DataCraft

4
SQL Basics

DataCraft

SQL-Abfragen (Queries) sind der zentrale Befehl, um Daten aus einer (oder mehrerer) Tabellen zu erhalten. Queries beginnen mit SELECT, FROM gibt die Tabelle an

```
-- die gesamte Tabelle album zurückgeben
select * from album;

-- die ersten 10 Zeilen der Tabelle album zurückgeben
select * from album limit 10;

-- die Spalten Title und ArtistId der Tabelle album zurückgeben
select
  Title as Albumtitel,
  ArtistId
from album;
```

SELECT DISTINCT

DataCraft

Mit dem Schlüsselwort DISTINCT werden nur eindeutige Werte/Zeilen zurückgegeben

```
select distinct ArtistId from album;

select distinct
  BillingCity,
  BillingCountry
from Invoice
```

Mit dem Zusatz ORDER BY kann eine Sortierung angegeben werden. Mehrere Spalten werden durch Komma getrennt. Mit DESC wird eine absteigende Reihenfolge gewählt

```
select
  Title as Albumtitel,
  ArtistId
from album
order by Title;

select
*
from Customer
order by LastName, FirstName desc;
```

Mit dem Zusatz WHERE kann auf die Zeilen gefiltert werden, die eine (oder mehrere Bedingungen) erfüllen.

```
select
*
from Customer
where City = 'Paris';
```

Strings werden in **einfachen** Hochkommata eingeschlossen. Im Gegensatz zu Python wird für die Gleichheit nur ein Gleichzeichen verwendet

- Gleichheit: `=`
- Ungleichheit: `!=` oder `<>`
- Größer (oder gleich), Kleiner (oder gleich): `>, <, >=, <=`
- Logische Operatoren: `and, or, not`
- `is null`

Mit LIKE können Muster in Strings gesucht werden. Dabei steht das Prozentzeichen % für beliebig viele Zeichen und der Unterstrich _ für ein beliebiges Zeichen.

'P%' bedeutet z.B., dass das Wort mit P anfängt. '%s' bedeutet, dass das Wort mit s aufhört.

```
select
*
from Customer
where City like 'P%';
```

DataCraft

Teilstrings können mit substr extrahiert werden (viele SQL-Dialekte verstehen auch LEFT und RIGHT, aber SQLite nicht)

```
-- substring hat als Argumente die Startposition und die Anzahl Zeichen
select
  substr(name,1,1) as anfangsbuchstabe,
  name
from artist
where substr(name,1,1) = 'B';

-- negative Zahlen entsprechen Positionen von rechts
select
*
from artist
where substr(name, -2, 2) = 'an';
```

Verknüpfung von Strings

DataCraft

Mit || oder concat() werden Strings miteinander verbunden.

```
select
  left(firstname,1) || '. ' || left(lastname,1) || '.' as initialen,
  concat(lastname, ', ', firstname) as name
from customer
```

Bei || wird der Ausdruck null, wenn ein String null ist. concat() ignoriert null.

```
select
  null || firstname as immernull,
  concat(null, firstname) as nurvorname
from customer
```

DataCraft

Die CASE-WHEN-Struktur ermöglicht Bedingungen für eine Spalte zu definieren, ähnlich zu if-else

```
select
  name,
  Milliseconds / 60000.0 as laenge,
  case when (Milliseconds/60000.0) > 4 then 'lang' else 'kurz' end as laenge_kat
from track;
```

Es sind zwei Strukturen möglich:

```
CASE WHEN BEDINGUNG1 THEN …
     WHEN BEDINGUNG2 THEN …
     ELSE …
```

```
CASE AUSDRUCK WHEN ERGEBNIS1 THEN …
              WHEN ERGEBNIS2 THEN …
     ELSE …
```

Tabellen miteinander verknüpfen

DataCraft

Um zwei Tabelle (mit den gleichen Spaltennamen und –typen) untereinander zu hängen, kann UNION verwendet werden.

```
select title, artistid  from album where left(title,1)='A'
union
select 'Black Album', 50
```

