# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 1

### DBMS Lab (KCA – 252)

**1.** Create following tables.

i) **Table name : Client_Master**

| Colum Name | DataType | Size |
|---|---|---|
| Client_No | Varchar | 6 |
| Name | Varchar | 20 |
| Address1 | Varchar | 30 |
| Address2 | Varchar | 30 |
| City | Varchar | 15 |
| State | Varchar | 15 |
| Pincode | Number | 6 |
| Bal_Due | Float | 10.2 |

ii) **Table Name: Product_Master**

| Colum Name | DataType | Size |
|---|---|---|
| Product_NO | Varchar | 6 |
| Description | Varchar | 20 |
| Profit_percentage | Number | 6 |
| Unit_Measure | Varchar | 10 |
| Qty_On_Hand | Number | 6 |
| Reorder_Lvl | Number | 6 |
| Sell_Price | Float | 7.2 |
| Cost_Price | Float | 7.2 |

2. Add a new column DOB to table Client_Master.
3. Change the data type of Client_No to number.
4. Drop the newly added column DOB from Client_Master.
5. Rename the column Sell_Price in Product_Master table to SellPrice.
6. Rename the table Product_Master to ProductMaster.
7. Delete both the tables.

# LAB ASSIGNMENT-1

Create following tables

CLIENT_MASTER

```
Create table CLIENT_MASTER (Client_No varchar2 (6),
Name varchar2(20), Address 1 varchar2 (30),
Address 2 varchar 2(30), City varchar 2(30),
State varchar2(15), Pincode number (6),
Bal_due number (10,2))
```

PRODUCT_MASTER

```
Create table PRODUCT_MASTER ( Product_No varchar
Description varchar2(20), Profit_percentage numbe
(6), Reorder_lvl number (6), Unit_ measurement
varchar2(10), Qty_on_hand number (6), sell_
price number (7,2), cost_price number (7,2))
```

```
ALTER TABLE CLIENT_MASTER ADD DOB NUMBER

ALTER TABLE CLIENT_MASTER MODIFY Client_No
NUMBER (6)

ALTER TABLE CLIENT_MASTER DROP COLUMN DOB

ALTER TABLE PRODUCT_MASTER RENAME column Sell_price
to Sellprice

ALTER TABLE PRODUCT_MASTER RENAME TO ProductMa

DROP TABLE CLIENT_MASTER
DROP TABLE ProductMaster
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 2

### DBMS Lab (KCA – 252)

1. Insert the following data into CLIENT_MASTER table:

| CLIENTNO | NAME | CITY | PINCODE | STATE | BALDUE |
|---|---|---|---|---|---|
| C00001 | Ivan Bayross | Mumbai | 400054 | Maharashtra | 15000 |
| C00002 | Mamta Mazumdar | Madras | 780001 | Tamil Nadu | 0 |
| C00003 | Chhaya Bankar | Mumbai | 400057 | Maharashtra | 5000 |
| C00004 | Ashwini Joshi | Bangalore | 560001 | Karnataka | 0 |
| C00005 | Hansel Colaco | Mumbai | 400060 | Maharashtra | 2000 |
| C00006 | Deepak Sharma | Mangalore | 560050 | Karnataka | 0 |

**2. Exercise on retrieving records from the table:**

a. Display the names of all the clients.

b. Retrieve the entire contents of the CLIENT_MASTER table.

c. Retrieve the list of names, city and the state of all the clients.

d. List all the clients who are located in Mumbai

e. Find the names of client who carries a balance of 15,000/-
f. List the details of the clients whose name starts with 'M'.
g. List the details of clients who are staying in either Maharastra or in Karnatak

**3. Exercise on updating records in a table:**

a. Change the city of clientno 'C00005' to Bangalore.

b. Change the balance due of clientno 'C00001' to Rs. 1000.

c. Change the city of the client to Pune

**4. Exercise on deleting records in a table:**

a. Delete all clients from the CLIENT_MASTER.

b. Delete from CLIENT_MASTER where the column state holds the value 'Tamil Nadu'.

# LAB ASSIGNMENT-2

Insert the data in CLIENT_MASTER Table

```
INSERT CLIENT_MASTER (Client No, Name, CITY, PINCODE,
   STATE BALDUE)
VALUES('C00001','Ivan Bayross','Mumbai',400054,'Maharashtra',
VALUES('C00002','Mamta Mazumdar','Madras',780001,'Tamil Na
VALUES('C00003','Chhaya Bankar','Mumbai',400057,'Maharasht
VALUES('C00004','Ashwini Joshi','Bangalore',560001,'Karnatka
VALUES('C00005','Hansel Colaco','Mumbai',400060,'Maharash
VALUES('C00006','Deepak Sharma','Mangalore',560050,'Karnat

) Select name from CLIENT_MASTER
) Select * from CLIENT_MASTER
) select name, city, state from CLIENT_MASTER
) Select * from CLIENT_MASTER where city = 'Mumbai'
  select name from CLIENT_MASTER where Baldue = 1500
  select * from CLIENT_MASTER where name like 'M%
  select * from CLIENT_MASTER where state = 'Maharas
  or state = 'Karnatka'

a) update CLIENT_MASTER SET CITY = 'Bangalore' where
   clientNo = 'C00005'

  update CLIENT_MASTER SET BALDUE = 1000 where
   clientNo = 'C00001'

  update CLIENT_MASTER SET CITY = 'Pune'

(a) DELETE FROM CLIENT_MASTER

  DELETE from CLIENT_MASTER where state = 'Tami
        Nadu'
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 3

### DBMS Lab (KCA – 252)

Table Name: PRODUCT_MASTER

Description: Used to store product information.

| Column Name | Data Type | Size |
|---|---|---|
| **PRODUCTNO** | VARCHAR2 | 6 |
| **DESCRIPTION** | VARCHAR2 | 15 |
| **PROFITPERCENT** | NUMBER | 4,2 |
| **UNITMEASURE** | VARCHAR2 | 10 |
| **QTYONHAND** | NUMBER | 8 |
| **REORDERLVL** | NUMBER | 8,2 |
| **SELLPRICE** | NUMBER | 8,2 |
| **COSTPRICE** | NUMBER | 8,2 |

Data for PRODUCT_MASTER table:

| PRODUCTNO | DESCRIPTION | PROFIT | UNIT | QTYON | REORDER | SELL | COS |
|---|---|---|---|---|---|---|---|
| | | PERCENT | MEASURE | HAND | LVL | PRICE | PRIC |
| P00001 | 1.44floppies | 5 | Piece | 200 | 50 | 350 | 250 |
| P03453 | Monitors | 6 | Piece | 150 | 50 | 500 | 350 |
| P06734 | Mouse | 5 | Piece | 100 | 20 | 600 | 450 |
| P07865 | 1.22floppies | 5 | Piece | 100 | 20 | 750 | 50 |
| P07868 | Keyboards | 2 | Piece | 150 | 50 | 850 | 55 |
| P07885 | CDDrive | 2.5 | Piece | 80 | 30 | 700 | 45 |
| P07965 | 540 HDD | 4 | Piece | 100 | 40 | 350 | 25 |
| P07975 | 1.44Drive | 5 | Piece | 70 | 30 | 300 | 17 |
| P08865 | 1.22Drive | 5 | Piece | 75 | 30 | 450 | 30 |

Create the table and insert records as given above.

