CAS SQL

WS01: Oracle Joins und Subqueries

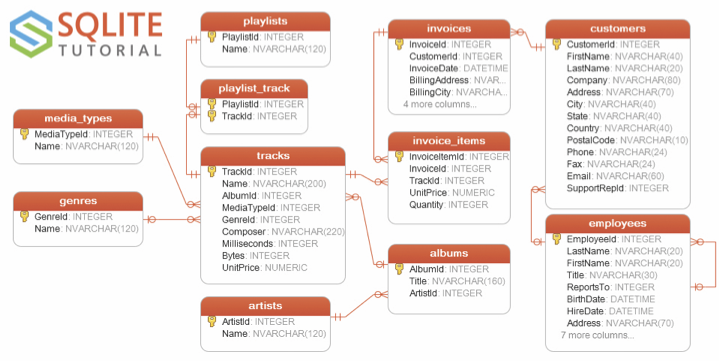
**Voraussetzung**

Du hast Zugriff auf Oracle (siehe SQL\_WS00)

* Schwierigkeit: einfach
* Prüfungsrelevant: ja
* Zeitaufwand: 45min

# Datenbank-Schema

Hier nochmals das Datenbankdiagram als Referenz.



# JOINS über mehrere Tabellen

Joins können auch mehrere Tabellen umfassen, indem einfach mehrere JOINS hintereinander aufgelistet werden.

*# Beispiel mit vier Tabellen:*

*# Zeige alle Tracks welche 'Isabelle Mercier' gekauft hat.*

SELECT tr.name

FROM customers cu

INNER JOIN invoices iv USING(CustomerId)

INNER JOIN invoice\_items ii USING(InvoiceId)

INNER JOIN tracks tr USING(TrackId)

WHERE cu.firstname = 'Isabelle' AND cu.lastname = 'Mercier';

## Aufgabe 9 : Joins über mehrere Tabellen

*Zeige für Tracks, die zwischen 470 und 480 Sekunden dauern, deren Namen, den Albumtitel und den Namen des Artisten. (WHERE milliseconds BETWEEN 470000 and 480000)*

*Erwartetes Resultat:*

*NAME TITLE NAME*

*---------------------- ----------------------------------- ------------------*

*Loverman Garage Inc. (Disc 1) Metallica*

*Am I Evil? Garage Inc. (Disc 2) Metallica*

*Flight Of The Rat Deep Purple In Rock Deep Purple*

*Evil Ways Santana Live Santana*

*Bye Bye Blackbird The Essential Miles Davis [Disc 1] Miles Davis*

*Nefertiti The Essential Miles Davis [Disc 2] Miles Davis*

*Old Love Unplugged Eric Clapton*

*Hallowed Be Thy Name A Real Dead One Iron Maiden*

*The Evil That Men Do Live At Donington 1992 (Disc 1) Iron Maiden*

*Running Free Live At Donington 1992 (Disc 2) Iron Maiden*

*Aeroplane Flies High Judas 0: B-Sides and Rarities Smashing Pumpkins*

*Out Of Control No Security The Rolling Stones*

## Aufgabe 9+ (Zusatzaufgabe)

*Finde die Anzahl Tracks mit Mediatype* '*MPEG audio file*'*, die pro Person/Customer gekauft wurden.*

*Erwartetes Resultat:*

*FIRSTNAME LASTNAME COUNT(TR.NAME)*

*---------------------------------------- -------------------- --------------*

*John Gordon 34*

*Hannah Schneider 38*

*Eduardo Martins 38*

*Jennifer Peterson 32*

*Edward Francis 38*

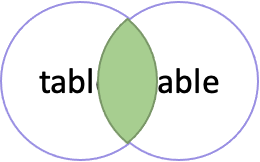
*Wyatt Girard 36*

*Helena Holý 26*

*Kara Nielsen 32*

*… … ..*

# Self-JOINS

Bei self-joins wird eine Tabelle table mit sich selber verknüpft. 

Damit werden Hierarchien abgebildet oder Einträge miteinander verglichen.

