**Ex. No.: 1**

**CREATION OF BASE TABLE AND DML OPERATIONS**

1.

CREATE TABLE MY\_EMPLOYEE (

ID NUMBER(4) NOT NULL,

Last\_name VARCHAR2(25),

First\_name VARCHAR2(25),

Userid VARCHAR2(25),

Salary NUMBER(9,2),

CONSTRAINT pk\_employee PRIMARY KEY (ID)

);

2.

INSERT INTO MY\_EMPLOYEE (ID, Last\_name, First\_name, Userid, Salary)

VALUES (1, 'Patel', 'Ralph', 'rpatel', 895);

INSERT INTO MY\_EMPLOYEE (ID, Last\_name, First\_name, Userid, Salary)

VALUES (2, 'Dancs', 'Betty', 'bdancs', 860);

3.

SELECT \* FROM MY\_EMPLOYEE;

4.

INSERT INTO MY\_EMPLOYEE (ID, Last\_name, First\_name, Userid, Salary)

VALUES (3, 'Biri', 'Ben', NULL, 1100);

INSERT INTO MY\_EMPLOYEE (ID, Last\_name, First\_name, Userid, Salary)

VALUES (4, 'Newman', 'Chad', NULL, 750);

UPDATE MY\_EMPLOYEE

SET Userid = LOWER(CONCAT(SUBSTR(First\_name, 1, 1), SUBSTR(Last\_name, 1, 7)))

WHERE ID = 3 OR ID = 4;

5.

DELETE FROM MY\_EMPLOYEE

WHERE First\_name = 'Betty' AND Last\_name = 'Dancs';

6.

UPDATE MY\_EMPLOYEE

SET Last\_name = NULL, First\_name = NULL, Userid = NULL, Salary = NULL

WHERE ID = 4;

7.

COMMIT;

8.

UPDATE MY\_EMPLOYEE

SET Last\_name = 'Drexler'

WHERE ID = 3;

9.

UPDATE MY\_EMPLOYEE

SET Salary = 1000

WHERE Salary < 900;

**Ex. No.: 2**

**DATA MANIPULATIONS**

**A.**

**Initial:**

CREATE TABLE EMPLOYEES (

Employee\_id NUMBER(6) NOT NULL,

First\_Name VARCHAR2(20),

Last\_Name VARCHAR2(25) NOT NULL,

Email VARCHAR2(25) NOT NULL,

Phone\_Number VARCHAR2(20),

Hire\_date DATE NOT NULL,

Job\_id VARCHAR2(10) NOT NULL,

Salary NUMBER(8,2),

Commission\_pct NUMBER(2,2),

Manager\_id NUMBER(6),

Department\_id NUMBER(4),

CONSTRAINT pk\_employee\_id PRIMARY KEY (Employee\_id)

);

INSERT INTO EMPLOYEES

VALUES (101, 'John', 'Doe', 'jdoe@example.com', '1234567890', TO\_DATE('2022-06-15', 'YYYY-MM-DD'), 'IT\_PROG', 5000, NULL, 100, 60);

INSERT INTO EMPLOYEES

VALUES (102, 'Jane', 'Austin', 'jaustin@example.com', '0987654321', TO\_DATE('2022-08-20', 'YYYY-MM-DD'), 'HR\_MAN', 4800, NULL, 101, 70);

INSERT INTO EMPLOYEES

VALUES (103, 'Mark', 'Smith', 'msmith@example.com', '1230984567', TO\_DATE('2023-01-10', 'YYYY-MM-DD'), 'SA\_REP', 4600, 0.10, 100, 80);

INSERT INTO EMPLOYEES

VALUES (104, 'Chad', 'Newman', 'cnewman@example.com', '7896541230', TO\_DATE('2021-11-03', 'YYYY-MM-DD'), 'FI\_MGR', 6000, NULL, 102, 60);

INSERT INTO EMPLOYEES

VALUES (105, 'Betty', 'Austin', 'baustin@example.com', '9874563210', TO\_DATE('2020-12-25', 'YYYY-MM-DD'), 'HR\_CLERK', 3900, NULL, 101, 70);

1.

SELECT Employee\_id, First\_Name, Last\_Name, Salary

FROM EMPLOYEES;

2.

SELECT Employee\_id, First\_Name, Last\_Name

FROM EMPLOYEES

WHERE Manager\_id = 100;

3.

SELECT First\_Name, Last\_Name

FROM EMPLOYEES

WHERE Salary >= 4800;

4.

SELECT First\_Name, Last\_Name

FROM EMPLOYEES

WHERE Last\_Name = 'AUSTIN';

5.

SELECT First\_Name, Last\_Name

FROM EMPLOYEES

WHERE Department\_id IN (60, 70, 80);

6.

SELECT DISTINCT Manager\_id

FROM EMPLOYEES;

**B.**

**Initial:**

CREATE TABLE EMP (

EmpNo NUMBER(6),

EmpName VARCHAR2(25),

Job VARCHAR2(20),

Basic NUMBER(8,2),

DA NUMBER(8,2),

HRA NUMBER(8,2),

PF NUMBER(8,2),

GrossPay NUMBER(8,2),

NetPay NUMBER(8,2),

Department\_id NUMBER(4)

);

1.

INSERT INTO EMP (EmpNo, EmpName, Job, Basic, Department\_id)

VALUES (1, 'John Doe', 'Manager', 5000, 60);

INSERT INTO EMP (EmpNo, EmpName, Job, Basic, Department\_id)

VALUES (2, 'Jane Austin', 'Clerk', 4000, 70);

INSERT INTO EMP (EmpNo, EmpName, Job, Basic, Department\_id)

VALUES (3, 'Mark Smith', 'Sales', 3500, 80);

INSERT INTO EMP (EmpNo, EmpName, Job, Basic, Department\_id)

VALUES (4, 'Chad Newman', 'Manager', 6000, 60);

INSERT INTO EMP (EmpNo, EmpName, Job, Basic, Department\_id)

VALUES (5, 'Betty Austin', 'HR', 3900, 70);

UPDATE EMP

SET

DA = 0.30 \* Basic,

HRA = 0.40 \* Basic,

PF = 0.12 \* Basic;

UPDATE EMP

SET

GrossPay = Basic + DA + HRA;

UPDATE EMP

SET

NetPay = GrossPay - PF;

2.

SELECT \*

FROM EMP e

WHERE Basic = (

SELECT MIN(Basic)

FROM EMP

WHERE Department\_id = e.Department\_id

);

3.

SELECT EmpName, NetPay

FROM EMP

WHERE NetPay < 7500;

**C.**

1.

CREATE TABLE DEPT (

ID NUMBER(7),

NAME VARCHAR2(25),

CONSTRAINT pk\_dept PRIMARY KEY (ID)

);

2.

CREATE TABLE EMP (

ID NUMBER(7),

LAST\_NAME VARCHAR2(25),

FIRST\_NAME VARCHAR2(25),

DEPT\_ID NUMBER(7),

CONSTRAINT pk\_emp PRIMARY KEY (ID)

);

3.

