1.1

CREATE TABLE Mitarbeiter (MitID CHAR(3) NOT NULL,

Nachname VARCHAR(20) NOT NULL,

Vorname VARCHAR(20),

Ort VARCHAR(30) NOT NULL,

Gebdat DATE NOT NULL,

Beruf VARCHAR(15) NOT NULL,

Telnr VARCHAR(20),

PRIMARY KEY(MitID))

CREATE TABLE Projekt (ProNr INT NOT NULL,

ProName VARCHAR(20) NOT NULL,

ProOrt VARCHAR(30) NOT NULL,

ProBeschreibung VARCHAR(30),

ProAufwand INT NOT NULL,

ProLeiter VARCHAR(20),

PRIMARY KEY (ProNr))

CREATE TABLE Zuordnung (MitID CHAR(3) NOT NULL,

ProNr INT NOT NULL,

Istanteil FLOAT NOT NULL,

PRIMARY KEY(MitID, ProNr),

FOREIGN KEY (MitID) REFERENCES Mitarbeiter (MitID),

FOREIGN KEY (ProNr) REFERENCES Projekt (ProNr))

1.2

1.3

ALTER TABLE Zuordnung ADD Plananteil FLOAT NOT NULL

1.4

INSERT INTO Mitarbeiter SELECT \* FROM depotN.dbo.quelleMitarbeiter

INSERT INTO Projekt (ProName, ProNr, ProOrt, ProLeiter, ProAufwand)

SELECT ProName, ProNr, ProOrt, Leiter, Aufwand FROM depotN.dbo.quelleProjekt

INSERT INTO Zuordnung SELECT \* FROM depotN.dbo.quelleZuordnung

1.6

insert into Mitarbeiter(MitID, Nachname, Vorname, Ort, Gebdat, Beruf, Telnr ) values ('200', 'Tsurkan', 'Oksana', 'Dresden', '30.04.1997', 'Praktikant', '00491746354686')

insert into Mitarbeiter(MitID, Nachname, Vorname, Ort, Gebdat, Beruf, Telnr ) values ('201', 'Shylina', 'Kateryna', 'Dresden', '30.04.1996', 'Praktikant', '00491754754686')

**23.05.2016**

1.7

CREATE UNIQUE INDEX Name\_idx ON Mitarbeiter (Nachname, Vorname)

DROP INDEX Beruf\_idx ON Mitarbeiter

1.8

CREATE VIEW vOrtBeruf (MitID, Nachname, Vorname, Ort, Beruf)

AS SELECT MitID, Nachname, Vorname, Ort, Beruf

FROM Mitarbeiter WHERE Beruf != 'Vertreter'

1.9

CREATE VIEW vPaz (MitID, Nachname, Projektname, Plananteil)

AS SELECT m.MitID, m.Nachname, p.Proname, z.Plananteil

FROM Mitarbeiter m, Projekt p, Zuordnung z

WHERE z.ProNr=p.ProNr AND m.MitID = z.MitID

* *Immer so viele Beziehungen (bei WHERE) wie es möglich ist (man soll nicht zu viele Daten zurückbekommen)*

1.10

select \* from vOrtBeruf

select \* from vPaz

2.1

1. SELECT \* FROM Mitarbeiter WHERE Ort = 'Dresden'
2. SELECT \* FROM Mitarbeiter WHERE Ort != 'Dresden'
3. SELECT \* FROM Projekt WHERE ProName LIKE 'R%'

*(SELECT Proname FROM Projekt WHERE ProName LIKE 'R%')*

1. SELECT \* FROM Mitarbeiter

WHERE Ort = 'Radebeul' AND Beruf LIKE 'Dipl%'

ORDER BY Nachname

1. SELECT \* FROM Zuordnung

WHERE Istanteil > 0.7 OR Plananteil > 0.7

1. SELECT \* FROM Projekt WHERE ProLeiter IS NULL

2.2

1. SELECT COUNT (\*) FROM Mitarbeiter
2. SELECT Ort, COUNT ( Ort) FROM Mitarbeiter Group By Ort
3. SELECT SUM (Istanteil) FROM Zuordnung WHERE ProNr = 31
4. SELECT MitID, SUM(Istanteil - Plananteil) FROM Zuordnung GROUP BY MitID
5. SELECT ProAufwand, MAX(ProAufwand), SUM(ProAufwand), COUNT(ProNr) FROM Projekt GROUP BY ProAufwand, ProNr