**Write the SQL queries for the following:**

1. Find out the names of all the clients.
2. Retrieve the list of names and cities of all the clients.
3. List the various products available from the product_master table.
4. List all the clients who are located in Bombay.
5. Display the information for client no C00001 and C 00002..
6. Find the products with description as '1.44 Drive' and '1.22 Drive'.
7. Find all the products whose sell price is greater than 5000.
8. Find the list of all clients who stay in city 'Bombay' or city 'Delhi' or 'Madras'.
9. Find the product whose selling price is greater than 2000 and less than or equal to :
10. List the name, city and state of clients not in the state of 'Maharashtra'.
11. Change the selling price of '1.44 floppy drive' to Rs.1150.00
12. Delete the record with client 0001 from the client_master table.
13. Find the products whose selling price is more than 1500 and also find the new sell price as original selling price*15.
14. Find out the clients who stay in a city whose second letter is a.
15. Find out the name of all clients having 'a' as the second letter in their names.
16. List the products in sorted order of their description.
17. Count the total number of product.
18. Calculate the average price of all the products
19. Calculate the minimum price of products.
20. Determine the maximum and minimum prices. Rename the tittle as 'max_price' : min_price respectively.
21. Count the number of products having price greater than or equal to 1500
22. List the products according to ascending order of their selling price.
23. List the products according to descending order of their selling price.

# LAB ASSIGNMENT- 3

Create a PRODUCT_MASTER table and insert records

INSERT INTO PRODUCT_MASTER (Product No, Description
Profitpercent, unitmeasure, Qtyonhand, Reorderlvl
sell price, Costprice).
Values ('P00001', '1.44 Floppies', 5, 'Piece', 200, 50
350, 250),
Values ('P03453', 'Monitors', 6, 'Piece', 150, 50,
500, 350),
Values ('P06734', 'Mouse', 5, 'Piece', 100, 20,
600, 450),
Values ('P07865', '1.22 floppies', 5, 'Piece', 100
20, 750, 500),
Values ('P07888', 'keyboards', 2, 'Piece', 150,
50, 850, 550),
Values ('P07885', 'CD Drive' 2.5, 'Piece', 80, 80
700, 450),
Values ('P07965', '540 HDD', 4, 'Piece', 100, 40,
350, 250),
Values ('P07975', '1.44Drive', 5, 'Piece', 70, 3
300, 175),
Values ('P08865', '1.22 drive', 5, 'Piece', 75,
30, 450, 300)

SQL Queries

. SELECT NAME FROM CLIENT_MASTER

. SELECT NAME, CITY from CLIENT_MASTER

. SELECT Description from PRODULT- MASTER

1. SELET * from CLIENT_MASTER where city
= 'BOMBAY'

```
.  SELECT * from CLIENT_MASTER where clientNo
    ('C00001','C00002)

.  SELECT * from PRODUCT_MASTER where descripti
   ('1.44 Drive', '1.22Drive')

.  SELET * from PRODUCT_MASTER where SELL PRI

.  SELECT * from PRODUCT_MASTER where SELL
    2000 and SELLPRICE <= 5000

.  SELECT * from CLIENT_MASTER where CITY IN
   ('Bombay', 'Delhi', 'Madras')

).  SELECT name, city, state from CLIENT_MA
    where state != 'Maharashtra'

.  Update PRODUCT_MASTER SET Sell price = 11
   where description = '1.44 Drive'

.  Delete from Client_Master where ClientNo:

).  Select description, sellprice, sellprice * 1.15
    new_selling_price from product-master
    sellprice > 1500

1.  select * from Client_Master where ci
     like '_a%'

5.  select name from client_master where
     like '_a%'
```

16. Select * from product_master order
ASC

17. Select count (*) from product_ master

18. Select Avg (sellprice) from product_m

19. Select min (sellprice) from product_

20. select max(sellprice) As max_price
as min_price from Product_Mast

21. Select count (*) from product_ma
sell price >= 1500

22. Select * from product_master order
ASC

23. select * from PRODUCT_MASTER ORD
sellprice DESC

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 4

### DBMS Lab (KCA – 252)

**Table name - EMPLOYEE**

| ID | F_NAME | L_NAME | DEPT | SALARY | DOJ | ADDRESS | Married |
|---|---|---|---|---|---|---|---|
| 1 | VINAY | KUMAR | MCA | 25000 | 27-09-2001 | GHAZIABAD | Y |
| 2 | SUMAN | VERMA | MCA | 15000 | 17-10-2006 | MEERUT | Y |
| 3 | AKASH | SINGH | CS | 20000 | 15-01-2005 | KANPUR | Y |
| 4 | SAGAR | KUMAR | IT | 20000 | 12-02-2019 | GHAZIABAD | N |
| 5 | ROHAN | SHARMA | CS | 21000 | 18-09-2005 | GHAZIABAD | |
| 6 | ROBIN | SINGH | IT | 18000 | 22-10-2021 | MEERUT | N |
| 7 | AKASH | RANJAN | CS | 22000 | 14-11-2019 | GHAZIABAD | |

Create the table and insert records as given above.

**Write the SQL queries for the following:**

1. Find the employee details who are from Ghaziabad.
2. Find the Department names.
3. List the employee id, complete name and the department of all the employees.
4. List all the employees' first name along with the department name.( Note the res should be displayed as " Vinay is working in MCA Department".)
5. Display the employee names who joined after 15-01-2005.
6. Update the salary of employee to 25000 whose employee id is 2 .
7. Display the employee id, name and Annual salary of all the employees. The colu name showing annual salary should be "Annual Salary".
8. List employee details with gross salary if every employee is getting a commission of 300.
9. List the employee details who are getting salary between 15000 and 30000.
10. List the employee details who joined between 27-09-2001 and 12-02-2019.
11. List the employees who works in MCA , CS Department.
12. List the employees who are not working in MCA department.
13. List the employees whose first name starts with 'R'.

