1. In Exercise 7 (2c) you haven’t been able to create the FOREIN KEY.

a. Do you know why? If not ask me and I will tell it to you. If yes, then explain it to me. Advice: Look at the attribute type!

**NOT POSSIBLE BECAUSE anrede.anrede is of type VARCHAR, meanwhile, the attribute person.anrede is of type INTEGER**

b. Drop your attribute anrede FROM person and CREATE a new attribute (anrede\_id) which you can use as your FOREIGN KEY!

**ALTER TABLE person**

**DROP anrede;**

**ALTER TABLE person**

**ADD anrede\_id INTEGER**

c. Retry to add the constraint with the name fk\_person2anrede and set the anrede\_id as a FOREIGN KEY from the table person which REFERENCES to the id from anrede. Please use RESTRICT for UPDATE & DELETE!

**ALTER TABLE person**

**ADD CONSTRAINT fk\_anrede2person**

**FOREIGN KEY (anrede\_id) REFERENCES anrede (id)**

**ON UPDATE RESTRICT**

**ON DELETE RESTRICT;**

2 You have received some new requisitions from your Marketing Manager. He decided to add some company informations. So please add the following attributes:

a. Attribute: company\_name Type: VARCHAR(30) b. Attribute: str\_company Type: VARCHAR (60)

c. Attribute: plz\_company\_id Type: INTEGER UNSIGNED d. Attribute: tel\_company Type: VARCHAR (40)

e. Attribute: email\_company Type: VARCHAR (40) f. Attribute: url Type: VARCHAR (45)

**CREATE TABLE company**

**(**

**company\_name VARCHAR(30),**

**str\_company VARCHAR(60),**

**plz\_company\_id INTEGER UNSIGNED,**

**tel\_company VARCHAR(40),**

**email\_company VARCHAR(40),**

**url VARCHAR(45)**

**);**

g. Add a constraint with the name fk\_person2plz\_company and set the plz\_company\_id as a FOREIGN KEY from the table person which REFERENCES to the id from plz. Please use RESTRICT for UPDATE & DELETE!

**ALTER TABLE company**

**ADD CONSTRAINT fk\_person2plz\_company**

**FOREIGN KEY (plz\_company\_id) REFERENCES plz (ID)**

**ON UPDATE RESTRICT**

**ON DELETE RESTRICT**

3. Your Marketing Manages is saying that all your datasets in your table person are so called SOHOs (Small Office / Home Office).

a. That’s why he wants you to fill the new fields (2b till 2d) with the existing private information (e.g. the address of the company is the same like the private).

**INSERT INTO company (str\_company, plz\_company\_id, tel\_company)**

**(SELECT DISTINCT str, plz\_id, tel FROM person);**

b. He also wants you to fill at least 5 datasets with a company name! (I only have 3)

**UPDATE company**

**SET company\_name = 'a' WHERE plz\_company\_id = 2;**

**UPDATE company**

**SET company\_name = 'b' WHERE plz\_company\_id = 3;**

**UPDATE company**

**SET company\_name = 'c' WHERE plz\_company\_id = 4;**

**Or, in case the prof wants totally new entries**

**INSERT INTO company (company\_name)**

**VALUES ('a');**

**INSERT INTO company (company\_name)**

**VALUES ('b');**

**INSERT INTO company (company\_name)**

**VALUES ('c');**

c. The Email of each employee should be generated automatically with the first three letters of the lastname and first three letters of the firstname, following by @companyname.com. Advice: Look at the command LEFT.

**ALTER TABLE company**

**ALTER email\_company SET DEFAULT ????????????**

**????? SELECT LEFT (nnam,3) AS nnam, LEFT (vnam,3) AS vnam, id FROM person WHERE str = 'Musterstr 1';**

4. While you are filling the company information you recognized that the field “company\_name” is not big enough.

a. That’s why you have to change the datatype in VARCHAR (60)!

**ALTER TABLE company**

**MODIFY COLUMN company\_name varchar(60);**

b. What should happen after changes if you don’t fill any information in the field “company\_name”? Please explain why it’s possible to let this field empty. Modify this column to prevent empty datasets.

**NOTHING WOULD HAPPEN, ERRORWISE, BUT NULL WILL BE INSERTED IN SAID FIELD.**

**If I try to modify company\_name to set NOT NULL, I would get an error, because a NULL value already exists in one of the rows.**

**FIRST OF ALL:**

**UPDATE company**

**SET company\_name = 'd' WHERE company\_name IS NULL;**

**SECONDLY:**

**ALTER TABLE company**

**MODIFY COLUMN company\_name varchar(60) NOT NULL;**