



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 2 Date of Implementation: 3. 5. 23

Q1. Implementation of DDL commands of SQL with suitable examples

Create table

Alter table

Drop Table

Create the following tables. Insert the appropriate data in these tables and solve the queries

1. Client_master(client_no, name, address1, address2, city, pincode, state, bal_due)
2. Product_master(product_no, description, profit_percent, unit_measure, qty_on_hand, reorder_lvl, sell_price, cost_price)
3. Salesman_master(salesman_name, address1, address2, city, pincode, state, .sal_amt tgt_to_get, ytd_sales, remarks)
4. Sales_order(order_no, order_date, client_no, dely_addr, salesman_no, dely_type, billed_yn, dely_date, order_status)
5. Sales_order_details(order_no, product_no, qty_ordered, qty_disp, product_rate)

Constraints are

- Client_master(Client_no is PK & first letter must start with 'C' , Name Not NULL)
- Product_master(product_no is PK & first letter must start with 'P')
- Salesman_master(salesman_no is PK & first letter must start with 'S' Name not NULL)
- Sales_Order(order_no is PK ; first letter must start with 'O', dely_type(Delivery: part(P)/full(f) Default (F), dely_date can not be less than order_date, order_status values ('In Process', 'Fulfilled' 'BackOrder', 'Cancelled'))

1. Alter table sales_order_details add column price.

2. Drop table sales_order_details

Program:

1. Creating Client_master Table Query

```
CREATE TABLE Client_master
(client_no VARCHAR(10) PRIMARY KEY CHECK (client_no LIKE 'C%') ,
name CHAR(20) NOT NULL ,
address1 VARCHAR(20) ,
address2 VARCHAR(20) ,
city CHAR(20) ,
pincode NUMBER(10) ,
state CHAR(20) ,
bal_due NUMBER(9,4));
```

DESC Client_Master

TABLE CLIENT_MASTER

Column	Null?	Type
CLIENT_NO	NOT NULL	VARCHAR2(10)
NAME	NOT NULL	CHAR(20)
ADDRESS1	-	VARCHAR2(20)
ADDRESS2	-	VARCHAR2(20)
CITY	-	CHAR(20)
PINCODE	-	NUMBER(10,0)
STATE	-	CHAR(20)
BAL_DUE	-	NUMBER(9,4)

Inserting Data into Client_master

```
INSERT INTO Client_master VALUES('C002', 'Arjun', 'b12',"", 'Madras', 780041, 'Tamilnadu', 600);
INSERT INTO Client_master VALUES('C003', 'Reena','c13',"", 'Bombay', 400057, 'Maharashtra', 3000);
INSERT INTO Client_master VALUES('C004', 'Kiran','d14',"", 'Bombay', 400056, 'Maharashtra', 0);
INSERT INTO Client_master VALUES('C005', 'Bhushan', 'e15',"", 'Delhi',100001, 'Delhi', 10000);
INSERT INTO Client_master VALUES('C006', 'Ronak','f16',"", 'Bombay', 400050, 'Maharashtra', 800);
```

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C002	Arjun	b12	-	Madras	780041	Tamilnadu	600
C003	Reena	c13	-	Bombay	400057	Maharashtra	3000
C004	Kiran	d14	-	Bombay	400056	Maharashtra	0
C005	Bhushan	e15	-	Delhi	100001	Delhi	10000
C006	Ronak	f16	-	Bombay	400050	Maharashtra	800

2. Creating Product_master Table Query

```
CREATE TABLE Product_master
(product_no VARCHAR2(6) PRIMARY KEY CHECK (product_no LIKE 'P%'),
description VARCHAR2(20),
profit_percent NUMBER(5,2),
unit_measure VARCHAR2(8),
qty_on_hand NUMBER(6),
reorder_lvl NUMBER(6),
sell_price NUMBER(10),
cost_price NUMBER(10));
```

DESC Product_master;

Table created.

TABLE PRODUCT_MASTER

Column	Null?	Type
PRODUCT_NO	NOT NULL	VARCHAR2(6)
DESCRIPTION	-	VARCHAR2(20)
PROFIT_PERCENT	-	NUMBER(5,2)
UNIT_MEASURE	-	VARCHAR2(8)
QTY_ON_HAND	-	NUMBER(6,0)
REORDER_LVL	-	NUMBER(6,0)
SELL_PRICE	-	NUMBER(10,0)
COST_PRICE	-	NUMBER(10,0)

Inserting Data into Product_master

```
INSERT INTO Product_master VALUES ('P1', 'Soaps', 5, 'piece', 100, 30, 40, 30);
INSERT INTO Product_master VALUES ('P2', 'books', 4, 'piece', 50, 50, 80, 60);
INSERT INTO Product_master VALUES ('P3', 'oil', 7, 'litres', 100, 80, 150, 110);
INSERT INTO Product_master VALUES ('P4', 'bottles', 2, 'piece', 70, 20, 50, 35);
INSERT INTO Product_master VALUES ('P5', 'dvd', 5, 'piece', 65, 30, 30, 15);
```

PRODUCT_NO	DESCRIPTION	PROFIT_PERCENT	UNIT_MEASURE	QTY_ON_HAND	REORDER_LVL	SELL_PRICE	COST_PRICE
P1	Soaps	5	piece	100	30	40	30
P2	books	4	piece	50	50	80	60
P3	oil	7	litres	100	80	150	110
P4	bottles	2	piece	70	20	50	35
P5	dvd	5	piece	65	30	30	15

3. Creating Salesman_master Table Query

```
CREATE TABLE Salesman_master
```

```
(salesman_no VARCHAR(10) PRIMARY KEY CHECK (salesman_no LIKE '%S'),  
salesman_name CHAR(20) NOT NULL,  
address1 VARCHAR(20) ,  
address2 VARCHAR(20) ,  
city CHAR(20) ,  
pincode NUMBER(10) ,  
state CHAR(20) ,  
sal_amt NUMBER(8,4) ,  
tgt_to_get NUMBER(5) ,  
ytd_sales NUMBER(10,4) ,  
remarks VARCHAR(30));
```

```
DESC Salesman_master
```

```
Table created.
```

