

GITHUB Link:

[lakshaygoel2000/DBMS: DBMS Tutorials and Practice \(github.com\)](#)

# KIET GROUP OF INSTITUTIONS

## DEPARTMENT OF COMPUTER APPLICATIONS



### DATABASE MANAGEMENT SYSTEM (DBMS)

#### (KCA-204) SESSION (2023-25) 2nd SEM

NAME: Lakshay Goel

CLASS ROLL: 27

UNIVERSITY ROLL: 2300290140095

SEM: 2

SEC: C

**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.

# Assignment - 1

```
CREATE TABLE CLIENT_MASTER (
    Client-No. VARCHAR2(6),
    Name VARCHAR2(20),
    Address1 VARCHAR2(30),
    Address2 VARCHAR2(30),
    City VARCHAR2(15),
    Pincode NUMBER(6),
    BalDue NUMBER(10,2)
);
```

```
CREATE TABLE PRODUCT_MASTER (
    Product-No. VARCHAR2(6),
    Description VARCHAR2(20),
    Profit-percentage NUMBER(6),
    Unit-Measure VARCHAR2(10),
    Qty-On-Hand NUMBER(6),
    Reorder-Level NUMBER(6),
    Sell-Price NUMBER(7,2),
    Cost-Price NUMBER(7,2)
);
```

ALTER TABLE CLIENT-MASTER ADD DOB DATE;

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 PRODUCT MASTER;

DROP TABLE CLIENT-MASTER AND PRODUCT-MASTER.

## 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:

- Display the names of all the clients.
- Retrieve the entire contents of the CLIENT\_MASTER table.
- Retrieve the list of names, city and the state of all the clients.
- List all the clients who are located in Mumbai
- Find the names of client who carries a balance of 15,000/-
- List the details of the clients whose name starts with 'M'.
- List the details of clients who are staying in either Maharashtra or in Karnataka.

3. Exercise on updating records in a table:

- Change the city of clientno 'C00005' to Bangalore.
- Change the balance due of clientno 'C00001' to Rs. 1000.
- Change the city of the client to Pune

4. Exercise on deleting records in a table:

- Delete all clients from the CLIENT\_MASTER.
- Delete from CLIENT\_MASTER where the column state holds the value 'Tamil Nadu'.

## Assignment - 2

INSERT INTO CLIENT\_MASTER (Client\_No, Name, City, Pincode, State, Bal\_Due) VALUES ('C00001', 'IVAN Bayross', 'Mumbai', '400054', 'Maharashtra', 15000);

INSERT INTO CLIENT\_MASTER (ClientNo., Name, City, Pincode, State, Bal-Due) VALUES ('C00002', 'MAMTA MAZUMDAR', 'Madras', '280001', 'Tamil Nadu', 0);

INSERT INTO CLIENT\_MASTER (Client-No., Name, City, Pincode, state, Bal-Due) VALUES ('C00003', 'Chhaya Bantekar', 'Mumbai', '400057', 'Maharashtra', 5000);

SELECT NAME FROM CLIENT\_MASTER;

SELECT \* FROM CLIENT\_MASTER;

SELECT NAME, CITY, STATE FROM CLIENT\_MASTER;

SELECT \* FROM CLIENT\_MASTER WHERE CITY = 'Mumbai';

SELECT \* FROM CLIENT\_MASTER WHERE Bal-Due = 15000;

SELECT \* FROM CLIENT\_MASTER WHERE NAME LIKE 'M.%';

SELECT \* FROM CLIENT\_MASTER WHERE STATE IN ('Maharashtra', 'Karnataka');

UPDATE CLIENT\_MASTER SET CITY = 'Bangalore' WHERE ClientNo = 'C00005';

UPDATE CLIENT\_MASTER SET Bal-Due = 1000 WHERE Client-No = 'C00001';

UPDATE CLIENT\_MASTER SET city = 'Pune' WHERE Client-No = 'C00001';

DELETE FROM CLIENT-MASTER WHERE STATE = 'Tami Madu';  
DELETE FROM CLIENT-MASTER;

✓ ✓

**LAB ASSIGNMENT 3**  
**DBMS Lab (KCA – 252)**

Table Name: **PRODUCT\_MASTER**

Description: Used to store product information.

Column Name	Data Type	Size
<b>PRODUCTNO</b>	VARCHAR2	6
<b>DESCRIPTION</b>	VARCHAR2	15
<b>PROFITPERCENT</b>	NUMBER	4,2
<b>UNITMEASURE</b>	VARCHAR2	10
<b>QTYONHAND</b>	NUMBER	8
<b>REORDERLVL</b>	NUMBER	8,2
<b>SELLPRICE</b>	NUMBER	8,2
<b>COSTPRICE</b>	NUMBER	8,2

Data for **PRODUCT\_MASTER** table:

PRODUCTNO	DESCRIPTION	PROFIT	UNIT	QTYON	REORDER	SELL	COST
		PERCENT	MEASURE	HAND	LVL	PRICE	PRICE
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	500
P07868	Keyboards	2	Piece	150	50	850	550
P07885	CDDrive	2.5	Piece	80	30	700	450
P07965	540 HDD	4	Piece	100	40	350	250
P07975	1.44Drive	5	Piece	70	30	300	175
P08865	1.22Drive	5	Piece	75	30	450	300

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 5000.
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 selling 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' and 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.

### Assignment -3

INSERT INTO PRODUCT\_MASTER (ProductNo, Description, Profitpercentage, UnitMeasure, Qtyonhand, SellPrice, Reorderlevel, CostPrice)  
VALUES ('P00001', '1.44floppies', 5, 'Piece', 200, 350, 50, 250);

INSERT INTO PRODUCT\_MASTER (ProductNo, Description, profitpercentage, UnitMeasure, Qtyonhand, SellPrice, Reorderlevel, CostPrice) VALUES  
('P03453', 'Monitors', 6, 'Piece', 150, 500, 50, 350);

INSERT INTO PRODUCT\_MASTER (ProductNo, Description, profitpercentage, UnitMeasure, Qtyonhand, SellPrice, Reorderlevel, CostPrice) VALUES  
('P06743', 'Mouse', 5, 'Piece', 100, 600, 20, 450);

SELECT NAME FROM CLIENT\_MASTER;

SELECT NAME, CITY FROM CLIENT\_MASTER;

SELECT DESCRIPTION "Available Products" FROM PRODUCT\_MASTER;

SELECT \* FROM CLIENT\_MASTER WHERE CITY = 'Mumbai';

SELECT \* FROM CLIENT\_MASTER WHERE CLIENT\_NO IN ('C00001',  
'C00002');

SELECT \* FROM PRODUCT\_MASTER WHERE DESCRIPTION IN ('1.44Drive',  
'1.22Drive');

