ORACLE/SQL

BEFEHLE:

ED => öffnet den Editor
R => führt den letzten Befehl aus der im Puffer ist
(der Puffer enthält immer den letzten Befehl Datei: afied.buf)
DESC tabellenname => zeigt den Inhalt der ausgewählten Tabelle an
DROP TABLE => Tabellenname löscht die ausgewählte Tabelle
ALTER TABLE tabellenname ADD (feld1 Datentyp); => Spalte zur bestehenden Tabelle hinzufügen
ALTER TABLE Tabellenname MODIFY (feldname datentyp (feldgröße) not null/ unique => bestehende Tabellenspalten ändern
DELETE From tabellenname WHERE bedingung => Datensätze löschen die diese Bedingung erfüllen

Seite einrichten (Ansicht)

SET LINESIZE 100
SET PAGESIZE 50

Anmelden an einem Anderen Rechner und Benutzer wenn Passwort bekannt ist
CONNECT Benutzer/Passwort@Fachinf

Passwort Ändern:

ALTER USER Benutzername IDENTIEFIED BY Neuer Benutzer

HOST DIR => listet das Verzeichnis auf wo man sich befindet

Abmelden aus Oracle und speichert Änderungen ab:
EXIT

Datei (SKRIPT) erstellen die der Editor nutz und wo man Befehle loswerden kann:

ED C:\Pfad wo hin\name der Datei
START C:\Pfad wo datei ist\name der Datei => startet die Abarbeitung dieser SKRIPT Datei
Oder
@ C:\Pfad wo hin\name der Datei => startet ebenfalls die Abarbeitung dieser SKRIPT Datei

Unterschiede zwischen Oracle und ACCESS

Wildcards:
* wird zu %
? wird zu _ (unterstrich)

Vergleichsoperatoren:
Identisch anstelle <> lässt sich auch != verwenden

Felddatentypen:

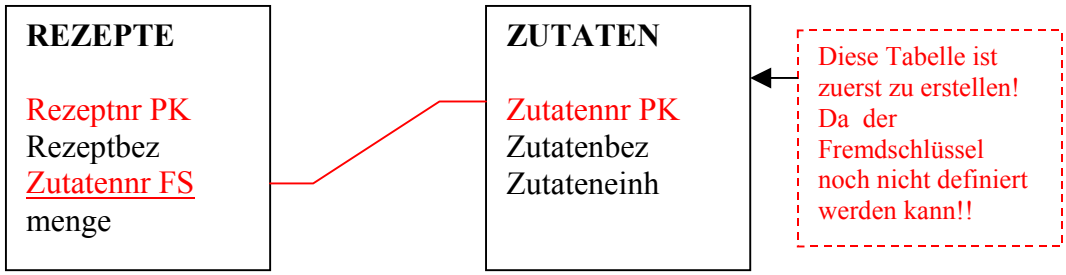
CHAR (10)	=> für TEXT bis 2000 Zeichen hier wird der komplette Platz reserviert (Speicherverschwendung)
VARCHAR2 (10)	=> für TEXT bis 4000 Zeichen verschenkt keinen Platz nur den der Eingegeben ist
DATE	=> für DATUM wird bei ORACLE so geschrieben '12.30.90'
NUMBER (2)	=> für Zahlen hier werden die Stellen eingetragen also (2) entspricht 0 bis 99
(5,2)	=> entspricht somit 5 Stellen insgesamt und davon 2 Nachkommastellen 0 bis 999,99
(6,3)	=> entspricht somit 6 Stellen insgesamt und davon 3 Nachkommastellen 0 bis 999,99

ROWID => ist eine Zahl die von Oracle zur Tabellen Erfassung dient

Rechnen in Oracle:
SELECT 1*1
FROM DUAL; => **DUAL** ist eine leere einspaltige Tabelle die in Oracle enthalten ist

Datumsausgabe:
SELECT SYSDATE
FROM DUAL;

