**UNIVERSITY OF DAR ES SALAAM**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING, COICT**

**IS 263/ IS 264** **TEST 2** **Duration 2 Hours**

The Rocky Concrete Company makes a range of concrete products from laundry tubs through park benches to garden gnomes. Rocky's regular customers include hardware shops, local councils, nurseries, farmers and other small businesses. These customers are considered to be the company's "bread and butter" and Rocky likes to satisfy their orders as quickly as possible. To this end the company tries to keep an adequate level of stock for each product made. Whenever the stock in hand falls below some predetermined level then another batch is made.

```
Customers(cust_no, cust_name, street, town, post_code, cr_limit, curr_bal)
Products(prod_cod, description, prod_group, list_price, qty_on_hand,
remake_level, remake_qty)
Orders(order_no, order_date, cust_no)
Order_details(order_no, prod_cod, order_qty, order_price)
```

Some columns, such as `cust_name`, are self explanatory; however, others need some definition:

`cr_limit`: The maximum that a customer is allowed to owe Rocky Concrete; this may be exceeded at the manager's discretion.

`curr_bal`: The amount currently owed by the customer.

`prod_group`: A code that indicates whether a product is grouped as agricultural (A), council (C), or domestic (D).

`list_price`: The advertised price for a single unit of a particular product; the price charged to a customer might vary from this.

`remake_level`: The level to which the quantity on hand is compared; if `qty_on_hand` falls below this level then Rocky will usually make another batch to avoid stockout.

`remake_qty`: The quantity usually made in any new batch.

`order_price`: The unit price charged on this order for this product.

`order_date`: The date on which the order was taken. This is held as 6 digits in DDMMYY form, so 31st March 1998 is held as 310398. (Note that this date format reflects poor design!)

## Rocky Concrete database instance

Customers:

| cust_no | cust_name | street | town | post_code | cr_limit | curr_bal |
|---|---|---|---|---|---|---|
| 1066 | Nev's Nursery | White Hart Lane | Bundoora | 3083 | 500 | 450 |
| 13144 | Preston City | High Street | Preston | 3072 | 3000 | 1000 |
| 1776 | Di Hunter | Thornton Farm | Whittlesea | 3757 | 500 | 500 |
| 2001 | Glad's Gladdies | Child's Road | Mill Park | 3082 | 500 | 0 |
| 2002 | Mill Park Garden | Betula Ave | Mill Park | 3082 | 1000 | 300 |

Products:

| prod_ cod | description | prod_ group | list_ price | qty_on_ hand | remake_ level | remake_ qty |
|---|---|---|---|---|---|---|
| MOO | Medium Cattle Trough | A | 150 | 6 | 3 | 5 |
| LOO | Large Cattle Trough | A | 250 | 1 | 1 | 3 |
| STANK | Small Septic Tank | D | 300 | 10 | 5 | 15 |
| LTANK | Large Septic Tank | D | 450 | 1 | 2 | 2 |
| LTUB | Laundry Tub | D | 100 | 20 | 15 | 20 |
| GNOME | Garden Gnome | D | 10 | 100 | 150 | 200 |
| STAND | Bicycle Stand | C | 50 | 50 | 35 | 20 |
| GABBY | Football Player Statue | C | 500 | 10 | 15 | 40 |

(A = Agricultural, D = Domestic, C = Council)

Orders:

| order_no | order_date | cust_no |
|---|---|---|
| 0001 | 01/07/93 | 13144 |
| 0002 | 02/07/93 | 13144 |
| 0003 | 02/07/93 | 1066 |

Order_details:

| order_no | prod_cod | order_qty | order_price |
|---|---|---|---|
| 0001 | STAND | 10 | 45 |
| 0001 | GABBY | 2 | 480 |
| 0002 | STAND | 5 | 45 |
| 0003 | GNOME | 10 | 10 |