SELECT \* FROM PRODUCT\_MASTER WHERE SELLPRICE > 500; AND

SELECT \* FROM CLIENT\_MASTER WHERE CITY IN ('Mumbai', 'Delhi',  
'Madras');

SELECT \* FROM PRODUCT\_MASTER WHERE SELLPRICE > 200 AND

SELLPRICE <= 600;

(3)

SELECT NAME, CITY, STATE FROM CLIENT-MASTER WHERE NOT STATE  
= 'Maharashtra';  
UPDATE PRODUCT-MASTER SET SellPrice = 1150 WHERE Description  
= '1.44 floppies';  
DELETE FROM CLIENT-MASTER WHERE CLIENT\_NO = 'C00001';  
SELECT ProductNo, Description, SellPrice \* 15 AS "Original selling  
Price" FROM PRODUCT-MASTER WHERE SellPrice > 150;  
SELECT NAME, CITY FROM CLIENT-MASTER WHERE CITY  
LIKE '% -a%.';  
SELECT NAME FROM CLIENT-MASTER WHERE NAME LIKE '% -a%.';  
SELECT \* FROM PRODUCT-MASTER ORDER BY Description;  
SELECT COUNT (Description) "Total Product" FROM PRODUCT-MASTER;  
SELECT AVG (costprice) AS Average-Price FROM PRODUCT-MASTER;  
SELECT MIN (costprice) AS Minimum-Price FROM PRODUCT-MASTER;  
SELECT MIN (costprice) AS min-price, MAX (costprice) AS max-price  
FROM PRODUCT-MASTER;  
SELECT COUNT (Description) FROM PRODUCT-MASTER WHERE  
costprice > 150;  
SELECT Description FROM PRODUCT-MASTER WHERE ORDER BY  
sellprice ASC;  
SELECT Description FROM PRODUCT-MASTER ORDER BY sellprice DESC;

## 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 result 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 column name showing annual salary should be "Annual Salary".
8. List employee details with gross salary if every employee is getting a commission of Rs. 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'.

- 11. List the employees whose first name starts with 'Ro'
- 12. List the employees whose first name must be 5 character long and starts with 'R'.
- 13. List the employees whose first name where the 2<sup>nd</sup> character must be 'o'.
- 14. List the employees who are married.
- 15. List the employees who are unmarried.
- 16. List the employees whose marriage status is not known.
- 17. List the employees who belongs to MCA department and are married

## Assignment - 4

CREATE TABLE EMPLOYEE (ID NUMBER PRIMARY KEY NOT NULL,  
F-NAME VARCHAR2(20), L-NAME VARCHAR2(20), DEPT VARCHAR2(20),  
SALARY NUMBER (15,2), DOJ VARCHAR2(20), ADDRESS VARCHAR(20),  
MARRIED CHAR);

INSERT INTO EMPLOYEE (ID, F-NAME, L-NAME, DEPT, SALARY,  
DOJ, ADDRESS, MARRIED) VALUES (1, 'VINAY', 'KUMAR', 'MCA',  
25000, '27-09-2001', 'Ghaziabad', 'Y');

INSERT INTO EMPLOYEE (ID, F-NAME, L-NAME, DEPT, SALARY, DOJ,  
ADDRESS, MARRIED) VALUES (2, 'SOMAN', 'VERMA', 'MCA', 15000,  
'17-10-2006', 'Meerut', 'Y');

INSERT INTO EMPLOYEE (ID, F-NAME, L-NAME, DEPT, SALARY,  
DOJ, ADDRESS, MARRIED) VALUES (3, 'Akash', 'singh', 'CS', 20000,  
'15-01-2005', 'KANPUR', 'Y');

SELECT \* FROM EMPLOYEE WHERE ADDRESS = 'Ghaziabad';

SELECT DISTINCT DEPT FROM EMPLOYEE;

SELECT ID, F-NAME || ' ' || L-NAME AS "FULLNAME", DEPT FROM  
EMPLOYEE;

SELECT INITCAP(F-NAME) || ' is working in' || UPPER(DEPT) ||  
'department' AS DETAILS FROM EMPLOYEE;

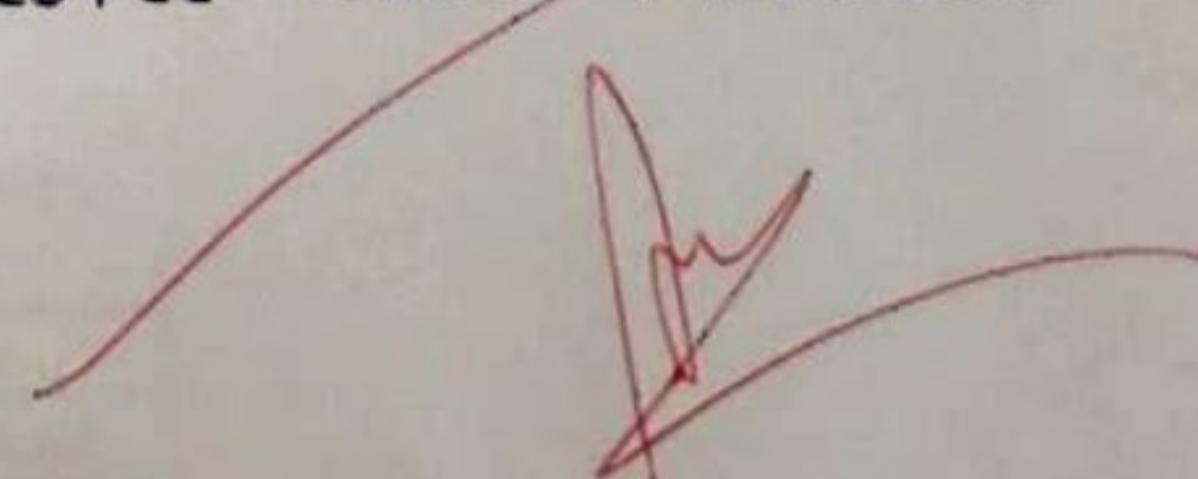
SELECT F-NAME || ' ' || L-NAME AS NAME FROM EMPLOYEE WHERE  
DOJ > '15-01-2005';

UPDATE EMPLOYEE SET SALARY = 25000 WHERE ID = 2;

SELECT ID, F-NAME || ' ' || L-NAME AS Name, SALARY \* 12 AS  
"Annual Salary" FROM EMPLOYEE;

⑨

SELECT \*, SALARY + 300 AS "Gross Salary" FROM EMPLOYEE;  
SELECT \* FROM EMPLOYEE WHERE SALARY BETWEEN 15000 AND 30000;  
  
SELECT \* FROM EMPLOYEE WHERE DOJ BETWEEN TO\_DATE('27-09-2001', 'DD-MM-YYYY') AND TO\_DATE('12-02-2019', 'DD-MM-YYYY');  
  