Tabellen erstellen:




```
SQL> Create Table zutaten (zutatennr Number (2),
                           zutatenbez varchar2 (15),
                           zutatenein varchar2 (3),
                           constraint pri_zutaten Primary Key (zutatennr));
Tabelle wurde angelegt.
```

```
SQL> desc zutaten
 Name                                      Null?    Typ
 ----------------------------------------- -------- ------------
 ZUTATENNR                                 NOT NULL NUMBER(2)
 ZUTATENBEZ                                         VARCHAR2(15)
 ZUTATENEIN                                         VARCHAR2(8)
```

```
SQL> Create Table rezepte (rezeptnr number (2),
                           rezeptbez varchar2 (15),
                           zutatennr number (2) not null,
                           menge number (4),
                           constraint pri_rezepte primary key (rezeptnr),
                           foreign key (zutatennr) references zutaten (zutatennr));
Tabelle wurde angelegt.
```

```
SQL> desc rezepte
 Name                                      Null?    Typ
 ----------------------------------------- -------- ------------
 REZEPTNR                                           NUMBER(2)
 REZEPTBEZ                                          VARCHAR2(15)
 ZUTATENNR                                 NOT NULL NUMBER(2)
 MENGE                                              VARCHAR2(8)
```

Primary Key löschen:

```
SQL>Alter Table rezepte drop constraint pri_rezepte;
```

Felddatentyp ändern :

```
SQL> Alter Table zutaten modify (zutatenein varchar2 (8));
```

Break on entfernt doppelte einträge in einer Spalte (so das Rührei und 1Person nicht 4 mal auftaucht sondern nur 1mal jeweils). Distinct funktioniert hier nicht da ja hier jeder Datensatz immer etwas anderes enthält!!!

SQL> break on menge on rezeptbez skip 1;

REZEPTBEZ	ZUTATENBEZ	MENGE	ZUTATENE
Rührei	ei	1Person	3ST
Rührei	Salz		1Pr
Rührei	Pfeffer		1Pr
Rührei	Milch		50ml

REZEPTBEZ	ZUTATENBEZ	MENGE	ZUTATENE
Rüheei	ei	1Person	3ST
	Salz		1Pr
	Pfeffer		1Pr
	Milch		50ml

SQL> **select table_name from user_tables; => zeigt alle Benutzertabellen an**

```
TABLE_NAME
------------------------------
BONUS
DEPT
DUMMY
EMP
EMP1
REZEPTE
SALGRADE
ZUTATEN
```

=> vorhandene Tabellen kopieren
Create Table tabellenname as select * from tabellenname der zu kopierenden Tabelle;

=> testen ob die Tabelle da ist mit
DESC tabellenname

=> löscht die Spalte
Alter Table tabellenname DROP Column spaltenname; **=>**

=> Felddatentypen ändern
Alter Table tabellname MODIFY (spaltenname neuer Felddatentyp (10));

=> Spalte erstellen

Alter Table tabellenname ADD (spaltenname Felddatentyp (10));

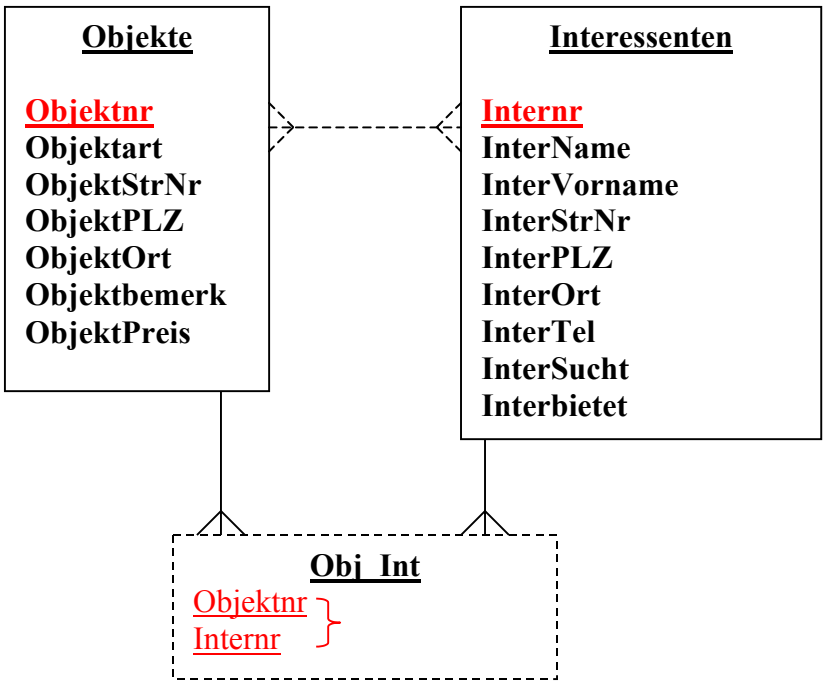
Skript erstellen:

Ed C:\pfad\Befehl1 => speichert den Befehl bzw. das Skript das nun im Editor erstellt wird!