14. List the employees whose first name starts with 'Ro'.

15. List the employees whose first name must be 5 character long and starts w

16. List the employees whose first name where the 2nd character must be 'o'.

17. List the employees who are married.

18. List the employees who are unmarried.

19. List the employees whose marriage status is not known.

20. List the employees who belongs to MCA department and are married

# LAB ASSIGNMENT - 4

```
CREATE TABLE employee_( id NUMBER(10), f-name
VARCHAR(30), l-name VARCHAR2(30), dept VARCHAR2(
Salary NUMBER (10), doj DATE, address VARCHAR:
married CHAR(1) );

DESCRIBE employee_;

INSERT INTO employee_(id, f-name, l-name, dept,
salary , doj , address , married)
VALUES (1, 'VINAY' , 'KUMAR', 'MCA' , 25000 , TO-DATE
('2001-09-27', 'YYYY-MM-DD'), 'GHAZIABAD' ,'Y');

INSERT INTO employee_(id , f-name, l-name , dept ,
salary , doj , address, married )
VALUES(2, 'VINAY', 'KUMAR', 'MCA', 15000 , TO_DATE
('2006- 10-17', 'YYYY-MM-DD'),'MEERUT' , 'Y');

INSERT INTO employee_(id, f-name, l-name, dept,
salary, doj , address , married)
VALUES(3, 'AKASH' , 'SINGH' ,'CS' , 20000, TO-DATE
('2005 -01-15','YYYY-MM-DD'), 'KANPUR', 'Y');

INSERT INTO employee_( id, f-name , L-name, dept,
salary , doj , address, married)
VALUES(4, 'SAGAR' , 'KUMAR' , 'IT', 20000 , TO_DATE
('2019-02-12', 'YYYY-MM-DD') , 'GHAZIABAD' , 'N');

INSERT INTO employee_( id, f-name , L-name , dept,
salary , doj, address, married)
VALUES( 5, 'ROHAN', 'SHARMA' , 'CS', 21000, TO-DA
('2005-09-18','YYYY-MM-DD'), 'GHAZIABAD');
```

```
INSERT INTO employee_(id, f-name, l-name, dep
salary, doj, address, married);
VALUES( 6, 'ROBIN', 'SINGH', 'IT', 18000, TO_DATE
('2021-10-21', 'YYYY-MM-DD'), 'GHAZIABAD');

SELECT * FROM employee_;

SELECT* from employee_ where address = 'GHAZIABA

SELECT dept from employee_;

SELECT id, f-name, || l-name as "employee name"
 from employee_;

Select f-name ||'is working in' || dept || 'depa
 from employee_;

Select * from employee_ where doj > to_date
('2005-01-15', 'YYYY-MM-DD');

update employee_ set salary = 25000 where

select * from employee_;

select id, f-name || l_name as "name", salau
as "Annual salary" from employee_;

Select salary +300 as gross from employee_

Select * from employee_ where salary betwe
15000 ~~to~~ and 30000 ;

select * from employee_ where doj between to_d
("27-09-2001" and "12-02-2019")
```

```
11. SELECT * FROM EMPLOYEE WHERE DEPT IN ('MCA',

12. SELECT* FROM EMPLOYEE WHERE DEPT!='MCA';

13. SELECT* FROM EMPLOYEE WHERE F_NAME LIKE

14. SELECT* FROM EMPLOYEE WHERE F_NAME LIKE

15. SELECT * FROM EMPLOYEE WHERE LENGTH (F_N
      5 AND F_NAME LIKE 'R%';

16. SELECT * FROM EMPLOYEE WHERE F_NAME LIKE

17. SELECT * FROM EMPLOYEE WHERE Married = 'y';

18. SELECT * FROM EMPLOYEE WHERE Married = 'N';

19. SELECT* FROM EMPLOYEE WHERE Married IS N

20. SELECT* FROM EMPLOYEE WHERE DEPT = 'MCA'
    Married= 'Y';
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 5

### DBMS Lab (KCA – 252)

1. Demonstrate the use of all string functions available in SQL.
2. Design the following table and solve the queries

| Roll | F_Name | M_Name | L_Name | Sec | City | Area | HouseNo | Div |
|---|---|---|---|---|---|---|---|---|
| 1 | Akash | Kumar | Jian | A | Ghaziabad | Rakesh Marg | C-355 | 1 |
| 2 | Manav | Mohan | Sharma | A | Ghaziabad | Raj nagar | D-211 | 2 |
| 3 | Chandra | Mohan | Batra | B | Meerut | Minto Road | A-201 | 3 |
| 4 | Rakesh | Chandra | Gupta | B | Kanpur | Nehru Marg | A-145 | |
| 5 | Sagar | Pratap | Singh | A | Meerut | Gandhi marg | C-35 | 0 |

i) Display Roll Number and complete name of all the students

ii) Display Roll Number, Name and complete address of all the students.

iii) Display Roll number, Name of all the students. (Note – The name should be displayed as A.K.Jain )

iv) Display all the student information according to ascending order of Section.

v) Display all the student information according to descending order of Section.

vi) Sort all the records according to section and then according to First Name.

vii) Display Roll number, name and city of all the students. (Note – City names should be left padded with 5 *s).

viii) Display those student details whose division has not been awarded.

ix) Display the cities from where the students are from.

x) Display Roll number, First Name and the division got. (Note – Division should be displayed as 1- FIRST, 2 – SECOND, 3 – THIRD, 0- FAIL, NULL – NOT AWARDED)

xi) Display Roll Number, First Name, Section of all the students. (Note – Section A should be displayed as 1 and Section B should be displayed as 2)

3. Solve the following query

| Empcode | Empname |
|---|---|
| E1 | 001Rajkumar |
| E2 | Ramkumar002 |
| E3 | Ravikumar003 |

Output should

| Empcode | Empname |
|---|---|
| E1 | Rajkumar |
| E2 | Ramkumar |
| E3 | Ravikumar |

4. Solve the following query

| First Name | Middle Name | Last Name |
|---|---|---|
| Pankaj | Kumar | Tiwari |
| Ashok | Kumar | Sharma |
| Arun | Kumar | Sharma |

**Output should**

| Name |
|---|
| P. K. Tiwari |
| A.  K. Sharma |
| A. K. Gupta |

5. Given the table structure with data.

| Sname | Marks |
|---|---|
| Raj | 65 |
| Amit | 32 |
| Sanjay | 45 |
| Rohit | 40 |
| Anil | 35 |

Output 1 Rule - : **Pass marks are 35.**

**Display the result as**

| Sname | Result |
|---|---|
| Raj | Pass |
| Amit | Fail |
| Sanjay | Pass |
| Rohit | Pass |
| Anil | Pass |

# ASSIGNMENT - 5

```
SELECT
 UPPER ('Himanshi Chopra') as UPPER_CASE, LOWER
 ('HIMANSHI') as LOWER_CASE, LENGTH ('HIMAN
as LENGTH, SUBSTR ('HIMANSHI CHOPRA', 1,6) ~~HIMANSHI'~~)
as SUBSTRING, REPLACE ('HIMANSHI CHOPRA', 'HIMA

 'HIMANSHIII') AS REPLACE_NAME
 TRIM ('HIMANSHI CHOPRA') AS TRIMSTR,
 CONCAT ('HIMANSHI', 'CHOPRA') AS CONCAT_STR
 FROM DUAL;
```

```
CREATE TABLE Students (
  ROLL NUMBER,
  f_name varchar2(20),
  L_name varchar2(20),
  Sec char(1)
  City varchar2(20)
  Area varchar2 (50)
  House No varchar2 (50)
  Div Number);
```