**Question One (20 marks)**

Consider the Rocky Concrete Database presented above and perform the following:

a) Create Database Rocky_*XXXX_XX_XXXXX* **(replace it wih your Reg Number)** **(4 marks)**

```
mysql> create database Rocky_2015_04_02536;
Query OK, 1 row affected (0.01 sec)
```

b) In the database you created in a) above, create the tables Customers, Products, Orders and Order_details as described above. Be certain to define appropriate data type, primary keys and referential integrity constraints. **(8 marks)**

```
mysql> create table Customers(cust_no int not null,cust_name varchar(20), street
 varchar(15), town varchar(10),post_code int, cr_limit int, curr_bal int, primar
y key(cust_no));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table Products(prod_cod varchar(10) not null, description varchar(
30),prod_group char(1),list_price int, qty_on_hand int, remake_level int, remake
_qty int, primary key (prod_cod));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table Orders(order_no char(4) not null,order_date char(8), cust_no
 int references customers(cust_no), primary key (order_no));
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> create table Order_details(order_no char(4) not null references orders(or
der_no),prod_cod varchar(10) references Products(prod_cod),order_qty int, order_
price int, primary key (order_no));
Query OK, 0 rows affected (0.00 sec)
```

c) For each table insert at least the first record of the respective table as in the Rocky Concrete Database Instance given above. **(8 marks)**

**Note:** You may need to insert more records for some tables due to referential integrity constraints.

```
mysql> insert into customers(cust_no,cust_name,street,town,post_code,cr_limit,cu
rr_bal) values (1066,"Nerv's Nursery",'white Hart Lane','Bundoora',3083,500,450)
;
Query OK, 1 row affected (0.00 sec)
```

```
mysql> select * from customers;
+---------+----------------+-----------------+----------+-----------+----------+
----------+
| cust_no | cust_name      | street          | town     | post_code | cr_limit |
 curr_bal |
+---------+----------------+-----------------+----------+-----------+----------+
----------+
|    1066 | Nerv's Nursery | white Hart Lane | Bundoora |      3083 |      500 |
     450 |
+---------+----------------+-----------------+----------+-----------+----------+
----------+
1 row in set (0.00 sec)

mysql> insert into products(prod_cod,description,prod_group,list_price,qty_on_ha
nd,remake_level,remake_qty) values ('MOO','Medium Cattle Trough','A',150,6,3,5);

Query OK, 1 row affected (0.01 sec)
```

```
mysql> select * from produts;
ERROR 1146: Table 'rocky_2015_04_02536.produts' doesn't exist
mysql> select * from products;
+----------+----------------------+------------+------------+-------------+-----
---------+------------+
| prod_cod | description          | prod_group | list_price | qty_on_hand | rema
ke_level | remake_qty |
+----------+----------------------+------------+------------+-------------+-----
---------+------------+
| MOO      | Medium Cattle Trough | A          |        150 |           6 |
       3 |          5 |
+----------+----------------------+------------+------------+-------------+-----
mysql> insert into orders(order_no,order_date,cust_no) values ('0001','01/07/93'
,13144);
Query OK, 1 row affected (0.00 sec)

mysql> select * from orders;
+----------+------------+----------+
| order_no | order_date | cust_no  |
+----------+------------+----------+
| 0001     | 01/07/93   |   13144  |
+----------+------------+----------+
1 row in set (0.00 sec)

mysql> insert into order_details(order_no,prod_cod,order_qty,order_price) values
 ('0001', 'STAND',10,45);
Query OK, 1 row affected (0.00 sec)

ysql> select * from order_details;
+----------+----------+-----------+-------------+
 order_no | prod_cod | order_qty | order_price |
+----------+----------+-----------+-------------+
 0001     | STAND    |        10 |          45 |
+----------+----------+-----------+-------------+
 row in set (0.00 sec)
```

**Question Two (80 marks)**

Consider that the Rocky Concrete Database presented above is already created using the name **Rocky** and with the database instance presented above. Perform the following activities:

a) Display the name, adddress (street, town, postcode), and credit limit of customer "Di Hunter". **(10 marks)**

```
mysql> select cust_name,street,town,postcode,cr_limit from customers where cus
name="Di Hunter";
+-----------+---------------+------------+----------+----------+
| cust_name | street        | town       | postcode | cr_limit |
+-----------+---------------+------------+----------+----------+
| Di Hunter | Thornton Farm | Whittlesea |     3757 |      500 |
+-----------+---------------+------------+----------+----------+
1 row in set (0.03 sec)
```

