

Kræsjkurs INF115



Lukas Schramm

Institutt for informatikk
Universitetet i Bergen

23 Mai 2022

Innføring

SQL

Relasjonsalgebra

ER-Diagrammer

Normalisasjon

Indekser (B-trær)

Transaksjoner

Alt mulig

Last meg ned



Figure: <https://tinyurl.com/inf115v22>

Temaer

- Select
- Aggregation
- Joins
- Update
- Insert
- Delete
- Create and Drop Tables
- Datatyper
- Constraints
- Alt mulig

Select

```
SELECT director, COUNT(*) as c -- Spalter
FROM movie                      -- Tabeller
WHERE country = "Dabendorf"    -- Betingelser
GROUP BY director              -- Gruppering etter spalter
HAVING c > 3                    -- Betingelser etter gruppering
ORDER BY c DESC                 -- Sortering etter spalter
LIMIT 3;                       -- Velg de første n output linjene
```

Aggregation

- Count: Teller antall elementer
- Sum: Summerer verdier
- Avg: Gjennomsnitt av verdier (tilsvarer count/sum)
- Min: Minimum
- Max: Maksimum
- Alle aggregasjonsfunksjoner ignorerer NULL verdier

Join (2 varianter)

```
SELECT *                -- Variante 1
FROM film, screening    -- Velg alle tabeller og bruk WHERE
WHERE film.film_id = screening.film_id;
```

```
SELECT *                -- Variante 2
FROM film               -- Velg en tabell og bruk JOIN og ON
JOIN screening ON film.film_id = screening.film_id;
```

Types of Joins

- INNER JOIN: Alle rad som finnes i begge tabeller
- LEFT (OUTER) JOIN: Alle rad som finnes i begge tabeller og alt fra venstre tabellen
- RIGHT (OUTER) JOIN: Alle rad som finnes i begge tabeller og alt fra høyre tabellen
- FULL (OUTER) JOIN: Alt fra begge tabeller
- *Alt* mener tilsvarende JOIN columns
- Ikke-eksisterende verdier fylles med NULL verdier
- ... SELF JOIN, CROSS JOIN, NATURAL JOIN, ...

Venn Diagram

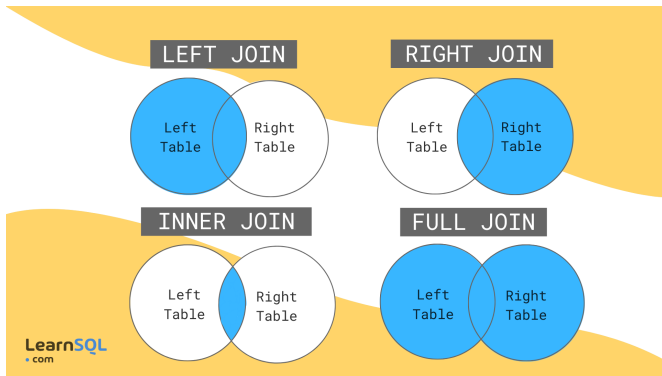


Figure: Kilde: <https://learnsql.com/blog/learn-and-practice-sql-joins/>

Update

```
UPDATE book                -- tabell
SET title = "Der satanarchäolügenialkohöllische Wunschpunsch"
WHERE book_id = 37;        -- betingelse
```

```
UPDATE music                -- tabell
SET title = "Supercalifragilisticexpialidocious",
    artist = "Mary Poppins" -- oppdatert informasjon
WHERE song_id = 42;        -- betingelse
```

Insert (2 varianter)

```
INSERT INTO film (film_id, title, director, genre)
VALUES (7, "Fantômas", "André Hunebelle", "Fransk tull");
```

```
INSERT INTO film      -- spaltenavn ikke nødvendig hvis alle blir brukt
VALUES (7, "Fantômas", "André Hunebelle", "Fransk tull");
```

Delete

```
DELETE FROM film      -- tabell  
WHERE country = "USA"; -- betingelse
```

Create Table

```
CREATE TABLE Persons (
    Person_id int,                -- column name, datatype
    LastName varchar(40) NOT NULL,
    FirstName varchar(40),
    Age int,
    Department_id int UNIQUE,
    PRIMARY KEY (Person_id),      -- primær- og fremmednøkler
    FOREIGN KEY (Department_id) REFERENCES Department(Department_id),
    CHECK (Age>=18)              -- constraints
);
```

Drop Table

```
DROP TABLE eksamenskarakterer; -- tabell som skal slettes
```

Datatyper

```
char(20)           -- 20 tegn med fast lengde
varchar(20)        -- opptil 20 tegn
text               -- tekst med opptil 2^16 tegn
int                -- integer (tinyint, smallint, bigint)
float(2)           -- float med 2 sifre bak komma
decimal(5,2)       -- 3 sifre før og 2 etter komma
enum ("INF102", "INF234", "INF237") -- en av n verdier
date               -- dato
timestamp          -- unix timestamp
```

Constraints

```
NOT NULL          -- verdien er ikke NULL
UNIQUE            -- spalten har ingen verdi flere ganger
PRIMARY KEY(key)  -- NOT NULL + UNIQUE
FOREIGN KEY(key) REFERENCES tabell(key) -- fremmednøkkel
AGE int CHECK (AGE >= 18) -- betingelse
AGE int DEFAULT 18 -- standardverdi
```


Alt mulig

```
SELECT DISTINCT           -- fjerner duplikater
WHERE navn LIKE "A\%"     -- alt som starter med A
-- \% 0, 1 eller flere tegn
-- _ eksakt et tegn
WHERE category IN (cake, bread) -- verdien er i en liste
WHERE age BETWEEN 18 and 65   -- verdien er i en range
WHERE EXISTS (SELECT ...)    -- verdien finnes i en annen query
(SELECT ...) UNION (SELECT ...) -- union av to queries
```

Alt mulig

```
SELECT Order_id, Price,  
CASE      -- if betingelser i SELECT  
    WHEN Price > 30 THEN "Won't pay freight"  
    ELSE "Freight costs 10\$"  
END AS FreightCost  
FROM OrderDetails;
```

Takk for oppmerksomheten
Lykke til med eksamenen