ALTER TABLE EMP

MODIFY LAST\_NAME VARCHAR2(50);

4.

CREATE TABLE EMPLOYEES2 AS

SELECT Employee\_id AS Id, First\_Name, Last\_Name, Salary, Department\_id AS Dept\_id

FROM EMPLOYEES;

5.

DROP TABLE EMP;

6.

ALTER TABLE EMPLOYEES2

RENAME TO EMP;

7.

COMMENT ON TABLE DEPT IS 'Department Table';

COMMENT ON TABLE EMP IS 'Employees Table';

DESC DEPT;

DESC EMP;

8.

ALTER TABLE EMP

DROP COLUMN First\_Name;

DESC EMP;

**Ex. No.: 3**

**WRITING BASIC SQL SELECT STATEMENTS**

**Initial:**

CREATE TABLE departments (

dept\_id NUMBER(4) PRIMARY KEY,

dept\_name VARCHAR2(30),

manager\_id NUMBER(6),

location\_id NUMBER(4)

);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (10, 'HR', 101, 1001);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (20, 'Sales', 102, 1002);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (30, 'IT', 103, 1003);

CREATE TABLE EMPLOYEES (

Employee\_id NUMBER(6) NOT NULL,

First\_Name VARCHAR2(20),

Last\_Name VARCHAR2(25) NOT NULL,

Email VARCHAR2(25) NOT NULL,

Phone\_Number VARCHAR2(20),

Hire\_date DATE NOT NULL,

Job\_id VARCHAR2(10) NOT NULL,

Salary NUMBER(8,2),

Commission\_pct NUMBER(2,2),

Manager\_id NUMBER(6),

Department\_id NUMBER(4),

CONSTRAINT pk\_employee\_id PRIMARY KEY (Employee\_id)

);

INSERT INTO EMPLOYEES

VALUES (101, 'John', 'Doe', 'jdoe@example.com', '1234567890', TO\_DATE('2022-06-15', 'YYYY-MM-DD'), 'IT\_PROG', 5000, NULL, 100, 60);

INSERT INTO EMPLOYEES

VALUES (102, 'Jane', 'Austin', 'jaustin@example.com', '0987654321', TO\_DATE('2022-08-20', 'YYYY-MM-DD'), 'HR\_MAN', 4800, NULL, 101, 70);

INSERT INTO EMPLOYEES

VALUES (103, 'Mark', 'Smith', 'msmith@example.com', '1230984567', TO\_DATE('2023-01-10', 'YYYY-MM-DD'), 'SA\_REP', 4600, 0.10, 100, 80);

INSERT INTO EMPLOYEES

VALUES (104, 'Chad', 'Newman', 'cnewman@example.com', '7896541230', TO\_DATE('2021-11-03', 'YYYY-MM-DD'), 'FI\_MGR', 6000, NULL, 102, 60);

INSERT INTO EMPLOYEES

VALUES (105, 'Betty', 'Austin', 'baustin@example.com', '9874563210', TO\_DATE('2020-12-25', 'YYYY-MM-DD'), 'HR\_CLERK', 3900, NULL, 101, 70);

1.

SELECT Employee\_id, Last\_Name, Salary \* 12 AS "ANNUAL SALARY"

FROM EMPLOYEES;

2.

DESC departments;

SELECT \* FROM departments;

3.

SELECT employee\_id, last\_name, job\_id, hire\_date

FROM employees;

4.

SELECT employee\_id, last\_name, job\_id, hire\_date AS "STARTDATE"

FROM employees;

5.

SELECT DISTINCT job\_id

FROM employees;

6.

SELECT last\_name || ', ' || job\_id AS "EMPLOYEE and TITLE"

FROM employees;

7.

SELECT employee\_id || ', ' || first\_name || ', ' || last\_name || ', ' || email || ', ' || phone\_number || ', ' || hire\_date || ', ' || job\_id || ', ' || salary || ', ' || commission\_pct || ', ' || manager\_id || ', ' || department\_id AS "THE\_OUTPUT"

FROM employees;

**Ex. No.: 4**

**WORKING WITH CONSTRAINTS**

**Initial:**

CREATE TABLE departments (

dept\_id NUMBER(4),

dept\_name VARCHAR2(30),

manager\_id NUMBER(6),

location\_id NUMBER(4)

);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (10, 'HR', 101, 1001);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (20, 'Sales', 102, 1002);

INSERT INTO departments (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (30, 'IT', 103, 1003);

CREATE TABLE EMP (

Employee\_id NUMBER(6) NOT NULL,

First\_Name VARCHAR2(20),

Last\_Name VARCHAR2(25) NOT NULL,

Email VARCHAR2(25) NOT NULL,

Phone\_Number VARCHAR2(20),

Hire\_date DATE NOT NULL,

Job\_id VARCHAR2(10) NOT NULL,

Salary NUMBER(8,2),

Commission\_pct NUMBER(2,2),

Manager\_id NUMBER(6),

Department\_id NUMBER(4)

);

INSERT INTO EMP

VALUES (101, 'John', 'Doe', 'jdoe@example.com', '1234567890', TO\_DATE('2022-06-15', 'YYYY-MM-DD'), 'IT\_PROG', 5000, NULL, 100, 60);

INSERT INTO EMP

VALUES (102, 'Jane', 'Austin', 'jaustin@example.com', '0987654321', TO\_DATE('2022-08-20', 'YYYY-MM-DD'), 'HR\_MAN', 4800, NULL, 101, 70);

INSERT INTO EMP

VALUES (103, 'Mark', 'Smith', 'msmith@example.com', '1230984567', TO\_DATE('2023-01-10', 'YYYY-MM-DD'), 'SA\_REP', 4600, 0.10, 100, 80);

INSERT INTO EMP

VALUES (104, 'Chad', 'Newman', 'cnewman@example.com', '7896541230', TO\_DATE('2021-11-03', 'YYYY-MM-DD'), 'FI\_MGR', 6000, NULL, 102, 60);

INSERT INTO EMP

VALUES (105, 'Betty', 'Austin', 'baustin@example.com', '9874563210', TO\_DATE('2020-12-25', 'YYYY-MM-DD'), 'HR\_CLERK', 3900, NULL, 101, 70);

1.

ALTER TABLE EMP

ADD CONSTRAINT my\_emp\_id\_pk PRIMARY KEY (Employee\_id);

2.

ALTER TABLE DEPARTMENTS

ADD CONSTRAINT my\_dept\_id\_pk PRIMARY KEY (dept\_id);

3.

ALTER TABLE EMP

ADD DEPT\_ID NUMBER(4);

ALTER TABLE EMP

ADD CONSTRAINT my\_emp\_dept\_id\_fk FOREIGN KEY (DEPT\_ID)

REFERENCES DEPARTMENTS(dept\_id);

4.