SELECT \* FROM EMPLOYEE WHERE DEPT = 'MCA' OR DEPT = 'CS';  
SELECT \* FROM EMPLOYEE WHERE DEPT != 'MCA';  
SELECT \* FROM EMPLOYEE WHERE F\_NAME LIKE 'R%';  
SELECT \* FROM EMPLOYEE WHERE F\_NAME LIKE 'RO%';  
SELECT \* FROM EMPLOYEE WHERE F\_NAME LIKE 'R---';  
SELECT \* FROM EMPLOYEE WHERE F\_NAME LIKE '\_O%';  
SELECT \* FROM EMPLOYEE WHERE MARRIED = 'Y';  
SELECT \* FROM EMPLOYEE WHERE MARRIED = 'N';  
SELECT \* FROM EMPLOYEE WHERE MARRIED IS NULL;  
SELECT \* FROM EMPLOYEE WHERE MARRIED = 'Y' AND DEPT = 'MCA';



## LAB ASSIGNMENT 5

### DBMS Lab (KCA – 252)

21. Demonstrate the use of all string functions available in SQL.
22. 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)

23. Solve the following query

Empcode	Empname
E1	001Rajkumar
E2	Ramkumar002
E3	Ravikumar003

Output should

Empcode	Empname
E1	Rajkumar
E2	Ramkumar
E3	Ravikumar

24. 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

25. 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

CREATE TABLE Students ( ROLL NUMBER, F\_NAME VARCHAR2(50), M\_NAME VARCHAR2(50), L\_NAME VARCHAR2(50), SEC CHAR(1), CITY VARCHAR2(50), AREA VARCHAR2(50), HOUSENO VARCHAR2(10), DN NUMBER );

INSERT INTO Students (ROLL, F\_NAME, M\_NAME, L\_NAME, SEC, CITY, AREA, HOUSENO, DN) VALUES (1, 'Akash', 'Kumar', 'Jain', 'A', 'Ghaziabad', 'Rakesh Marg', 'C-355', 1);

INSERT INTO Students (ROLL, F\_NAME, M\_NAME, L\_NAME, SEC, CITY, AREA, HOUSENO, DN) VALUES (2, 'MANAV', 'MOHAN', 'SHARMA', 'A', 'Ghaziabad', 'Raj Nagar', 'D-211', 2);

SELECT ROLL, F\_NAME || ' ' || M\_NAME || ' ' || L\_NAME AS "NAME"  
FROM STUDENTS;

SELECT ROLL, F\_NAME || ' ' || M\_NAME || ' ' || L\_NAME AS "NAME",  
HOUSENO || ' ' || AREA || ' ' || CITY AS "ADDRESS" FROM STUDENTS;

SELECT ROLL, SUBSTR(F\_NAME, 1, 1) || '.' || SUBSTR(M\_NAME, 1, 1)  
|| ' ' || L\_NAME AS "NAME" FROM STUDENTS;

SELECT \* FROM STUDENTS ORDER BY SEC ASC;

SELECT \* FROM STUDENTS ORDER BY SEC DESC;

SELECT \* FROM STUDENTS ORDER BY SEC, F\_NAME;

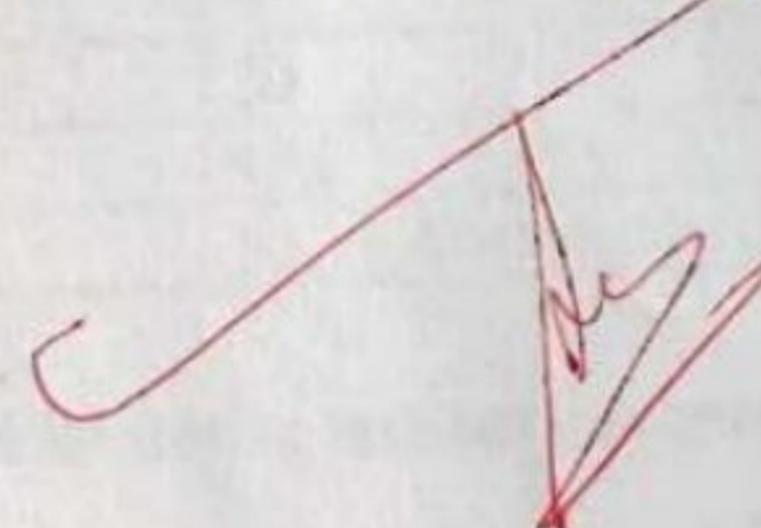
③

```
SELECT * FROM STUDENTS WHERE DIV IS NULL;
```

```
SELECT CITY FROM STUDENT STUDENTS;
```

```
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;
```

```
SELECT ROLL, F.NAME, CASE WHEN SEC = 'A' THEN '1' WHEN  
SEC = 'B' THEN '2' END AS SEC FROM STUDENTS;
```



## 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 with 'O'
Order Date	Date		Not Null
Client_No	Varchar2	6	Foreign Key reference client no of Client Master table
Dely_add	Varchar2	25	
Salesman_No	Varchar2	6	Foreign Key references Salesman_No of 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)'; 'Fulfilled(F)'; 'Canceled(C)')

**v) SALES ORDER DETAILS**

Column Name	Data Type	Size	Attribute
Order_No	Varchar	6	Primary Key/Foreign Key references Order No of Sales Order
Product_No	Varchar2	6	Primary Key/Foreign Key references Product No of Product Master
Qty_Order	Number	8	
Qty_disp	Number	8	

**Data for CLIENT MASTER**

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

**Data for PRODUCT\_MASTER**

Product_No	Name	Unit Measure	Qty OnHand	Reorder Level	Sell Price	Cost Price
P00001	T-Shirt	Piece	200	50	350	250
P03453	Shirt	Piece	150	50	500	350
P06734	Cotton Jeans	Piece	100	20	600	450
P07865	Jeans	Piece	100	20	750	500
P07868	Trouser	Piece	150	50	850	550
P07885	Pull Over	Piece	80	30	700	450
P07965	Denim Shirts	Piece	100	40	350	250
P07975	Lycra Tops	Piece	70	30	300	175
P08865	Skirts	Piece	75	30	450	300

**Data For SALESMA\_N\_MASTER**

Salesman_no	Name	Address	City	PIN	Stat e	Sala mt	Tgt _To_ge t	Ytd_Sal e	Remark
S00001	Kiran	A/14 worli	Mumbai	400002	Mah	3000	100	50	Good
S00002	Manish	65,narima n	Mumbai	400001	Mah	3000	200	100	Good
S00003	Ravi	P-7 Bandra	Mumbai	400032	Mah	3000	200	100	Good
S00004	Ashish	A/5 Juhu Bombay	Mumbai	400044	Mah	3500	200	150	Good