=> Skript speichern unter.....z.B. Befehl1 Editor schließen!

SQL> r => führt den gerade geschriebenen Befehl aus!

Übung zum erstellen einer Datenbank IMOBILIEN



Erstellung der IMOBILIEN- Datenbank in ORACLE

```
Drop Table Objekte;
Create Table Objekte      (Objnr Number (3),
                           Objart Varchar2 (20),
                           Objstrnr Varchar2 (20),
                           Objplz Varchar2 (8),
                           Objort Varchar2 (15),
                           Objbem Varchar2 (100),
                           Objpreis Number (9,2),
Constraint pri_Objnr Primary Key (Objnr));


Drop Table Interressenten;
Create Table Interressenten      (Intnr Number (4),
                                  Intname Varchar2 (15),
                                  Intvorname Varchar2 (15),
                                  Intstrassenr Varchar2 (25),
                                  Intplz Varchar2 (8),
                                  Intort Varchar2 (20),
                                  Inttel Varchar2 (15),
                                  Intsucht Varchar2 (50),
                                  Intbietet Number (10,2),
Constraint pri_Internr Primary Key (Intnr));


Insert into Objekte Values (1,'Einfamilienhaus','Eisenbahnstr.2','12345','Essen','mit: Garten,Garage',255000.00);
Insert into Objekte Values (2,'Einfamilienhaus','Schillerstr.17','67548','Grünau','mit: Teich,Doppelgarage',378000.00);
Insert into Objekte Values (3,'Einfamilienhaus','Grothestr.8','12553','Düsseldorf','mit: Garten,Terasse',235674.35);
Insert into Objekte Values (4,'Einfamilienhaus','Grunewalderstr.26','15537','Tempelhof','mit: Wintergarten,Garage',365800.00);
Insert into Objekte Values (5,'Einfamilienhaus','Südowerstr.4','38920','München','mit: Pool,Garage',545700.98);
Insert into Objekte Values (6,'Mehrfamilienhaus','Storkowerstr.37','23450','Köln','mit: Tiefgarage',998800.48);
Insert into Objekte Values (7,'Doppelhaushälfte','Sperlingsgasse 10','12320','Reutlingen','mit: Pool,Garage,KFZ Stellplatz',545700.98);

Insert into Interressenten Values (1,'Jablinskie','Jan','Wuhlheiderstr.4','23543','Berlin','030-638589','Einfamilienhaus mit Garage',455000.00);
Insert into Interressenten Values (2,'Kretschmer','Mario','Am Walde 10','15537','Erkner','03362-700339','Einfamilienhaus mit Garage und Pool',500000.00);
Insert into Interressenten Values (3,'Schröder','Torsten','Friedrichstr.67','95747','Berlin','030-753839','Doppelhaushälfte mit Garten',333000.00);
Insert into Interressenten Values (4,'Knebel','Sören','Friedensstr.22','53389','Schönhausen','030-86589','Mehrfamilienhaus mit Garten',255000.00);
Insert into Interressenten Values (5,'Deich','Host','Seestr.3','34653','Bielefeld','15538-343589','Einfamilienhaus',275000.00);
```

Befehle unter Oracle

SELECT
SQL>SELECT_Feldanzeige
SQL>FROM_Tabellenname;

SELECT
Distinct keine doppelten Werte anzeigen
SQL>SELECT_DISTINCT_(Spaltenname)
SQL>FROM_Tabellenname
SQL>WHERE_.....
Beispiel:
SQL>SELECT_DISTINCT_(Abteilung)
SQL>FROM_Belegschaft
SQL>WHERE_.....