ALTER TABLE EMP

ADD COMMISSION NUMBER(2,2);

ALTER TABLE EMP

ADD CONSTRAINT chk\_commission\_gt\_zero CHECK (COMMISSION > 0);

**Ex. No.: 5**

**CREATING VIEWS**

**Initial:**

CREATE TABLE JOB\_GRADE (

Grade\_level VARCHAR2(2),

Lowest\_sal NUMBER,

Highest\_sal NUMBER

);

INSERT INTO JOB\_GRADE (Grade\_level, Lowest\_sal, Highest\_sal)

VALUES ('A', 3000, 4999);

INSERT INTO JOB\_GRADE (Grade\_level, Lowest\_sal, Highest\_sal)

VALUES ('B', 5000, 6999);

INSERT INTO JOB\_GRADE (Grade\_level, Lowest\_sal, Highest\_sal)

VALUES ('C', 7000, 9999);

CREATE TABLE DEPARTMENTS (

dept\_id NUMBER(4) PRIMARY KEY,

dept\_name VARCHAR2(30),

manager\_id NUMBER(6),

location\_id NUMBER(4)

);

INSERT INTO DEPARTMENTS (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (80, 'HR', 101, 1001);

INSERT INTO DEPARTMENTS (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (20, 'Sales', 102, 1002);

INSERT INTO DEPARTMENTS (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (30, 'IT', 103, 1003);

INSERT INTO DEPARTMENTS (dept\_id, dept\_name, manager\_id, location\_id)

VALUES (50, 'Support', 104, 1004);

CREATE TABLE EMPLOYEES (

Employee\_id NUMBER(6) NOT NULL,

First\_Name VARCHAR2(20),

Last\_Name VARCHAR2(25) NOT NULL,

Email VARCHAR2(25) NOT NULL,

Phone\_Number VARCHAR2(20),

Hire\_date DATE NOT NULL,

Job\_id VARCHAR2(10) NOT NULL,

Salary NUMBER(8,2),

Commission NUMBER(2,2),

Manager\_id NUMBER(6),

Dept\_ID NUMBER(4),

CONSTRAINT pk\_employee\_id PRIMARY KEY (Employee\_id),

CONSTRAINT fk\_department FOREIGN KEY (Dept\_ID) REFERENCES DEPARTMENTS(dept\_id)

);

INSERT INTO EMPLOYEES (Employee\_id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_date, Job\_id, Salary, Commission, Manager\_id, Dept\_ID)

VALUES (101, 'John', 'Doe', 'jdoe@example.com', '1234567890', TO\_DATE('2022-06-15', 'YYYY-MM-DD'), 'IT\_PROG', 5000, 0.05, 100, 80);

INSERT INTO EMPLOYEES (Employee\_id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_date, Job\_id, Salary, Commission, Manager\_id, Dept\_ID)

VALUES (102, 'Jane', 'Austin', 'jaustin@example.com', '0987654321', TO\_DATE('2022-08-20', 'YYYY-MM-DD'), 'HR\_MAN', 4800, NULL, 101, 50);

INSERT INTO EMPLOYEES (Employee\_id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_date, Job\_id, Salary, Commission, Manager\_id, Dept\_ID)

VALUES (103, 'Mark', 'Smith', 'msmith@example.com', '1230984567', TO\_DATE('2023-01-10', 'YYYY-MM-DD'), 'SA\_REP', 4600, 0.10, 100, 30);

INSERT INTO EMPLOYEES (Employee\_id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_date, Job\_id, Salary, Commission, Manager\_id, Dept\_ID)

VALUES (104, 'Chad', 'Matos', 'cnewman@example.com', '7896541230', TO\_DATE('2021-11-03', 'YYYY-MM-DD'), 'FI\_MGR', 6000, NULL, 102, 50);

INSERT INTO EMPLOYEES (Employee\_id, First\_Name, Last\_Name, Email, Phone\_Number, Hire\_date, Job\_id, Salary, Commission, Manager\_id, Dept\_ID)

VALUES (105, 'Betty', 'Austin', 'baustin@example.com', '9874563210', TO\_DATE('2020-12-25', 'YYYY-MM-DD'), 'HR\_CLERK', 3900, NULL, 101, 20);

1.

CREATE VIEW EMPLOYEE\_VU AS

SELECT Employee\_id,

First\_Name || ' ' || Last\_Name AS EMPLOYEE,

Dept\_ID

FROM EMPLOYEES;

2.

SELECT \* FROM EMPLOYEE\_VU;

3.

SELECT VIEW\_NAME, TEXT

FROM USER\_VIEWS

WHERE VIEW\_NAME = 'EMPLOYEE\_VU';

4.

SELECT EMPLOYEE, Dept\_ID

FROM EMPLOYEE\_VU;

5.

CREATE VIEW DEPT50 AS

SELECT Employee\_id AS EMPNO,

Last\_Name AS EMPLOYEE,

Dept\_ID AS DEPTNO

FROM EMPLOYEES

WHERE Dept\_ID = 50;

6.

DESC DEPT50;

SELECT \* FROM DEPT50;

7.

UPDATE EMPLOYEES

SET Dept\_ID = 80

WHERE Last\_Name = 'Matos';

8.

CREATE VIEW SALARY\_VU AS

SELECT E.Last\_Name AS Employee,

D.dept\_name AS Department,

E.Salary AS Salary,

J.Grade\_level AS Grade

FROM EMPLOYEES E

JOIN DEPARTMENTS D ON E.Dept\_ID = D.dept\_id

JOIN JOB\_GRADE J ON E.Salary BETWEEN J.Lowest\_sal AND J.Highest\_sal;

**Ex. No.: 6**

**RESTRICTING AND SORTING DATA**

**Initial:**

CREATE TABLE EMPLOYEES (

Employee\_id NUMBER(6) NOT NULL,

Last\_Name VARCHAR2(25) NOT NULL,

First\_Name VARCHAR2(20),

Email VARCHAR2(25) NOT NULL,

Phone\_Number VARCHAR2(20),

Hire\_date DATE NOT NULL,

Job\_id VARCHAR2(10) NOT NULL,

Salary NUMBER(8,2),

Commission\_pct NUMBER(2,2),

Manager\_id NUMBER(6),

Department\_id NUMBER(4),

CONSTRAINT pk\_employee\_id PRIMARY KEY (Employee\_id)

);

INSERT INTO EMPLOYEES

VALUES (176, 'Smith', 'John', 'jsmith@example.com', '555-1234', TO\_DATE('1994-07-15', 'YYYY-MM-DD'), 'SA\_REP', 13000, 0.10, NULL, 30);

INSERT INTO EMPLOYEES

VALUES (177, 'Doe', 'Jane', 'jdoe@example.com', '555-5678', TO\_DATE('1998-03-25', 'YYYY-MM-DD'), 'IT\_PROG', 11000, NULL, 176, 20);

INSERT INTO EMPLOYEES

VALUES (178, 'Johnson', 'Emily', 'ejohnson@example.com', '555-8765', TO\_DATE('1995-11-30', 'YYYY-MM-DD'), 'ST\_CLERK', 2500, NULL, 176, 50);

INSERT INTO EMPLOYEES

VALUES (179, 'Miller', 'Tom', 'tmiller@example.com', '555-4321', TO\_DATE('1996-09-10', 'YYYY-MM-DD'), 'SA\_REP', 8000, 0.15, 176, 20);

INSERT INTO EMPLOYEES

VALUES (180, 'Matos', 'Daniel', 'dmatos@example.com', '555-7890', TO\_DATE('1994-05-23', 'YYYY-MM-DD'), 'HR\_CLERK', 3000, NULL, NULL, 50);

INSERT INTO EMPLOYEES

VALUES (196, 'Sharukesh', 'John', 'jsharuk@example.com', '555-1274', TO\_DATE('1999-07-15', 'YYYY-MM-DD'), 'SA\_REP', 16000, 0.10, NULL, 60);

1.