```
Select Roll, f_name ||' '|| l_name as Full_name
Students.
```

```
Select Roll, f_name ||' '|| L_name Full_name, f
||' '|| HouseNo ||' '|| City as Address from
students.
```

```
Select Roll, Substr (F_name, 1,1) ||'.'|| Substr (L_
1,1) As Name from Students
```

```
(iv)    select * from students order by sec;

(v)     select * from students order by sec desc

(vi)    select * from students order by sec, f-no

(vii)   select, Roll, f-name || ' ' || L_name As N
          (city, length (city) + 5, '*') as city

(viii)  Select * from students where DIV IS NU

(ix)    select distinct city from students;

(x)     select Roll, f-name,
         CASE
               when DIV = 1  THEN 'FIRST'
               when DIV = 2. THEN 'SECOND'
               when DIV = 3  THEN 'THIRD'
               when DIV = 0  THEN 'FAIL'
                Else 'Not Awarded'
        END As Division
         from students;

(xi)     select Roll, f-name
           CASE
          when sec = 'A' THEN 1
          when sec = 'B' THEN 2
          End As Section
          from students;
```

```
select Empcode,
  TRIM (THanslate ( Empname '0123456789')) AS

 Empname from Employees


Select Substr (firstname, 1, 1) 11 '.' 11 Substr
(Last_name 1,1) AS Name from students.

Select Sname,
case
   when marks >= 35 then 'Pass'
    Else 'fail'
  END As Result
   . from students ;
```

# KIET GROUP OF INSTITUTIONS

# DEPARTMENT OF COMPUTER APPLICATIONS

## LAB ASSIGNMENT 6

## DBMS Lab (KCA – 252)

**Create the following tables & insert the records as given.**

**i) CLIENT_MASTER (Used for storing client information)**

| Column Name | Data Type | Size | Attribute |
|---|---|---|---|
| Client_No | Varchar2 | 6 | Primary Key / Must Start with 'C' |
| Name | Varchar2 | 20 | Not Null |
| CITY | Varchar2 | 20 | Must be either from Delhi or Mumbai or Chennai only. |
| Pincode | Number | 6 | |
| State | Varchar2 | 20 | |
| BALDUE | Number | 10,2 | |
| Email | Varchar2 | 30 | Always take unique value |

**ii) PRODUCT_MASTER (Used for storing Product Information)**

| Column Name | Data Type | Size | Attribute |
|---|---|---|---|
| Product_No | Varchar2 | 6 | Primary Key / Must Start with 'P' |
| Name | Varchar2 | 20 | Not Null |
| UnitMeasure | Varchar2 | 10 | Not Null |
| QtyOnHand | Number | 8 | Not Null, cannot less than ReorderLevel |
| ReorderLevel | Number | 8 | Not Null |
| SellPrice | Number | 8,2 | Not Null, cannot be 0 or -ve |
| CostPrice | Number | 8,2 | Not Null, cannot be 0 or -ve |

**iii) SALESMAN_MASTER (Used for storing salesman information)**

| Column Name | Data Type | Size | Attribute |
|---|---|---|---|
| Salesman_No | Varchar | 6 | Primary Key/First Letter must start with 'S' |
| Sal_name | Varchar | 20 | Not Null |
| Address | Varchar | 40 | Not Null |
| City | Varchar | 20 | |
| State | Varchar | 20 | |
| Pincode | Number | 6 | |
| Sal_Amount | Number | 8,2 | Not Null, Cannot be 0 or -ve |
| Tgt_to_get | Number | 8,2 | Not Null, Cannot be 0 or -ve |
| Ytd_Sales | Number | 8,2 | Not Null, Cannot be 0 or -ve |
| Remarks | Varchar | 30 | |

### iv) SALES_ORDER (Used to store client's orders)

| Column Name | Data Type | Size | Attribute |
|---|---|---|---|
| Order_NO | Varchar2 | 6 | Primary Key/ First letter must start v |
| Order_Date | Date | | Not Null |
| Client_No | Varchar2 | 6 | Foreign Key reference client Client_Master table |
| Dely_add | Varchar2 | 25 | |
| Salesman_No | Varchar2 | 6 | Foreign Key references Salesman Salesman_Master table |
| Dely_type | Char | 1 | Delivery part(P)/full(F), default F |
| Billed_yn | Char | 1 | Values must be 'Y'/'N' |
| Dely_Date | Date | | Can not be less than Order_Date |
| Order_Status | Varchar2 | 10 | Values ('In Process(IP)'; 'Fulfi 'Canceled(C)' |

### v) SALES_ORDER_DETAILS

| Column Name | Data Type | Size | Attribute |
|---|---|---|---|
| Order_No | Varchar | 6 | Primary Key/Foreign Key referenc Order_No of Sales_Order |
| Product_No | Varchar2 | 6 | Primary Key/Foreign Key referenc Product_No of Product_Master |
| Qty_Order | Number | 8 | |
| Qty_disp | Number | 8 | |

**Data for CLIENT_MASTER**

| CLIENTNO | NAME | CITY | PINCODE | STATE | BALDUE | EMAI |
|---|---|---|---|---|---|---|
| C00001 | Pankaj Sharma | Delhi | 400054 | Delhi | 15000 | pk@gmai |
| C00002 | Yogesh Sharma | Delhi | 780001 | Delhi | 0 | ys@gmai |
| C00003 | Aditya Singh | Mumbai | 400057 | Maharashtra | 5000 | as@gmai |
| C00004 | Ashwini Joshi | Chennai | 560001 | Tamil Nādu | 0 | aj@gmai |
| C00005 | Neha Sharma | Mumbai | 400060 | Maharashtra | 2000 | ns@gmai |
| C00006 | Divya Gupta | Chennai | 560050 | Tamil Nādu | 0 | |

**Data for PRODUCT_MASTER**

| Product_No | Name | Unit Measure | Qty OnHand | Reorder Level | Sell Price |
|---|---|---|---|---|---|
| P00001 | T-Shirt | Piece | 200 | 50 | 350 |
| P03453 | Shirt | Piece | 150 | 50 | 500 |
| P06734 | Cotton Jeans | Piece | 100 | 20 | 600 |
| P07865 | Jeans | Piece | 100 | 20 | 750 |
| P07868 | Trouser | Piece | 150 | 50 | 850 |
| P07885 | Pull Over | Piece | 80 | 30 | 700 |
| P07965 | Denim Shirts | Piece | 100 | 40 | 350 |
| P07975 | Lycra Tops | Piece | 70 | 30 | 300 |
| P08865 | Skirts | Piece | 75 | 30 | 450 |

**Data For SALESMAN_MASTER**

| Salesman_no | Name | Address | City | PIN | State | Salamt | Tgt_To_get | Yt e |
|---|---|---|---|---|---|---|---|---|
| S00001 | Kiran | A/14 worli | Mumbai | 400002 | Mah | 3000 | 100 | 50 |
| S00002 | Manish | 65,nariman | Mumbai | 400001 | Mah | 3000 | 200 | 10 |
| S00003 | Ravi | P-7 Bandra | Mumbai | 400032 | Mah | 3000 | 200 | 10 |
| S00004 | Ashish | A/5 Juhu Bombay | Mumbai | 400044 | Mah | 3500 | 200 | 15 |

**Data for SALES_ORDER table:**

| Orderno | S_Orderdate | ClientNo | Delytype | Bill YN | Salesmanno | Delydat |
|---|---|---|---|---|---|---|
| O19001 | 12-jan-21 | 0001 | F | N | S00001 | 20-jan-2 |
| O19002 | 25-jan-21 | 0002 | P | N | S00002 | 27-jan-2 |
| O16865 | 18-feb-21 | 0003 | F | Y | S00003 | 20-feb-2 |
| O19003 | 03-apr-21 | 0001 | F | Y | S00001 | 07-apr-2 |
| O46866 | 20-may-21 | 0004 | P | N | S00002 | 22-may- |
| O10008 | 24-may-21 | 0005 | F | N | S00004 | 26-may- |

**Data for SALES_ORDER_DETAILS table:**

| S_order no | Productno | Qtyordered | Qtydisp |
|---|---|---|---|
| O19001 | P00001 | 4 | 4 |
| O19001 | P07965 | 2 | 1 |
| O19001 | P07885 | 2 | 1 |
| O19002 | P00001 | 10 | 0 |
| O46865 | P07868 | 3 | 3 |
| O46865 | P07885 | 10 | 10 |
| O19003 | P00001 | 4 | 4 |
| O19003 | P03453 | 2 | 2 |

# LAB ASSIGNMENT-6