TABLE SALESMAN_MASTER

Column	Null?	Type
SALESMAN_NO	NOT NULL	VARCHAR2(10)
SALESMAN_NAME	NOT NULL	CHAR(20)
ADDRESS1	-	VARCHAR2(20)
ADDRESS2	-	VARCHAR2(20)
CITY	-	CHAR(20)
PINCODE	-	NUMBER(10,0)
STATE	-	CHAR(20)
SAL_AMT	-	NUMBER(8,4)
TGT_TO_GET	-	NUMBER(5,0)
YTD_SALES	-	NUMBER(10,4)
REMARKS	-	VARCHAR2(30)

Inserting Data into Product_master

```
INSERT INTO Salesman_master VALUES ('S01', 'Ranveer', 'Mumbai', '', 'Mumbai', '400050', 'Maharashtra', 5000, 4000, 240000.00, 'Target achieved');
```

```
INSERT INTO Salesman_master VALUES ('S02', 'Tanvi', 'Camp', '', 'Pune', '400001', 'Maharashtra', 1000, 2000, 100000.00, 'Target not achieved');
```

```
INSERT INTO Salesman_master VALUES ('S03', 'Mahesh', '', '', 'Bhubaneshwar', '400001', 'Odisha', 750, 500, 300000.00, 'Target achieved');
```

SALESMAN_NO	SALESMAN_NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	SAL_AMT	TGT_TO_GET	YTD_SALES	REMARKS
S01	Ranveer	Mumbai	-	Mumbai	400050	Maharashtra	5000	4000	240000	Target achieved
S02	Tanvi	Camp	-	Pune	400001	Maharashtra	1000	2000	100000	Target not achieved
S03	Mahesh	-	-	Bhubaneshwar	400001	Odisha	750	500	300000	Target achieved

4. Creating Sales_order Table Query

```
CREATE TABLE Sales_order
```

```
(order_no VARCHAR(5) PRIMARY KEY CHECK (order_no LIKE 'O%') NOT NULL,  
order_date DATE,  
client_no VARCHAR(6) REFERENCES Client_master(client_no) NOT NULL,  
dely_addr VARCHAR(10),  
salesman_no VARCHAR(20) REFERENCES Salesman_master(salesman_no) NOT NULL,  
dely_type CHAR(1) DEFAULT 'F' CHECK (dely_type IN ('P', 'F')),  
billed_yn CHAR(1),  
dely_date DATE,  
order_status VARCHAR(10) CHECK (order_status IN ('In Process', 'Fulfilled', 'BackOrder', 'Cancelled')),  
CONSTRAINT chk_dely_date CHECK (dely_date > order_date))
```

```
DESC Sales_order
```

Table created.

TABLE SALES_ORDER

Column	Null?	Type
ORDER_NO	NOT NULL	VARCHAR2(5)
ORDER_DATE	-	DATE
CLIENT_NO	NOT NULL	VARCHAR2(6)
DELY_ADDR	-	VARCHAR2(10)
SALESMAN_NO	NOT NULL	VARCHAR2(20)
DELY_TYPE	-	CHAR(1)
BILLED_YN	-	CHAR(1)
DELY_DATE	-	DATE
ORDER_STATUS	-	VARCHAR2(10)

Inserting Data into Sales_order

```
INSERT INTO Sales_order VALUES ('O001', '4-MAY-23', 'C005', 'Delhi', 'S01', 'P', 'Y', '5-MAY-23', 'Fulfilled');  
INSERT INTO Sales_order VALUES ('O002', '2-MAY-23', 'C004', 'Bombay', 'S03', 'F', 'Y', '5-MAY-23', 'Cancelled');  
INSERT INTO Sales_order VALUES ('O003', '8-MAY-23', 'C003', 'Bombay', 'S03', 'P', 'N', '10-MAY-23',  
'BackOrder');  
INSERT INTO Sales_order VALUES ('O004', '5-MAY-23', 'C002', 'Madras', 'S02', 'F', 'Y', '8-MAY-23', 'Fulfilled');  
INSERT INTO Sales_order VALUES ('O005', '1-MAY-23', 'C006', 'Bombay', 'S03', 'P', 'Y', '13-MAY-23', 'In  
Process');
```

ORDER_NO	ORDER_DATE	CLIENT_NO	DELY_ADDR	SALESMAN_NO	DELY_TYPE	BILLED_YN	DELY_DATE	ORDER_STATUS
0001	04-MAY-23	C005	Delhi	S01	P	Y	05-MAY-23	Fulfilled
0002	02-MAY-23	C004	Bombay	S03	F	Y	05-MAY-23	Cancelled
0003	08-MAY-23	C003	Bombay	S03	P	N	10-MAY-23	BackOrder
0004	05-MAY-23	C002	Madras	S02	F	Y	08-MAY-23	Fulfilled
0005	01-MAY-23	C006	Bombay	S03	P	Y	13-MAY-23	In Process

5. Creating Sales_order_details Table Query

```
CREATE TABLE Sales_order_details
(order_no VARCHAR(5) PRIMARY KEY REFERENCES Sales_order(order_no) NOT NULL,
product_no VARCHAR(6) REFERENCES Product_master(product_no) NOT NULL,
qty_ordered NUMBER(3),
qty_display NUMBER(3),
product_rate NUMBER(5));
```

DESC Sales_order_details;

Table created.

