Coordinator: Pär Douhan, pdo@du.se

Lab assignment # 1 - DDL, DML, constraints and transaction processing

How do you write the lab report? You can put your answers in this document and provide your code with comments where you think it's necessary. If you can't use this document I would like you to include the task text in your answer. These goes for all lab reports.

During this lab you will acquire knowledge required to create database objects in the form of tables and sequences. Furthermore, you will see that certain integrity rules mentioned in the tasks is maintained by constraints on the table level.

Do all labs here: https://livesql.oracle.com

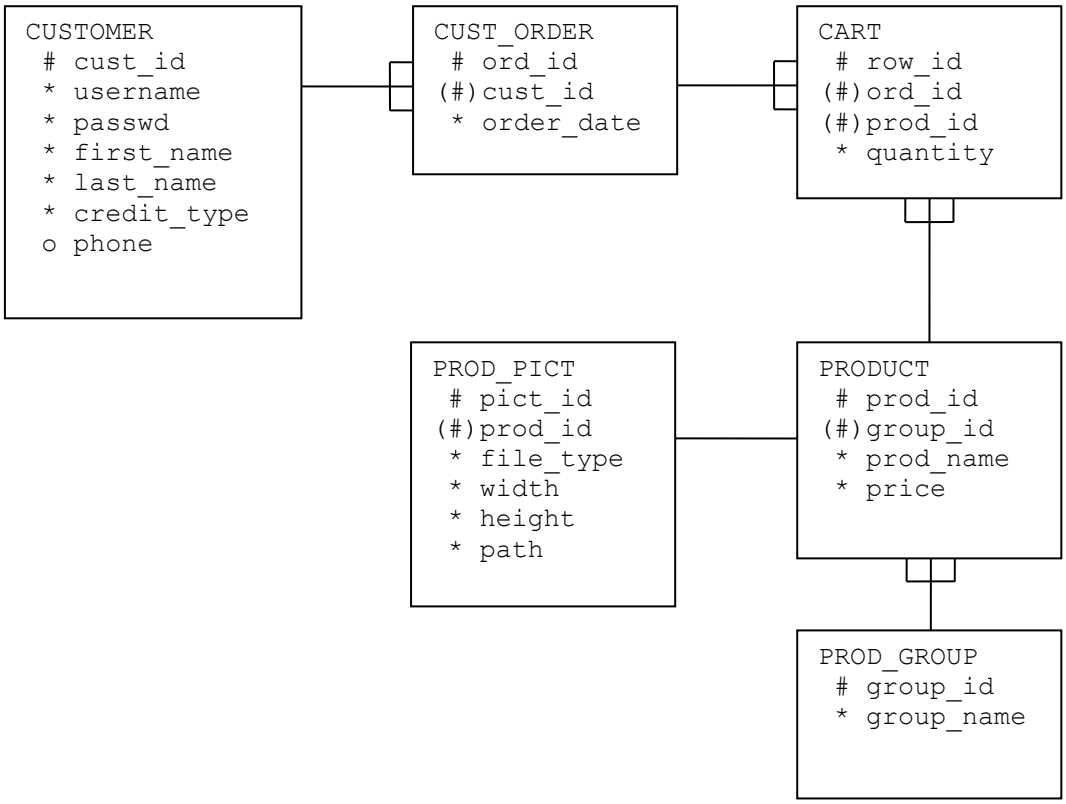
Task 1
Create a sequence object with the name **my_seq**. It should start with 1 and increase by 1. The sequence method NEXTVAL returns a numeric data type.

```
create sequence my_seq
start with 1
increment by 1;


-- The sequence is used during INSERT with NEXT method --
```

Coordinator: Pär Douhan, pdo@du.se

Task 2
Create a table structure according to the drawing below:



Explanation of notation

```
 #   = Primary key
(#)  = Foreign key
 *   = Mandatory (must contain a value => NOT NULL)
 o   = Optional (must not contain a value can be NULL)
```

customer.credit_type CHECK ('high','average','low')
prod_pict.file_type CHECK ('gif','jpg')
cust_order.ord_id (generated by the sequence my_seq)
cart.row_id (generated by the sequence my_seq)
cust_order.order_date (data type = DATE, DEFAULT SYSDATE)
customer.username (should be unique, constraint UNIQUE)
All Foreign Key columns should have the column constraint **NOT NULL**

Declare all constraints except NOT NULL at the table level! Suggestion for a constraint naming convention: **table_column_constraint**, you can use the following abbreviations

Coordinator: Pär Douhan, pdo@du.se

```
--  Table CUSTOMER --
create table CUSTOMER (
    cust_id number(9),
    username varchar2(20) not null,
    passwd varchar2(20) not null,
    first_name varchar2(30) not null,
    last_name varchar2(30) not null,
    credit_type varchar2(20) not null,
    phone varchar2(15)
);
-- Add named constraints for  table CUSTOMER
alter table CUSTOMER
add constraint customer_id_pk primary key (cust_id)
add constraint customer_username_uq unique(username)
add constraint customer_credit_type_ck check (credit_type in ('high','low','average'));
-------------------------------------------------------------------------
-- Table CUST-ORDER --
create table CUST_ORDER(
    ord_id number(9),
    cust_id number(9) not null,
    order_date date default sysdate not null
);
-- Add named constraints for  table CUST_ORDER
alter table CUST_ORDER
add constraint cust_order_ord_id_pk primary key (ord_id)
add constraint cust_order_cust_id_fk foreign key (cust_id) references customer(cust_id);
-------------------------------------------------------------------------
-- Table PROD-GROUP --
create table PROD_GROUP(
    group_id number(9),
    group_name varchar(255) not null
);


-- Add named table constraints
alter table PROD_GROUP
add constraint prod_group_group_id_pk primary key (group_id);
```

