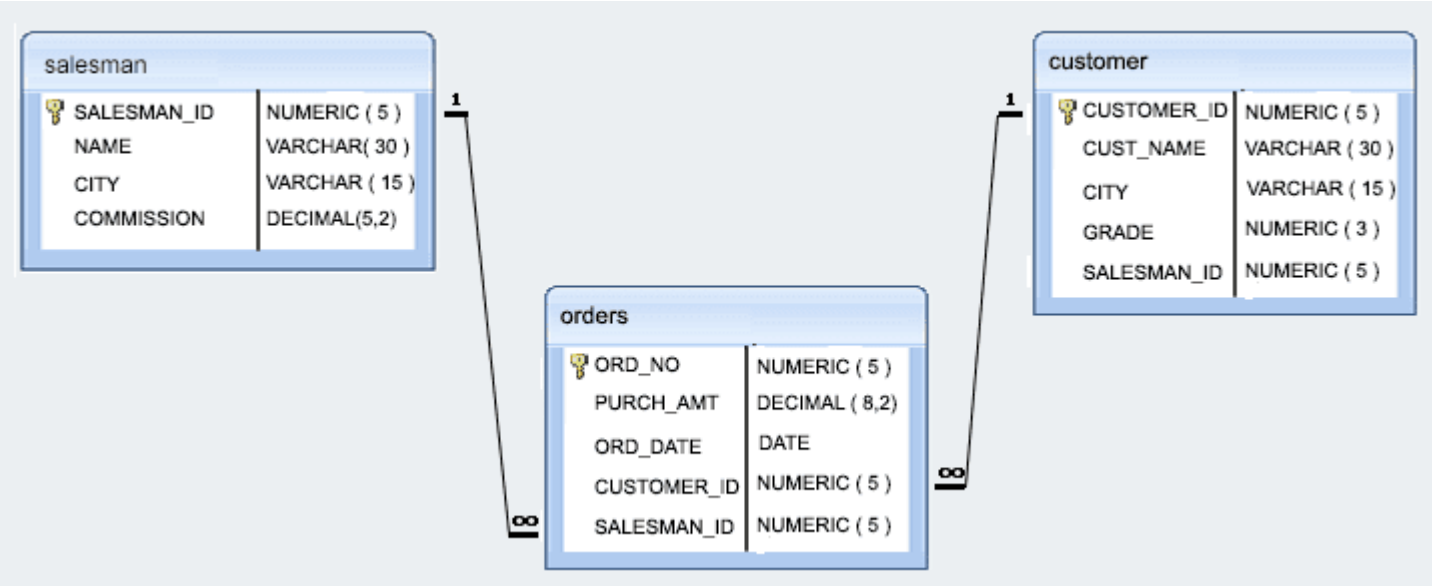
# DBMS LAB. ASSIGNMENT

| Roll Number: | 22051437 | Branch/Section: | CSE-45 |
|---|---|---|---|
| Name in Capital: | Mridul Moona | | |

Consider the structure of **inventory database** as follows:




**Compose queries based on the provided inventory database and the reference given below:**
**Reference: https://www.w3resource.com/sql-exercises/sql-joins-exercises.php**
**Visit the link to access SQL exercises that focus on JOINS, comprising a total of 29 questions with corresponding solutions. It is recommended to practice all the exercises, but for the purpose of lab record submission, choose any 10 questions from the set of 29.**

## Instruction

For each question and its solution, given tables data will be same, you change either the question for the query that must cover atleast two tables data and write SQL query for the same or write a SQL query that is different from the given solution/answer for the given question. One sample assignment solution is given as follows: Suppose you are solving assignment no. 6.3 wrt to the question no-1, then write as follows.