b) Write an SQL query to find the product with the highest list_price. Return the prod_cod and list_price of the product. **(10 marks)**

```
mysql> select max(list_price) from products;
+-----------------+
| max(list_price) |
+-----------------+
|             500 |
+-----------------+
1 row in set (0.00 sec)
```

```
mysql> select prod_cod,list_price from products where list_price=(select max(lit
s_price) from products);
```

c) Calculate the total number of customers in each town. Output should be a list of town and total-number-of-customers pairs, in alphabetical order of town. **(10 marks)**

```
ysql> select town,count(cust_name) astotalcustomers
    -> from customers
    -> group by town
    -> order by town;
+-------------+--------------------+
| town        | astotalcustomers   |
+-------------+--------------------+
| Bundoora    |                  1 |
| Mill Park   |                  2 |
| Preston     |                  1 |
| Whittlesea  |                  1 |
+-------------+--------------------+
```

d) The account officer of Rocky Concrete needs to regularly find customers who reach their credit limit and issue a notice of payment. A customer reaches his/her credit limit if the difference between his/her credit limit and current balance is no more than $50. Which customers have reached their respective credit limit? List the name and address of these customers. **(10 marks)**

```
mysql> select cust_name as name,street,town,postcode from customers where (cr_limit-curr_balance)<=50;
+--------------+--------------+------------+----------+
| name         | street       | town       | postcode |
+--------------+--------------+------------+----------+
| Nevs Nursery | White Hart   | Bundoora   |     3083 |
| Di Hunter    | Thornton Farm | Whittlesea |     3757 |
+--------------+--------------+------------+----------+
2 rows in set (0.00 sec)
```

e) For each group of products, calculate the average price for the group. Output the product groups and their average product price. **(10 marks)**

```
mysql> select prod_group, avg(list_price) as Average_price_for_group from Produc
ts group by prod_group;
+------------+--------------------------+
| prod_group | Average_price_for_group  |
+------------+--------------------------+
| A          |                200.0000  |
| C          |                275.0000  |
| D          |                215.0000  |
+------------+--------------------------+
3 rows in set (0.03 sec)
```

f) Produce a list of product groups where each group has a total product value of at least $2000. The total product value should also be output for each of these groups. **(10 marks)**

```
mysql> select prod_group,sum(list_price*qty_on_hand) as totalvalue
    -> from products
    -> group by prod_group
    -> having totalvalue >=2000;
+------------+------------+
| prod_group | totalvalue |
+------------+------------+
| C          |       7500 |
| D          |       6450 |
+------------+------------+
2 rows in set (0.02 sec)
```

```
mysql> select prod_group,SUM(list_price*qty_on_hand) as Total_product_value from products group by prod_group having sum(list_price*qty_on_hand)>=2000;
+------------+---------------------+
| prod_group | Total_product_value |
+------------+---------------------+
| C          |                7500 |
| D          |                6450 |
+------------+---------------------+
2 rows in set (0.00 sec)

mysql>
```

g) A customer complained that his/her order of Garden Gnomes was not delivered. Write an SQL query to find the details of all orders for Garden Gnomes, including the order_no, order_date, cust_no, and prod_cod, quantity and price of products of the orders. **(10 marks)**

h) Find the orders whose total value is at least $1000. Output the order_no and order_date of these orders with their cust_no and cust_name. **(10 marks)**

```
mysql> select orders.order_no,order_date,customers.cust_no,cust_name from orders,order_details,customers where orders.cust_no=customers.cust_no and orders.order_no=order_details
.order_no group by order_no having sum(order_qty*order_price)>=1000;
+----------+------------+---------+-------------+
| order_no | order_date | cust_no | cust_name   |
+----------+------------+---------+-------------+
|        1 | 1993-07-01 |   13144 | Preston City |
+----------+------------+---------+-------------+
1 row in set (0.00 sec)

mysql>
```

**Note:** Recommended to Write a Join SQL query as you can run it in this lab **OR** you may write SQL queries using operators IN and EXISTS respectively, however you will not be able to execute this in this lab but I can accept that as long as the SQL Query you wrote is correct.