```
. Create table CLIENT_MASTER ( Client_No VARCHAR 2
  primary Key , Name VARCHAR2(20) NOT NULL,
  city VARCHAR2(20) CONSTRAINT CITY_ CONSTRAINT ch
  city in ('Delhi', 'Mumbai', 'Chennai')), Pincode
  NUMBER (6) , State VARCHAR2 (20), Balalue NU
  (10,2) , Email VARCHAR2 (30) UNIQUE , CHECH
  (client_No LIKE 'C%')';
```

```
Create table PRODUCT_MASTER ( Product_No VARCH
 (6) PRIMARY KEY, Name VARCHAR2 (20) NOT NU
 Unit_Measure VARCHAR2 (10) NOT NULL , QTYONHAN
 NUMBER (8) NOT NULL , SELLPRICE NUMBER (8,2 )
 NULL , COSTPRICE NUMBER (8,2) NOT NULL , CHE
 (Product_No Like 'P%') , CHECH (QTYONHAN>
 REORDERLEVEL ) , CHECK (SELLPRICE >0) , CHECK
 (COSTPRIEE >0 ));
```

```
. Create table SALESMAN_MASTER ( Salesman_ No
  VARCHAR 2(6) primary Key , Sal-Name VARCHAR2(
  NOT NULL , Address VARCHAR2 (40) NOT NULL ,
  Address VARCHAR2 (20) NOT NULL , CITY VARCHAR
  STATE VARCHAR2 (20) , PINCODE NUMBER (6), Sal-Am
  NUMBER (8,2 ), NOT NULL, TGT_TO_GET NUMBER
  (8,2 ) NOT NULL , YTD - SALES NUMBER (8,2) NOT
  REMARKS VARCHAR2 (30) , CHECK ( Salesman -N
     lik 'S%) , CHECH (Sal-Amount >0), CHECK
(TGT_TO_GET >0), CHECK (YTD_sales >0), CHECK
 (REMARKS >0));
```

```
CREATE TABLE SALES_ORDER( ORDER_NO VARCHAR2(6)
PRIMARY KEY, ORDER_DATE NOT NULL, CLIENT_N
VARCHAR2(6) REFERENCES CLIENT_MASTER (CLIENT_N
DELY_ADD VARCHAR2(25), SALESMAN_NO VARCHAR
REFERENCES SALESMAN_MASTER (SALESMAN_NO), DE
TYPE CHAR(1) DEFAULT 'F', BILLED_YN CHAR(1),
DELY_DATE DATE, ORDER_STATUS VARCHAR2(10
CHECH (ORDER_NO LIKE '%O'), CHECH (BILLED_Y
IN ('P','F')), CHECK(DELY_DATE ≥ ORDER_DATE
CHECH (ORDER_STATUS IN ('IN PROCESS (IP)', 'FUL
(F)','CANCELED(C)')));
```

```
CREATE TABLE SALES_ORDER_DETAILS
(ORDER_NO VARCHAR(6) REFERENCES SALES_ORDER (ORD
NO), PRODUCT_NO VARCHAR2(6) REFERENCES PRODU
MASTER (PRODUCT_NO), QTY_ORDER NUMBER(8),
QTY_DISP NUMBER(8), CHECK (QTY_ORDER >= 0)
CHECH (QTY_DISP >= QTY_ORDER), PRIMARY KEY (
ORDER_NO, PRODUCT_NO));
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 7

### DBMS Lab (KCA – 252)

### Assignments on Date Functions

1. Implement the date functions of SQL.

2. Create the following table and perform the following queries.

| Field | Datatype | Remark |
|---|---|---|
| EmpId | Varchar2(3) | Primary key, Should start with 'E' |
| Ename | Varchar2(25) | Not Null |
| DOB | Date | Not Null |
| DOJ | Date | Not Null, Should later than DOB |
| Salary | Number(7) | Not Null, Should take only +ve value |
| Adhar | Number(12) | Unique, should always take 12 digit number |

2.1 List the employee details who joined on a particular date.

2.2 Write a query to list the employees with Hiredate in the format like February 22, 1991.

2.3 Write a query to list the employees who joined before 2018.

2.4 Write a query to list the employees who joined in the month January.

2.5 Write a query to list the employees who have joined in the year 2019.

2.6 Write a query to list the employee details according to the date of joining (recent joining should come first).

2.7 Write a query to list the employee details along with their experience.

2.8 Write a query to list those employees whose salary is an odd value.

2.9 Write a query to list the employees of id E001 or E10, joined in the year 1991.

2.10 Write a query in SQL to list the employees who joined in any year except the month February.

2.11 Write a query to display happy birthday to those employees who are eligible to get this message.

2.12 Display the employee id, Name and Date of birth of all the employees( **Note - if the dob is 3 – jan-1985 then it should be displayed as 3rd January 1985)**

