

Some SQL Queries

% you can use them with the project Buch/Verlag

select Name, ort From Verkauf;
selektiert Name und Ort von Tabelle Verkauf

select DISTINCT Name, ort From Verkauf;
selektiert Name und Ort von Tabelle Verkauf - DISTINCT steht dass keine Zeile mehrfach vorkommt

select DISTINCT Name, ort **AS SITZ** From Verkauf;
wie oben, die Tabellenüberschrift Ort wird mit Sitz ersetzt

select * from buch **Where** FACHGEBIET='Datenbanken';
selektiert alle (Stern) Bücher mit Fachgebiet Datenbanken

select isbn from buch **Where** FACHGEBIET='Datenbanken';
selektiert alle ISBN von Bücher mit Fachgebiet Datenbanken

select isbn from buch **Where** Titel=Fachgebiet;
selektiert ISBN von den Büchern mit dem Titel Fachgebiet

select name, Telefon from Person where Strasse **IS NULL** or ort **IS NULL**;
selektiert Name und Telefon von Tabelle Person welche keine Strassen- oder Ortsangabe haben

select * from buch Where FACHGEBIET='Datenbanken'
and Jahr **BETWEEN** 1990 AND 1999 AND ((Auflage >1) OR (Auflage **IS NULL**));
selektiert alle Bücher mit Fachgebiet Datenbanken und mit Jahr zwischen 1990 und 1999 und einer Auflage grösser als 1 oder 0

select Name,Ort From Person Where Ort LIKE '%**Bonn**';
selektiert Name und Ort von Tabelle Person welche die in Bonn oder ähnlich lautendem Ort wohnen (% entspricht dem Stern im Dos-Modus)

select Name,Ort From Person Where Ort Like '%**Bo_n**';
wie oben, zusätzlich ist „_“ ein Platzhalter bei dem mir der Buchstabe egal ist aber nicht die Anzahl

select Name,Ort From Verkauf **Where** Ort IN ('Freiburg', 'Konstanz');
selektiert Name und Ort von Tabelle Verkauf mit Ort Freiburg oder Konstanz

select AVG(Gehalt) From Lektor;
selektiert den Durchschnitt vom Gehalt der Tabelle Lektor

select SUM(Gehalt)/**Count**(Gehalt) From Lektor;
selektiert die Summe der Gehälter und dividiert diese durch die Anzahl der Gehälter in der Tabelle Lektor

select **Count(DISTINCT** Lektor) From Buch;
selektiert die Anzahl der Lektoren in der Tabelle Buch ohne Doppelnennung

select FACHGEBIET, ISBN From Buch **Group By** FACHGEBIET, ISBN;
selektiert Fachgebiet und ISBN von Tabelle Buch und gruppiert dann erstens nach Fachgebiet und als zweites nach ISBN

select FACHGEBIET, **Count(*)** From Buch **Group By** FACHGEBIET;
selektiert Fachgebiet von Tabelle Buch und zählt diese und gruppiert nach Fachgebiet

select FACHGEBIET, **Count(*) As** Anzahl From Buch **Group By** FACHGEBIET;
wie oben, anstelle von Count steht Anzahl als Überschrift

select FACHGEBIET, Count(*) As Anzahl, Sum(AUFLAGE) As Summe From Buch
Group by FACHGEBIET;
selektiert Fachgebiet und zählt, benennt die Überschrift mit Anzahl, zählt die Summe der Auflage und benennt das als Summe und gruppiert dann nach Fachgebiet

select Name , Ort , SUM(Anzahl) As Summe From Verkauf Where Monat Between
9001 And 9812
Group By Name, Ort;
selektiert Name und Ort, zählt die Summe verkaufter Bücher zwischen den Monaten 9001 und 9812, benennt die Spalte als Summe und gruppiert nach Name und Ort

select ISBN, Monat, Anzahl From Verkauf Group By ISBN, Monat,
Anzahl Having SUM(Anzahl) > 50;
selektiert ISBN, Monat und Anzahl von Verkauf und gruppiert nach ISBN, Monat und Anzahl (nur mit Anzahl über 50)

Select ISBN From Buch **UNION** Select ISBN From Verkauf;
Selektiert ISBN von Buch und Verkauf (union steht für Gesamtmenge)

Select Lektor As Kuerzel From Buch **INTERSECT** Select Kuerzel From Lektor;
Selektiert Schnittmenge Lektor benannt als Kuerzel von Buch und Kuerzel von Lektor