**SELECT** **\*** **FROM** **table** t1

**INNER** **JOIN** **table** t2

**ON** t1.id **=** t2.other\_id;

*# Im folgenden Beispiel wird eine hierarchische Beziehung zwischen Employee und Manager abgefragt.*

SELECT m.firstname || ' ' || m.lastname AS "Manager",

e.firstname || ' ' || e.lastname AS "Direct report"

FROM employees e

INNER JOIN employees m ON m.employeeid = e.reportsto

ORDER BY "Manager";

## Aufgabe 10: Self-join

*Folgende query gibt uns alle Personen nach Stadt geordnet.*

*Der ||-Operator verbindet zwei Strings.*

SELECT DISTINCT

e1.city,

e1.firstName || ' ' || e1.lastname AS fullname

FROM

employees e1

ORDER BY

e1.city;

*Wie drucken wir nun nur diejenigen Städte, in welchen mindestens zwei Personen wohnen?*

*Tipp: self-join verwenden und sicherstellen, dass die Stadt gleich ist, die Person aber nicht.*

# Subqueries

**SELECT** column\_1

**FROM** table\_1

**WHERE** column\_1 **=**

(

**SELECT** column\_2

**FROM** table\_2

);

* Subqueries sind erlaubt nach SELECT, FROM , HERE und JOIN.
* Subqueries geben oftmals einen einzigen Wert zurück (Ausnahme: IN-Operator)

*# Beispiel 1: finde das Album mit title='Let There Be Rock' und zeige alle tracks in diesem Album.*

*# PS: Diese Query könnte auch mit einem Join implementiert werden. Ausprobieren ist freiwillig.*

SELECT trackid, name, albumid

FROM tracks

WHERE albumid =

(

SELECT albumid

FROM albums

WHERE title = 'Let There Be Rock'

);

*# Beispiel 2: finde alle Kunden mit einem oder einer Supportverantwortlichen in 'Canada'.*

SELECT customerid, firstname, lastname

FROM customers

WHERE supportrepid IN

(

SELECT employeeid FROM employees

WHERE country = 'Canada'

);

*# Beispiel 3: Finde alle Kunden mit einer Rechnung.*

SELECT

FirstName || ' ' || LastName name, Company

FROM

Customers c

WHERE

EXISTS (

SELECT

1

FROM

Invoices

WHERE

CustomerId = c.CustomerId

)

ORDER BY

name;

*# Beispiel 4: Finde die durchschnittliche Album-Grösse in MB. Dazu müssen zuerst alle Tracks zusammengezählt werden.*

SELECT

ROUND(AVG(album.album\_size) / 1024 / 1024)

FROM

(

SELECT albumid, SUM(bytes) album\_size

FROM tracks

GROUP BY albumid

) album;

## Aufgabe 11: Subqueries mit IN

*Folgende Query listet alle Genres vom Typ* '*Alternative*'

SELECT genreid, name

FROM genres WHERE name LIKE 'Alt%';

*Augabe: Zähle die Anzahl Tracks die zum Genre* '*Alternative*' *gehören.*

## Aufgabe 12: Subqueries mit [NOT] EXISTS

*Aufgabe 12a: Suche die Anzahl Artists die mindestens ein Album zugewiesen haben.*

*Aufgabe 12b: Suche die Anzahl Artists die keine Album zugewiesen haben.*

# Lösungen

Aufgabe 9

SELECT tr.name, al.Title, ar.Name

FROM tracks tr

INNER JOIN albums al USING(AlbumId)

INNER JOIN artists ar USING(ArtistId)

WHERE milliseconds BETWEEN 470000 and 480000;

Aufgabe 9+

SELECT cu.firstname, cu.lastname, count(tr.name)

FROM customers cu

INNER JOIN invoices iv ON(cu.CustomerId = iv.CustomerId)

INNER JOIN invoice\_items ii ON(iv.InvoiceId = ii.InvoiceId)

INNER JOIN tracks tr ON(ii.TrackId = tr.TrackId)

INNER JOIN media\_types mt ON(tr.MediaTypeId = mt.MediaTypeId)

WHERE mt.name = 'MPEG audio file'

GROUP BY cu.firstname, cu.lastname;

Aufgabe 10

SELECT DISTINCT

e1.city,

e1.firstName || ' ' || e1.lastname AS fullname

FROM

employees e1

INNER JOIN employees e2 ON e2.city = e1.city

AND (e1.firstname <> e2.firstname AND e1.lastname <> e2.lastname)

ORDER BY

e1.city;

Aufgabe 11

SELECT COUNT(\*)

FROM tracks t

WHERE t.genreid IN

(SELECT genreid

FROM genres WHERE name LIKE 'Alt%');

Aufgabe 12a

SELECT count(\*) FROM

artists a

WHERE EXISTS (

SELECT 1 FROM albums

WHERE a.artistid = artistid

);

Aufgabe 12b

SELECT count(\*) FROM

artists a

WHERE NOT EXISTS (

SELECT 1 FROM albums

WHERE a.artistid = artistid

);