# LAB ASSIGNMENT - 7

```
.  SELECT SYSDATE FROM dual;
   SELECT ADD_MONTHS (SYSDATE, 3) AS "MONTH" FROM du
   SELECT LAST_DAY (SYSDATE) FROM dual;
   SELECT MONTHS_BETWEEN (SYSDATE, TO_DATE ('2023-01-
   'YYYY-MM-DD')) AS "MONTHS FROM dual;
   SELECT ROUND (SYSDATE, 'MONTH') AS "ROUNDED Date"
    FROM dual;

-' CREATE TABLE EMPLOYEE ( EmpId. Varchar2(3) PRIMARY
   CHECH (EmpId LIKE 'E%'), Ename VARCHAR2 (25) NOT
   DOB DATE NOT NULL, DOJ DATE NOT NULL CHECH
   (DOJ>DOB) Salary NUMBER (7) NOT NULL CHECK
   (Salary>0), Adhar NUMBER (12) UNIQUE CHECK (LEN
   (Adhar) = 12));

.1 SELECT * FROM EMPLOYEE WHERE DOJ= TO_DATE ('2015
   'YYYY-MM DD');

2  SELECT Ename, To_char (DOJ, 'MONTH DD, YYYY)
     AS Hiredate FROM EMPLOYEE;

3  SELECT * FROM EMPLOYEE WHERE DOJ < TO_DATE ('2018
   'YYYY - MM - DD');

1  SELECT * FROM EMPLOYEE WHERE TO_CHAR (DOJ, 'M
       = '01';

5  SELECT * FROM EMPLOYEE WHERE TO_CHAR (DOJ, 'YY

.6 SELECT * FROM EMPLOYEE ORDERBY DOJ DESC;
```

```
2.7   SELECT Ename , DOJ , ROUND (MONTHS_B
      122,1 ) AS Experience FROM EMPLOY

2.8   SELECT * FROM EMPLOYEE WHERE MOD (Salar

2.9   SELECT * FROM EMPLOYEE WHERE (EmpId= '
      'E10' ) AND TO_CHAR (DOJ , 'YYYY') = '19

2.10  SELECT * FROM EMPLOYEE WHERE TO_CHA
      <> '02';

2.11  SELECT 'Happy Birthday' || Ename AS M
      EMPLOYEE WHERE TO_CHAR (DOB, 'MM-
      (SYSDATE , 'MM-DD');

2.12  SELECT EmpId , Ename , TO_CHAR (DOB
      AS Date_of_Birth From EMPLOYEE
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 8

### DBMS Lab (KCA – 252)

### Assignments on Join

**Note – Questions from 1 to 14 refer the sample tables Salesman, Customer,**

***Sample table*: 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 |

***Sample table*: customer**

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

***Sample table*: orders**

| ord_no | purch_amt | ord_date | customer_id | salesman_ic |
|---|---|---|---|---|
| 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 |

1. Write a SQL statement to prepare a list with salesman name, customer name and their cities for the salesmen and customer who belongs to the same city.

2. Write a SQL statement to make a list with order no, purchase amount, customer nam and their cities for those orders which order amount between 500 and 2000.

3. Write a SQL statement to know which salesman are working for which customer.

4. Write a SQL statement to find the list of customers who appointed a salesman for the jobs who gets a commission from the company is more than 12%.

5. Write a SQL statement to find the list of customers who appointed a salesman fo jobs who does not live in the same city where their customer lives, and gets a comr is above 12%.

6. Write a SQL statement to find the details of a order i.e. order number, order date, a of order, which customer gives the order and which salesman works for that custon how much commission he gets for an order.

7. Write a SQL statement to make a list in ascending order for the customer who work through a salesman or by own.

8. Write a SQL statement to make a list in ascending order for the customer who grade less than 300 and works either through a salesman or by own.

9. Write a SQL statement to make a report with customer name, city, order numb date, and order amount in ascending order according to the order date to find th any of the existing customers have placed no order or placed one or more orders.

10. Write a SQL statement to make a report with customer name, city, order numb date, order amount salesman name and commission to find that either any of th customers have placed no order or placed one or more orders by their salesr own.

11. Write a SQL statement to make a list in ascending order for the salesmen who w for one or more customer or not yet join under any of the customers.
12. Write a SQL statement to make a list for the salesmen who works either for or customer or not yet join under any of the customers who placed either one or m or no order to their supplier.

13. Write a SQL statement to make a list for the salesmen who either work for o customers or yet to join any of the customer. The customer may have placed, ei more orders on or above order amount 2000 and must have a grade, or he m placed any order to the associated supplier.
14. Write a SQL statement to make a cartesian product between salesman and c each salesman will appear for all customer and vice versa.

**Note – For questions 15 to 19 use sample table comp item_mast**

*Sample table*: company_mast

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

*Sample table*: item_mast

| PRO_ID | PRO_NAME | PRO_PRICE | PRO_CO |
|---|---|---|---|
| 101 | Mother Board | 3200 | |
| 102 | Key Board | 450 | |
| 103 | ZIP drive | 250 | |
| 104 | Speaker | 550 | |
| 105 | Monitor | 5000 | |
| 106 | DVD drive | 900 | |
| 107 | CD drive | 800 | |
| 108 | Printer | 2600 | |
| 109 | Refill cartridge | 350 | |
| 110 | Mouse | 250 | |

15. Write a SQL query to display all the data from the item_mast, inclu each item's producer company.

16. Write a SQL query to display the item name, price, and company nam
17. Write a SQL query to display the average price of items of each cc name of the company.
18. Write a SQL query to display the names of the company whose prod price larger than or equal to Rs. 350.

19. Write a SQL query to display the name of each company along witl their most expensive product.

Note – For questions 20 to 23 use the sample tables emp_deptmen,

*Sample table*: emp_departmen

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

```
EMP_IDNO EMP_FNAME       EMP_LNAME       EMP_D
--------- --------------- --------------- -----
   127323 Michale         Robbin
   526689 Carlos          Snares
   843795 Enric           Dosio
   328717 Jhon            Snares
   444527 Joseph          Dosni
   659831 Zanifer         Emily
   847674 Kuleswar        Sitaraman
   748681 Henrey          Gabriel
   555935 Alex            Manuel
   539569 George          Mardy
   733843 Mario           Saule
   631548 Alan            Snappy.
   839139 Maria           Foster
```

20. Write a query in SQL to display all the data of empl

21. Write a query in SQL to display the first name and the name and sanction amount for their departme

22. Write a query in SQL to find the first name an departments with a budget more than Rs. 50000.

23. Write a query in SQL to find the names of depart are working

# LAB ASSIGNMENT - 8

```
.   select s.name, c.cust_name, s.city FROM salesman s
    customer c ON s.city = c.city AND s.salesman.
    c.salesman_id;

.   select o.ord_no, o.purch_amt, c.cust_name, c.c
    FROM order o JOIN customer c ON o.customer_ic
    c.customer_id WHERE o.purch_amt BETWEEN 500
    2000;

3.  select s.name AS salesman_name, c.cust_name
    salesman s JOIN customer c ON s.salesman_id
    c.salesman_id;

4.  select c.cust_name FROM customer c JOIN sales
    s ON c.salesman_id = s.salesman id WHERE
    s.commission >0.12;

5.  select c.cust_name FROM customer c JOIN
    s ON c.salesman_id = s.salesman_id WHERE
    s.city <> c.city AND s.comission >0.12;

6.  select o.ord_no, o.ord_date, o.purch_amt,
    name, s.name, s.comission FROM order o
    customer c ON o.customer_id = c.customer_id
    salesman s ON o.salesman_id = s.salesman_ic

7.  select c.cust_name FROM customer c LEFT JOI
    order o ON c.customer_id = o.customer_id O
    BY c.cust_name ASC;
```

8.