Coordinator: Pär Douhan, pdo@du.se

```
-------------------------------------------------------------------------
-- Table PRODUCT --
create table PRODUCT(
    prod_id number(9),
    group_id number(9) not null,
    prod_name varchar(255) not null,
    price number(9,2) not null
);

-- add named table constraints
alter table PRODUCT
add constraint product_prod_id_pk primary key (prod_id)
add constraint product_group_id_fk foreign key (group_id) references prod_group(group_id);
-------------------------------------------------------------------------
-- Table CART --
create table CART(
    row_id number(9),
    ord_id number(9) not null,
    prod_id number(9) not null,
    quantity number(9) not null
);
-- Add named table constraints
alter table CART
add constraint cart_row_id_pk primary key (row_id)
add constraint cart_ord_id_fk foreign key (ord_id) references cust_order(ord_id)
add constraint cart_prod_id_fk foreign key (prod_id) references product(prod_id);
-------------------------------------------------------------------------
-- Table PRODUCT_PICT
create table PROD_PICT(
    pict_id number(9),
    prod_id number(9) not null,
    file_type varchar(255) not null,
    width number(9) not null,
    height number(9) not null,
    path varchar(255) not null
);
```

Coordinator: Pär Douhan, pdo@du.se

```
-- add named table constraints
alter table PROD_PICT
add constraint prod_pict_pict_id_pk primary key (pict_id)
add constraint prod_pict_prod_id_fk foreign key (prod_id) references PRODUCT(prod_id)
add constraint prod_pict_file_type check (file_type in ('gif', 'jpg'));
-------------------------------------------------------------------------------
```

Task 3

Insert three rows in the **customer** table.

```
-- insert three rows in the customer table
insert into customer(cust_id, username, passwd, first_name, last_name, credit_type, phone)
values (1, 'frener', '98438er', 'Fred', 'Nerks', 'low', '0798312771');

insert into customer(cust_id, username, passwd, first_name, last_name, credit_type, phone)
values(2, 'jandoe', '988fkd-f', 'Jane', 'Doe', 'high', '0708316522');

insert into customer(cust_id, username, passwd, first_name, last_name, credit_type, phone)
values(3, 'joeblo', '6l-eg5fs', 'Joe', 'Bloggs', 'average', '');
```

Task 4

Insert two rows in the **prod_group** table.

```
--insert two rows in the product group table
insert into prod_group (group_id, group_name)
values (1, 'mobile');

insert into prod_group (group_id, group_name)
values (2, 'laptop');
```

Task 5

Insert two rows in the **product** table.

```
--insert two rows in the product table
insert into product (prod_id, group_id, prod_name, price)
values (1, 1, 'samsung s22', 8990);

insert into product (prod_id, group_id, prod_name, price)
values (2, 2, 'DELL XPS13', 13990);
```

Task 6

Coordinator: Pär Douhan, pdo@du.se

Perform a sale by creating **one row** in the **cust_order** table and **two rows** in the **cart** table. **Remember** to use the sequence to generate primary key in the tables.

NOTE that when you have created the cust_order you must check what value the sequence put in the ord_id column (i.e. the Primary Key value). Then take that number and use it in the insert on the cart table FK-column. **DO NOT USE** the sequence to generate a number to the foreign key ord_id in the cart table!

```
-- insert one  row in cust order
insert into cust_order (ord_id, cust_id, order_date)
values (my_seq.nextval, 1, sysdate);

-- Get the ord_id (PK) from the cust_order table, use as a FK in cart table
select * from cust_order

-- insert two rows in the cart table
insert into cart (row_id, ord_id, prod_id, quantity)
values (my_seq.nextval, 1, 1, 3);

insert into cart (row_id, ord_id, prod_id, quantity)
values (my_seq.nextval, 1, 2, 5);
```

Task 7

Increase the price on all articles by 12%.

```
-- increasing the price on all articles by 12%
update product
set price = price + price * 0.12;
```

Task 8

Update the phone number for an optional customer.

```
-- Update the phone number for an optional customer
update customer
set phone = '0700977131'
where first_name = 'Jane' and last_name = 'Doe';
```

Task 9

Delete all rows from the cust_order table, by using DML. **What happens and why!**

```
-- Delete all rows from the cust_order table
delete from cust_order;
```

An integrity constraint violation error is shown, and the rows are not deleted. An attempt to delete records from the parent table(cust_order) which has a record in the child table (cart) referenced by foreign keys will break data integrity constraints (in this case referential integrity). Because the cart table records exist in the cust_order table (through foreign keys), the user needs to first delete the records from the cart table to be able to delete the records in the cust_order table.