**Assignment - 6.3**

| Q. No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 1 | Given => | From the following tables write a SQL query to find the salesperson and customer who reside in the same city. Return Salesman, cust_name and city. | SELECT salesman.name AS "Salesman", customer.cust_name, customer.city<br>FROM salesman,customer<br>WHERE salesman.city=customer.city; |
| | **I have Changed =>** | **Changed as follows:**<br>Write a SQL query to find the salesperson and customer who reside in the same city and **they have the same name**. Return Salesman, cust_name and city. | **My Answer**<br>SELECT salesman.name AS "Salesman", customer.cust_name, customer.city<br>FROM salesman,customer<br>WHERE salesman.city=customer.city and salesman.name=customer.cust_name; |

**The Tables are as follows:**

**Input**

**Table-1: customer**

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

**Table-2: salesman**

| salesman_id | name | city | commission |
|---|---|---|---|
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5007 | Paul Adam | Rome | 0.13 |
| 5003 | Lauson Hen | San Jose | 0.12 |

**Output**

**With reference to the above sample solution, solve assignment-6 and Create the above tables and practise the queries, you may cross check your query online through the above link after each question by clicking** Go to the editor. Write everything in A4 size paper including the table data and following the sample format for each question. This type of questions you may find in sessional exam. Submit this file (typed, not hand writen) in google classroom after solving each question similar to sample given above.

## Assignmnet-6

### Assignment - 6.1

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 5 | Given => | From the following tables write a SQL query to locate those salespeople who do not live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission. | `SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.city,b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12 AND a.city<>b.city;` |
| | **I have Changed =>** | From the following tables write a SQL query to locate those salespeople who live in the same city where their customers live and have received a commission of more than 12% from the company. Return Customer Name, customer city, Salesman, salesman city, commission. | `SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.city,b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12 AND a.city=b.city;` |

**The Tables are as follows:**

**INPUT**

Customer

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

salesman

| salesman_id | name | city | commission |
|---|---|---|---|
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5007 | Paul Adam | Rome | 0.13 |
| 5003 | Lauson Hen | San Jose | 0.12 |

**OUTPUT**

| Customer Name | city | Salesman | commission |
|---|---|---|---|
| Nick Rimando | New York | James Hoog | 0.15 |
| Brad Davis | New York | James Hoog | 0.15 |
| Graham Zusi | California | Nail Knite | 0.13 |
| Julian Green | London | Nail Knite | 0.13 |
| Fabian Johnson | Paris | Mc Lyon | 0.14 |
| Geoff Cameron | Berlin | Lauson Hen | 0.12 |
| Jozy Altidor | Moscow | Paul Adam | 0.13 |
| Brad Guzan | London | Pit Alex | 0.11 |

**Assignment - 6.2**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 9 | Given => | From the following tables write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, Salesman, salesmancity. The result should be ordered by ascending customer_id. | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a LEFT OUTER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE a.grade<300 ORDER BY a.customer_id ; |
| | **I have Changed =>** | From the following tables write a SQL query to find those customers with a grade less than 300. Return cust_name, customer city, grade, | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a LEFT OUTER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE a.grade<300 ORDER BY a.customer_id |

| | | | |
|---|---|---|---|
| | | Salesman, salesmancity. The result should be ordered by descending customer_id. | DESC; |

**The Tables are as follows:**

**INPUT**

Customer

```
 customer_id |   cust_name    |    city    | grade | salesman_id
-------------+----------------+------------+-------+-------------
        3002 | Nick Rimando   | New York   |   100 |        5001
        3007 | Brad Davis     | New York   |   200 |        5001
        3005 | Graham Zusi    | California |   200 |        5002
        3008 | Julian Green   | London     |   300 |        5002
        3004 | Fabian Johnson | Paris      |   300 |        5006
        3009 | Geoff Cameron  | Berlin     |   100 |        5003
        3003 | Jozy Altidor   | Moscow     |   200 |        5007
        3001 | Brad Guzan     | London     |       |        5005
```

salesman

```
 salesman_id |    name    |   city   | commission
-------------+------------+----------+------------
        5001 | James Hoog | New York |       0.15
        5002 | Nail Knite | Paris    |       0.13
        5005 | Pit Alex   | London   |       0.11
        5006 | Mc Lyon    | Paris    |       0.14
        5007 | Paul Adam  | Rome     |       0.13
        5003 | Lauson Hen | San Jose |       0.12
```

