hvilke karakterer i hvilke fag. Vi har brug for data fra tre tabeller, nemlig students (first_name og last_name), enrollments (grade) og courses (course_name).

```
SELECT first_name, last_name, grade, c.course_name
FROM enrollments e
JOIN students s ON e.student_id = s.student_id
JOIN courses c ON e.course_id = c.course_id;
```

Bemærk at vi giver tabellerne "navne" i ovenstående. Tabellen enrollments forkortes til e, students til s og courses til c. Det kan være en fordel i mere komplicerede queries, hvor man ikke gider skrive det fulde tabelnavn hele tiden.

For at løse denne opgave, skal vi kigge på hvordan flere joins fungerer. Der sker det, at man først joiner tabellerne enrollments og students og dernæst joiner denne virtuelle, joinede tabel med courses.

Hvordan ville en virtuel tabel, som er joinet mellem enrollments og students se ud (vi starter med at se bort fra SELECT og tabellen courses)? Den lavede I i opgave 1, så vi snyder lidt og giver jer den her (kolonnerne står i en anden rækkefølge, fordi enrollments kommer først i vores join, men det betyder intet for hvordan tabellen fungerer).

| enrollment_id | student_id | course_id | grade | student_id | first_name | last_name | role_id |
|---------------|------------|-----------|--------------|------------|------------|------------|---------|
| 1 | 1 | 1 | 7 | 1 | Emil | Hansen | 1 |
| 2 | 1 | 2 | 10 | 1 | Emil | Hansen | 1 |
| 3 | 2 | 1 | 4 | 2 | Sofie | Nielsen | 2 |
| 4 | 2 | 3 | 7 | 2 | Sofie | Nielsen | 2 |
| 5 | 3 | 2 | 12 | 3 | Mads | Kristensen | 1 |
| 6 | 3 | 3 | 10 | 3 | Mads | Kristensen | 1 |
| 7 | 4 | 1 | 02 | 4 | Anna | Møller | 3 |
| 8 | 4 | 2 | 4 | 4 | Anna | Møller | 3 |
| 9 | 5 | 3 | 7 | 5 | Frederik | Olsen | 2 |
| 10 | 6 | 1 | 10 | 6 | Julie | Andersen | 1 |

Hvordan ser det så ud, når vi joiner vores virtuelle enrollments-students-tabel med courses?

| enrollme nt_id | studen t_id | course_i d | grade | studen t_id | first_name | last_name | role_id | | | |
|-------------------|----------------|---------------|-------|----------------|------------|------------|---------|--|--|--|
| 1 | 1 | 1 | 7 | 1 | Emil | Hansen | 1 | | | |
| 2 | 1 | 2 | 10 | 1 | Emil | Hansen | 1 | | | |
| 3 | 2 | 1 | 4 | 2 | Sofie | Nielsen | 2 | | | |
| 4 | 2 | 3 | 7 | 2 | Sofie | Nielsen | 2 | | | |
| 5 | 3 | 2 | 12 | 3 | Mads | Kristensen | 1 | | | |
| 6 | 3 | 3 | 10 | 3 | Mads | Kristensen | 1 | | | |
| 7 | 4 | 1 | 02 | 4 | Anna | Møller | 3 | | | |
| 8 | 4 | 2 | 4 | 4 | Anna | Møller | 3 | | | |
| 9 | 5 | 3 | 7 | 5 | Frederik | Olsen | 2 | | | |
| 10 | 6 | 1 | 10 | 6 | Julie | Andersen | 1 | | | |

Og hvordan ser det endelige resultatsæt ud, når vi tilføjer SELECT?

[illegible]

Opgave 4

I den sidste opgave skal I lave opgaver til hinanden. I skal gå sammen to og to (hvis I ikke allerede har lavet ovenstående i par) og hver lave en lille join opgave til den anden med udgangspunkt i de udleverede tabeller. Lav også en vejledende løsning.

Løs hinandens opgaver og kig på dem i fællesskab for at se om løsningerne stemmer overens med jeres vejledende løsninger.