SELECT Last\_Name, Salary

FROM EMPLOYEES

WHERE Salary > 12000;

2.

SELECT Last\_Name, Department\_id

FROM EMPLOYEES

WHERE Employee\_id = 176;

3.

SELECT Last\_Name, Salary

FROM EMPLOYEES

WHERE Salary NOT BETWEEN 5000 AND 12000;

4.

SELECT Last\_Name, Job\_id, Hire\_date

FROM EMPLOYEES

WHERE Hire\_date BETWEEN TO\_DATE('1998-02-20', 'YYYY-MM-DD') AND TO\_DATE('1998-05-01', 'YYYY-MM-DD')

ORDER BY Hire\_date;

5.

SELECT Last\_Name, Department\_id

FROM EMPLOYEES

WHERE Department\_id IN (20, 50)

ORDER BY Last\_Name;

6.

SELECT Last\_Name AS EMPLOYEE, Salary AS "MONTHLY SALARY"

FROM EMPLOYEES

WHERE Salary BETWEEN 5000 AND 12000

AND Department\_id IN (20, 50)

ORDER BY Last\_Name;

7.

SELECT Last\_Name, Hire\_date

FROM EMPLOYEES

WHERE TO\_CHAR(Hire\_date, 'YYYY') = '1994';

8.

SELECT Last\_Name, Job\_id

FROM EMPLOYEES

WHERE Manager\_id IS NULL;

9.

SELECT Last\_Name, Salary, Commission\_pct

FROM EMPLOYEES

WHERE Commission\_pct IS NOT NULL

ORDER BY Salary DESC, Commission\_pct DESC;

10.

SELECT Last\_Name

FROM EMPLOYEES

WHERE Last\_Name LIKE '\_\_a%';

11.

SELECT Last\_Name

FROM EMPLOYEES

WHERE Last\_Name LIKE '%a%' AND Last\_Name LIKE '%e%';

12.

SELECT Last\_Name, Job\_id, Salary

FROM EMPLOYEES

WHERE Job\_id IN ('SA\_REP', 'ST\_CLERK')

AND Salary NOT IN (2500, 3500, 7000);

**Ex. No.: 7**

**USING SET OPERATORS**

**Initial:**

CREATE TABLE EMPLOYEES (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(50),

job\_id VARCHAR2(10),

department\_id NUMBER,

hire\_date DATE

);  
  
CREATE TABLE DEPARTMENTS (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50),

country\_id VARCHAR2(10)

);

CREATE TABLE JOB\_HISTORY (

employee\_id NUMBER,

job\_id VARCHAR2(10) PRIMARY KEY,

hire\_date DATE

);

CREATE TABLE COUNTRIES (

country\_id VARCHAR2(10) PRIMARY KEY,

country\_name VARCHAR2(50)

);

INSERT INTO EMPLOYEES VALUES

(101, 'Smith', 'ST\_CLERK', 10, TO\_DATE('2015-06-01', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(102, 'Johnson', 'SA\_MAN', 50, TO\_DATE('2018-03-12', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(103, 'Williams', 'ST\_CLERK', 20, TO\_DATE('2019-07-14', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(104, 'Brown', 'IT\_PROG', 30, TO\_DATE('2017-11-25', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(105, 'Jones', 'HR\_REP', 40, TO\_DATE('2020-01-03', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(106, 'Garcia', 'ST\_CLERK', 50, TO\_DATE('2015-04-19', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(107, 'Davis', 'IT\_PROG', 20, TO\_DATE('2019-01-01', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(108, 'Taylor', 'SA\_MAN', 10, TO\_DATE('2021-09-12', 'YYYY-MM-DD'));

INSERT INTO EMPLOYEES VALUES

(109, 'Clark', 'IT\_PROG', 30, TO\_DATE('2018-08-01', 'YYYY-MM-DD'));

INSERT INTO DEPARTMENTS

VALUES (10, 'Administration', 'US');

INSERT INTO DEPARTMENTS

VALUES (20, 'Marketing', 'US');

INSERT INTO DEPARTMENTS

VALUES (30, 'IT', 'UK');

INSERT INTO DEPARTMENTS

VALUES (40, 'HR', 'FR');

INSERT INTO DEPARTMENTS

VALUES (50, 'Sales', 'DE');

INSERT INTO DEPARTMENTS

VALUES (60, 'Finance', 'IN');

INSERT INTO JOB\_HISTORY

VALUES (101, 'ST\_CLERK', TO\_DATE('2015-06-01', 'YYYY-MM-DD'));

INSERT INTO JOB\_HISTORY

VALUES (102, 'SA\_MAN', TO\_DATE('2018-03-12', 'YYYY-MM-DD'));

INSERT INTO JOB\_HISTORY

VALUES (107, 'IT\_PROG',TO\_DATE('2019-01-01', 'YYYY-MM-DD'));

INSERT INTO COUNTRIES

VALUES ('US', 'United States');

INSERT INTO COUNTRIES

VALUES ('UK', 'United Kingdom');

INSERT INTO COUNTRIES

VALUES ('FR', 'France');

INSERT INTO COUNTRIES

VALUES ('DE', 'Germany');

INSERT INTO COUNTRIES

VALUES ('IN', 'India');

INSERT INTO COUNTRIES

VALUES ('JP', 'Japan');

1.

SELECT department\_id

FROM DEPARTMENTS

MINUS

SELECT department\_id

FROM EMPLOYEES

WHERE job\_id = 'ST\_CLERK';

2.

SELECT country\_id, country\_name

FROM COUNTRIES

WHERE country\_id IN (

SELECT country\_id FROM COUNTRIES

MINUS

SELECT DISTINCT country\_id FROM DEPARTMENTS

WHERE department\_name='HR'

);

3.

SELECT job\_id, department\_id

FROM EMPLOYEES

WHERE department\_id = 10

UNION ALL

SELECT job\_id, department\_id

FROM EMPLOYEES

WHERE department\_id = 50

UNION ALL

SELECT job\_id, department\_id

FROM EMPLOYEES

WHERE department\_id = 20;

4.

SELECT employee\_id, job\_id, hire\_date

FROM EMPLOYEES

INTERSECT

SELECT employee\_id, job\_id, hire\_date

FROM JOB\_HISTORY

ORDER BY hire\_date ASC;

5.