**Data for SALES\_ORDER table:**

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

**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
O46866	P06734	1	1
O46866	P07965	1	0
O10008	P07975	1	0
O10008	P00001	10	5

## Assignment - 6

CREATE TABLE CLIENT-MASTER ( client-No VARCHAR2(6) PRIMARY KEY, NAME VARCHAR2(20) NOT NULL, CITY VARCHAR2(20) CHECK (CITY IN ('Delhi', 'Mumbai', 'Chennai')), Pincode NUMBER(6), STATE VARCHAR2(20), BalDue NUMBER(10,2), Email VARCHAR2(30) UNIQUE);

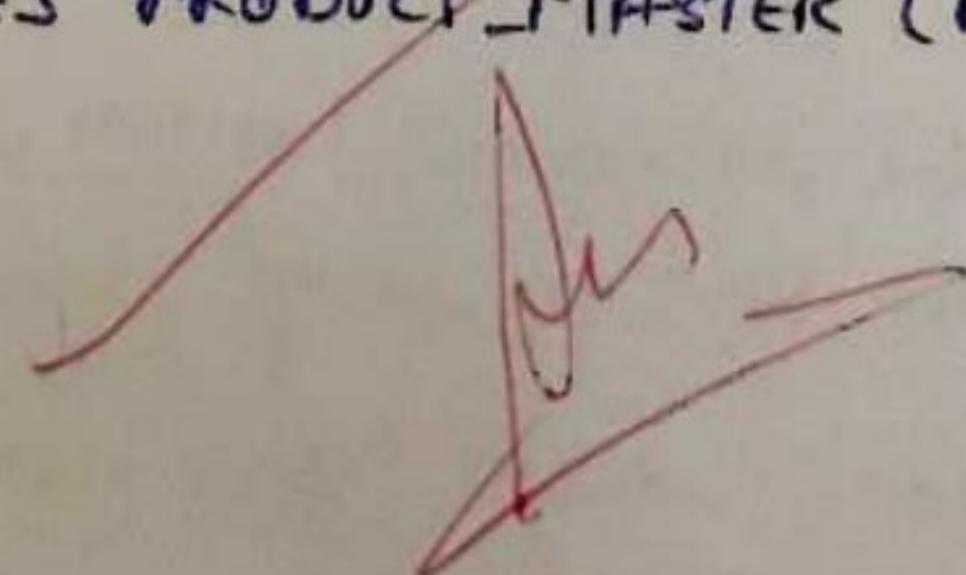
CREATE TABLE PRODUCT-MASTER (Product-No VARCHAR2(6) PRIMARY KEY, Name VARCHAR2(20) NOTNULL, UNIT MEASURE VARCHAR2(10) NOTNULL, Qtyonhand NUMBER(8) NOTNULL, Reorderlevel NUMBER(8) NOT NULL, SellPrice NUMBER(8,2) NOT NULL CHECK (SellPrice > 0), CostPrice NUMBER(8,2) NOTNULL CHECK (CostPrice > 0), CHECK (Qtyonhand >= Reorderlevel);

CREATE TABLE SALESMAN-MASTER (SalesmanNo VARCHAR2(6) PRIMARY KEY CHECK (SalesmanNo like 'S%.'), Sal-name VARCHAR2(20) NOT NULL, Address VARCHAR2(40) NOT NULL, City VARCHAR2(20), State VARCHAR2(20), Pincode NUMBER(6), Sal-Amount NUMBER(8,2) NOTNULL CHECK (Sal-Amount > 0), Tgt-to-get NUMBER(8,2) NOTNULL CHECK (Tgt-to-get > 0), Ytd-Sales NUMBER(8,2) NOTNULL CHECK (Ytd-Sales > 0), Remarks VARCHAR2(30));

⑥

```
CREATE TABLE SALES_ORDER ( Order_No VARCHAR2(6)
    PRIMARY KEY CHECK (Order_No LIKE 'O%'), Order_Date DATE
    NOT NULL, Client_No VARCHAR2(6), Dely_addr VARCHAR2(25),
    Salesman_No VARCHAR2(6), Dely_type CHAR(1) DEFAULT 'P'
    CHECK (Dely_type IN ('P', 'F')), Billed_yn CHAR(1) CHECK
    (Billed_yn IN ('Y', 'N')), Dely_Date DATE, Order_Status
    VARCHAR2(10) CHECK (OrderStatus IN ('In Process (IP)', 'Fulfilled
    (F)', 'Cancelled (C)'), FOREIGN KEY (Client_No) REFERENCES
    CLIENT_MASTER (Client_No), FOREIGN KEY (Salesman_No)
    REFERENCES SALESMAN_MASTER (Salesman_No), CHECK (Dely_Date
    >= OrderDate));
```

```
CREATE TABLE SALES_ORDER_DETAILS ( Order_No VARCHAR2(6),
    Product_No VARCHAR2(6), Qty_Order NUMBER(8), Qty_Dispatched
    NUMBER(8), PRIMARY KEY (Order_No, Product_No), FOREIGN KEY
    (Order_No) REFERENCES SALES_ORDER (Order_No), FOREIGN KEY
    (Product_No) REFERENCES PRODUCT_MASTER (Product_No));
```



## ⑥ Data for CLIENT-MASTER

```
INSERT INTO CLIENT-MASTER (client-No, Name, CITY, Pincode, State,  
Balduo, Email) Values ('C00001', 'Pankaj Sharma', 'Delhi', 400054,  
'Delhi', 15000, 'pk@gmail.com');
```

```
INSERT INTO CLIENT-MASTER (client-No, Name, CITY, Pincode, State,  
Balduo, Email) values ('C00002', 'Yogesh Sharma', 'Delhi', 780001,  
'Delhi', 0, 'ys@gmail.com');
```

## Data for Product-MASTER

### PRODUCT-MASTER

```
INSERT INTO PRODUCT-MASTER (Product-No, Name, UnitMeasure, Qtyonhand, Reorderlevel,  
SellPrice, CostPrice) values ('P00001', 'T-shirt', 'Piece', 200, 50,  
350, 250);
```

```
INSERT INTO PRODUCT-MASTER (Product-No, Name, UnitMeasure, Qtyonhand,  
Reorderlevel, SellPrice, CostPrice) VALUES ('P03453', 'Shirt', 'Piece', 150,  
50, 500, 350);
```

## Data for Salesman-Master

```
INSERT INTO SALESMAN-MASTER (Salesman-No, Sal-name, Address, city,  
Pincode, State, Sal-Amount, Tgt-to-get, ytd-Sales, Remarks) VALUES  
('S00001', 'Kiran', 'A114 worti', 'Mumbai', 400002, 'Mah', 3000, 100,  
50, 'good');
```

```
INSERT INTO SALESMAN-MASTER (Salesman-No, Sal-name, Address, city,  
Pincode, State, Sal-Amount, Tgt-to-get, ytd-Sales, Remarks) VALUES ('S00002',  
'Manish', '65, naman', 'Mumbai', 400001, 'Mah', 3000, 200, 100, 'good');
```