TABLE SALES_ORDER_DETAILS

Column	Null?	Type
ORDER_NO	NOT NULL	VARCHAR2(5)
PRODUCT_NO	NOT NULL	VARCHAR2(6)
QTY_ORDERED	-	NUMBER(3,0)
QTY_DISPLAY	-	NUMBER(3,0)
PRODUCT_RATE	-	NUMBER(5,0)

Inserting Data into Sales_order_details

INSERT INTO Sales_order_details VALUES ('O001', 'P2', '4', '2', '500');

INSERT INTO Sales_order_details VALUES ('O003', 'P4', '2', '1', '150');

INSERT INTO Sales_order_details VALUES ('O004', 'P1', '10', '0', '100');

ORDER_NO	PRODUCT_NO	QTY_ORDERED	QTY_DISPLAY	PRODUCT_RATE
0001	P2	4	2	500
0003	P4	2	1	150
0004	P1	10	0	100

Alter table sales_order_details add column price

```
ALTER TABLE Sales_order_details ADD price NUMBER(5);
```

```
DESC Sales_order_details;
```

Table altered.

TABLE SALES_ORDER_DETAILS

Column	Null?	Type
ORDER_NO	NOT NULL	VARCHAR2(5)
PRODUCT_NO	NOT NULL	VARCHAR2(6)
QTY_ORDERED	-	NUMBER(3,0)
QTY_DISPLAY	-	NUMBER(3,0)
PRODUCT_RATE	-	NUMBER(5,0)
PRICE	-	NUMBER(5,0)

Drop table sales_order_details

```
DROP TABLE Sales_order_details;
```

```
DESC Sales_order_details;
```

Table dropped.

ORA-20001: object SALES_ORDER_DETAILS does not exist



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23
(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 3 Date of Implementation: 18. 5. 23

Implementation of DML commands of SQL with suitable examples

Insert

Update

Delete

1. Add a record to department table with values (50,'PERSONNEL','BOSTON').
2. Make a table called 'Bonus1' having fields name, job, salary and store records from the existing employee table for Analysts and Manager only.
3. If a new person HERALD joins the organization in place of TURNER on the 5th of September 1992 with employee number 7999, make the required changes.
4. Make changes in the EMP table so that CLARK reports to BLAKE instead of KING

SOLUTION:

1. Add a record to department table with values (50,'PERSONNEL','BOSTON').

QUERY:

INSERT INTO DEPT VALUES (50, 'PERSONNEL', 'BOSTON') ;

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON
50	PERSONNEL	BOSTON



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 3 Date of Implementation: 18. 5. 23

2. Make a table called 'Bonus1' having fields name, job, salary and store records from the existing employee table for Analysts and Manager only.

QUERY:

```
CREATE TABLE Bonus1 (name VARCHAR(15), job VARCHAR(10), salary NUMBER(9,3));
```

Table created.

TABLE BONUS1

Column	Null?	Type
NAME	-	VARCHAR2(15)
JOB	-	VARCHAR2(10)
SALARY	-	NUMBER(9,3)

```
INSERT INTO Bonus1 SELECT ename, job, sal FROM emp WHERE job='ANALYST' OR job='MANAGER';
```

NAME	JOB	SALARY
JONES	MANAGER	2975
BLAKE	MANAGER	2850
CLARK	MANAGER	2450
SCOTT	ANALYST	3000
FORD	ANALYST	3000



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 3 Date of Implementation: 18. 5. 23

3. If a new person HERALD joins the organization in place of TURNER on the 5th of September 1992 with employee number 7999, make the required changes.

QUERY:

```
UPDATE emp
```

```
SET empno=7999, ename='HERALD', hiredate='05-SEP-92'
```

```
WHERE ename='TURNER';
```

Before Query:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10

After Query:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7999	HERALD	SALESMAN	7698	05-SEP-92	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 3 Date of Implementation: 18. 5. 23

4. Make changes in the EMP table so that CLARK reports to BLAKE instead of KING.

QUERY:

UPDATE emp

SET MGR = 7698

WHERE ename='CLARK';

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7698	09-JUN-81	2450	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7999	HERALD	SALESMAN	7698	05-SEP-92	1500	0	30
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 4

Date of Implementation: 24. 5. 23

Implementation of different types of function with suitable examples

Number function

Aggregate Function

Character Function

Conversion Function

Date Function

1. Find out details of employees where commission is greater than 7% of salary.
2. Select names of all employees whose name start with 'S'.
3. Display the department name which has more than 3 employees in it.
4. Write a query which concatenates the employee number, his name and manager number. Display the output with spaces in between the columns.
5. List names of people who have salary less than the average salary for department 20.
6. Display details of all employees who were hired earliest and latest
7. Find out the department in which the maximum number of employees works
8. Display information about employees who have the maximum number of employees reporting to him
9. Find the day of the week, time of the day, month and century on which SMITH joined.
10. Find number of months the PRESIDENT has worked for the company. Round the months to the nearest whole number.

SOLUTION:

- 1. Find out details of employees where commission is greater than 7% of salary.**

QUERY:

```
SELECT * FROM emp
WHERE comm > 0.07*sal;
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 4

Date of Implementation: 24. 5. 23

2. Select names of all employees whose name start with 'S'.

QUERY:

```
SELECT ename FROM emp  
WHERE ename LIKE 'S%';
```

ENAME
SMITH
SCOTT

3. Display the department name which has more than 3 employees in it

QUERY:

```
SELECT dname  
      FROM dept WHERE deptno IN  
(SELECT deptno FROM emp  
      GROUP BY deptno HAVING COUNT(*)>3);
```

DNAME
RESEARCH
SALES



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 4

Date of Implementation: 24. 5. 23

4. Write a query which concatenates the employee number, his name and manager number. Display the output with spaces in between the columns.

QUERY:

```
SELECT CONCAT (CONCAT (CONCAT(CONCAT(empno,''),ename), ' '),mgr)
FROM emp;
```

7369 SMITH 7902
7499 ALLEN 7698
7521 WARD 7698
7566 JONES 7839
7654 MARTIN 7698
7698 BLAKE 7839
7782 CLARK 7839
7788 SCOTT 7566
7839 KING
7844 TURNER 7698
7876 ADAMS 7788
7900 JAMES 7698
7902 FORD 7566
7934 MILLER 7782

5. List names of people who have salary less than the average salary for department 20.

QUERY:

```
SELECT ename
FROM emp
WHERE sal < (SELECT AVG(sal) FROM emp)
      AND deptno = 20;
```

ENAME
SMITH
ADAMS



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 4 Date of Implementation: 24. 5. 23

6. Display details of all employees who were hired earliest and latest.

QUERY:

```
SELECT * FROM emp
WHERE hiredate= (SELECT MIN(hiredate) FROM EMP)
OR hiredate= (SELECT MAX(hiredate) FROM EMP);
```

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20

7. Find out the department in which the maximum number of employees works.

QUERY:

```
SELECT dname FROM dept
WHERE deptno= (SELECT deptno FROM
(SELECT deptno, COUNT(deptno)
FROM emp GROUP BY deptno
ORDER BY COUNT(deptno) DESC)
WHERE ROWNUM = 1);
```

DNAME
SALES



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 4 Date of Implementation: 24. 5. 23

8. Display information about employees who have the maximum number of employees reporting to him.

QUERY:

```
SELECT e.ename,
       COUNT(*)
      FROM emp m,
           emp e
     WHERE m.mgr = e.empno
   GROUP BY e.ename
HAVING COUNT(*) =
(SELECT MAX (mycount)
   FROM
(SELECT COUNT(*) mycount
      FROM emp
  GROUP BY mgr) );
```

ENAME	COUNT(*)
BLAKE	5

9. Find the day of the week, time of the day, month and century on which SMITH joined.

QUERY:

```
SELECT ename, hiredate,
       TO_CHAR(hiredate, 'D') AS Day_of_week,
       TO_CHAR (hiredate,'MM') AS Month,
       TO_CHAR (hiredate,'cc') AS Century,
       TO_CHAR (hiredate,HH:MM:SS') AS Time
      FROM emp
     WHERE ename LIKE 'SMITH';
```

ENAME	HIREDATE	DAY_OF_WEEK	MONTH	CENTURY	TIME
SMITH	17-DEC-80	4	12	20	12:12:00



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 4 **Date of Implementation:** 24. 5. 23

10. Find number of months the PRESIDENT has worked for the company. Round the months to the nearest whole Number.

QUERY:

```
SELECT ename, hiredate,
       ROUND(MONTHS_BETWEEN(sysdate,hiredate)) "Months"
  FROM emp
 WHERE job LIKE 'PRESIDENT' ;
```

ENAME	HIREDATE	Months
KING	17-NOV-81	498



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 5

Date of Implementation: 31. 5. 23

Implementation of different types of operators in SQL

Arithmetic Operators

Logical Operators

Comparison Operator

Special Operator

Set Operation

1. Display names, departments, and positions who work as Analyst or Clerk.
2. Display different types of jobs / positions available in the company.
3. Display the names of people and their departments working in either Sales or Research department.
4. Find the list of all clients who stay in 'Bombay' or 'Delhi' or 'Chennai'.
5. Print the list of clients whose bal_due is greater than value 10000.
6. Display the order information for client_no 'C00001' and 'C00002'.
7. Find products whose selling price is greater than 2000 and less than or equal to 5000.
8. Find products whose selling price is more than 1500. Calculate a new selling price as, original selling price * .15. Rename the new column in the above query new_price.

SOLUTION:

1. Display names, departments, and positions who work as Analyst or Clerk.

QUERY:

```
SELECT emp.ename, emp.job position, dept.dname department_name
FROM dept LEFT JOIN emp ON dept.deptno= emp.deptno
WHERE emp.job LIKE 'ANALYST' OR emp.JOB LIKE 'CLERK'
ORDER BY dept.deptno;
```

ENAME	POSITION	DEPARTMENT_NAME
MILLER	CLERK	ACCOUNTING
SCOTT	ANALYST	RESEARCH
SMITH	CLERK	RESEARCH
ADAMS	CLERK	RESEARCH
FORD	ANALYST	RESEARCH
JAMES	CLERK	SALES



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 5 Date of Implementation: 31. 5. 23

2. Display different types of jobs / positions available in the company.

QUERY:

```
SELECT UNIQUE job FROM emp;
```

JOB
CLERK
SALESMAN
ANALYST
MANAGER
PRESIDENT

3. Display the names of people and their departments working in either Sales or Research department.

QUERY:

