

**Name: Maha Vajeeshwaran Navaneethan**

## **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.

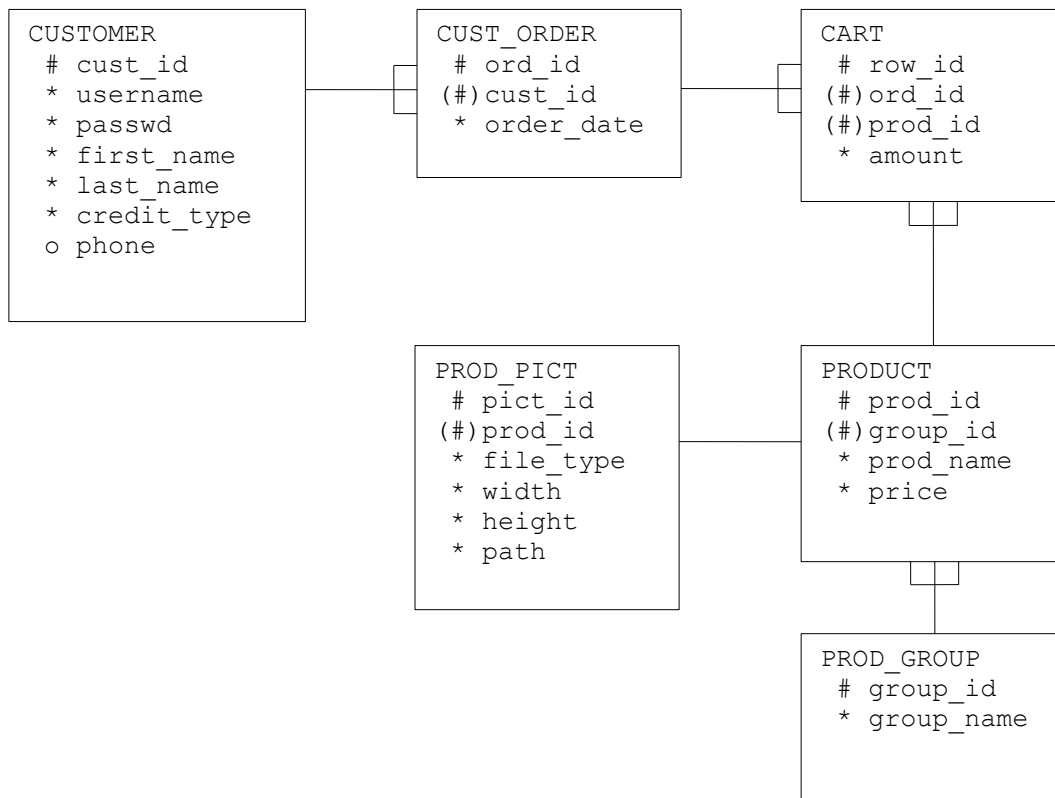
----- TASK 1

```
CREATE SEQUENCE my_seq  
START WITH 1  
INCREMENT BY 1;
```

```
---- Created a sequence object with the name my_seq  
---- The sequence method NEXTVAL returns a numeric data type this is  
used in below task
```

## 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, 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

if you like : **CK** = CHECK, **PK** = PRIMARY KEY, **FK** = FOREIGN KEY and finally **UQ** = UNIQUE, or whatever you like as long as you are consistently.

For the customer table above, a primary key constraint would be named:  
**customer\_cust\_id\_pk**