SELECT last\_name, department\_id, NULL AS department\_name

FROM EMPLOYEES

UNION

SELECT NULL AS last\_name, department\_id, department\_name

FROM DEPARTMENTS;

**Ex. No.: 8**

**WORKING WITH MULTIPLE TABLES**

**Initial:**

CREATE TABLE EMPLOYEES (

EMPLOYEE\_ID NUMBER(6) PRIMARY KEY,

FIRST\_NAME VARCHAR2(20),

LAST\_NAME VARCHAR2(25) NOT NULL,

EMAIL VARCHAR2(50) UNIQUE NOT NULL,

PHONE\_NUMBER VARCHAR2(20),

HIRE\_DATE DATE NOT NULL,

JOB\_ID VARCHAR2(10) NOT NULL,

SALARY NUMBER(8,2),

COMMISSION\_PCT NUMBER(2,2),

MANAGER\_ID NUMBER(6),

DEPARTMENT\_ID NUMBER(4)

);

CREATE TABLE DEPARTMENTS (

DEPARTMENT\_ID NUMBER(4) PRIMARY KEY,

DEPARTMENT\_NAME VARCHAR2(30) NOT NULL,

MANAGER\_ID NUMBER(6),

LOCATION\_ID NUMBER(4)

);

CREATE TABLE JOBS (

JOB\_ID VARCHAR2(10) PRIMARY KEY,

JOB\_TITLE VARCHAR2(35) NOT NULL,

MIN\_SALARY NUMBER(8,2),

MAX\_SALARY NUMBER(8,2)

);

CREATE TABLE LOCATIONS (

LOCATION\_ID NUMBER(4) PRIMARY KEY,

STREET\_ADDRESS VARCHAR2(40),

POSTAL\_CODE VARCHAR2(12),

CITY VARCHAR2(30) NOT NULL,

COUNTRY VARCHAR2(25),

COUNTRY\_CODE VARCHAR2(20)

);

CREATE TABLE JOB\_GRADES (

GRADE\_LEVEL CHAR(1) PRIMARY KEY,

LOW\_SALARY NUMBER(8,2),

HIGH\_SALARY NUMBER(8,2)

);

INSERT INTO LOCATIONS VALUES

(1000, '123 Main St', '560001', 'Toronto', 'Ontario', 'CA');

INSERT INTO LOCATIONS VALUES

(1001, '456 Park Ave', '110020', 'New York', 'New York', 'US');

INSERT INTO LOCATIONS VALUES

(1002, '789 King Rd', '700008', 'London', 'England', 'UK');

INSERT INTO LOCATIONS VALUES

(1003, '696 VOC Rd', '600098', 'Chennai', 'India', 'IND');

INSERT INTO DEPARTMENTS VALUES

(10, 'Administration', NULL, 1001);

INSERT INTO DEPARTMENTS VALUES

(20, 'Marketing', 101, 1002);

INSERT INTO DEPARTMENTS VALUES

(30, 'IT', 102, 1001);

INSERT INTO DEPARTMENTS VALUES

(40, 'HR', 103, 1000);

INSERT INTO DEPARTMENTS VALUES

(50, 'Sales', 104, 1000);

INSERT INTO DEPARTMENTS VALUES

(80, 'Finance', 105, 1003);

INSERT INTO JOBS VALUES

('AD\_PRES', 'President', 20000, 40000);

INSERT INTO JOBS VALUES

('MK\_MAN', 'Marketing Manager', 10000, 20000);

INSERT INTO JOBS VALUES

('IT\_PROG', 'Programmer', 5000, 15000);

INSERT INTO JOBS VALUES

('HR\_REP', 'HR Representative', 6000, 12000);

INSERT INTO JOBS VALUES

('FI\_MGR', 'Finance Manager', 12000, 25000);

INSERT INTO JOBS VALUES

('SA\_REP', 'Sales Representative', 5000, 10000);

INSERT INTO JOB\_GRADES VALUES

('A', 5000, 7000);

INSERT INTO JOB\_GRADES VALUES

('B', 7001, 12000);

INSERT INTO JOB\_GRADES VALUES

('C', 12001, 15000);

INSERT INTO JOB\_GRADES VALUES

('D', 15001, 20000);

INSERT INTO JOB\_GRADES VALUES

('E', 20001, 40000);

INSERT INTO EMPLOYEES VALUES

(101, 'John', 'King', 'JKing@example.com', '1234567890', TO\_DATE('2010-01-01', 'YYYY-MM-DD'), 'AD\_PRES', 30000, NULL, NULL, 10);

INSERT INTO EMPLOYEES VALUES

(102, 'Sara', 'Davies', 'SDavies@example.com', '2234567890', TO\_DATE('2013-05-10', 'YYYY-MM-DD'), 'MK\_MAN', 15000, NULL, 101, 20);

INSERT INTO EMPLOYEES VALUES

(103, 'Mike', 'Smith', 'MSmith@example.com', '3234567890', TO\_DATE('2012-03-15', 'YYYY-MM-DD'), 'IT\_PROG', 9000, NULL, 102, 80);

INSERT INTO EMPLOYEES VALUES

(104, 'Anna', 'Brown', 'ABrown@example.com', '4234567890', TO\_DATE('2013-09-20', 'YYYY-MM-DD'), 'HR\_REP', 7000, 0.10, 102, 40);

INSERT INTO EMPLOYEES VALUES

(105, 'James', 'Wilson', 'JWilson@example.com', '5234567890', TO\_DATE('2014-07-23', 'YYYY-MM-DD'), 'FI\_MGR', 18000, NULL, 101, 80);

INSERT INTO EMPLOYEES VALUES

(106, 'Sophia', 'Johnson', 'SJohnson@example.com', '6234567890', TO\_DATE('2015-11-05', 'YYYY-MM-DD'), 'SA\_REP', 8000, 0.15, 103, 50);

INSERT INTO EMPLOYEES VALUES

(107, 'Emily', 'Taylor', 'ETaylor@example.com', '7234567890', TO\_DATE('2016-04-18', 'YYYY-MM-DD'), 'SA\_REP', 8500, 0.12, 104, 50);

1.

SELECT e.LAST\_NAME, e.DEPARTMENT\_ID, d.DEPARTMENT\_NAME

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID;

2.

SELECT DISTINCT e.JOB\_ID, d.LOCATION\_ID, l.COUNTRY

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

JOIN LOCATIONS l ON d.LOCATION\_ID=l.LOCATION\_ID

WHERE e.DEPARTMENT\_ID = 80;

3.

SELECT e.LAST\_NAME, d.DEPARTMENT\_NAME, d.LOCATION\_ID, l.CITY

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

JOIN LOCATIONS l ON d.LOCATION\_ID = l.LOCATION\_ID

WHERE e.COMMISSION\_PCT IS NOT NULL;

4.

SELECT e.LAST\_NAME, d.DEPARTMENT\_NAME

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

WHERE LOWER(e.LAST\_NAME) LIKE '%a%';

5.