DDL
Rename (Umbenennen)
SQL>RENAME_Tabellenname_TO_neuer-Tabellenname;
Beispiel:
SQL>RENAME_Abteilung_TO_Rechnungswesen;

DDL
Alter Table (Tabelle Spalte zufügen);
SQL>ALTER_TABLE_Tabellenname_ADD_(neuer Spaltenname_Felddatentyp_());
Beispiel:
SQL>ALTER_TABLE_Abteilung_ADD_(Zurichtung_VARCHAR2_(20));

Eine Spalte mit der Eigenschaft NOT NULL können nur dann hinzugeführt werden, wenn die Tabelle leer ist.

DDL
Alter Table (Tabelle Spalte Löschen)
SQL>ALTER TABLE_Tabelle_DROP_COLUMN_Spalte;

DDL
Alter Table (Felddatentyp vergrößern)
SQL>ALTER_TABLE_Tabellenname_MODIFY_(Feldname_Felddatentyp ()_NOT NULL);
Beispiel:
SQL>ALTER_TABLE_Abteilung_MODIFY_(Zurichtung_VARCHAR2_(20)_NOT NULL);
Verkleinern der Felddatengröße geht nur wenn die Spalte noch leer ist.

DDL
Drop Table (Tabellen Löschen)
SQL>DROP_TABLE_Tabellenname;
Beispiel:
SQL>DROP TABLE Abteilung;

Primärschlüsselfeld Löschen:
ALTER_TABLE_Tabellenname_DROP_CONSTRAINT_Indexname;

Fremdschlüsselfeld Löschen:
ALTER_TABLE_Tabellenname_DROP_CONSTRAINT_Indexname;
DML
Insert Into (Daten in Tabellen Einfügen)
SQL>INSERT_INTO_Tabellenname_(Feld1_Feld2_Feld3_Feld4)_VALUES_(Daten1_Date
n2_
Daten3_Daten4);
Beispiel:
SQL>INSERT
INTO_Abteilung_(Abtnr_Abtgr_Abtmitgl_Abtumf)_VALUES_(10_Groß_19_3);

oder kurze Schreibweise:

SQL>INSERT_INTO_Tabellenname_VALUES (Daten1_Daten2_Daten3_Daten4);
Beispiel:
SQL>INSERT INTO_Abteilung_VALUES_(10_Groß_19_3);

Bei der kurzen Schreibweise darf kein Feld ausgelassen werden.

DML
Update
Set
Where (Werte in Feldern ändern)
SQL>UPDATE_Tabellenname
SQL>SET_Spaltenname_=_Neuer-Eintrag
SQL>WHERE_Welches-Feld;
Beispiel:
SQL>UPDATE_Abteilung
SQL>SET_name_'Müller'
SQL>WHERE_Arbeitnr_=_102;

DML
Delete (Löscht eine oder mehrere Zeilen aus einer Tabelle)
SQL>DELETE
SQL>FROM_Tabellenname
SQL>WHERE_Spaltenname = Nummer;
Beispiel:
SQL>DELETE
SQL>FROM_Abteilung
SQL>WHERE_Arbeiternr_=_102;

DDL
Create Table (Tabellen herstellen)
SQL>CREATE_TABLE_Tabellenname_(Feld1_Felddatentyp ()_NOT_NULL);
Beispiel:
SQL>CREATE_TABLE_Mitarbeiter_(Mitnr_NUMBER (3)_NOT_NULL);

oder mit PRIMARY KEY

SQL>CREATE_TABLE_Mitarbeiter_(Mitnr_NUMBER (3)_PRIMARY KEY);

oder mit PRIMARY KEY und Fremdschlüssel (wenn Vorhanden) am Schluss der Feldeintragungen und gleichzeitigen INDEX

SQL>CREATE_TABLE_Mitarbeiter_(Mitnr_NUMBER (3), Abteilung_VARCHAR2_(20), CONSTRAINT_PSname_PRIMARY_KEY_(Primärfeld1,Primärfeld2),CONSTRAINT_FSname_FOREIGN_KEY_(Fremdschlüsselfeld),REFERENES_Primärschlüsseltabelle_(Primärschlüsselfeld)