⑥

## Data for Sales-Order Table

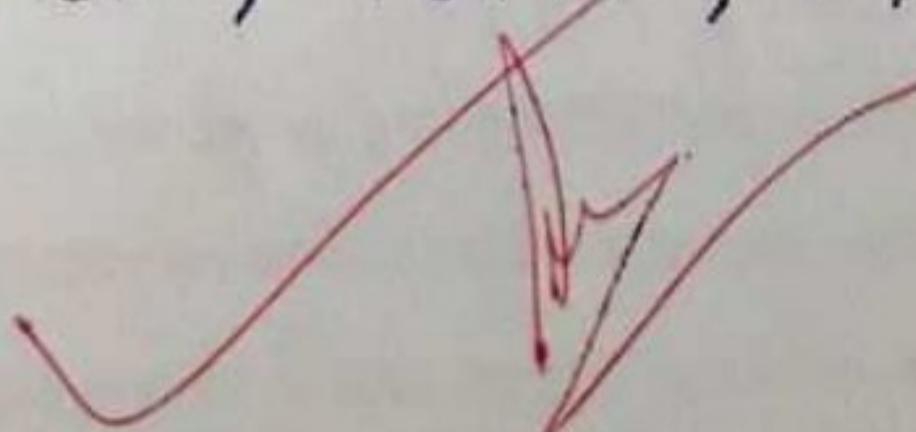
```
INSERT INTO SALES-ORDER (Order-No, Order-Date, Client-No, Dely-type,  
Billed-yn, Salesman-No, Dely-Date, Order-Status) VALUES ('019001',  
TO-DATE('12-Jan-21', 'DD-MON-YY'), 'C00001', 'F', 'N', 'S00001',  
TO-DATE('20-Jan-21', 'DD-MON-YY'), 'IP');
```

```
INSERT INTO SALES-ORDER (Order-No, Order-Date, Client-No, Dely-type,  
Billed-yn, Salesman-No, Dely-Date, Order-Status) VALUES ('019002',  
TO-DATE('25-Jan-21', 'DD-MON-YY'), 'C00002', 'P', 'N', 'S00002',  
TO-DATE('27-Jan-21', 'DD-MON-YY'), 'C');
```

## DATA for sales\_order Details

```
INSERT INTO SALES-ORDER DETAILS (Order-No, Product-No, Qty-Order,  
Qty-disp) VALUES ('019001', 'P00001', 4, 4);
```

```
INSERT INTO SALES-ORDER DETAILS (Order-No, Product-No, Qty-Order,  
Qty-disp) VALUES ('019002', 'P07965', 2, 1);
```



**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
Empld	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 3<sup>rd</sup> January 1985)

# Assignment - 7

SELECT SYSDATE FROM dual;

- SELECT ADD\_MONTHS(SYSDATE, 3) AS "MONTH" FROM dual;
- SELECT LAST\_DAY(SYSDATE) FROM dual;
- SELECT MONTHS\_BETWEEN(SYSDATE, TO\_DATE('2023-01-01', 'YYYY-MM-DD')) AS "MONTHS" FROM dual;
- SELECT ROUND(SYSDATE, 'MONTH') AS "ROUNDED DATE" FROM dual;

CREATE TABLE EMPLOYEE

EMPID VARCHAR2(3) PRIMARY KEY CHECK (EMPID LIKE 'E%'),  
ENAME VARCHAR2(25) NOT NULL,  
DOB DATE NOTNULL,  
DOJ DATE NOT NULL CHECK (DOJ > DOB),  
SALARY NUMBER(7) NOT NULL CHECK (SALARY > 0),  
ADHAR NUMBER(12) UNIQUE CHECK (LENGTH(ADHAR)=12)

);

SELECT \* FROM EMPLOYEE WHERE DOJ = TO\_DATE('2015-03-15', 'YYYY-MM-DD');

SELECT ENAME, TO\_CHAR(DOJ, 'Month DD, YYYY') AS Hiredate  
FROM EMPLOYEE;

SELECT \* FROM EMPLOYEE WHERE DOJ < TO\_DATE('2018-01-01', 'YYYY-MM-DD');

SELECT \* FROM EMPLOYEE WHERE TO\_CHAR(DOJ, 'MM')='01';

SELECT \* FROM EMPLOYEE WHERE TO\_CHAR(DOJ, 'YYYY')='2018';

SELECT \* FROM EMPLOYEE ORDER BY DOJ DESC;

SELECT Ename , DOJ, ROUND(MONTHS\_BETWEEN(SYSDATE, DOJ)

(12,1) AS Experience FROM EMPLOYEE;

SELECT \* FROM EMPLOYEE WHERE MOD(Salary, 2) = 1;

SELECT \* FROM EMPLOYEE WHERE (Empld = 'E01' OR Empld = 'E01')

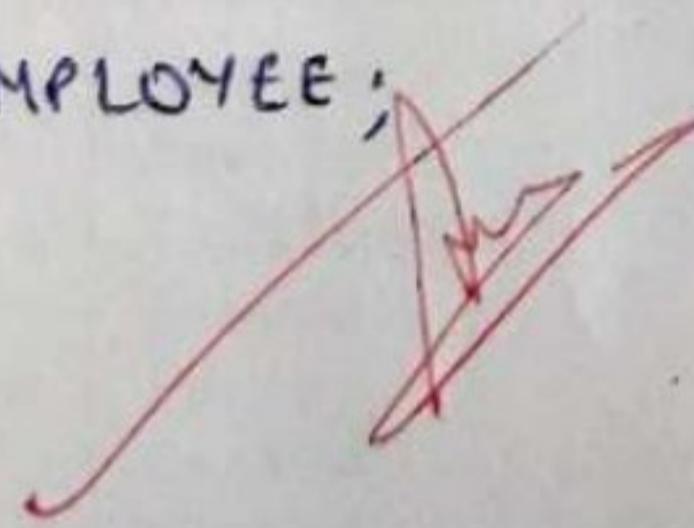
AND TO\_CHAR(DOJ, 'YYYY') = '1991';

SELECT \* FROM EMPLOYEE WHERE TO\_CHAR(DOJ, 'MM') <> '02';

SELECT 'Happy Birthday' || Ename AS MESSAGE FROM EMPLOYEE  
WHERE TO\_CHAR(DOB, 'MM-DD') = TO\_CHAR(SYSDATE, 'MM-DD');

SELECT Empld, Ename, TO\_CHAR(DOB, 'DDth Month YYYY')

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, Order.**

**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	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

**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 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 name 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 their 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 for their jobs who does not live in the same city where their customer lives, and gets a commission is above 12%.
6. Write a SQL statement to find the details of a order i.e. order number, order date, amount of order, which customer gives the order and which salesman works for that customer and how much commission he gets for an order.
7. Write a SQL statement to make a list in ascending order for the customer who works either through a salesman or by own.
8. Write a SQL statement to make a list in ascending order for the customer who holds a 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 number, order date, and order amount in ascending order according to the order date to find that either 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 number, order date, order amount salesman name and commission to find that either any of the existing customers have placed no order or placed one or more orders by their saiesman or by own.
11. Write a SQL statement to make a list in ascending order for the salesmen who works either 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 one or more customer or not yet join under any of the customers who placed either one or more orders or no order to their supplier.
13. 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.
14. Write a SQL statement to make a cartesian product between salesman and customer i.e. each salesman will appear for all customer and vice versa.