```
SELECT emp.ename, dept.deptno, dept.dname
FROM dept LEFT JOIN emp
ON dept.deptno = emp.deptno
WHERE dept.dname LIKE 'SALES' OR dept.dname LIKE 'RESEARCH'
ORDER BY dept.deptno;
```

ENAME	DEPTNO	DNAME
SMITH	20	RESEARCH
ADAMS	20	RESEARCH
JONES	20	RESEARCH
SCOTT	20	RESEARCH
FORD	20	RESEARCH
JAMES	30	SALES
TURNER	30	SALES
MARTIN	30	SALES
WARD	30	SALES
ALLEN	30	SALES
BLAKE	30	SALES



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 5 Date of Implementation: 31. 5. 23

4. Find the list of all clients who stay in 'Bombay' or 'Delhi' or 'Chennai'.

QUERY:

SELECT * FROM CLIENT_MASTER

WHERE CITY ='Bombay' OR CITY='Delhi' OR CITY='Chennai';

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00002	Arjun	b12	-	Chennai	780041	Tamilnadu	600
C00003	Reena	c13	-	Bombay	400057	Maharashtra	3000
C00004	Kiran	d14	-	Bombay	400056	Maharashtra	5000
C00005	Bhushan	e15	-	Delhi	100001	Delhi	10000
C00006	Ronak	f16	-	Bombay	400050	Maharashtra	800
C00009	Rohit	i19	-	Chennai	780041	Tamilnadu	20000

5. Print the list of clients whose bal_due is greater than value 10000.

QUERY:

SELECT * FROM CLIENT_MASTER WHERE BAL_DUE>10000;

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00001	Manish	a11	-	Pune	411001	Maharashtra	25500
C00007	Tina	g17	-	Agra	402938	Uttar Pradesh	10700
C00009	Rohit	i19	-	Chennai	780041	Tamilnadu	20000

6. Display the order information for client_no 'C00001' and 'C00002'.

QUERY:

SELECT * FROM SALES_ORDER

WHERE CLIENT_NO='C00001' OR CLIENT_NO='C00002';

ORDER_NO	ORDER_DATE	CLIENT_NO	DELY_ADDR	SALESMAN_NO	DELY_TYPE	BILLED_YN	DELY_DATE	ORDER_STATUS
0004	05-MAY-23	C00002	Chennai	S02	F	Y	08-MAY-23	Fulfilled
0006	01-MAY-23	C00001	Pune	S03	F	N	13-MAY-23	In Process



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 5 Date of Implementation: 31. 5. 23

7. Find products whose selling price is greater than 2000 and less than or equal to 5000.

QUERY:

```
SELECT DESCRIPTION AS PRODUCT_NAME, SELL_PRICE  
FROM PRODUCT_MASTER  
WHERE SELL_PRICE > 2000 AND SELL_PRICE <= 5000;
```

PRODUCT_NAME	SELL_PRICE
Keyboards	2500
RAM	4500

8. Find products whose selling price is more than 1500. Calculate a new selling price as, original selling price * 15. Rename the new column in the above query new_price.

QUERY:

```
SELECT DESCRIPTION, SELL_PRICE, (SELL_PRICE)*15 "NEW_PRICE"  
FROM PRODUCT_MASTER  
WHERE SELL_PRICE > 1500;
```

DESCRIPTION	SELL_PRICE	NEW_PRICE
Keyboards	2500	37500
RAM	4500	67500
PC	10000	150000



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23
(410917) Database Management System Laboratory

Class: FY-MCA Shift / Div: A Batch: F2 Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 6 Date of Implementation: 1. 6. 23

Implementation of different types of Joins

Inner Join

Outer Join

Natural Join etc.

1. Display the locations of the employees.
2. Display name, salary and location of employees who stay in CHICAGO.
3. List number, name, job, manager number and manager job of each employee.

Solution:

1. Display the locations of the employees.

QUERY:

```
SELECT e.ename, e.deptno, d.loc
FROM dept d
    LEFT JOIN emp e
        ON e.deptno=d.deptno
ORDER BY d.loc;
```

ENAME	DEPTNO	LOC
-	-	BOSTON
TURNER	30	CHICAGO
WARD	30	CHICAGO
ALLEN	30	CHICAGO
MARTIN	30	CHICAGO
BLAKE	30	CHICAGO
JAMES	30	CHICAGO
JONES	20	DALLAS
SMITH	20	DALLAS
SCOTT	20	DALLAS
FORD	20	DALLAS
ADAMS	20	DALLAS
CLARK	10	NEW YORK
MILLER	10	NEW YORK
KING	10	NEW YORK



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23
(410917) Database Management System Laboratory

Class: FY-MCA Shift / Div: A Batch: F2 Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 6 **Date of Implementation:** 1. 6. 23

2. Display name, salary and location of employees who stay in CHICAGO.

QUERY:

```
SELECT e.ename, e.sal, d.loc
  FROM dept d
  LEFT JOIN emp e
    ON e.deptno=d.deptno
 WHERE d.loc LIKE 'CHICAGO' ;
```

ENAME	SAL	LOC
ALLEN	1600	CHICAGO
WARD	1250	CHICAGO
MARTIN	1250	CHICAGO
BLAKE	2850	CHICAGO
TURNER	1500	CHICAGO
JAMES	950	CHICAGO



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23
(410917) Database Management System Laboratory

Class: FY-MCA Shift / Div: A Batch: F2 Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 6 Date of Implementation: 1. 6. 23

3. List number, name, job, manager number and manager job of each employee.

QUERY:

```
SELECT e.empno, e.ename , m.empno manager_id ,
m.ename manager_name, m.job manager_job
FROM emp e join emp m
ON (e.mgr = m.empno);
```

EMPNO	ENAME	MANAGER_ID	MANAGER_NAME	MANAGER_JOB
7788	SCOTT	7566	JONES	MANAGER
7902	FORD	7566	JONES	MANAGER
7499	ALLEN	7698	BLAKE	MANAGER
7521	WARD	7698	BLAKE	MANAGER
7654	MARTIN	7698	BLAKE	MANAGER
7844	TURNER	7698	BLAKE	MANAGER
7900	JAMES	7698	BLAKE	MANAGER
7934	MILLER	7782	CLARK	MANAGER
7876	ADAMS	7788	SCOTT	ANALYST
7566	JONES	7839	KING	PRESIDENT
7698	BLAKE	7839	KING	PRESIDENT
7782	CLARK	7839	KING	PRESIDENT
7369	SMITH	7902	FORD	ANALYST



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 7

Date of Implementation: 1. 6. 23

Study and Implementation of

Group By & having clause

Order by clause

Indexing

Views

1. Give a list of employees sorted on their names.
2. Display names and departments of employees who work in department 10.
3. Accept department number from user and display details of all employees in that department.
4. Create an index that would permit each salesperson to retrieve his or her orders grouped by date quickly.
5. Create a view on emp table which will display empno, ename, sal, deptno, dname

Solution:

1. Give a list of employees sorted on their names.

QUERY:

SELECT * FROM emp ORDER BY ename;

SQL Worksheet

1 | SELECT * FROM emp ORDER BY ename;

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7876	ADAMS	CLERK	7788	12-JAN-83	1100	-	20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850	-	30
7782	CLARK	MANAGER	7839	09-JUN-81	2450	-	10
7902	FORD	ANALYST	7566	03-DEC-81	3000	-	20
7900	JAMES	CLERK	7698	03-DEC-81	950	-	30
7566	JONES	MANAGER	7839	02-APR-81	2975	-	20
7839	KING	PRESIDENT	-	17-NOV-81	5000	-	10
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7934	MILLER	CLERK	7782	23-JAN-82	1300	-	10
7788	SCOTT	ANALYST	7566	09-DEC-82	3000	-	20
7369	SMITH	CLERK	7902	17-DEC-80	800	-	20
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 7

Date of Implementation: 1. 6. 23

2. Display names and departments of employees who work in department 10.

QUERY:

```
SELECT emp.ename, dept.dname, dept.deptno
FROM dept
LEFT JOIN emp ON emp.deptno=dept.deptno
WHERE emp.deptno=10;
```

ENAME	DNAME	DEPTNO
CLARK	ACCOUNTING	10
KING	ACCOUNTING	10
MILLER	ACCOUNTING	10



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 7

Date of Implementation: 1. 6. 23

3. Accept department number from user and display details of all employees in that department.

QUERY:

DECLARE

 department_number NUMBER;

BEGIN

 -- Accepting department number from user

 department_number := :department_number ;

 -- Query to fetch employee details

DBMS_OUTPUT.PUT_LINE('Department Number: ' || department_number);

 FOR emp IN (

 SELECT empno, ename, sal

 FROM emp

 WHERE deptno = department_number

)

 LOOP

 -- Displaying employee details

 DBMS_OUTPUT.PUT_LINE('Employee ID: ' || emp.empno);

 DBMS_OUTPUT.PUT_LINE('Employee Name: ' || emp.ename);

 DBMS_OUTPUT.PUT_LINE('Employee Salary: ' || emp.sal);

 END LOOP;

END;

Results Explain Describe Saved SQL History

Department Number: 10
Employee ID: 7782
Employee Name: CLARK
Employee Salary: 2450
Employee ID: 7839
Employee Name: KING
Employee Salary: 5000
Employee ID: 7934
Employee Name: MILLER
Employee Salary: 1300

Statement processed.



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 7

Date of Implementation: 1. 6. 23

4. Create an index that would permit each salesperson to retrieve his or her orders grouped by date quickly.

QUERY:

```
CREATE INDEX o_date ON Sales_order(order_date);
```

SQL Worksheet

```
1 create index o_date on Sales_order(order_date);
2
3
```

Index created.

5. Create a view on emp table which will display empno, ename, sal, deptno, dname

QUERY:

```
CREATE OR REPLACE VIEW view_1 AS
SELECT e.empno, e.ename, e.sal, e.deptno, d.dname
FROM emp e
JOIN dept d ON e.deptno = d.deptno;
```

```
SELECT * FROM view_1;
```

SQL Worksheet

```
View created.
```

EMPNO	ENAME	SAL	DEPTNO	DNAME
7369	SMITH	800	20	RESEARCH
7499	ALLEN	1600	30	SALES
7521	WARD	1250	30	SALES
7566	JONES	2975	20	RESEARCH
7654	MARTIN	1250	30	SALES
7698	BLAKE	2850	30	SALES
7782	CLARK	2450	10	ACCOUNTING
7788	SCOTT	3000	20	RESEARCH
7839	KING	5000	10	ACCOUNTING
7844	TURNER	1500	30	SALES
7876	ADAMS	1100	20	RESEARCH
7900	JAMES	950	30	SALES
7902	FORD	3000	20	RESEARCH
7934	MILLER	1300	10	ACCOUNTING



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 8

Date of Implementation: 7. 6. 23

1. Write a PL/SQL block of code that first inserts a record in an ‘emp’ table. Update the salary by Rs. 2000. Then check to see that the total salary does not exceed 20000. If so, undo the updates made to the salaries.

QUERY:

```
DECLARE
    v_total_salary NUMBER;
BEGIN
    INSERT INTO EMP VALUES
        (7338, 'RAM', 'SALESMAN', 7566,
         TO_DATE('08-DEC-1988', 'DD-MON-YYYY'), 700, NULL, 10);

    SAVEPOINT no_update;

    UPDATE emp
    SET sal = sal + 2000;
    SELECT sum(sal) INTO v_total_salary
    FROM emp;

    IF v_total_salary > 20000
    THEN
        DBMS_OUTPUT.PUT_LINE('Salary limit exceeded. Updates have been rolled back.');
        DBMS_OUTPUT.PUT_LINE('Sum of Salary: '|| v_total_salary );
        ROLLBACK TO SAVEPOINT no_update;
    ELSE
        DBMS_OUTPUT.PUT_LINE('Changes saved');
    END IF;
    COMMIT;
END;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Salary limit exceeded. Updates have been rolled back.
Sum of Salary: 62425

Statement processed.

0.03 seconds



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 8

Date of Implementation: 7. 6. 23

2. HRD manager has decided to raise the salary of employees by 0.15. Write a PL/SQL block to accept the employee number update the salary of that emp. Display message based on the existence of record in employee table.