**OUTPUT**

| cust_name | city | grade | Salesman |
|---|---|---|---|
| Geoff Cameron | San Jose | 100 | Lauson Hen |
| Brad Davis | New York | 200 | James Hoog |
| Graham Zusi | Paris | 200 | Nail Knite |
| Jozy Altidor | Rome | 200 | Paul Adam |
| Nick Rimando | New York | 100 | James Hoog |

**Assignment - 6.3**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 23 | Given => | From the following tables write a SQL query to calculate the average price of items of each company. Return average value and company name. | SELECT AVG(pro_price), company_mast.com_name FROM item_mast INNER JOIN company_mast ON item_mast.pro_com= company_mast.com_id GROUP BY company_mast.com_name; |
| | **I have Changed =>** | From the following tables write a SQL query to calculate the sum price of items of each company. Return sum and company name. | SELECT SUM(pro_price), company_mast.com_name FROM item_mast INNER JOIN company_mast ON item_mast.pro_com= company_mast.com_id GROUP BY company_mast.com_name; |

**The Tables are as follows:**

**INPUT**

company_mast

```
COM_ID COM_NAME
------ -------------
    11 Samsung
    12 iBall
    13 Epsion
    14 Zebronics
    15 Asus
    16 Frontech
```

item_mast

```
 PRO_ID PRO_NAME                   PRO_PRICE    PRO_COM
------- ------------------------ ------------ ----------
    101 Mother Board                 3200.00         15
    102 Key Board                     450.00         16
    103 ZIP drive                     250.00         14
    104 Speaker                       550.00         16
    105 Monitor                      5000.00         11
    106 DVD drive                     900.00         12
    107 CD drive                      800.00         12
    108 Printer                      2600.00         13
    109 Refill cartridge              350.00         13
    110 Mouse                         250.00         12
```

**OUTPUT**

| sum | com_name |
|---|---|

| | |
|---|---|
| 5000.00 | Samsung |
| 250.00 | Zebronics |
| 1000.00 | Frontech |
| 2950.00 | Epsion |
| 1950.00 | iBall |
| 3200.00 | Asus |

**Assignment - 6.4**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 10 | Given => | Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the order date to determine whether any of the existing customers have placed an order or not. | SELECT a.cust_name,a.city, b.ord_no, b.ord_date,b.purch_amt AS "Order Amount" FROM customer a LEFT OUTER JOIN orders b ON a.customer_id=b.customer_id order by b.ord_date; |
| | **I have Changed =>** | Write a SQL statement to make a report with customer name, city, order number, order date, and order amount in ascending order according to the purch_amt to determine whether any of the existing customers have placed an order or not. | SELECT a.cust_name,a.city, b.ord_no, b.ord_date,b.purch_amt AS "Order Amount" FROM customer a LEFT OUTER JOIN orders b ON a.customer_id=b.customer_id order by b.purch_amt; |

**The Tables are as follows:**

**INPUT**

Orders:

| ord_no | purch_amt | ord_date | customer_id | salesman_id |
|---|---|---|---|---|
| 70001 | 150.5 | 2012-10-05 | 3005 | 5002 |
| 70009 | 270.65 | 2012-09-10 | 3001 | 5005 |

| | | | | |
|---|---|---|---|---|
| 70002 | 65.26 | 2012-10-05 | 3002 | 5001 |
| 70004 | 110.5 | 2012-08-17 | 3009 | 5003 |
| 70007 | 948.5 | 2012-09-10 | 3005 | 5002 |
| 70005 | 2400.6 | 2012-07-27 | 3007 | 5001 |
| 70008 | 5760 | 2012-09-10 | 3002 | 5001 |
| 70010 | 1983.43 | 2012-10-10 | 3004 | 5006 |
| 70003 | 2480.4 | 2012-10-10 | 3009 | 5003 |
| 70012 | 250.45 | 2012-06-27 | 3008 | 5002 |
| 70011 | 75.29 | 2012-08-17 | 3003 | 5007 |
| 70013 | 3045.6 | 2012-04-25 | 3002 | 5001 |

Customer:

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

## OUTPUT

| cust_name | city | ord_no | ord_date | Order Amount |
|---|---|---|---|---|
| Nick Rimando | New York | 70002 | 2012-10-05 | 65.26 |
| Jozy Altidor | Moscow | 70011 | 2012-08-17 | 75.29 |
| Geoff Cameron | Berlin | 70004 | 2012-08-17 | 110.50 |
| Graham Zusi | California | 70001 | 2012-10-05 | 150.50 |
| Julian Green | London | 70012 | 2012-06-27 | 250.45 |
| Brad Guzan | London | 70009 | 2012-09-10 | 270.65 |
| Graham Zusi | California | 70007 | 2012-09-10 | 948.50 |
| Fabian Johnson | Paris | 70010 | 2012-10-10 | 1983.43 |
| Brad Davis | New York | 70005 | 2012-07-27 | 2400.60 |
| Geoff Cameron | Berlin | 70003 | 2012-10-10 | 2480.40 |
| Nick Rimando | New York | 70013 | 2012-04-25 | 3045.60 |
| Nick Rimando | New York | 70008 | 2012-09-10 | 5760.00 |

## Assignment - 6.5

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 4 | Given => | From the following tables write a SQL query to find salespeople who received commissions of more than 12 percent from the company. Return Customer Name, customer city, Salesman, commission. | SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission>.12; |
| | **I have Changed =>** | From the following tables write a SQL query to find salespeople who received commissions of less than 12 percent from the company. Return Customer Name, customer city, Salesman, commission. | SELECT a.cust_name AS "Customer Name", a.city, b.name AS "Salesman", b.commission FROM customer a INNER JOIN salesman b ON a.salesman_id=b.salesman_id WHERE b.commission<.12; |

**The Tables are as follows:**

**INPUT**

customer

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

salesman

| salesman_id | name | city | commission |
|---|---|---|---|
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5007 | Paul Adam | Rome | 0.13 |
| 5003 | Lauson Hen | San Jose | 0.12 |

**OUTPUT**

| Customer Name | city | Salesman | commission |
|---|---|---|---|
| Brad Guzan | London | Pit Alex | 0.11 |

## Assignment - 6.6

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 6 | Given => | From the following tables write a SQL query to find the details of an order. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission. | Select a.order_no,a.ord_date,a.purch_amt,b.cust_name AS “CUSTOMER NAME”,b.grade,c.name as “SALESMAN” ,c.commission From orders a inner join customer b on a.customer_id=b.customer_id inner join salesman c ON a.salesman_id=c.salesman_id; |
| | **I have Changed =>** | From the following tables write a SQL query to find the details of an order where purchasing amount exceeds 700. Return ord_no, ord_date, purch_amt, Customer Name, grade, Salesman, commission. | SELECT a.ord_no,a.ord_date,a.purch_amt, b.cust_name AS "Customer Name", b.grade, c.name AS "Salesman", c.commission FROM orders a INNER JOIN customer b ON a.customer_id=b.customer_id INNER JOIN salesman c ON a.salesman_id=c.salesman_id where a.purch_amt>700; |

**The Tables are as follows:**

**INPUT**

orders

| ord_no | purch_amt | ord_date | customer_id | salesman_id |
|---|---|---|---|---|
| 70001 | 150.5 | 2012-10-05 | 3005 | 5002 |
| 70009 | 270.65 | 2012-09-10 | 3001 | 5005 |
| 70002 | 65.26 | 2012-10-05 | 3002 | 5001 |
| 70004 | 110.5 | 2012-08-17 | 3009 | 5003 |
| 70007 | 948.5 | 2012-09-10 | 3005 | 5002 |
| 70005 | 2400.6 | 2012-07-27 | 3007 | 5001 |
| 70008 | 5760 | 2012-09-10 | 3002 | 5001 |
| 70010 | 1983.43 | 2012-10-10 | 3004 | 5006 |
| 70003 | 2480.4 | 2012-10-10 | 3009 | 5003 |
| 70012 | 250.45 | 2012-06-27 | 3008 | 5002 |
| 70011 | 75.29 | 2012-08-17 | 3003 | 5007 |
| 70013 | 3045.6 | 2012-04-25 | 3002 | 5001 |