```
Select c.cust_name FROM customer c LEFT
O ON c.customer_id = O.customer_id
<300 ORDER BY c.cust_name ASC;
```

9.

```
select c.cust_name , c.city , o.ord_no, o
o.purch_amt FROM customer c LEFT J
o ON c.customer_id = o.customer_id ord
o.order_date ASC;
```

10.

```
select c.cust_name , c.city , o.ord_no , c
o.purch_amt, s.name , s.commision FRON
c.customer_id = o.customer_id ORDER BY c
```

11.

```
select s.name FROM salesman s LEFT JOIN
c ON s.salesman_id = c.salesman_id ORDER P
```

12.

```
Select s.name, COUNT (c.customer_id) , COU
FROM salesman s LEFT JOIN customer c O
id = c.salesman_id LEFT JOIN orders o
= o.customer_id GROUP BY s.name ORDEF
ASG;
```

13.

```
Select s.name, COUNT (DISTINCT c.custome
( o.ord_no) FROM salesman s LEFT JOI
c ON s.salesman_id= c.salesman_id
c.grade IS NOT NULL OR o.ord_no IS
BY s.name ORDER BY s.name ASC;
```

14.

```
Select s.name AS salesman_name , c.cust
FROM salesman s CROSS JOIN customer
```

15. SELECT i.*, c.* FROM item_mast i JOIN compar
c on i.PRO_COM = c.com_ID;

16. SELECT i.PRO_NAME, i.PRO_PRICE, C.COM_NA
item_mast i JOIN company_mast c ON
= C.COM_ID;

17. select C.COM_NAME, AVG (i.PRO_PRICE) AS avg_
FROM item_mast i JOIN company_mast
i.PRO_COM = C.COM_ID GROUP BY C.COM_N

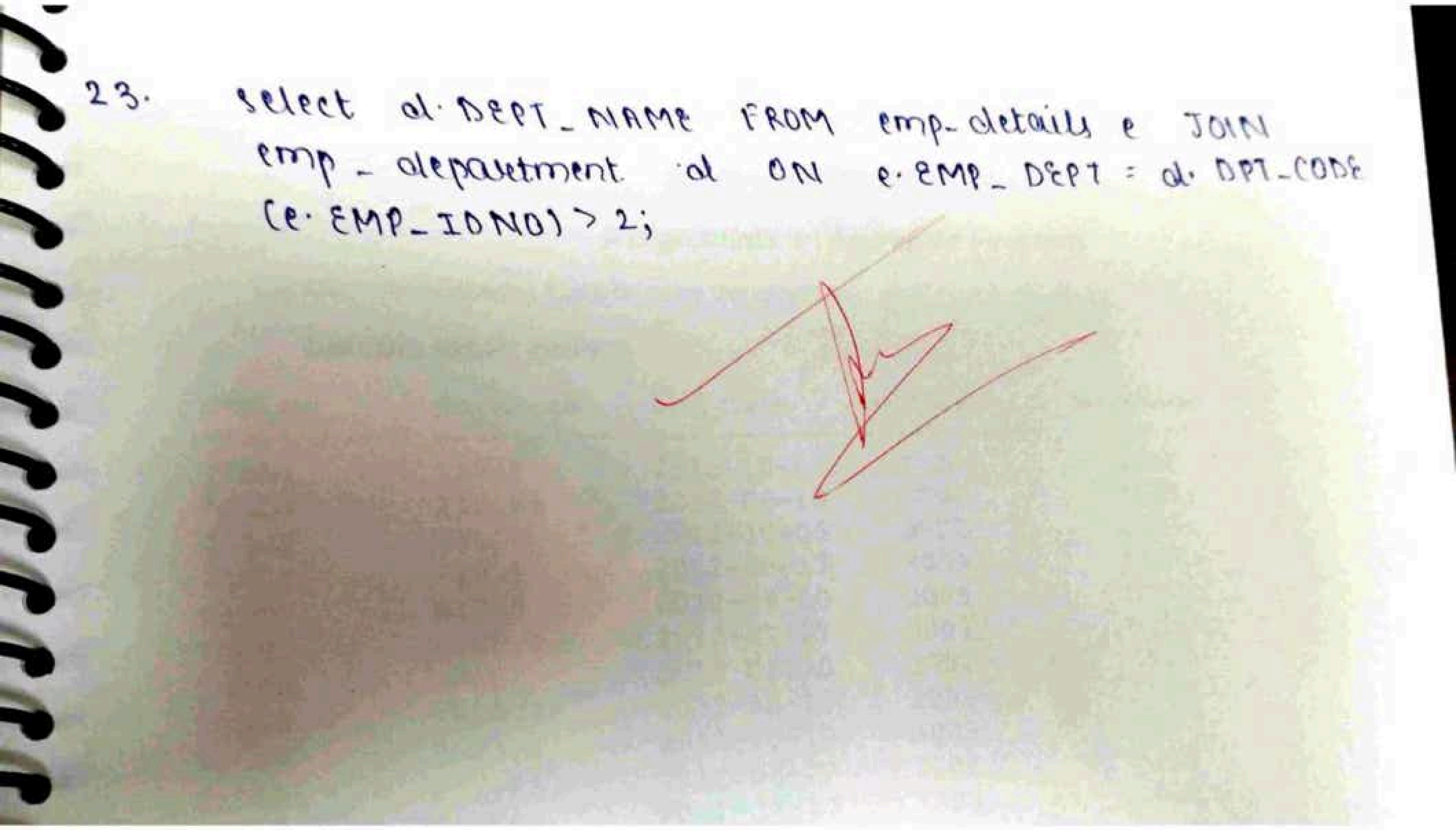
18. Select C.COM_NAME FROM item_mast i JO
_mast C JON i_PROM_COM = C.COM_ID GR
c.com_NAME HAVING AVG (i.PRO_PRICE)

19. SELECT C.COM_NAME, i.PRO_ID, MAX (i.PRO_
item_mast i JOIN company_mast c ON i
= C.COM.ID GROUP BY c.com_name, i.PRO_I
BY max_price DESC;

20. select e.*, d.* FROM emp_details e JOI
department d on e.emp_dept = d.dept

21. select e.emp_fname, e.emp_lname, d.de
d.dept_allotment FROM emp_details e
emp_department d ON e.emp_DEPT = c

22. select e. EMP_FNAME, e.emp_Lname FROM
e JOIN emp_department d ON e.EMP
_CODE WHERE d.DPT_ALLOTMENT > 50000