**Note – For questions 15 to 19 use sample table company\_mast and item\_mast**  
**Sample table: company\_mast**

COM_ID	COM_NAME
11	Samsung
12	iBall
13	Epsilon
14	Zebronics
15	Asus
16	Frontech

**Sample table: item\_mast**

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200	15
102	Key Board	450	16
103	ZIP drive	250	14
104	Speaker	550	16
105	Monitor	5000	11
106	DVD drive	900	12
107	CD drive	800	12
108	Printer	2600	13
109	Refill cartridge	350	13
110	Mouse	250	12

15. Write a SQL query to display all the data from the item\_mast, including all the data for each item's producer company.
16. Write a SQL query to display the item name, price, and company name of all the products.
17. Write a SQL query to display the average price of items of each company, showing the name of the company.
18. Write a SQL query to display the names of the company whose products have an average price larger than or equal to Rs. 350.
19. Write a SQL query to display the name of each company along with the ID and price for their most expensive product.

Note – For questions 20 to 23 use the sample tables emp\_deptmen, emp\_details.

**Sample table: emp\_deptmen**

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

**Sample table: 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

20. Write a query in SQL to display all the data of employees including their department.
21. Write a query in SQL to display the first name and last name of each employee, along with the name and sanction amount for their department.
22. Write a query in SQL to find the first name and last name of employees working for departments with a budget more than Rs. 50000.
23. Write a query in SQL to find the names of departments where more than two employees are working

## Assignment - 8

```
SELECT s.name, c.cust_name, s.city FROM Salesman S JOIN  
Customer C ON s.city = c.city AND s.Salesman_id =  
c.Salesman_id;
```

```
SELECT o.ord_no, o.purch_amt, c.cust_name, c.city  
FROM Order O JOIN Customer C ON o.customer_id =  
c.customer_id WHERE o.purch_amt BETWEEN 500 AND 2000;
```

```
SELECT s.name AS Salesman_name, c.cust_name FROM  
Salesman S JOIN Customer C ON s.Salesman_id = c.Salesman_id
```

```
SELECT c.cust_name FROM Customer C JOIN Salesman S ON  
c.Salesman_id = s.Salesman_id WHERE s.commission > 0.12;
```

```
SELECT c.cust_name FROM Customer C JOIN Salesman S ON  
c.Salesman_id = s.Salesman_id WHERE s.city <> c.city AND  
s.commission > 0.12
```

```
SELECT o.order_no, o.ord_date, o.purch_amt, c.cust_name,  
s.name, s.commission FROM Order O JOIN Customer C ON  
o.customer_id = c.customer_id JOIN Salesman S ON  
o.Salesman_id = s.Salesman_id;
```

```
SELECT c.cust_name FROM Customer C LEFT JOIN Order O  
ON c.customer_id = o.customer_id ORDER BY c.cust_name ASC;
```

8  
SELECT c.cust-name FROM customer c LEFT JOIN orders o  
ON c.customer-id = o.customer-id WHERE c.grade < 300  
ORDER BY c.cust-name ASC;

SELECT c.cust-name, c.city, o.ord-no, o.ord-date, o.purch-  
amt FROM customer c LEFT JOIN orders o ON c.customer-  
id = o.customer-id ORDER BY o.ord-date ASC;

SELECT c.cust-name, c.city, o.ord-no, o.ord-date, o.Purch-amt,  
s.name, s.commission FROM customer c LEFT JOIN orders o  
ON c.customer-id = o.customer-id ORDER BY o.order-date ASC;

SELECT s.name FROM Salesman s LEFT JOIN customer c ON  
s.Salesman-id = c.Salesman-id ORDER BY s.name ASC;

SELECT s.name, COUNT(c.customer-id), COUNT(o.order-no)  
FROM Salesman s LEFT JOIN customer c ON s.Salesman-id =  
c.Salesman-id LEFT JOIN orders o ON c.customer-id =  
o.customer-id GROUP BY s.name ORDER BY s.name ASC;

SELECT s.name, COUNT(DISTINCT c.customer-id), COUNT  
(o.ord-no) FROM Salesman s LEFT JOIN customer c ON  
s.Salesman-id = c.Salesman-id WHERE c.grade IS NOT  
NULL OR o.ord-no. IS NULL GROUP BY s.name ORDER  
BY s.name ASC;

SELECT s.name AS Salesman-name, c.cust-name FROM Salesman  
S CROSS JOIN Customer C;

SELECT i.\* , c.\* FROM item-mast i JOIN COMPANY-mast c ON  
i.PRO-COM = c.COM-ID;

SELECT i.PRO-NAME , i.PRO-PRICE , c.COM-NAME FROM item-mast i  
JOIN Company-mast c ON i.PRO-COM = c.COM-ID;

SELECT c.COM-NAME , AVG(i.PRO-PRICE) AS avg-Price FROM item-mast  
i JOIN Company-mast c ON i.PRO-COM = c.COM-ID GROUP BY  
c.COM-NAME;

SELECT c.COM-NAME FROM item-mast i JOIN Company-mast c ON  
i.PRO-COM = c.COM-ID GROUP BY c.COM-NAME HAVING AVG  
(i.PRO-PRICE) >= 350;

SELECT c.COM-NAME , i.PRO-ID , MAX(i.PRO-PRICE) FROM item-mast i  
JOIN Company-mast c ON i.PRO-COM = c.COM-ID GROUP BY  
c.COM-Name , i.PRO-ID ORDER BY Max-Price DESC;