Customer:

| customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |

```
3003 | Jozy Altidor   | Moscow    |   200 |        5007
3001 | Brad Guzan     | London    |       |        5005
```

Salesman:

```
salesman_id |    name     |   city    | commission
------------+-------------+-----------+------------
       5001 | James Hoog  | New York  |       0.15
       5002 | Nail Knite  | Paris     |       0.13
       5005 | Pit Alex    | London    |       0.11
       5006 | Mc Lyon     | Paris     |       0.14
       5007 | Paul Adam   | Rome      |       0.13
       5003 | Lauson Hen  | San Jose  |       0.12
```

**OUPUT**

| ord_no | ord_date | purch_amt | Customer Name | grade | Salesman | commission |
|---|---|---|---|---|---|---|
| 70005 | 2012-07-27 | 2400.60 | Brad Davis | 200 | James Hoog | 0.15 |
| 70008 | 2012-09-10 | 5760.00 | Nick Rimando | 100 | James Hoog | 0.15 |
| 70010 | 2012-10-10 | 1983.43 | Fabian Johnson | 300 | Mc Lyon | 0.14 |
| 70003 | 2012-10-10 | 2480.40 | Geoff Cameron | 100 | Lauson Hen | 0.12 |
| 70013 | 2012-04-25 | 3045.60 | Nick Rimando | 100 | James Hoog | 0.15 |
| 70007 | 2012-09-10 | 948.50 | Graham Zusi | 200 | Nail Knite | 0.13 |

**Assignment - 6.7**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 12 | Given => | Write a SQL statement to generate a list in ascending order of salespersons who work either for one or more customers or have not yet joined any of the customers. | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id ORDER BY b.salesman_id; |
| | **I have Changed =>** | Write a SQL statement to generate a list in ascending order of salespersons commissions who work either for one or more customers or have not yet joined any of the customers. | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", b.city FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id ORDER BY b.commission; |

**The Tables are as follows:**

## INPUT

Customer:

| Customer_id | cust_name | city | grade | salesman_id |
|---|---|---|---|---|
| 3002 | Nick Rimando | New York | 100 | 5001 |
| 3007 | Brad Davis | New York | 200 | 5001 |
| 3005 | Graham Zusi | California | 200 | 5002 |
| 3008 | Julian Green | London | 300 | 5002 |
| 3004 | Fabian Johnson | Paris | 300 | 5006 |
| 3009 | Geoff Cameron | Berlin | 100 | 5003 |
| 3003 | Jozy Altidor | Moscow | 200 | 5007 |
| 3001 | Brad Guzan | London | | 5005 |

Salesman:

| salesman_id | name | city | commission |
|---|---|---|---|
| 5001 | James Hoog | New York | 0.15 |
| 5002 | Nail Knite | Paris | 0.13 |
| 5005 | Pit Alex | London | 0.11 |
| 5006 | Mc Lyon | Paris | 0.14 |
| 5007 | Paul Adam | Rome | 0.13 |
| 5003 | Lauson Hen | San Jose | 0.12 |

## OUTPUT

| cust_name | city | grade | Salesman |
|---|---|---|---|
| Brad Guzan | London | | Pit Alex |
| Geoff Cameron | San Jose | 100 | Lauson Hen |
| Graham Zusi | Paris | 200 | Nail Knite |
| Julian Green | Paris | 300 | Nail Knite |
| Jozy Altidor | Rome | 200 | Paul Adam |
| Fabian Johnson | Paris | 300 | Mc Lyon |
| Nick Rimando | New York | 100 | James Hoog |
| Brad Davis | New York | 200 | James Hoog |

