**Assignment Hand In**

2. semester

( 5 studypoints)

Relational Model and SQL

**Hand out:** Friday 5/2 2016 at 8.30 on Fronter

**Hand in:** Sunday 7/2 2016 at 12.00 AM  
Send your answer as an attachment by mail to "tutor2sem2016f@gmail.com" (i.e. not via Fronter!)

**IMPORTANT:**

**Individual**: Must be answered and handed in individually

**1 single document:** Must be handed in as one single document (word/pdf).

**Student info on first page:** Daniel Grønbjerg, 16cos2v15f, Relational Model and SQL

**Assignment 1**

a)

Underline all primary keys and use **bold** to indicate foreign keys, for these schemas:

employees (eno, ename, **zip**, hdate)

parts (pno, pname, qoh, price, olevel)

customers (cno, cname, street, **zip**, phone)

orders (ono, **cno**, **eno**, received, shipped)

odetails (**ono**, **pno**, qty)

zipcodes (zip, city)

*Attribute names explained:*

Table employees: hdate = hire date

Table parts: qoh = Quantity on Hand

Table parts: olevel = Order Level – min. amount in stock before reordering.

b)

Write an SQL-script with all necessary statements to create tables in assignment 1.a

* You decide appropriate data types
* Assign fitting integrity constraints (not null, default)
* Determine primary- and foreign keys
* Run the script to create the database

DROP TABLE if exists odetails;

DROP TABLE if exists orders;

DROP TABLE if exists parts;

DROP TABLE if exists customers;

DROP TABLE if exists employees;

DROP TABLE if exists zipcodes;

CREATE TABLE zipcodes (

zip integer not null,

city varchar(30),

PRIMARY KEY (zip)

);

CREATE TABLE employees (

eno integer auto\_increment,

ename varchar(30) not null,

zip integer,

hdate date,

PRIMARY KEY (eno),

FOREIGN KEY (zip) REFERENCES zipcodes (zip)

);

CREATE TABLE parts (

pno integer auto\_increment,

pname varchar(30) not null,

qoh integer,

price double,

olevel integer,

PRIMARY KEY (pno),

CHECK (qoh>0)

);

CREATE TABLE customers (

cno integer auto\_increment,

cname varchar(30) not null,

street varchar (30),

zip integer,

phone varchar(30),

PRIMARY KEY (cno),

FOREIGN KEY (zip) REFERENCES zipcodes (zip)

);

CREATE TABLE orders (

ono integer auto\_increment,

cno integer,

eno integer,

received Date,

shipped date,

PRIMARY KEY (ono),

FOREIGN KEY (cno) REFERENCES customers (cno),

FOREIGN KEY (eno) REFERENCES employees (eno)

);

CREATE TABLE odetails (

ono integer,

pno integer,

qty integer,

CHECK (qty>parts.qoh),

PRIMARY KEY (ono,pno),

FOREIGN KEY (ono) REFERENCES orders (ono),

FOREIGN KEY (pno) REFERENCES parts (pno)

);

commit;

c)

Run the script ”insert\_mail.sql” (Can be found on Fronter), which populates the database.

Hint: If the script fails, the cause might be a mismatch in the data types or constraints. Adjust the script or the schemas.

**Assingment 2**

The following SQL statements will generate errors. Identify the cause and fix the error. Write down what you did and why.

a)

insert into employees values (1002, 'Olsen', 67226, '2006-09-13');

-- Error: Dublicate entry: primary key must be unique, changed eno.

insert into employees values (1007, 'Olsen', 67226, '2006-09-13');

b)

insert into odetails values (1020, 10900);

-- Error: no qty specified, changed qty to 1.

insert into odetails values (1020, 10900, 1);

c)

insert into employees values (1004, 'Jensen', 66666, '2006-09-15');

-- Error: zipcode not in the system. Punished the jerk by moving him to Kansas.

insert into employees values (1004, 'Jensen', 50302, '2006-09-15');

d)

insert into parts values (11000, Harry Potter, 12, 23.25, 12);

-- No quotation marks. Added Quotation marks.

insert into parts values (11000, 'Harry Potter', 12, 23.25, 12);

e)

insert into parts values (11001, 'Marx Brothers', 10, -22.99, 20);

-- Error: Negative price. Can’t fix through constrains, so doesn’t actually produce an error.

insert into parts values (11001, 'Marx Brothers', 10, 22.99, 20);

f)

update zipcodes set values (city='Los Angeles') where zip=67226;

-- Error syntax for updating was wrong.

update zipcodes set city='Los Angeles' where zip=67226;

**Assignment 3**

Insert 1 additional row / tuple in each table. Run the 6 statements (6 in total) in a sequence, so that the integrity of the database is maintained at all time.

insert into zipcodes values (10007, 'New York');

insert into parts values (1,'Beer', 12, 5,2);

insert into employees values (1,'Batman', 10007, '2006-09-13');

insert into customers values (1,'Robin','Gotham', 10007, '2008-05-13');

insert into orders values (1,1111, 1007, '2006-09-13',null);

insert into odetails values (1,1,2);

**Assignment 4**

Write queries to fetch the results requested below:

1. The names of all customers  
   select cname from customers;
2. The names of products of which there are at least 150 pieces in stock  
   select pname from parts where qoh>150;
3. Names and zipcodes of all customers with a phone number which ends with ’55’.  
   select cname, zip from customers where phone like '%55';
4. Names of products which cost less than 18.00  
   select pname from parts where price <18;
5. The name and city of all customers  
   select cname,city FROM customers NATURAL JOIN zipcodes;
6. Order numbers for orders made by an employee named ’Jones’  
   select ono from orders natural join employees where ename='Jones';
7. Customer name and order number for all orders, where the customers address begins with ‘1’.  
   select cname,ono from orders natural join customers where street like '1%';
8. All information about employees and the cities they live in. Also include information about cities without employees (Hint: outer join).   
   select \* from employees e right outer join zipcodes z on e.zip=z.zip;
9. Customer names and order numbers for all orders where the customer lives in ”Los Angeles”  
   select cname, ono from customers c natural join zipcodes, orders o where c.cno=o.cno and city='Los Angeles';
10. A list with name, quantity, price and total price for all products on the order with order number 1020.  
    select pname,qty, price, qty\*price from odetails natural join parts where ono=1020;
11. The price of all orders combined.  
    select sum(qty\*price) from odetails natural join parts;