SELECT e.LAST\_NAME, e.JOB\_ID, e.DEPARTMENT\_ID, d.DEPARTMENT\_NAME

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

JOIN LOCATIONS l ON d.LOCATION\_ID = l.LOCATION\_ID

WHERE l.CITY = 'Toronto';

6.

SELECT e.LAST\_NAME AS "Employee", e.EMPLOYEE\_ID AS "Emp#",

m.LAST\_NAME AS "Manager", m.EMPLOYEE\_ID AS "Mgr#"

FROM EMPLOYEES e

JOIN EMPLOYEES m ON e.MANAGER\_ID = m.EMPLOYEE\_ID;

7.

SELECT e.LAST\_NAME AS "Employee", e.EMPLOYEE\_ID AS "Emp#",

m.LAST\_NAME AS "Manager", m.EMPLOYEE\_ID AS "Mgr#"

FROM EMPLOYEES e

LEFT JOIN EMPLOYEES m ON e.MANAGER\_ID = m.EMPLOYEE\_ID

ORDER BY e.EMPLOYEE\_ID;

8.

SELECT e1.LAST\_NAME AS "Employee", e1.DEPARTMENT\_ID, e2.LAST\_NAME AS "Co-Workers"

FROM EMPLOYEES e1

JOIN EMPLOYEES e2 ON e1.DEPARTMENT\_ID = e2.DEPARTMENT\_ID

WHERE e1.EMPLOYEE\_ID = 106 AND e1.EMPLOYEE\_ID <> e2.EMPLOYEE\_ID;

9.

DESCRIBE JOB\_GRADES;

SELECT e.LAST\_NAME, e.JOB\_ID, d.DEPARTMENT\_NAME, e.SALARY, jg.GRADE\_LEVEL

FROM EMPLOYEES e

JOIN DEPARTMENTS d ON e.DEPARTMENT\_ID = d.DEPARTMENT\_ID

JOIN JOB\_GRADES jg ON e.SALARY BETWEEN jg.LOW\_SALARY AND jg.HIGH\_SALARY;

10.

SELECT e.LAST\_NAME AS "Employee", e.HIRE\_DATE AS "Hire Date"

FROM EMPLOYEES e

JOIN EMPLOYEES r ON r.LAST\_NAME = 'Davies'

WHERE e.HIRE\_DATE > r.HIRE\_DATE;

11.

SELECT e.LAST\_NAME AS "Employee", e.HIRE\_DATE AS "Emp Hired",

m.LAST\_NAME AS "Manager", m.HIRE\_DATE AS "Mgr Hired"

FROM EMPLOYEES e

JOIN EMPLOYEES m ON e.MANAGER\_ID = m.EMPLOYEE\_ID

WHERE e.HIRE\_DATE < m.HIRE\_DATE AND e.EMPLOYEE\_ID <> m.EMPLOYEE\_ID;

**Ex. No.: 9**

**SUB QUERIES**

**Initial:**

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(100),

location\_id NUMBER

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(100),

first\_name VARCHAR2(100),

hire\_date DATE,

salary NUMBER(10, 2),

department\_id NUMBER,

job\_id VARCHAR2(10),

manager\_id NUMBER,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

INSERT INTO departments VALUES (10, 'Executive', 1700);

INSERT INTO departments VALUES (20, 'HR', 1800);

INSERT INTO departments VALUES (30, 'IT', 1700);

INSERT INTO departments VALUES (40, 'Finance', 1600);

INSERT INTO employees

VALUES (1, 'King', 'John', TO\_DATE('2000-01-01', 'YYYY-MM-DD'), 10000, 10, 'CEO', NULL);

INSERT INTO employees

VALUES (2, 'Zlotkey', 'Jane', TO\_DATE('2001-02-15', 'YYYY-MM-DD'), 8000, 10, 'VP', 1);

INSERT INTO employees

VALUES (3, 'Smith', 'Anna', TO\_DATE('2005-03-10', 'YYYY-MM-DD'), 8500, 10, 'Manager', 1);

INSERT INTO employees

VALUES (4, 'Green', 'Tom', TO\_DATE('2010-05-20', 'YYYY-MM-DD'), 4500, 20, 'HR Rep', 2);

INSERT INTO employees

VALUES (5, 'Brown', 'Lily', TO\_DATE('2011-06-22', 'YYYY-MM-DD'), 4200, 20, 'HR Rep', 2);

INSERT INTO employees

VALUES (6, 'Turner', 'Michael', TO\_DATE('2012-07-13', 'YYYY-MM-DD'), 5000, 30, 'Developer', 3);

INSERT INTO employees

VALUES (7, 'Miller', 'Sandra', TO\_DATE('2014-08-25', 'YYYY-MM-DD'), 5500, 30, 'Developer', 3);

INSERT INTO employees

VALUES (8, 'Jones', 'Peter', TO\_DATE('2018-09-15', 'YYYY-MM-DD'), 6000, 40, 'Accountant', 1);

INSERT INTO employees

VALUES (9, 'Austin', 'James', TO\_DATE('2014-06-13', 'YYYY-MM-DD'), 7500, 30, 'Developer', 1);

1.

SELECT last\_name, hire\_date

FROM employees

WHERE department\_id = (

SELECT department\_id FROM employees

WHERE last\_name = 'Zlotkey'

)

AND last\_name != 'Zlotkey';

2.

SELECT employee\_id, last\_name, salary

FROM employees

WHERE salary > (

SELECT AVG(salary) FROM employees

)

ORDER BY salary;

3.

SELECT employee\_id, last\_name

FROM employees

WHERE department\_id IN (

SELECT department\_id FROM employees

WHERE last\_name LIKE '%u%'

);

4.

SELECT last\_name, department\_id, job\_id

FROM employees

WHERE department\_id IN (

SELECT department\_id FROM departments

WHERE location\_id=1700

);

5.

SELECT last\_name, salary

FROM employees e

WHERE EXISTS(

SELECT last\_name FROM employees m

WHERE e.manager\_id = m.employee\_id

AND m.last\_name='King'

);

6.

SELECT department\_id, last\_name, job\_id

FROM employees

WHERE department\_id = (

SELECT department\_id

FROM departments

WHERE department\_name = 'Executive'

);

7.