## Assignment - 6.8

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 25 | Given => | From the following tables write a SQL query to find the most expensive product of each company. Return pro_name, | SELECT A.pro_name, A.pro_price, F.com_name FROM item_mast A INNER JOIN company_mast F ON A.pro_com = F.com_id AND A.pro_price = ( SELECT MAX(A.pro_price) FROM item_mast A |

| | | | |
|---|---|---|---|
| | | pro_price and com_name. | WHERE A.pro_com = F.com_id ); |
| | **I have Changed =>** | From the following tables write a SQL query to find the most cheap product of each company. Return pro_name, pro_price and com_name. | SELECT A.pro_name, A.pro_price, F.com_name FROM item_mast A INNER JOIN company_mast F ON A.pro_com = F.com_id AND A.pro_price = ( SELECT MIN(A.pro_price) FROM item_mast A WHERE A.pro_com = F.com_id ); |

**The Tables are as follows:**

**INPUT**

company_mast

```
COM_ID COM_NAME
------ ------------
    11 Samsung
    12 iBall
    13 Epsion
    14 Zebronics
    15 Asus
    16 Frontech
```

item_mast

```
 PRO_ID PRO_NAME                   PRO_PRICE    PRO_COM
------- ------------------------- ------------ ----------
    101 Mother Board                  3200.00         15
    102 Key Board                      450.00         16
    103 ZIP drive                      250.00         14
    104 Speaker                        550.00         16
    105 Monitor                       5000.00         11
```

**OUTPUT**

| pro_name | pro_price | com_name |
|---|---|---|
| Mother Board | 3200.00 | Asus |
| Key Board | 450.00 | Frontech |
| ZIP drive | 250.00 | Zebronics |
| Monitor | 5000.00 | Samsung |

| | | |
|---|---|---|
| Refill cartridge | 350.00 | Epsion |
| Mouse | 250.00 | iBall |

**Assignment - 6.9**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 27 | Given => | From the following tables write a SQL query to display the first and last names of each employee, as well as the department name and sanction amount. | SELECT emp_details.emp_fname AS "First Name", emp_lname AS "Last Name", emp_department.dpt_name AS "Department", dpt_allotment AS "Amount Allotted" FROM emp_details INNER JOIN emp_department ON emp_details.emp_dept = emp_department.dpt_code; |
| | **I have Changed =>** | From the following tables write a SQL query to display the first and last names of each employee, as well as the department name and sanction amount.(Using RIGHT JOIN) | SELECT emp_details.emp_fname AS "First Name", emp_lname AS "Last Name", emp_department.dpt_name AS "Department", dpt_allotment AS "Amount Allotted" FROM emp_details RIGHT JOIN emp_department ON emp_details.emp_dept = emp_department.dpt_code; |

**The Tables are as follows:**

**INPUT**

emp_department

| DPT_CODE | DPT_NAME | DPT_ALLOTMENT |
|---|---|---|
| 57 | IT | 65000 |
| 63 | Finance | 15000 |
| 47 | HR | 240000 |
| 27 | RD | 55000 |
| 89 | QC | 75000 |

emp_details

| EMP_IDNO | EMP_FNAME | EMP_LNAME | EMP_DEPT |
|---|---|---|---|
| 127323 | Michale | Robbin | 57 |
| 526689 | Carlos | Snares | 63 |
| 843795 | Enric | Dosio | 57 |
| 328717 | Jhon | Snares | 63 |
| 444527 | Joseph | Dosni | 47 |
| 659831 | Zanifer | Emily | 47 |
| 847674 | Kuleswar | Sitaraman | 57 |
| 748681 | Henrey | Gabriel | 47 |

```
555935 Alex            Manuel                57
539569 George          Mardy                 27
733843 Mario           Saule                 63
631548 Alan            Snappy                27
839139 Maria           Foster                57
```

## OUTPUT

| First Name | Last Name | Department | Amount Allotted |
|---|---|---|---|
| Alan | Snappy | RD | 55000 |
| Maria | Foster | IT | 65000 |
| Michale | Robbin | IT | 65000 |
| Carlos | Snares | Finance | 15000 |
| Enric | Dosio | IT | 65000 |
| Jhon | Snares | Finance | 15000 |
| Joseph | Dosni | HR | 240000 |
| Zanifer | Emily | HR | 240000 |
| Kuleswar | Sitaraman | IT | 65000 |
| Henrey | Gabriel | HR | 240000 |
| Alex | Manuel | IT | 65000 |

| | | | |
|---|---|---|---|
| George | Mardy | RD | 55000 |
| Mario | Saule | Finance | 15000 |

