

```

DROP TABLE Studenten CASCADE CONSTRAINTS;
DROP TABLE Personal CASCADE CONSTRAINTS;
DROP TABLE Projekte CASCADE CONSTRAINTS;
DROP TABLE Gruppen CASCADE CONSTRAINTS;
DROP TABLE Angemeldet CASCADE CONSTRAINTS;
DROP SEQUENCE Studenten_pk_seq;
DROP SEQUENCE Personal_pk_seq;

CREATE TABLE Studenten (
StudID          INTEGER          PRIMARY
KEY,
Nachname        VARCHAR2(30)     NOT NULL,
Vorname         VARCHAR2(30)     NOT NULL,
Email           VARCHAR2(50)     NOT NULL,
Studiengang     VARCHAR2(20)     NULL,
Matrikelnummer  NUMBER(10)       NOT NULL);

CREATE TABLE Personal (
PersID          INTEGER          PRIMARY
KEY,
Nachname        VARCHAR2(30)     NOT NULL,
Vorname         VARCHAR2(30)     NOT NULL,
Telefonnr       VARCHAR2(15)     NULL,
Email           VARCHAR2(50)     NOT NULL,
Raum            VARCHAR2(5)       NULL,
Typ             CHAR(1)          NOT NULL);

CREATE TABLE Projekte (
ProjID          INTEGER          GENERATED ALWAYS AS IDENTITY NOT NULL,
Bezeichnung     VARCHAR2(30)     NOT NULL,
Semester        VARCHAR2(10)     NOT NULL,
maxGGroesse     INTEGER          NULL,
Fach            VARCHAR2(20)     NOT NULL,
WMA_PersID      INTEGER          NOT NULL,
Prof_PersID     INTEGER          NOT NULL,
Freigeschaltet  DATE             NOT NULL,
CONSTRAINT PKProjekte PRIMARY KEY (ProjID));

CREATE TABLE Gruppen (
GrupID          INTEGER          GENERATED ALWAYS AS IDENTITY NOT NULL,
Kennung         VARCHAR2(10)     NOT NULL,
ProjID          INTEGER          NOT NULL,
PersID          INTEGER          NOT NULL,
CONSTRAINT PKGruppen PRIMARY KEY (GrupID));

CREATE TABLE Angemeldet (
AngID           INTEGER          GENERATED ALWAYS AS IDENTITY PRIMARY KEY
,
StudID          INTEGER          NOT NULL,
GrupID          INTEGER          NOT NULL);

ALTER TABLE Projekte ADD (CONSTRAINT FK_Proj_PersProf FOREIGN KEY
(Prof_PersID) REFERENCES Personal (PersID));

ALTER TABLE Projekte ADD (CONSTRAINT FK_Proj_PersWMA FOREIGN KEY
(WMA_PersID) REFERENCES Personal (PersID));

ALTER TABLE Gruppen ADD (CONSTRAINT FK_Grup_Pers FOREIGN KEY (PersID) REF-
ERENCES Personal (PersID));

ALTER TABLE Gruppen ADD (CONSTRAINT FK_Grup_Proj FOREIGN KEY (ProjID) REF-
ERENCES Projekte (ProjID));

```

----- Praktikum 3 Aufg.1 -----

--Aufgabe 1

-- Seq für Studenten und Personal

-- Rest mit Identity Column s.o : Create Table...Spaltenname Datentyp <identity klausel>

-- <identity klausel> ::= GENERATED ALWAYS/BY DEFAULT AS IDENTITY

CREATE SEQUENCE Studenten_pk_seq;

CREATE SEQUENCE Personal_pk_seq;

--Aufgabe 2

-- Syntax: **INSERT INTO** Tabellename (Spaltennamen,..) **VALUES** (WERTE)

-- alt. ohne Argumentenliste dann by default alle Spalten ausgewählt

INSERT INTO Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Hans', 'Peter', 'Hans.Peter@smail.fh-koeln.de', 'MI', 15245512);

INSERT INTO Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Harald', 'Katz', 'Harald.Katz@smail.fh-koeln.de', 'AI', 22421224);

INSERT INTO Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Franz', 'Kanns', 'Franz.Kanns@smail.fh-koeln.de', 'AI', 11421444);

INSERT INTO Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Franz', 'Hanns', 'Franz.Kanns@smail.fh-koeln.de', 'AI', 11421555);

INSERT INTO Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Franz', 'Kanns', 'Franz.Kanns2@smail.fh-koeln.de', 'AI', 11421666);

-- mit select ggf. testen **SELECT * FROM** Tabellename;

INSERT INTO Personal (PersID, Nachname, Vorname, Telefonnr, Email, Raum, Typ) **VALUES** (Person-

al_pk_seq.NEXTVAL, 'Franz', 'Kanns', 01573564854, 'Franz.Kanns@smail.fh-koeln.de', 0515, 'S');

INSERT INTO Personal (PersID, Nachname, Vorname, Telefonnr, Email, Raum, Typ) **VALUES** (Person-

al_pk_seq.NEXTVAL, 'Hermann', 'Hats', 01543544494, 'Hermann.Hats@smail.fh-koeln.de', 0519, 'P');

-- same here

-- Bei identity column muss man Argumentenliste füllen

INSERT INTO

Projekte (Bezeichnung, Semester, maxGGroesse, Fach, WMA_PersID, Prof_PersID, Freigeschaltet) **VALUES** ('DB-

Praktikum', '3', '1000', 'Datenbanken', 1, 2, TO_DATE('2018/09/03 21:02:44', 'yyyy/mm/dd hh24:mi:ss'));

