Database (Individual study point exercises ‐ 14 study points)

**Hand‐in**: In moodleroom. Solution document must contain headings for each exercise solution with solutions under each heading. Hand in must be in pdf format.

**Deadline**: Sunday February 19th at 23.59

# Exercise 1

1. Underline all primary keys and use **bold** to indicate foreign keys, for these tables:

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)

*Explanations of attribute names:*

|  |  |
| --- | --- |
| *hdate* in employees table:  *qoh* in parts table: *olevel* in table parts: | hire date  quantity on hand (dk: antal på lager) order level (e.g. min. amount in stock before reordering) |

1. Write an SQL‐script with all necessary statements to create tables in exercise 1.a
   * Define appropriate data types
   * Define primary keys and foreign keys
   * Assign necessary integrity constraints (not null, default)
   * Run the script to create the database

drop tables if exists odetails;

drop tables if exists orders;

drop tables if exists customers;

drop tables if exists parts;

drop tables if exists employees;

drop tables if exists zipcodes;

CREATE TABLE `zipcodes` (

`zip` int(11) NOT NULL,

`city` varchar(45),

PRIMARY KEY (`zip`)

);

CREATE TABLE `employees` (

`eno` int(11) NOT NULL,

`ename` varchar(45),

`zip` int(11),

`hdate` varchar(45),

PRIMARY KEY (`eno`),

CONSTRAINT `fk\_emp` FOREIGN KEY (`zip`) REFERENCES `zipcodes` (`zip`)

);

CREATE TABLE `parts` (

`pno` int(11) NOT NULL,

`pname` varchar(45),

`qoh` int(11),

`price` double,

`olevel` varchar(45),

PRIMARY KEY (`pno`)

);

CREATE TABLE `customers` (

`cno` int(11) NOT NULL,

`cname` varchar(45),

`street` varchar(45),

`zip` int(11),

`phone` varchar(45),

PRIMARY KEY (`cno`),

CONSTRAINT `fk\_cus` FOREIGN KEY (`zip`) REFERENCES `zipcodes` (`zip`)

);

CREATE TABLE `orders` (

`ono` int(11) NOT NULL,

`cno` int(11) NOT NULL,

`eno` int(11) NOT NULL,

`recieved` varchar(45),

`shipped` varchar(45),

PRIMARY KEY (`ono`),

CONSTRAINT `fk\_orders` FOREIGN KEY (`cno`) REFERENCES `customers` (`cno`)

);

CREATE TABLE `odetails` (

`ono` int(11) NOT NULL,

`pno` int(11) NOT NULL,

`qty` int(11),

PRIMARY KEY (`ono`, `pno`),

CONSTRAINT `fk\_details` FOREIGN KEY (`pno`) REFERENCES `parts` (`pno`)

);

1. Run the script ”insert\_mail.sql” (can be found on github), which populates the database. Hint: If the script fails, there might be a mismatch in the data types or constraints between the database that you just created and the insert statements in the SQL script. Adjust the script or the database tables.

insert into zipcodes values

(67226,'Wichita');

insert into zipcodes values

(60606,'Fort Dodge');

insert into zipcodes values

(50302,'Kansas City');

insert into zipcodes values

(54444,'Columbia');

insert into zipcodes values

(66002,'Liberal');

insert into zipcodes values

(61111,'Fort Hays');

insert into employees values

(1000,'Jones',67226,'1995-12-12');

insert into employees values

(1001,'Smith',60606,'1992-01-01');

insert into employees values

(1002,'Brown',50302,'1944-09-01');

insert into parts values

(10506,'Land Before Time I',200,19.99,'20');

insert into parts values

(10507,'Land Before Time II',156,19.99,'20');

insert into parts values

(10508,'Land Before Time III',190,19.99,'20');

insert into parts values

(10509,'Land Before Time IV',60,19.99,'20');

insert into parts values

(10601,'Sleeping Beauty',300,24.99,'20');

insert into parts values

(10701,'When Harry Met Sally',120,19.99,'30');

insert into parts values

(10800,'Dirty Harry',140,14.99,'30');

insert into parts values

(10900,'Dr. Zhivago',100,24.99,'30');