QUERY:

```
DECLARE
    v_emp_id emp.empno%TYPE;
    v_emp_name emp.ename%TYPE;
    v_old_salary emp.sal%TYPE;
BEGIN
    -- Accept employee number from user
    v_emp_id := :employee_number;

    -- Check if the employee exists
    SELECT ename, sal INTO v_emp_name, v_old_salary
    FROM emp
    WHERE empno = v_emp_id;

    -- Update the salary
    UPDATE emp
    SET sal = sal + (sal* 0.15)
    WHERE empno = v_emp_id;

    -- Display appropriate message
    IF SQL%ROWCOUNT = 1 THEN
        DBMS_OUTPUT.PUT_LINE('Salary updated successfully for Employee ' || v_emp_name);
    ELSE
        DBMS_OUTPUT.PUT_LINE('Employee ' || v_emp_id || ' does not exist.');
    END IF;

    -- Display old and new salary
    DBMS_OUTPUT.PUT_LINE('Old Salary: ' || v_old_salary);
    DBMS_OUTPUT.PUT_LINE('New Salary: ' || (v_old_salary + (v_old_salary * 0.15)));

    COMMIT;
EXCEPTION
    WHEN NO_DATA_FOUND THEN
        DBMS_OUTPUT.PUT_LINE('Employee ' || v_emp_id || ' does not exist.');
    ROLLBACK;
END;
```



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 8

Date of Implementation: 7. 6. 23

When Employee number exists in Record:

Results Explain Describe Saved SQL History

Salary updated successfully for Employee JAMES

Old Salary: 950

New Salary: 1092.5

1 row(s) updated.

0.00 seconds

When Employee number doesn't exist in Record:

Results Explain Describe Saved SQL History

Employee 523 does not exist.

1 row(s) updated.

0.00 seconds



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 9

Date of Implementation: 15. 6. 23

Create a table and perform the following

1. Increase salary of each customer by 5000

QUERY:

```
DECLARE
    total_rows number;
BEGIN
    UPDATE CUSTOMERS
    SET Salary= Salary + 5000;

    total_rows := SQL%ROWCOUNT;

    DBMS_OUTPUT.PUT_LINE(total_rows || ' rows updated.');
END;
```

Results Explain Describe Saved SQL History

8 rows updated.

1 row(s) updated.

0.01 seconds

Before Query:

Results Explain Describe Saved SQL History

ID	NAME	AGE	ADDRESS	SALARY
1	RIYA	24	PUNE	3000.5
2	AKASH	45	MUMBAI	7000
3	TINA	35	LUCKNOW	6000
4	KARAN	26	KOLKATA	2000
5	TANYA	19	SURAT	4500
6	ROHIT	67	PUNE	3200.5
7	RONAK	22	LONAVALA	8100.5
8	BHUSHAN	29	PUNE	10000

8 rows returned in 0.01 seconds

[CSV Export](#)

After Query:

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

ID	NAME	AGE	ADDRESS	SALARY
1	RIYA	24	PUNE	8000.5
2	AKASH	45	MUMBAI	12000
3	TINA	35	LUCKNOW	11000
4	KARAN	26	KOLKATA	7000
5	TANYA	19	SURAT	9500
6	ROHIT	67	PUNE	8200.5
7	RONAK	22	LONAVALA	13100.5
8	BHUSHAN	29	PUNE	15000

8 rows returned in 0.02 seconds

[CSV Export](#)



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 9

Date of Implementation: 15. 6. 23

2. Write a program to retrieve the customer name and address.

QUERY:

```
DECLARE
    c_id customers.id%type;
    c_name customers.name%type;
    c_addr customers.address%type;
    CURSOR c_customers is
        SELECT id, name, address FROM customers;
BEGIN
    OPEN c_customers;
    dbms_output.put_line('ID'|| ' ' || 'Name'|| ' ' || 'Address');
    LOOP
        FETCH c_customers into c_id, c_name, c_addr;
        EXIT WHEN c_customers%notfound;
        dbms_output.put_line(c_id || ' ' || c_name || ' ' || c_addr);
    END LOOP;
    CLOSE c_customers;
END;
```

Results Explain Describe Saved SQL History

ID	Name	Address
1	RIYA PUNE	
2	AKASH MUMBAI	
3	TINA LUCKNOW	
4	KARAN KOLKATA	
5	TANYA SURAT	
6	ROHIT PUNE	
7	RONAK LONAVALA	
8	BHUSHAN PUNE	

Statement processed.



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More Assignment No: 10 Date of Implementation: 27. 6. 23

Q1. Create a stored function to perform item_id check operation. Which accepts a item_id returns a flag as per the id exist or not.

TABLE:

Results Explain Describe Saved SQL History		
ITEM_ID	I_NAME	I_QUANTITY
1001	Lcd Screens	24
1005	Hard drives	50
1012	Mouse	30
1123	Keyboards	65
1009	CPU	70
1010	RAM	40

6 rows returned in 0.02 seconds

[CSV Export](#)

FUNCTION:

```
CREATE OR REPLACE FUNCTION check_item(p_item_id NUMBER) RETURN NUMBER IS
v_flag NUMBER;
BEGIN
SELECT COUNT(*) INTO v_flag
FROM items
WHERE item_id = p_item_id;
RETURN v_flag;
END;
```

Autocommit Display 10

```
CREATE OR REPLACE FUNCTION check_item(p_item_id NUMBER) RETURN NUMBER IS
v_flag NUMBER;
BEGIN
SELECT COUNT(*) INTO v_flag
FROM items
WHERE item_id = p_item_id;
RETURN v_flag;
END;
```

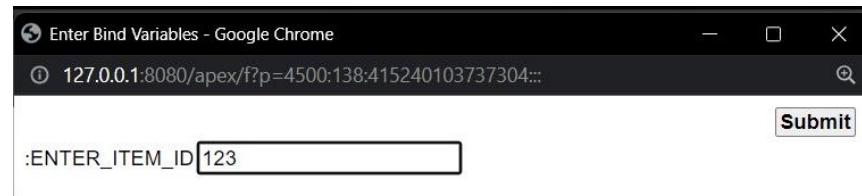
Results Explain Describe Saved SQL History

Function created.