**30.05.2016**

**2.3**

a. SELECT Nachname, Vorname, Gebdat, DATEDIFF(yy, Gebdat, GETDATE()) AS Alt FROM Mitarbeiter

b. SELECT Nachname, Vorname FROM Mitarbeiter WHERE DATEDIFF(yy, Gebdat, GETDATE()) > 30

c. SELECT Nachname, Vorname, Gebdat FROM Mitarbeiter WHERE MONTH(Gebdat) = 5

d. SELECT Gebdat, Nachname, Vorname FROM Mitarbeiter ORDER BY DAY(Gebdat), MONTH(Gebdat)

e. SELECT AVG(DATEDIFF(yy, Gebdat, GETDATE())) AS Durchschnittsalter FROM Mitarbeiter

f. SELECT Ort, AVG(DATEDIFF(yy, Gebdat, GETDATE())) AS Durchschnittsalter FROM Mitarbeiter GROUP BY Ort

**2.4**

a)

SELECT Ort, AVG(DATEDIFF(yy, Gebdat, GETDATE())) AS Durchschnittsalter FROM Mitarbeiter GROUP BY Ort

SELECT m.MitID, m.Nachname, z.ProNr, z.Istanteil FROM Mitarbeiter m INNER JOIN Zuordnung z ON m.MitID = z.MitID

b)

SELECT z.ProNr, p.ProName, m.Nachname, m.Vorname, m.Beruf FROM Mitarbeiter m INNER JOIN Zuordnung z ON m.MitID = z.MitID INNER JOIN Projekt p ON z.ProNr = p.ProNr ORDER BY ProNr

SELECT z.ProNr, p.ProName, m.Nachname, m.Vorname, m.Beruf FROM Mitarbeiter m JOIN Zuordnung z ON m.MitID = z.MitID JOIN Projekt p ON z.ProNr = p.ProNr ORDER BY ProNr

c)

SELECT m.MitID, m.Nachname, m.Vorname, z.ProNr, p.ProName, p.ProLeiter, z.Plananteil FROM Mitarbeiter m INNER JOIN Zuordnung z ON m.MitID=z.MitID INNER JOIN Projekt p ON z.ProNr=p.ProNr WHERE ProName = 'Reportgenerator'

SELECT m.MitID, m.Nachname, m.Vorname, z.ProNr, p.ProName, p.ProLeiter, z.Plananteil FROM Mitarbeiter m, Zuordnung z, Projekt p WHERE z.ProNr=p.ProNr AND m.MitID=z.MitID AND ProName = 'Reportgenerator'

d)

SELECT m.MitID, m.Nachname, m.Vorname, p.ProNr, p.ProName, p.ProLeiter FROM Mitarbeiter m, Projekt p WHERE m.Nachname=p.Proleiter

SELECT m.MitID, m.Nachname, m.Vorname, p.ProNr, p.ProName, p.ProLeiter FROM Mitarbeiter m INNER JOIN Projekt p ON m.Nachname=p.Proleiter

e)

SELECT z.MitID, z.Plananteil, p.ProNr FROM Zuordnung z RIGHT OUTER JOIN Projekt p ON p.ProNr = z.ProNr ORDER BY MitID, ProNr

SELECT z.MitID, z.Plananteil, p.ProNr FROM Projekt p LEFT OUTER JOIN Zuordnung z ON p.ProNr = z.ProNr ORDER BY MitID, ProNr

**2.5**

a)

SELECT \* FROM Mitarbeiter m WHERE m.Nachname NOT IN (SELECT p.ProLeiter FROM Projekt p WHERE p.ProLeiter = m.Nachname )

b)

SELECT \* FROM Mitarbeiter WHERE Gebdat = (SELECT MAX(m.Gebdat) FROM Mitarbeiter m WHERE Beruf = 'Dipl.-Ing.')

31.05.2016

c)

UNKORRELIERT!

SELECT MitId, Plananteil, z.ProNr FROM Zuordnung z,

(SELECT ProNr, MAX(Plananteil) maximal FROM Zuordnung GROUP BY ProNr ) Plan\_Anteil

WHERE Plananteil = maximal AND z.ProNr= Plan\_Anteil.ProNr

KORRELIERT!

SELECT MitId, Plananteil, z.ProNr FROM Zuordnung z

WHERE Plananteil = (SELECT MAX(Plananteil) maximal FROM Zuordnung WHERE Zuordnung.ProNr = z.ProNr )

**2.6**

a)

SELECT MitID, SUM(Istanteil) AS ist\_anteil FROM Zuordnung GROUP BY MitID HAVING SUM(Istanteil)>0.7

b)

SELECT ProName, SUM(ProAufwand) AS pro\_aufwand, SUM(Plananteil) AS plan\_anteil FROM Projekt, Zuordnung GROUP BY ProName HAVING SUM(Plananteil)<SUM(ProAufwand)

**2.7**

a)