insert into customers values

(1111,'Charles','123 Main St.',67226,'316-636-5555');

insert into customers values

(2222,'Bertram','237 Ash Avenue',67226,'316-689-5555');

insert into customers values

(3333,'Barbara','111 Inwood St.',60606,'316-111-1234');

insert into orders values

(1020,1111,1000,'1994-12-10','1994-12-12');

insert into orders values

(1021,1111,1000,'1994-12-01','1995-01-15');

insert into orders values

(1022,2222,1001,'1995-02-13','1995-02-20');

insert into orders values

(1023,3333,1001,'1995-02-13','1995-02-20');

insert into orders values

(1024,2222,1002,'1995-02-13','1995-02-20');

insert into orders values

(1025,3333,1002,'1995-02-13','1995-02-20');

insert into odetails values

(1020,10506,1);

insert into odetails values

(1020,10507,1);

insert into odetails values

(1020,10508,2);

insert into odetails values

(1020,10509,3);

insert into odetails values

(1021,10601,4);

insert into odetails values

(1022,10601,1);

insert into odetails values

(1022,10701,1);

insert into odetails values

(1023,10800,1);

insert into odetails values

(1023,10900,1);

# Exercise 2

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

1. insert into employees values (1002, 'Olsen', 67226, '2006-09-13');
2. insert into odetails values (1020, 10900);
   1. mangler at definere en foreign key da not null (mangler eno)
3. insert into employees values (1004, 'Jensen', 66666, '2006-09-15');
4. insert into parts values (11000, Harry Potter, 12, 23.25, 12);
   1. mangler gåseøjne ved harry potter
5. insert into parts values (11001, 'Marx Brothers', 10, -22.99, 20);
6. update zipcodes set values (city='Los Angeles') where zip=67226;
   1. Forkert syntax med update. Rigtig svar: update zipcodes set city='Los Angeles' where zip=67226;

# Exercise 3

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

/\*opgave 3\*/

insert into zipcodes values

(2635,'Ishoej');

insert into employees values

(1003,'Oztank',2635,'1993-12-12');

insert into parts values

(11000,'xxx',100,20.88,'50');

insert into customers values

(5555,'hasan','gildbrovej',2635,'316-636-5559');

insert into orders values

(1029,5555,1003,'1994-12-10','1994-12-12');

insert into odetails values

(1029,11000,3);

commit;

# Exercise 4

Write queries to retrieve the results requested below from the database:

1. The names of all customers

SELECT cname from customers;

1. The names of products of which there are at least 150 pieces in stock

SELECT pname from parts where qoh <= 150;

1. Names and zip codes of all customers with a phone number which ends with ’55’.

select \* from customers where phone like '%55'

1. Names of products which cost less than 18.00

select \* from parts where price < 18.00;

1. The name and city of all customers

select c.cname, z.city from customers c, zipcodes z where c.zip = z.zip;

1. Order numbers for orders made by an employee named ’Jones’

select o.ono from orders o, employees e where o.eno = e.eno;

1. Customer name and order number for all orders, where the customers address begins with ‘1’.

select c.cname, o.ono from customers c, orders o where c.cno = o.cno and c.street like '1%';

1. All information about employees and the cities they live in. Include information about cities without employees (Hint: outer join).

select \* from employees e right join zipcodes z on e.zip = z.zip;

1. Customer names and order numbers for all orders with customer that live in ”Los Angeles”

select customers.cname, orders.ono from customers inner join orders inner join zipcodes where zipcodes.city='Los Angeles';

1. A list with name, quantity, price and total price for all products on the order with order number 1020.

select \* from parts inner join odetails where ono=1020;

1. The price of all orders combined.

select sum(price) from parts inner join orders where ono=1020;

1. Order numbers for orders which have not yet been delivered.

select ono from orders where shipped;

1. The order number, number of line items, the customer’s name, street, zip code and city

select odetails.ono, count(odetails.ono)

as numberOfLineItems, customers.cname, customers.street, customers.zip, zipcodes.city

from customers

inner join orders on orders.cno=customers.cno

inner join odetails on odetails.ono=orders.ono

inner join zipcodes on zipcodes.zip=customers.zip

group by odetails.ono;