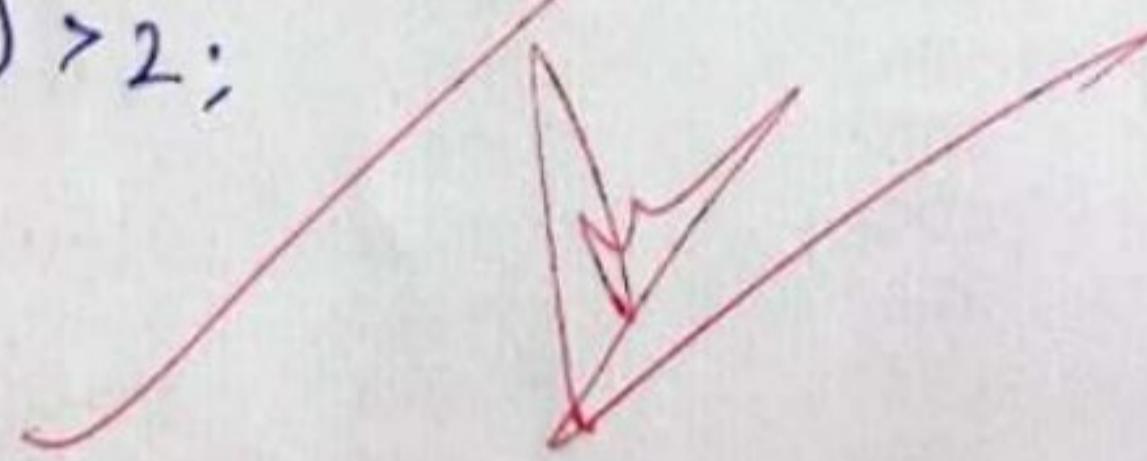
~~SELECT e.\* , d.\* FROM emp-details e JOIN emp-department d ON  
e.emp-dept = d.dept code;~~

~~SELECT e.emp-fname , e.emp-lname , d.dept-name , d.dpt-allotment FROM emp-details e JOIN emp-department d ON  
e.EMP-DEPT = d.DEPT-Code;~~

8

```
SELECT e.EMP-FNAME , e.emp-LNAME FROM emp-details e  
JOIN emp-department d ON e.EMP-DEPT = d.DEP-T-CODE  
WHERE d.DPT-ALLOTMENT > 50000;
```

```
SELECT d.DEP-T-NAME FROM emp-details e JOIN EMP-department  
d ON e.EMP-DEPT = d.DEP-T-CODE group BY d.DPT-NAME  
HAVING COUNT(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 purchase amount.
2. write a SQL query to calculate average purchase amount of all orders. Return average 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. Return customer ID, maximum purchase amount.
7. write a SQL query to find the highest purchase amount ordered by each customer on a 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 each salesperson. Return salesperson ID, purchase amount.
9. write a SQL query to find highest order (purchase) amount by each customer in a particular order date. Filter the result by highest order (purchase) amount above 2000.00. Return customer id, order date and maximum purchase amount.
10. write a SQL query to find the maximum order (purchase) amount in the range 2000, 6000 (Begin and end values are included.) by combination of each customer and order date. Return customer id, order date and maximum purchase amount.
11. write a SQL query to find the maximum order (purchase) amount by each customer. The customer ID should be in the range 3002 and 3007(Begin and end values are included.). Return customer id and maximum purchase amount.
12. write a SQL query to count all the orders generated on '2012-08-17'. Return number of orders

**Sample table: 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

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 gradation for his/her activity.

15. write a SQL query to find the highest grade of the customers for each of the city. Return 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. Return number of salespeople.

**Sample table: item\_mast**

PRO_ID	PRO_NAME	PRO_PRICE	PRO_COM
101	Mother Board	3200	15
102	Key Board	450	16
103	ZIP drive	250	14
104	Speaker	550	16
105	Monitor	5000	11
106	DVD drive	900	12
107	CD drive	800	12
108	Printer	2600	13
109	Refill cartridge	350	13
110	Mouse	250	12

17. write a SQL query to count number of products where product price is higher than or equal 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 1<sup>st</sup> 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 Department.
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.

## Assignment - 9

SELECT SUM (Purch-amt) FROM ORDERS;

SELECT AVG (Purch-amt) FROM ORDERS;

SELECT COUNT (DISTINCT Salesman-id) FROM ORDERS;

SELECT MAX (Purch-amt) FROM ORDERS;

SELECT MIN (Purch-amt) FROM ORDERS;

SELECT customer\_id , MAX (Purch-amt) FROM ORDERS GROUP BY customer\_id;

SELECT customer\_id , ord-date , MAX (Purch-amt) FROM ORDERS GROUP BY customer\_id , ord-date;

SELECT Salesman-id , MAX (Purch-amt) FROM ORDERS  
WHERE ord-date = '2012-08-17' GROUP BY Salesman-id;

SELECT customer\_id , ord-date , MAX (Purch-amt) FROM orders GROUP BY customer\_id , ord-date Having MAX (Purch-amt) > 2000;

SELECT customer\_id , ord-date , MAX (Purch-amt) FROM ORDERS WHERE Purch-amt BETWEEN 2000 and 6000  
GROUP BY customer\_id , ord-date;

SELECT customer\_id , MAX (Purch-amt) FROM ORDERS WHERE customer\_id BETWEEN 3002 and 3007 GROUP BY customer\_id;

9  
SELECT COUNT(\*) FROM ORDERS WHERE ord\_date = '2012-08-17';

SELECT COUNT(\*) FROM CUSTOMER;

SELECT COUNT(\*) FROM CUSTOMER WHERE grade IS NOT NULL;

SELECT city, MAX(grade) FROM CUSTOMER GROUP BY city;

SELECT city, COUNT(\*) FROM Salesman GROUP BY city;

SELECT COUNT(\*) FROM item\_mart WHERE PRO\_PRICE >= 350;

SELECT MAX(DOJ) FROM EMPLOYEE;

SELECT MIN(DOJ) FROM EMPLOYEE;

SELECT COUNT(DISTINCT DEPT) FROM EMPLOYEE;

SELECT COUNT(\*) FROM EMPLOYEE WHERE DEPT = 'MCA';

SELECT DEPT, MAX(AVG(Salary)) FROM EMPLOYEE GROUP BY Dept;

SELECT COUNT(\*) FROM EMPLOYEE;

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
Empld		Number
Name		Varchar2
DOJ		Date
JobId		Number
Salary		Number

Empld	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 more salary than the employee whose id is 104.
- 1.2 Write a query to display the name, salary, department id, job id for those employees who works in the same designation as the employee works whose id is 103.
- 1.3 Write a query to display the name, salary, department id for those employees who earn such amount of salary which is the smallest salary of any of the departments.
- 1.4 Write a query to display the employee details who are having the same date of birth as of employee having id 106.
- 1.5 Write a query to display the employee details who are having the date of birth after the date of birth of employee having ID 102

## Assignment-10