Select Name, Strasse, Ort From Person, Lektor Where
Person.Kuerzel=lektor.Kuerzel;
Selektiert Name, Strasse und Ort von Person wo

Select ISBN,Anzahl, (Select AVG(Anzahl) From Verkauf) From verkauf
Where Anzahl>(Select AVG(Anzahl) From Verkauf);
Selektieren wie gewohnt, in Klammern stehen jetzt einfach Abfragen

Select ISBN,Titel From Buch Where ISBN<> **All**(Select ISBN From Verkauf);
<> ungleich aller selektierten in klammern

Select ISBN, Titel From Buch Where ISBN= **ANY**(Select ISBN From Verkauf);
= in einer von()

Select ISBN, Titel From Buch Where ISBN **IN** (Select ISBN From Verkauf);
muss in () sein

Select ISBN, Titel From Buch Where ISBN **NOT IN** (Select ISBN From Verkauf);
nicht in ()

create table emp
(empno NUMBER(3),
ename CHAR(15),
evorname CHAR(10));
Tabelle mit Name „emp“ erstellt, die Felder definiert und ihre Länge

truncate table emp;
Tabelle leeren (Inhalte entfernen)

truncate table emp **drop storage**;
Speicher von Tabelle freigeben

truncate table emp **reuse storage**;
Speicher reservieren um neue Daten zu schreiben

drop table emp;
Löschen der Tabelle

create user gerard **identified by** amigo;
neuen Benutzer erstellen, gerard ist username und amigo ist passwort

create user gerard identified by amigo **default**
tablespace emp **temporary**
tablespace emp **quota 1 M** on emp;
erstellt neuer default Benutzer, darf Tabelle emp benutzen und hat recht für 1mb in dieser Tabelle zu speichern

alter user gerard identified by amie;
Passwort ändern – neues Passwort amie

alter user gerard identified by amie
default tablespace emp quota 2 M on
emp
quota 0 M on Buch;
Passwort geändert und Quota von 1mb auf 2mb erhöht für emp und 0mb für buch

drop user gerard **cascade**;
Benutzer gerard löschen – mit cascade werden alle seine Tabellen gelöscht

select * from **user_users**;
alle eingelogten Benutzer anzeigen

select * from **all_users**;
selektiert alle Benutzer

select * from dba_users;
selektiert

select * from user_ts_quotas;
selektiert die Quoten von den Benutzern

select **sid**, **serial#**, username from v\$session;
selektiert alle offenen Sessions

alter system kill session '6,1';
Session mit sid 6 und serial-nr 1 löschen

grant alter any table to gerard;
Ermittelt Recht für gerard dass er in alle Tabellen schreiben darf

grant create user, alter user, drop user to gerard **with admin option**;
rechte verteilen von user, ändern von Passwörtern, mit admin Optionen

select * from sys.dba_sys_privs order by grantee, privilege;
selektiere alle Benutzer und Liste Rechte auf

revoke alter user, drop user from gerard;
rechte rückgängig machen von gerard

grant select, **insert**(empno, ename), **update** (ename) on emp to gerard;
Benutzer kann selektieren ausführen auf tabelle emp von feldern empno und ename und kann die Felder aktualisieren

grant select, insert(empno, ename), update (ename) on emp to gerard **with grant option**;
wie oben mit Option dass er Rechte weitergeben kann wenn er neuer Benutzer erstellt

create table emp (empno Number primary key, ename varchar(10), foreign key (nr) references dept(nr));
erstelle Tabelle emp mit Feldern und Schlüsseln

create table person3 as select * from Person;
gesamte Tabelle Person kopieren und unter neuem Namen person3 speichern

create table person3 as select * from Buch where Lektor='WA' and Auflage=1;
Tabelle kopieren wie oben aber nur bestimmte Felder nach folgenden Bedingungen

SELECT 'Bonjour mon cher '|| name FROM Person;
Text „Bonjour mon cher“ mit Name von der Tabelle Person verknüpfen

select * from Person ORDER BY name;
selektiert alle von Person geordnet nach Name

SELECT table_name from USER_TABLES;
Selektiert alle Tabellen die in der Datenbank sind

select * from global_name;
selektiert Datenbankname

DROP USER nouri **CASCADE**;
user nouri mit allen Tabellen löschen