UPDATE Mitarbeiter SET Ort='Dresden' WHERE Nachname = 'Fuchs' OR Nachname = 'Elster'

b)

*Überprüfen des Ergebnisses:* SELECT \* FROM Mitarbeiter WHERE Nachname = 'Fuchs'

UPDATE Mitarbeiter SET Beruf='Dipl.-Ing.' WHERE Nachname = 'Fuchs'

c)

**DELETE FROM Zuordnung WHERE MitID IN (SELECT MitID FROM Mitarbeiter WHERE Beruf = 'Industriekauf.')**

**DELETE FROM Mitarbeiter WHERE Beruf = 'Industriekauf.'**

**06.06.2016**

d)

SELECT MitID INTO temploesch

FROM Zuordnung GROUP BY MitId HAVING SUM(Plananteil)<0.3

SELECT \* FROM temploesch

*// von beiden Tabellen löschen, aber zuerst von der Tabelle Zuordnung die betroffenen Datensätze rausholen*

DELETE FROM Zuordnung WHERE MitID IN (SELECT MitID FROM temploesch)

DELETE FROM Mitarbeiter WHERE MitID IN (SELECT MitID FROM temploesch)

DROP TABLE temploesch

**2.8**

e)

SELECT Ort FROM Mitarbeiter

EXCEPT

SELECT ProOrt FROM Projekt

f)

SELECT Ort FROM Mitarbeiter

INTERSECT

SELECT ProOrt FROM Projekt

g)

SELECT Ort FROM Mitarbeiter

UNION

SELECT ProOrt FROM Projekt

**2.9**

a)

**EQUI JOIN**

*((SELECT Mitarbeiter.\*, Projekt.\*, Zuordnung.\* FROM Mitarbeiter, Projekt, Zuordnung WHERE Projekt.ProNr = Zuordnung.ProNr AND Zuordnung.MitID= Mitarbeiter.MitID))*

SELECT Mitarbeiter.\*, Projekt.\* FROM Mitarbeiter, Projekt WHERE Projekt.ProLeiter= Mitarbeiter.Nachname

**NATURALJOIN**

SELECT Mitarbeiter.\*, Projekt.ProAufwand FROM Mitarbeiter, Projekt WHERE Projekt.ProLeiter= Mitarbeiter.Nachname

**Ein kartesisches Produkt**

SELECT Mitarbeiter.\*, Projekt.\* FROM Mitarbeiter CROSS JOIN Projekt

((((((((((((SELECT Mitarbeiter.\*, Projekt.\* FROM Mitarbeiter CROSS JOIN Projekt WHERE Mitarbeiter.Nachname=Projekt.ProLeiter)))))))))))))))) – WIRGUNG ANALOG INNER JOIN

**2.10**

*SELECT MitID, Nachname, DATEDIFF(yy, gebdat, GETDATE()) AS Age FROM Mitarbeiter*

*SELECT \* FROM Mitarbeiter WHERE MONTH(GETDATE())<MONTH(Gebdat) AND DAY(GETDATE())<DAY(Gebdat)*

*loesung*

SELECT MitID, Nachname, DATEDIFF(yy, gebdat, GETDATE()) AS Age,

CASE

WHEN MONTH(GETDATE())<MONTH(Gebdat) AND DAY(GETDATE())<DAY(Gebdat) THEN 'TRUE'

ELSE 'FALSE'

END AS Loesung

FROM Mitarbeiter

**07.06.2016**

**3.1**

BEGIN

SELECT Nachname, ProName FROM Mitarbeiter, Projekt WHERE Projekt.ProLeiter = Mitarbeiter.Nachname

END

**3.2**

BEGIN

SELECT Nachname, Beruf, ProName FROM Mitarbeiter m, Projekt p WHERE p.ProNr=31

END

**3.3**

--SELECT \* INTO Mitarbeiterkopie FROM Mitarbeiter

DROP PROCEDURE Loesch\_Ma

CREATE Procedure Loesch\_Ma @alter INT

AS

WHILE(@alter>0)

BEGIN

WAITFOR DELAY '00:00:01'

SELECT \* FROM Mitarbeiterkopie

--SELECT @zahl = COUNT(\*) FROM Mitarbeiterkopie

PRINT @alter

SELECT @alter = @alter - 1

UPDATE Mitarbeiterkopie SET gebdat=DATEADD(yy, -1, gebdat)

DELETE FROM Mitarbeiterkopie WHERE DATEDIFF(yy, gebdat, GETDATE()) >= 60

END

--man muss den Alter der MA aktualisieren -> UPDATE… SET gebdat=DATEADD(yy, -1,gebdat)

--EXECUTE Loesch\_Ma 50 //einen kleineren Wert eingeben

-- DROP PROCEDURE Loesch\_Ma;