```
23.   select d.DEPT_NAME FROM emp-details e JOIN
      emp_department d ON e.EMP_DEPT = d.DPT_CODE
      (e.EMP_IDNO) > 2;
```

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 9

### DBMS Lab (KCA – 252)

### Assignments on Aggregate Function

Consider the following table to solve the queries.

***Sample table*: 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 |

1. write a SQL query to calculate total purchase amount of all orders. Return total pur amount.
2. write a SQL query to calculate average purchase amount of all orders. Return av purchase amount.
3. write a SQL query to find the number of salespeople.
4. write a SQL query to find the maximum purchase amount.
5. write a SQL query to find the minimum purchase amount.
6. write a SQL query to find the highest purchase amount ordered by each customer. R customer ID, maximum purchase amount.
7. write a SQL query to find the highest purchase amount ordered by each customer particular date. Return, order date and highest purchase amount.
8. write a SQL query to find the highest purchase amount on '2012-08-17' by salesperson. Return salesperson ID, purchase amount.
9. write a SQL query to find highest order (purchase) amount by each customer in a part order date. Filter the result by highest order (purchase) amount above 2000.00. R customer id, order date and maximum purchase amount.
10. write a SQL query to find the maximum order (purchase) amount in the range 2000, (Begin and end values are included.) by combination of each customer and order Return customer id, order date and maximum purchase amount.
11. write a SQL query to find the maximum order (purchase) amount by each custome customer ID should be in the range 3002 and 3007(Begin and end values are inclu Return customer id and maximum purchase amount.
12. write a SQL query to count all the orders generated on '2012-08-17'. Return numb orders

**_Sample table_: customer**

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

13. write a SQL query to count the number of customers.
14. write a SQL query to find the number of customers who got at least a grada
    activity.
15. write a SQL query to find the highest grade of the customers for each of
    city, maximum grade.

**_Sample table_: 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
```

16. write a SQL query to count number of salespeople who belongs to a city. F
    salespeople.

**_Sample table_: item_mast**

```
PRO_ID PRO_NAME                        PRO_PRICE    PRO_COM
------- ------------------------------ ----------  ----------
    101 Mother Board                        3200          1
    102 Key Board                            450          1
    103 ZIP drive                            250          1
    104 Speaker                              550          1
    105 Monitor                             5000          1
    106 DVD drive                            900          1
    107 CD drive                             800          1
    108 Printer                             2600          1
    109 Refill cartridge                     350          1
    110 Mouse                                250          1
```

17. write a SQL query to count number of products where product price is h
    to 350. Return number of products.

Consider a table named **Employee( Eid, Name, Dept, Salary, DOJ)**
Solve the following queries.

18. Display the latest date on which an employee had joined.
19. Display the 1st date on which an employee had joined.
20. List out how many numbers of departments are there.
21. Display how many numbers of employees are there in MCA Departm
22. Display department wise maximum average salary.
23. Display the total number of employees in the organization.
24. Display department wise the numbers of employees working.

## LAB ASSIGNMENT- 9

1. select SUM (purch-amt) FROM orders;

2. select AVG( purch-amt) FROM orders;

3. select COUNT(DISTINCT salesman_id ) FROM

4. select MAX (purch-amt) FROM orders;

5. select MIN (purch-amt) FROM orders;

6. select customer_id, MAX( purch_amt ) FR
GROUP BY customer_id;

7. select customer_id, ord-date , MA
FROM orders GROUP BY customer_id

8. select salesman_id , MAX (purch_am
WHERE ord-date = '2012-08-17' GROUP

9. select customer_id , ord-date , MAX (
orders GROUP BY customer-id , ord
MAX (purch-amt)>2000 ~~BETWEEN 2000~~
~~GROUP BY~~

10. select customer-id , ord-date , M
FROM orders WHERE purch-amt
2000 and 6000 GROUP BY cus
date;

```
11.  Select customer_id , MAX (purch - amt ) F
     WHERE customer-id BETWEEN 3002 a
     GROUP BY customer_id ;

12.  Select COUNT (*) FROM orders WHERE ord-d
     08-17' ;

13.  select COUNT (*) FROM customer ;

14.  select COUNT (*) FROM customer WHERE
     NULL;

15   select city , MAX(grade) FROM customer
     city .

16   select city , COUNT (* ) FROM salesman
     city .

17   select count (*) FROM item_mast W
     PRICE >= 350' ,

18   Select MAX (DOJ) FROM Employee ;

19   select MIN (DOJ) FROM Employee ;

20   select COUNT ( DISTINCT DEPT ) FROM

21   Select count (*) from Employee
     Dept = 'MCA'
```

22. select Dept , MAX (AVG (salary)) FROM employee
GROUP BY Dept ;

23. select COUNT (*) FROM employee ;

24. select Dept , COUNT (*) FROM employee GROUP BY Dept;

## KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS

### LAB ASSIGNMENT 10

### DBMS Lab (KCA – 252)

### Assignments on Sub Query

1. Create the following table and insert some records.
   Table Name: Employee

| Field | Datatype |
|---|---|
| EmpId | Number |
| Name | Varchar2 |
| DOJ | Date |
| JobId | Number |
| Salary | Number |

| EmpId | Name | DOJ | JobId | Salary |
|---|---|---|---|---|
| 100 | Aman Jian | 17-June-2017 | AD_PRES | 24000.00 |
| 101 | Yash Kumar | 15-July-2019 | AD_VP | 17000.00 |
| 102 | Ayushi | 12-Aug-2017 | IT_PROG | 9000.00 |
| 103 | Kamal | 15-Sept-2016 | IT_PROG | 6000.00 |
| 105 | Madhav Mohan | 14-July-2018 | IT_PROG | 4000.00 |
| 106 | Astha Sharma | 27-June-2017 | PU_CLERK | 2500.00 |

Write the following queries over the said table

1.1 Write a query to display the name for those employees who gets m salary than the employee whose id is 104.

1.2 Write a query to display the name,salary,department id, job id for tl employees who works in the same designation as the employee w whose id is 103.

1.3 Write a query to display the name,salary,department id for t employees who earn such amount of salary which is the smallest sala any of the departments.

1.4 Write a query to display the employee details who are having the : date of birth as of employee having id 106.

# LAB ASSIGNMENT-10

```
CREATE TABLE Employee ( EmpId NUMBER PRIMAR
KEY, Name VARCHAR2 (50), DOJ DATE , JobId
VARCHAR2 (10) salary NUMBER);
```

```
1.1 SELECT Name FROM Employee WHERE Salary > (
    Salary FROM Employee WHERE EmpId = 104);
```

```
·2  SELECT Name , salary , JobId FROM Employee W
    JobId = (SELECT JobId FROM Employee WHERE
    EmpId = 103);
```

```
1.3 SELECT Name , Salary , JobId FROM Employee
    salary = (SELECT MIN (salary) FROM Employee
```

```
1.4 SELECT * FROM Employee WHERE DOJ = (SELECT
    FROM Employee WHERE EmpId = 106);
```

```
1.5 Select * FROM Employee WHERE DOJ > (SELECT DO
    FROM Employee WHERE EmpId = 102);
```