```
CREATE TABLE EMPLOYEE (
    EMPID NUMBER PRIMARY KEY,
    NAME VARCHAR2(50),
    DOJ DATE,
    JOBID VARCHAR2(20),
    SALARY NUMBER
);
```

SELECT NAME FROM EMPLOYEE WHERE SALARY > (SELECT SALARY FROM EMPLOYEE WHERE EMPID = 104);

SELECT NAME, SALARY, JOBID FROM EMPLOYEE WHERE JOBID = (SELECT JOBID FROM EMPLOYEE WHERE EMPID = 103);

SELECT NAME, SALARY, JOBID FROM EMPLOYEE WHERE SALARY = (SELECT MIN(SALARY) FROM EMPLOYEE);

SELECT \* FROM EMPLOYEE WHERE DOJ = (SELECT DOJ FROM EMPLOYEE WHERE EMPID = 106);

SELECT \* FROM EMPLOYEE WHERE DOJ > (SELECT DOJ FROM EMPLOYEE WHERE EMPID = 102);

16/28

## Assignment - 11

Procedure

Procedure to find minimum -

Create or Replace Procedures find-min (

    num1 in number,  
    num2 in number  
    min-val out number );

IS

BEGIN

    If num1 < num2 THEN,

        min-val := num1;

    ELSE

        min-val := num2;

    End If;

END;

DECLARE

    num1 number := 10;

    num2 number := 20;

    result number;

BEGIN

    find-min (num1, num2, result);

    DBMS-OUTPUT.PUT\_LINE (result);

END;

-function

CREATE OR REPLACE FUNCTION TOTAL

RETURN NUMBER IS

TOTAL NUMBER (2) := 0;

BEGIN

SELECT COUNT \* INTO total  
FROM DUAL;

RETURN TOTAL;

END;

DECLARE

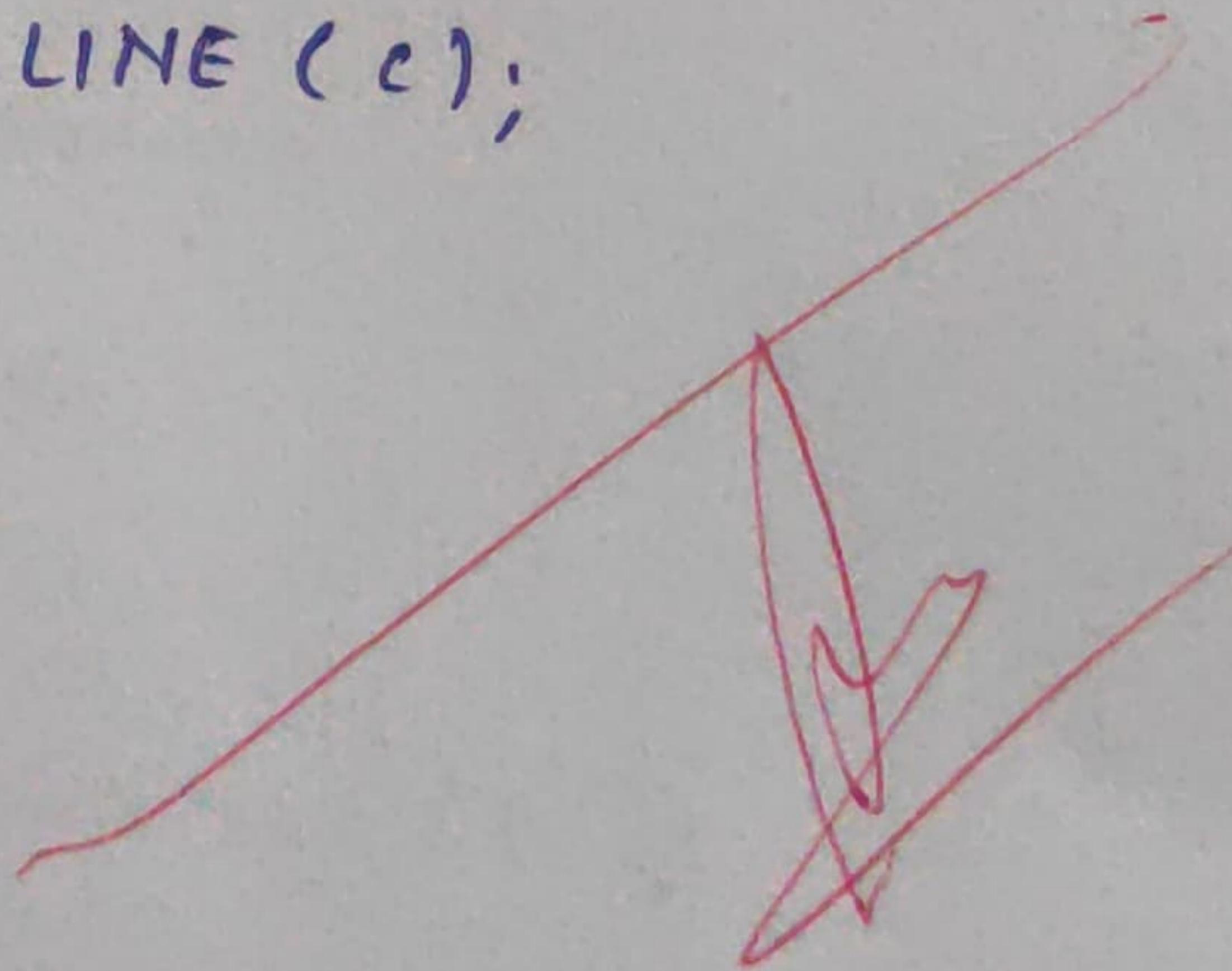
C NUMBER (2);

BEGIN

C := TOTAL();

DBMS-OUTPUT.PUT-LINE (C);

END;



## Assignment - 12

Trigger -

```
CREATE TABLE Supplier (
    sup-id number Primary key,
    sup-name Varchar2(500),
    Sub-address Varchar2(500)
);
```

```
Create TABLE Audits (
    aud-name Varchar2(100),
    trans-name Varchar2(100),
    trans-date date );
```

```
CREATE OR REPLACE trigger aut-audit-trig after
```

```
INSERT ON Suppliers
```

```
for each row
```

```
DECLARE
```

```
l.transaction Varchar2(10);
```

```
BEGIN
```

```
l.transaction := 'Insert';
```

```
Insert into audits (table-name, trans-name, trans-date )
Values ('Supplier', l.transaction, sysdate );
```

```
END;
```

INSERT INTO Supplier (sup-id, sup-name, sup-address)

Values (1, 'Book Supplier', '12 street');

Select \* from Audits;

Implicit Cursor

Declare

total-rows numbers (2);

Begin

update supplier

Set sup-id := sup-id + 500;

If SQL%NOT FOUND THEN

DBMS-OUTPUT.PUT-LINE ('NOT FOUND');

ELSE If SQL%FOUND THEN

total-rows := SQL%ROW COUNT;

DBMS-OUTPUT.PUT-LINE (total-rows || 'row(s) updated');

END IF;

END;

~~for~~