**Assignment - 6.10**

| Q.No. Wrt above link | | Question for the query | Answer |
|---|---|---|---|
| 14 | Given => | Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and must have a grade, or he may not have placed any order to the associated supplier. | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id LEFT OUTER JOIN orders c ON c.customer_id=a.customer_id WHERE c.purch_amt>=2000 AND a.grade IS NOT NULL; |
| | **I have Changed =>** | Write a SQL statement to make a list for the salesmen who either work for one or more customers or yet to join any of the customer. The customer may have placed, either one or more orders on or above order amount 2000 and should generate a commission of more than 11% for the supplier must have a grade, or he may not have placed any order to the associated supplier. | SELECT a.cust_name,a.city,a.grade, b.name AS "Salesman", c.ord_no, c.ord_date, c.purch_amt FROM customer a RIGHT OUTER JOIN salesman b ON b.salesman_id=a.salesman_id LEFT OUTER JOIN orders c ON c.customer_id=a.customer_id WHERE c.purch_amt>=2000 AND a.grade IS NOT NULL AND b.commission>.11; |

**The Tables are as follows:**

## INPUT

Customer:

```
customer_id | cust_name    |    city    | grade | salesman_id
------------+--------------+------------+-------+------------
        3002 | Nick Rimando | New York   |   100 |        5001
        3007 | Brad Davis   | New York   |   200 |        5001
        3005 | Graham Zusi  | California |   200 |        5002
        3008 | Julian Green | London     |   300 |        5002
        3004 | Fabian Johnson | Paris    |   300 |        5006
```

```
3009 | Geoff Cameron  | Berlin     |   100 |        5003
3003 | Jozy Altidor   | Moscow     |   200 |        5007
3001 | Brad Guzan     | London     |       |        5005
```

Salesman:

```
salesman_id |    name    |   city   | commission
------------+------------+----------+------------
        5001 | James Hoog | New York |       0.15
        5002 | Nail Knite | Paris    |       0.13
        5005 | Pit Alex   | London   |       0.11
        5006 | Mc Lyon    | Paris    |       0.14
        5007 | Paul Adam  | Rome     |       0.13
        5003 | Lauson Hen | San Jose |       0.12
```

Orders:

| ord_no | purch_amt | ord_date | customer_id | salesman_id |
|---|---|---|---|---|
| 70001 | 150.5 | 2012-10-05 | 3005 | 5002 |
| 70009 | 270.65 | 2012-09-10 | 3001 | 5005 |
| 70002 | 65.26 | 2012-10-05 | 3002 | 5001 |
| 70004 | 110.5 | 2012-08-17 | 3009 | 5003 |
| 70007 | 948.5 | 2012-09-10 | 3005 | 5002 |
| 70005 | 2400.6 | 2012-07-27 | 3007 | 5001 |
| 70008 | 5760 | 2012-09-10 | 3002 | 5001 |
| 70010 | 1983.43 | 2012-10-10 | 3004 | 5006 |
| 70003 | 2480.4 | 2012-10-10 | 3009 | 5003 |
| 70012 | 250.45 | 2012-06-27 | 3008 | 5002 |
| 70011 | 75.29 | 2012-08-17 | 3003 | 5007 |
| 70013 | 3045.6 | 2012-04-25 | 3002 | 5001 |

## OUTPUT

| cust_name | city | grade | Salesman | ord_no | ord_date | purch_amt |
|---|---|---|---|---|---|---|
| Brad Davis | New York | 200 | James Hoog | 70005 | 2012-07-27 | 2400.60 |
| Nick Rimando | New York | 100 | James Hoog | 70008 | 2012-09-10 | 5760.00 |
| Geoff Cameron | Berlin | 100 | Lauson Hen | 70003 | 2012-10-10 | 2480.40 |
| Nick Rimando | New York | 100 | James Hoog | 70013 | 2012-04-25 | 3045.60 |