0.52 seconds

QUERY:

```
DECLARE
v_item_id NUMBER;
v_item_exists NUMBER;
BEGIN
v_item_id := :Enter_Item_ID;
v_item_exists := check_item(v_item_id);
IF v_item_exists = 1 THEN
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' exists.');
ELSE
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' does not exist.');
END IF;
END;
```



```
DECLARE
v_item_id NUMBER;
v_item_exists NUMBER;
BEGIN
v_item_id := :Enter_Item_ID;
v_item_exists := check_item(v_item_id);
IF v_item_exists = 1 THEN
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' exists.');
ELSE
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' does not exist.');
END IF;
END;
```

Results Explain Describe Saved SQL History

Item ID 123 does not exist.

Statement processed.

0.02 seconds

```
DECLARE
v_item_id NUMBER;
v_item_exists NUMBER;
BEGIN
v_item_id := :Enter_Item_ID;
v_item_exists := check_item(v_item_id);
IF v_item_exists = 1 THEN
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' exists.');
ELSE
DBMS_OUTPUT.PUT_LINE('Item ID ' || v_item_id || ' does not exist.');
END IF;
END;
```

Results Explain Describe Saved SQL History

Item ID 1001 exists.

Statement processed.

0.00 seconds



Progressive Education Society's
Modern College of Engineering, Pune
MCA Department
A.Y.2022-23

(410917) Database Management System Laboratory

Class: FY-MCA

Shift / Div: A

Batch: F2

Roll Number: 51043

Name: Vanessa Reetu Prashant More **Assignment No:** 11 **Date of Implementation:** 28. 6. 23

Q1. Application using database triggers – Create a transparent audit system for a table Client_master.
The system must keep track of the records that are being deleted or updated. When the record is deleted or modified the original record details & date of operation are stored in audit table & then the delete update is allowed to go.

QUERY:

Client_master1 Table Creation:

```
CREATE TABLE Client_master1 (client_no varchar(10) primary key , name char(20) NOT NULL ,address1 varchar(20) ,address2 varchar(20) ,city char(20) ,pincode number(10) ,state char(20) ,bal_due number(9,4));
```

Client_master_audit1 Table Creation:

```
CREATE TABLE Client_master_audit1 (client_no varchar(10) primary key , name char(20) NOT NULL ,address1 varchar(20) ,address2 varchar(20) ,city char(20) ,pincode number(10) ,state char(20) ,bal_due number(9,4));
```

Client_master1 Table:

Results Explain Describe Saved SQL History							
CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00001	Manish	a11	-	Pune	411001	Maharashtra	25500
C00002	Arjun	b12	-	Chennai	780041	Tamilnadu	600
C00003	Reena	c13	-	Bombay	400057	Maharashtra	3000
C00004	Kiran	d14	-	Bombay	400056	Maharashtra	5000
C00005	Bhushan	e15	-	Delhi	100001	Delhi	10000
C00006	Ronak	f16	-	Bombay	400050	Maharashtra	800

6 rows returned in 0.00 seconds

[CSV Export](#)

Trigger:

```
CREATE OR REPLACE TRIGGER Client_audit
BEFORE INSERT OR DELETE OR UPDATE ON Client_Master1
FOR EACH ROW
BEGIN
    IF DELETING THEN
        INSERT INTO Client_master_audit1 VALUES
        (:OLD.client_no, :OLD.name, :OLD.address1, :OLD.address2, :OLD.city, :OLD.pincode, :OLD.state,
        :OLD.bal_due);
    ELSIF UPDATING THEN
        INSERT INTO Client_master_audit1 VALUES
        (:OLD.client_no, :OLD.name, :OLD.address1, :OLD.address2, :OLD.city, :OLD.pincode, :OLD.state,
        :OLD.bal_due);
    END IF;
END;
```

Home > SQL > SQL Commands

Autocommit Display 10 ▾

```
CREATE OR REPLACE TRIGGER Client_audit
BEFORE INSERT OR DELETE OR UPDATE ON Client_Master1
FOR EACH ROW
BEGIN
    IF DELETING THEN
        INSERT INTO Client_master_audit1 VALUES
        (:OLD.client_no, :OLD.name, :OLD.address1, :OLD.address2, :OLD.city, :OLD.pincode, :OLD.state, :OLD.bal_due);
    ELSIF UPDATING THEN
        INSERT INTO Client_master_audit1 VALUES
        (:OLD.client_no, :OLD.name, :OLD.address1, :OLD.address2, :OLD.city, :OLD.pincode, :OLD.state, :OLD.bal_due);
    END IF;
END;
```

Results Explain Describe Saved SQL History

Trigger created.

2.33 seconds

Deleting from Client_master1:

```
DELETE FROM Client_master1 WHERE name='Ronak' ;
```

Deleted Record gets stored in Client_master_audit1:

Results Explain Describe Saved SQL History

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00006	Ronak	f16	-	Bombay	400050	Maharashtra	800

1 rows returned in 0.00 seconds [CSV Export](#)

Updating record in Client_master1:

```
UPDATE Client_master1 SET bal_due=1000 WHERE client_no='C00005';
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00001	Manish	a11	-	Pune	411001	Maharashtra	25500
C00002	Arjun	b12	-	Chennai	780041	Tamilnadu	600
C00003	Reena	c13	-	Bombay	400057	Maharashtra	3000
C00004	Kiran	d14	-	Bombay	400056	Maharashtra	5000
C00005	Bhushan	e15	-	Delhi	100001	Delhi	10000

5 rows returned in 0.14 seconds

[CSV Export](#)

Updated Record gets stored in Client_master_audit1:

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

CLIENT_NO	NAME	ADDRESS1	ADDRESS2	CITY	PINCODE	STATE	BAL_DUE
C00006	Ronak	f16	-	Bombay	400050	Maharashtra	800
C00005	Bhushan	e15	-	Delhi	100001	Delhi	10000

2 rows returned in 0.00 seconds

[CSV Export](#)