Fremdschlüssel separat Vergeben:
ALTER TABLE_Tabellenname_ADD_(CONSTRAINT_FSname_FOREIGN KEY_(Fremdschlüsselspalte)_REFERENCES_Tabellenname_(Primärschlüsselspalte));

Alle Tabellen Anzeigen lassen
SELECT TABLE_NAME FROM USER_TABLES
SQL>SELECT_Table_name
SQL>FROM_User_Tables;

Werte die Mehrfach Angezeigt werden Ausblenden lassen:

BREAK_ON_Spalte1_ON_Spalte2_SKIP1;
Kopiert die Tabelle mit inhalt:
SQL>CREATE TABLE Tabellenname AS SELECT Spalte FROM Tabellenname;
Kopiert die Spalte wo hier im Beispiel Spiegelei Drin Steht.
SQL>CREATE TABLE Tabellenname AS SELECT Spalte FROM Tabellenname
WHERE Spalte = 'Spiegelei';
oder wenn nur die Struktur übernommen wird:
SQL>CREATE TABLE Tabellenname AS SELECT Spalte FROM Tabellenname
WHERE 7 = 5;
7 = 5 ist ein Unwahrer Ausdruck dadurch wird nur die Struktur der Tabelle Gespeichert.

Nach Alias suchen hier im Beispiel nach unterschiedlichen Speisen.
select menge,rezeptbez,zuteilnr,zutateinheit
from zutaten z,rezepte r
where z.zutatnr = r.zuteilnr and rezeptbez = '&Suchen'

Datum Rechnen:

select sysdate-to_date ('28.08.1970','dd.mm.yyyy')/365
from dual

Ausdruck ist :
11871,4972 Tage Alt.

select (sysdate-to_date ('28.08.1970','dd.mm.yyyy'))/365
from dual

Ausdruck ist:
32,5246559 Monate Alt.

Tabellen löschen mit Primärschlüssel:

DROP_TABLE_Tabellenname_CASCADE

Kopiert aus Feld3 und Feld4 der Tabellenname1 die Daten in Feld1 und Feld2 der Tabellenname

INSERT INTO_Tabellenname_(Feld1,Feld2)
(SELECT_Feld3,Feld4
FROM_Tabellenname1)

Spalten zusammenführen mittels Pipezeichen

SELECT_ename_||'_-_'||_Job as “Name und Beruf“

Constraints

NOT NULL
UNIQUE
PRIMARY KEY
FOREIGN KEY
CHECK

Beispiel:

CREATE TABLE Dept
(deptno NUMBER (8) NOT NULL,
loc VARCHAR2 (20),
Dname VRACHAR2 (20),
CONSTRAINT dept_Dname_con UNIQUE (Dname),
CONSTRAINT ps_dept PRIMARY KEY (deptno)

CREATE TABLE Emp
(Ename VARCHAR (10) NOT NULL,
sal NUMBER (8,2) NOT NULL,
deptno NUMBER (8),
CONSTRAINT fs_emp FOREIGN KEY (deptno) REFERENCES Dept (deptno));
CONSTRAINT ck_emp CHECK (deptno between 10 AND 99)

CONSRAINTS löschen mit:
DROP TABLE Emp CASCADE CONSTRAINT;

Sequencen

CREATE SEQUENCE Sequencename...START WITH 1...INCREMENT BY 1
MAXVALUE Maximalerwert
MINVALUE Minimalwert
CYRCLE Zyklus
NO CYCLE Standard
CACHE N (Im Cache Speichert zum schnellen Zugriff)
NO CACHE ; (Nicht im Cache Spreichern erst wenn gebraucht)

INSERT INTO Tabellenname VALUES (Sequencename.NEXTVAL,....

Beispiel:
CREATE SEQUENCE seq_dept_deptno START WITH 91 INCREMENT BY 1
MAXVALUE 100
NO CACHE
NO CYCLE

Sequenzen müssen vor den Eintragungen mit INSERT INTO vergeben sein.
Müssen aber nicht unter der Jeweiligen Tabelle stehen.

Dictionary für Sequencen

SELECT Tabellenname
FROM USER_SEQUENCES;

Beispiel
SELECT emp FROM USER_SEQUENCES;

Daten eintragen

INSERT INTO emp
(Sequencename.nextvalue