INSERT INTO

Projekte (Bezeichnung, Semester, maxGGroesse, Fach, WMA_PersID, Prof_PersID, Freigeschaltet) **VALUES** ('KTN-

Praktikum', '3', '1000', 'Kommunikationstech', 1, 2, TO_DATE('2018/09/03 21:02:44', 'yyyy/mm/dd hh24:mi:ss'));

--testen

INSERT INTO Gruppen (Kennung, ProjID, PersID) **VALUES** ('GRP001', 1, 1);

INSERT INTO Gruppen (Kennung, ProjID, PersID) **VALUES** ('GRP002', 1, 2);

--same

INSERT INTO Angemeldet (StudID, GrupID) **VALUES** (1, 1);

commit; -- commit zum persistenten abspeichern

-- Integritätsprüfung zum testen c) beide finden sofort statt

--**INSERT INTO** Studenten **VALUES**

(Studenten_pk_seq.NEXTVAL, 'Florian', 'Wolf', 'Florian.Wolf@smail.fh-koeln.de', 'TI', 23421334);

--**UPDATE** Studenten set StudID=1 WHERE StudID=2;

--commit;

```

--SELECT * FROM Projekte;
--SELECT * FROM Personal;
--commit;
--commit;
--UPDATE Projekte set ProjID=1 WHERE ProjID=1;
--commit;
--INSERT INTO
Projekte(ProjID,Bezeichnung,Semester,maxGGroesse,Fach,WMA_PersID,Prof_PersID,Freigeschaltet) VALUES (2,'KTN-Praktikum','3','1000','Kommunikationstech',1,2,TO_DATE('2018/09/03 21:02:44','yyyy/mm/dd hh24:mi:ss'));
--commit;

--Aufgabe d)
-- Integritätsprüfung zum testen FK-Constraint
--(Fremdschlüssel existiert nicht)
-- Zum Testen Tabelle neu erstellen mit deferrable initially deferred beim
Constraint und Update laufen lassen
--UPDATE Gruppen SET PersID=3 WHERE PersID=1;
--(Fremdschlüssel hat falschen Datentyp)
--UPDATE Gruppen SET PersID="Hans" WHERE PersID=1;
--(Fremdschlüssel) Master Entity wird gelöscht
--DELETE FROM Personal WHERE PersID=1;
--commit;

-- e) Email Check mit Deffered Check
ALTER TABLE Studenten ADD CHECK (EMAIL LIKE '%@%.%') deferrable initially
deferred;
--UPDATE Studenten SET Email='Hans@koelnde' WHERE StudID=1;
--SELECT * FROM Studenten;
commit;

--f)
ALTER TABLE Personal ADD CHECK((Typ IN ('P','p','S','s','W','w','V','v')));
UPDATE Personal SET Typ='p' WHERE PersID=1;
SELECT * FROM Personal;
ROLLBACK;

--g)
ALTER TABLE Projekte MODIFY freigeschaltet NULL;
ALTER TABLE Projekte MODIFY fach NULL;
ALTER TABLE Projekte MODIFY Prof_PersID NULL;
ALTER TABLE Projekte ADD CHECK((freigeschaltet IS NOT NULL AND fach IS NOT
NULL AND Prof_PersID IS NOT NULL) OR (freigeschaltet IS NULL)) deferrable
initially deferred;
--commit;
--INSERT INTO
Projekte(Bezeichnung,Semester,maxGGroesse,WMA_PersID,freigeschaltet) VALUES
('DB2-Praktikum','3','1000',1,TO_DATE('2018/09/03 21:02:44','yyyy/mm/dd
hh24:mi:ss'));
--INSERT INTO
Projekte(Bezeichnung,Semester,maxGGroesse,Fach,WMA_PersID,Prof_PersID) VAL-
UES ('DB2-Praktikum2','5','1000','Datenbanken',1,2);
--INSERT INTO Projekte(Bezeichnung,Semester,maxGGroesse,WMA_PersID) VALUES
('DB2-Praktikum4','5','1000',1);
--INSERT INTO
Projekte(Bezeichnung,Semester,maxGGroesse,Fach,WMA_PersID,Prof_PersID) VAL-
UES ('DB2-Praktikum5','5','1000','Datenbanken',1,2);
--INSERT INTO
Projekte(Bezeichnung,Semester,maxGGroesse,Fach,WMA_PersID,Prof_PersID,Freig-
eschaltet) VALUES ('DB2-
Praktikum3','5','1000','Datenbanken',1,2,TO_DATE('2018/09/03

```

```

21:02:44','yyyy/mm/dd hh24:mi:ss')));
--SELECT * FROM Projekte;
--commit;

--h)
--ALTER TABLE Personal ADD CHECK((Typ IN ('S','s')));
SELECT Nachname, Vorname, Email FROM Personal WHERE (Nachname, Vorname,
Email) IN (SELECT Studenten.Nachname, Studenten.Vorname, Studenten.Email
From Studenten) AND (Typ = 'S' OR Typ = 's');
SELECT Nachname, Vorname, Email From Personal
MINUS
SELECT Nachname, Vorname, Email FROM Studenten;

select * FROM Studenten;
select * FROM Personal;
select * FROM Projekte;
select * FROM Gruppen;
select * FROM Angemeldet;

DELETE FROM Studenten;
DELETE FROM Gruppen;
DELETE FROM Projekte;
DELETE FROM Personal;
DELETE FROM Angemeldet;

```