SELECT e.employee\_id, e.last\_name, e.salary

FROM employees e

WHERE e.salary > (SELECT AVG(salary) FROM employees)

AND EXISTS (

SELECT \*

FROM employees e2

WHERE e.department\_id = e2.department\_id

AND e2.last\_name LIKE '%u%'

);

**Ex. No.: 10**

**AGGREGATING DATA USING GROUP FUNCTIONS**

**Initial:**

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(100),

location\_id NUMBER

);

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

last\_name VARCHAR2(100),

first\_name VARCHAR2(100),

hire\_date DATE,

salary NUMBER(10, 2),

department\_id NUMBER,

job\_id VARCHAR2(10),

manager\_id NUMBER,

FOREIGN KEY (department\_id) REFERENCES departments(department\_id)

);

INSERT INTO departments VALUES (10, 'Executive', 1700);

INSERT INTO departments VALUES (20, 'HR', 1800);

INSERT INTO departments VALUES (30, 'IT', 1700);

INSERT INTO departments VALUES (40, 'Finance', 1600);

INSERT INTO employees

VALUES (1, 'King', 'John', TO\_DATE('1998-01-01', 'YYYY-MM-DD'), 10000, 10, 'CEO', NULL);

INSERT INTO employees

VALUES (2, 'Zlotkey', 'Jane', TO\_DATE('1995-02-15', 'YYYY-MM-DD'), 8000, 10, 'VP', 1);

INSERT INTO employees

VALUES (3, 'Smith', 'Anna', TO\_DATE('1996-03-10', 'YYYY-MM-DD'), 8500, 10, 'Manager', 1);

INSERT INTO employees

VALUES (4, 'Green', 'Tom', TO\_DATE('1998-05-20', 'YYYY-MM-DD'), 7500, 20, 'HR Rep', 2);

INSERT INTO employees

VALUES (5, 'Brown', 'Lily', TO\_DATE('1997-06-22', 'YYYY-MM-DD'), 7200, 20, 'HR Rep', 2);

INSERT INTO employees

VALUES (6, 'Turner', 'Michael', TO\_DATE('1995-07-13', 'YYYY-MM-DD'), 5000, 30, 'Developer', 3);

INSERT INTO employees

VALUES (7, 'Miller', 'Sandra', TO\_DATE('1992-08-25', 'YYYY-MM-DD'), 5500, 30, 'Developer', 3);

INSERT INTO employees

VALUES (8, 'Jones', 'Peter', TO\_DATE('1997-09-15', 'YYYY-MM-DD'), 6500, 40, 'Accountant', 1);

INSERT INTO employees

VALUES (9, 'Austin', 'James', TO\_DATE('1996-06-13', 'YYYY-MM-DD'), 7500, 30, 'Developer', 1);

1. TRUE

2. FALSE

3. TRUE

4.

SELECT

ROUND(MAX(salary)) AS Maximum,

ROUND(MIN(salary)) AS Minimum,

ROUND(SUM(salary)) AS Sum,

ROUND(AVG(salary)) AS Average

FROM employees;

5.

SELECT

job\_id,

ROUND(MIN(salary)) AS Minimum,

ROUND(MAX(salary)) AS Maximum,

ROUND(SUM(salary)) AS Sum,

ROUND(AVG(salary)) AS Average

FROM employees

GROUP BY job\_id;

6.

SELECT

job\_id,

COUNT(\*) AS Number\_of\_People

FROM employees

WHERE job\_id = 'Developer'

GROUP BY job\_id;

7.

SELECT

COUNT(DISTINCT manager\_id) AS Number\_of\_Managers

FROM employees

WHERE manager\_id IS NOT NULL;

8.

SELECT

ROUND(MAX(salary) - MIN(salary)) AS DIFFERENCE

FROM employees;

9.

SELECT

manager\_id,

MIN(salary) AS Lowest\_Salary

FROM employees

WHERE manager\_id IS NOT NULL

GROUP BY manager\_id

HAVING MIN(salary) > 6000

ORDER BY Lowest\_Salary DESC;

10.

SELECT

COUNT(\*) AS Total\_Employees,

SUM(CASE WHEN EXTRACT(YEAR FROM hire\_date) = 1995 THEN 1 ELSE 0 END) AS Employees\_1995,

SUM(CASE WHEN EXTRACT(YEAR FROM hire\_date) = 1996 THEN 1 ELSE 0 END) AS Employees\_1996,

SUM(CASE WHEN EXTRACT(YEAR FROM hire\_date) = 1997 THEN 1 ELSE 0 END) AS Employees\_1997,

SUM(CASE WHEN EXTRACT(YEAR FROM hire\_date) = 1998 THEN 1 ELSE 0 END) AS Employees\_1998

FROM employees;

11.

SELECT

job\_id,

department\_id,

SUM(salary) AS Total\_Salary,

AVG(salary) AS Average\_Salary

FROM employees

WHERE department\_id IN (20, 50, 80, 90)

GROUP BY job\_id, department\_id

ORDER BY department\_id, job\_id;

12.

SELECT

d.department\_name AS "Name-Location",

d.location\_id AS Location,

COUNT(e.employee\_id) AS "Number of People",

ROUND(AVG(e.salary), 2) AS Salary

FROM departments d

LEFT JOIN employees e ON d.department\_id = e.department\_id

GROUP BY d.department\_name, d.location\_id;

**Ex. No.: 11**

**PL SQL PROGRAMS**

**Initial:**

CREATE TABLE employees (

employee\_id NUMBER PRIMARY KEY,

first\_name VARCHAR2(50),

last\_name VARCHAR2(50),

job\_id VARCHAR2(10),

salary NUMBER(8, 2),

hire\_date DATE,

department\_id NUMBER

);

CREATE TABLE departments (

department\_id NUMBER PRIMARY KEY,

department\_name VARCHAR2(50),

manager\_id NUMBER

);

CREATE TABLE jobs (

job\_id VARCHAR2(10) PRIMARY KEY,

job\_title VARCHAR2(50),

min\_salary NUMBER(8, 2), max\_salary NUMBER(8, 2)

);

CREATE TABLE job\_history (

employee\_id NUMBER,

start\_date DATE, end\_date DATE,

job\_id VARCHAR2(10),

department\_id NUMBER

);

BEGIN

INSERT INTO employees VALUES (110, 'John', 'Doe', 'IT\_PROG', 60000, TO\_DATE('2020-01-15', 'YYYY-MM-DD'), 50);

INSERT INTO employees VALUES (122, 'Jane', 'Smith', 'SA\_REP', 55000, TO\_DATE('2019-07-10', 'YYYY-MM-DD'), 80);

INSERT INTO departments VALUES (50, 'IT', 110);

INSERT INTO departments VALUES (80, 'Sales', 122);

INSERT INTO jobs VALUES ('IT\_PROG', 'Programmer', 40000, 80000);

INSERT INTO jobs VALUES ('SA\_REP', 'Sales Representative', 30000, 60000);

INSERT INTO job\_history VALUES (110, TO\_DATE('2018-05-01', 'YYYY-MM-DD'), TO\_DATE('2020-01-14', 'YYYY-MM-DD'), 'HR\_REP', 60);

INSERT INTO job\_history VALUES (122, TO\_DATE('2017-03-01', 'YYYY-MM-DD'), TO\_DATE('2019-07-09', 'YYYY-MM-DD'), 'SA\_REP', 80);

END;/

1.

DECLARE

emp\_salary employees.salary%TYPE;

incentive NUMBER(8,2);

BEGIN

SELECT salary INTO emp\_salary FROM employees WHERE employee\_id = 110;

incentive := emp\_salary \* 0.1;

DBMS\_OUTPUT.PUT\_LINE('Incentive for Employee ID 110: ' || incentive);

END; /

2.