----- TASK 2

```
CREATE TABLE CUSTOMER(  
    cust_id NUMBER(7) ,  
    username VARCHAR2(20) NOT NULL,  
    passwd VARCHAR2(20) NOT NULL,  
    first_name VARCHAR2(40) NOT NULL,  
    last_name VARCHAR2(40) NOT NULL,  
    credit_type VARCHAR2(7) NOT NULL,  
    phone VARCHAR2(13));  
  
ALTER TABLE CUSTOMER  
ADD CONSTRAINT CUSTOMER_cust_id_pk PRIMARY KEY (cust_id)  
ADD CONSTRAINT CUSTOMER_credit_type_ck CHECK (credit_type IN ('high',  
'average', 'low'))  
ADD CONSTRAINT CUSTOMER_username_uq UNIQUE(username);  
  
CREATE TABLE CUST_ORDER(  
    ord_id NUMBER(9) ,  
    cust_id NUMBER(7) NOT NULL,  
    order_date DATE NOT NULL);  
  
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);  
  
CREATE TABLE PROD_GROUP(  
    group_id NUMBER(10) ,  
    group_name VARCHAR2(20));  
  
ALTER TABLE PROD_GROUP  
ADD CONSTRAINT PROD_GROUP_group_id_pk PRIMARY KEY(group_id);  
  
CREATE TABLE PRODUCT(  
    prod_id NUMBER(8) ,  
    group_id NUMBER(10) NOT NULL,  
    prod_name VARCHAR2(25) NOT NULL,  
    price NUMBER(9,2) NOT NULL);  
  
ALTER TABLE PRODUCT  
ADD CONSTRAINT PRODUCT_prod_id_pk PRIMARY KEY(prod_id)  
ADD CONSTRAINT PRODUCT_group_id_pk FOREIGN KEY(group_id) REFERENCES  
PROD_GROUP (group_id);
```



```
CREATE TABLE CART(  
  row_id NUMBER(9),  
  ord_id NUMBER(9) NOT NULL,  
  prod_id NUMBER(8) NOT NULL,  
  amount NUMBER(6) NOT NULL);  
  
ALTER TABLE CART  
ADD CONSTRAINT CART_row_id_pk PRIMARY KEY (row_id)  
ADD CONSTRAINT CART_ord_id_pk FOREIGN KEY (ord_id) REFERENCES  
CUST_ORDER (ord_id)  
ADD CONSTRAINT CART_prod_id_pk FOREIGN KEY (prod_id) REFERENCES PRODUCT  
(prod_id);  
  
CREATE TABLE PROD_PICT(  
  pict_id NUMBER(10),  
  prod_id NUMBER(8) NOT NULL,  
  file_type VARCHAR2(5) NOT NULL,  
  width NUMBER(10) NOT NULL,  
  height NUMBER(10) NOT NULL,  
  path VARCHAR2(80) NOT NULL);  
  
ALTER TABLE PROD_PICT  
ADD CONSTRAINT PROD_PICT_pict_id_pk PRIMARY KEY (pict_id)  
ADD CONSTRAINT PROD_PICT_prod_id_pk FOREIGN KEY (prod_id) REFERENCES  
PRODUCT (prod_id)  
ADD CONSTRAINT PROD_PICT_file_type_ck CHECK (file_type IN ('gif',  
'jpg'));
```

### Task 3

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

----- TASK 3

```
INSERT INTO CUSTOMER (cust_id, username, passwd, first_name,  
last_name, credit_type, phone)  
VALUES (1, 'Tom', '123456789', 'David', 'Tom', 'high',  
'+918754194593');
```

```
INSERT INTO CUSTOMER (cust_id, username, passwd, first_name,  
last_name, credit_type, phone)  
VALUES (2, 'WadeDavid', '789456123', 'David', 'Wade', 'low',  
'+934578965412');
```

```
INSERT INTO CUSTOMER (cust_id, username, passwd, first_name,  
last_name, credit_type, phone)  
VALUES (3, 'Steevesmith', '456789123', 'Steve', 'Smith', 'high',  
'+967894561231');
```

---- Inserted three rows in the customer table.

### Task 4

Insert two rows in the **prod\_group** table.

----- Task 4

```
INSERT INTO PROD_GROUP (group_id, group_name)  
VALUES (1, 'Group A');
```

```
INSERT INTO PROD_GROUP (group_id, group_name)  
VALUES (2, 'Group B');
```

----- Insert two rows in the prod\_group table

### Task 5

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

----- Task 5

```
INSERT INTO PRODUCT (prod_id, group_id, prod_name, price)  
VALUES (1, 1, 'Toy', 12.56);
```

```
INSERT INTO PRODUCT (prod_id, group_id, prod_name, price)  
VALUES (2, 2, 'Apple', 22.50);
```

----- Inserted two rows in the product table.



### Task 6

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!

---- Task 6

```
INSERT INTO CUST_ORDER(ord_id, cust_id, order_date)
VALUES (my_sql.NEXTVAL, 2, SYSDATE);
```

```
INSERT INTO CART(row_id, ord_id, prod_id, amount)
VALUES (my_sql.NEXTVAL, 1, 1, 379);
```

```
INSERT INTO CART(row_id, ord_id, prod_id, amount)
VALUES (my_sql.NEXTVAL, 1, 2, 687);
```

---- Performed a sale by creating one row in the cust\_order table and two rows in the cart table

### Task 7

Increase the price on all articles by 12%.

---- TASK 7

```
UPDATE PRODUCT
SET price = price + (price * 0.12);
```

----- Increase the price on all articles by 12%.

### Task 8

Update the phone number for an optional customer.

---- TASK 8

```
-- Updated the phone number for an optional customer for the cust_id = 1
UPDATE CUSTOMER
SET phone = '+4187541978'
WHERE cust_id = 1;
```

### Task 9

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

-----TASK 9

```
DELETE  
FROM CUST_ORDER;
```

-- By running the above code we get an error. Because of the parent  
child relationship we must first delete the Cart table

```
DELETE  
FROM CART;
```

```
DELETE  
FROM CUST_ORDER;
```

---- Now all rows from the CUST\_ORDER table will be deleted

**All the task code are checked while running in live SQL.**

**Put the lab report in Learn (Blackboard). Click on the link Assignments.**