

Databaser

Programmering-B

Thomas Birk Abildgaard

January 8, 2026

HTX Frederikshavn

SQL (Structured Query Language) er et programmeringssprog, der bruges til at administrere og manipulere relationelle databaser.

- **SELECT:** Udtrækning af data fra en database.
- **INSERT:** Indsættelse af nye data i en tabel.
- **UPDATE:** Opdatering af eksisterende data i en tabel.
- **DELETE:** Sletning af data fra en tabel.

Chinook dataset

Vi vil bruge det dataset kaldt 'Chinook'.

Chinook-databasen indeholder fiktive data om en musikbutik, inkl. tabeller som: Customer, Invoice, InvoiceLine, Track, Album, Artist, Genre, Employee osv.

Datatyper i MariaDB

Almindelige datatyper i MariaDB

Datatype	Beskrivelse	Eksempel / brug
INT / INTEGER	Heltal	Id'er, antal
BIGINT	Store heltal	Store id'er eller tællere
FLOAT / DOUBLE	Decimaltal	Pris, vægt
CHAR(n)	Fast længde tekst	Initialer, korte koder
VARCHAR(n)	Variabel længde tekst	Navn, e-mail, adresse
TEXT	Lange tekster	Produktbeskrivelser
DATE	Dato	Fødselsdag, ordre dato
DATETIME	Dato + tid	Ordre tidspunkt
TIMESTAMP	Auto. opdatering	Sidst ændret dato
BOOLEAN	Sandt/falsk	Aktiv kunde, betalt

SQL Syntaks

SELECT Statement

Bruges til at udtrække data fra en eller flere tabeller.

```
SELECT column1, column2, ...  
FROM table  
WHERE condition;
```

Eksempel:

```
SELECT first_name, last_name  
FROM employees  
WHERE department = 'IT';
```

INSERT Statement

Bruges til at indsætte nye rækker i en tabel.

```
INSERT INTO table (column1, column2, ...)
VALUES (value1, value2, ...);
```

Eksempel:

```
INSERT INTO customers (first_name, last_name,
    email)
VALUES ('John', 'Doe', 'john.doe@example.com'
    );
```

UPDATE Statement

Bruges til at opdatere eksisterende data i en tabel.

```
UPDATE table  
SET column1 = value1, column2 = value2, ...  
WHERE condition;
```

Eksempel:

```
UPDATE employees  
SET salary = salary * 1.1  
WHERE department = 'IT';
```

DELETE Statement

Bruges til at slette data fra en tabel.

```
DELETE FROM table  
WHERE condition;
```

Eksempel:

```
DELETE FROM customers  
WHERE last_purchase_date < '2023-01-01';
```

Spørsmål ?

- Importering af Chinook dataset
- Basic SQL øvelser

SQL WHERE LIKE

```
-- Eksempel: Find Kunder med 'vej' i adressen  
SELECT * FROM customer  
WHERE Address LIKE '%vej%';
```

SQL WHERE LIKE (Fortsat)

- % bruges som et jokertegn til at matche nul eller flere tegn.
- _ bruges til at matche præcis ét tegn.
- LIKE 'a%' matcher alle, der starter med 'a'.
- LIKE '%a' matcher alle, der slutter med 'a'.

```
-- Eksempel: Vis kun de foerste 5 raekker fra  
      ordrer  
SELECT * FROM orders  
LIMIT 5;
```

SQL AND og OR

```
-- Eksempel: Find produkter med enhedspris  
mellem 10 og 20
```

```
SELECT * FROM products  
WHERE UnitPrice >= 10 AND UnitPrice <= 20;
```

```
-- Eksempel: Find ordrer fra kunde 'ALFKI'  
eller 'BLONP'
```

```
SELECT * FROM orders  
WHERE CustomerID = 'ALFKI' OR CustomerID = '  
    BLONP';
```

Keys & Foreign keys

Konvention for navngivning

Konsistens er vigtigere end den enkelte konvention.

Hver tabel og kolonne i databasen skal følge samme navngivnings "mønster"

En database med ensartede navne er:

- nemmere at læse
- nemmere at fejlfinde
- nemmere at samarbejde om

De efterfølgende retningslinjer er en **anbefalet standard** i undervisningen.

- Brug kun bogstaver (a–z), tal og underscore (`_`)
- Undgå mellemrum og specialtegn (herunder `æ`, `ø`, `å`)
- Undgå SQL-reserverede ord (fx `order`, `group`, `select`)

SQLite skelner ikke mellem store og små bogstaver, men vi gør det for læsbarhed.

- Brug **Stort begyndelsesbogstav (PascalCase)**
- Navngives i **ental**
- Tabellen repræsenterer én type objekt (fx en kunde)

Eksempler:

Customer

CustomerOrder

Product

CustomerAddress

Felter (columns)

- Brug **lowercase** (snake_case)
- Navne skal være beskrivende

Eksempler:

```
id
username
email
created_at
order_date
```

Primærnøglen hedder altid:

```
id
```

Eksempel:

```
id INTEGER PRIMARY KEY
```

Navngives som: `<tabel>_id`

Eksempler:

```
customer_id  
order_id  
product_id
```

Eksempel på tabel

```
CREATE TABLE Customer (  
    id INTEGER PRIMARY KEY,  
    name TEXT NOT NULL,  
    email TEXT NOT NULL,  
    created_at DATETIME  
);  
  
CREATE TABLE CustomerOrder (  
    id INTEGER PRIMARY KEY,  
    customer_id INTEGER NOT NULL,  
    order_date DATETIME,  
    FOREIGN KEY (customer_id) REFERENCES  
        Customer(id)  
);
```