Der häufigere Fall ist aber ein JOIN anhand eines Schlüssels. Die vier JOIN-Arten habt ihr schon beim Verknüpfen von DataFrames kennengelernt (siehe nächste Seite)

DataCraft

Es gibt vier Arten, Tabellen miteinander zu verbinden

INNER JOIN

behalte nur Zeilen, die in beiden Tabellen vorkommen

ID	X
1	Jahr
2	Tag
3	Monat

+

ID	Y
2	365
3	12
4	52

=

ID	X	Y
2	Tag	365
3	Monat	12

FULL OUTER JOIN

behalte alle Zeilen aus beiden Tabellen

ID	X
1	Jahr
2	Tag
3	Monat

+

ID	Y
2	365
3	12
4	52

=

ID	X	Y
1	Jahr	NA
2	Tag	365
3	Monat	12
4	NA	52

LEFT JOIN

behalte alle Zeilen aus der linken Tabelle

ID	X
1	Jahr
2	Tag
3	Monat

+

ID	Y
2	365
3	12
4	52

=

ID	X	Y
1	Jahr	NA
2	Tag	365
3	Monat	12

RIGHT JOIN

behalte alle Zeilen aus der rechten Tabelle

ID	X
1	Jahr
2	Tag
3	Monat

+

ID	Y
2	365
3	12
4	52

=

ID	X	Y
2	Tag	365
3	Monat	12
4	NA	52

Tabellen miteinander verknüpfen

DataCraft

Um zwei Tabelle per JOIN zu verbinden, wird nach from der join definiert.
Tabellen können einen Alias (Benennung) bekommen

```
select
  *
from album a
inner join artist b on a.artistid = b.artistid;
```

Auswahl der Spaltennamen über den Tabellennamen bzw. –alias

```
select
  a.title as album_title,
  b.name as artist_name
from album a
inner join artist b on a.ArtistId = b.ArtistId
```

DataCraft

Verknüpfung von mehreren Tabellen gleichzeitig ist möglich

```
SELECT
  t.trackid,
  t.name as track_name,
  a.Title as album_name,
  art.Name as artist_name
from Track t
inner join Album a on t.AlbumId = a.AlbumId
inner join Artist art on a.ArtistId = art.ArtistId ;
```

DataCraft

Durch Klammerung können auch Unterabfragen realisiert werden

```
SELECT
  t.trackid,
  t.name as track_name,
  a.album_name,
  a.artist_name
from Track t
inner join (
  select
    a.AlbumId,
    a.title as album_name,
    b.name as artist_name
  from album a
  inner join artist b on a.ArtistId = b.ArtistId
) a on t.AlbumId = a.AlbumId
```

Subquery

DataCraft

Aggregation erfolgt mittels group by (am Ende, aber vor order)

```
select
  b.name as artist_name,
  count(1) as anzahl_alben
from album a
inner join artist b on a.ArtistId = b.ArtistId
group by b.name
order by anzahl_alben desc;
```

Nur die Felder, die in group by auftauchen, dürfen ohne Aggregations-Funktion erscheinen. Ggf. solche Felder aufnehmen, auch wenn sich Gruppierung nicht ändert (z.B. artistId)

Aggregations-Funktionen: MIN, MAX, SUM, COUNT, AVG, GROUP_CONCAT

DataCraft

Bedingungen an die aggregierten Felder mittels HAVING

```
select
  b.name as artist_name,
  count(1) as anzahl_alben
from album a
inner join artist b on a.ArtistId = b.ArtistId
group by b.name
having anzahl_alben > 1
order by anzahl_alben desc;
```

GROUP_CONCAT erlaubt das Verbinden von Strings

```
select
  b.name as artist_name,
  count(1) as anzahl_alben,
  group_concat(a.Title, ";") as alben
from album a
inner join artist b on a.ArtistId = b.ArtistId
group by b.name
having anzahl_alben = 2;
```

Um nur eindeutige Werte zu verbinden, kann das Schlüsselwort DISTINCT hinzugefügt werden:

```
group_concat(distinct a.Title, ";")
```