DECLARE

"EmployeeID" NUMBER := 110;

BEGIN

DBMS\_OUTPUT.PUT\_LINE(EmployeeID);

END; /

3.

BEGIN

UPDATE employees SET salary = salary + 5000 WHERE employee\_id = 122;

DBMS\_OUTPUT.PUT\_LINE('Salary adjusted for Employee ID 122');

END; /

4.

CREATE OR REPLACE PROCEDURE CheckNullAndOperator IS

value1 BOOLEAN := TRUE;

value2 BOOLEAN := TRUE;

BEGIN

IF value1 IS NOT NULL AND value2 IS NOT NULL AND value1 AND value2 THEN

DBMS\_OUTPUT.PUT\_LINE('Both conditions are TRUE');

ELSE

DBMS\_OUTPUT.PUT\_LINE('One or both conditions are FALSE');

END IF;

END; /

5.

DECLARE

emp\_name employees.first\_name%TYPE;

BEGIN

FOR rec IN (SELECT first\_name FROM employees WHERE first\_name LIKE 'J%')

LOOP

DBMS\_OUTPUT.PUT\_LINE('Employee name starting with J: ' || rec.first\_name);

END LOOP;

END; /

6.

DECLARE

num1 NUMBER := 10;

num2 NUMBER := 5;

num\_small NUMBER;

num\_large NUMBER;

BEGIN

IF num1 < num2 THEN

num\_small := num1;

num\_large := num2;

ELSE

num\_small := num2;

num\_large := num1;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Small Number: ' || num\_small || ', Large Number: ' || num\_large);

END; /

7.

CREATE OR REPLACE PROCEDURE UpdateIncentive IS

target NUMBER := 100000;

sales NUMBER := 120000;

incentive NUMBER;

BEGIN

IF sales >= target THEN

incentive := sales \* 0.1;

DBMS\_OUTPUT.PUT\_LINE('Incentive updated to ' || incentive);

ELSE

DBMS\_OUTPUT.PUT\_LINE('Target not met. No incentive.');

END IF;

END; /

8.

CREATE OR REPLACE PROCEDURE CalculateIncentive(sales\_limit IN NUMBER) IS

incentive NUMBER;

BEGIN

IF sales\_limit > 50000 THEN

incentive := sales\_limit \* 0.15;

ELSE

incentive := sales\_limit \* 0.1;

END IF;

DBMS\_OUTPUT.PUT\_LINE('Incentive: ' || incentive);

END; /

9.

DECLARE

emp\_count NUMBER;

vacancies NUMBER := 45;

BEGIN

SELECT COUNT(\*) INTO emp\_count FROM employees WHERE department\_id = 50;

IF emp\_count < vacancies THEN

DBMS\_OUTPUT.PUT\_LINE('Vacancies available: ' || (vacancies - emp\_count));

ELSE

DBMS\_OUTPUT.PUT\_LINE('No vacancies');

END IF;

END; /

10.

DECLARE

emp\_count NUMBER;

dept\_id NUMBER := 80;

vacancies NUMBER := 45;

BEGIN

SELECT COUNT(\*) INTO emp\_count FROM employees WHERE department\_id = dept\_id;

IF emp\_count < vacancies THEN

DBMS\_OUTPUT.PUT\_LINE('Vacancies in Department ' || dept\_id || ': ' || (vacancies - emp\_count));

ELSE

DBMS\_OUTPUT.PUT\_LINE('No vacancies');

END IF;

END; /

11.

DECLARE

CURSOR emp\_cursor IS

SELECT employee\_id, first\_name, job\_id, hire\_date, salary FROM employees;

BEGIN

FOR emp IN emp\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('ID: ' || emp.employee\_id || ', Name: ' || emp.first\_name || ', Job: ' || emp.job\_id || ', Hire Date: ' || emp.hire\_date || ', Salary: ' || emp.salary);

END LOOP;

END; /

12.

DECLARE

CURSOR emp\_dept\_cursor IS

SELECT e.employee\_id, e.first\_name, d.department\_name

FROM employees e

JOIN departments d ON e.department\_id = d.department\_id;

BEGIN

FOR emp IN emp\_dept\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('ID: ' || emp.employee\_id || ', Name: ' || emp.first\_name || ', Dept: ' || emp.department\_name);

END LOOP;

END; /

13.

DECLARE

CURSOR job\_cursor IS

SELECT job\_id, job\_title, min\_salary FROM jobs;

BEGIN

FOR job IN job\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Job ID: ' || job.job\_id || ', Title: ' || job.job\_title || ', Min Salary: ' || job.min\_salary);

END LOOP;

END; /

14.

DECLARE

CURSOR job\_hist\_cursor IS

SELECT employee\_id, start\_date FROM job\_history;

BEGIN

FOR job\_hist IN job\_hist\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || job\_hist.employee\_id || ', Start Date: ' || job\_hist.start\_date);

END LOOP;

END; /

15.

DECLARE

CURSOR job\_hist\_cursor IS

SELECT employee\_id, end\_date FROM job\_history;

BEGIN

FOR job\_hist IN job\_hist\_cursor LOOP

DBMS\_OUTPUT.PUT\_LINE('Employee ID: ' || job\_hist.employee\_id || ', End Date: ' || job\_hist.end\_date);

END LOOP;

END; /

**Ex. No.: 12**

**WORKING WITH CURSOR, PROCEDURES AND FUNCTIONS**

**1.**

CREATE OR REPLACE FUNCTION factorial(n NUMBER) RETURN NUMBER IS

result NUMBER := 1;

BEGIN

IF n < 0 THEN

RETURN NULL;

ELSIF n = 0 THEN

RETURN 1;

ELSE

FOR i IN 1..n LOOP

result := result \* i;

END LOOP;

END IF;

RETURN result;

END factorial; /

DECLARE

num NUMBER := 5;

fact NUMBER;

BEGIN

fact := factorial(num);

DBMS\_OUTPUT.PUT\_LINE('Factorial of ' || num || ' is: ' || fact);

END; /

**2. Initial:**

CREATE TABLE books (

book\_id NUMBER PRIMARY KEY,

title VARCHAR2(100),

author VARCHAR2(100),

genre VARCHAR2(50),

publication\_year NUMBER

); /

BEGIN

INSERT INTO books VALUES (1, '1984', 'George Orwell', 'Dystopian', 1949);

INSERT INTO books VALUES (2, 'To Kill a Mockingbird', 'Harper Lee', 'Fiction', 1960);

INSERT INTO books VALUES (3, 'The Great Gatsby', 'F. Scott Fitzgerald', 'Classic', 1925);

INSERT INTO books VALUES (4, 'Moby-Dick', 'Herman Melville', 'Adventure', 1851);

INSERT INTO books VALUES (5, 'Pride and Prejudice', 'Jane Austen', 'Romance', 1813);

END; /

**2.**

CREATE OR REPLACE PROCEDURE get\_book\_info (

p\_book\_id IN NUMBER,

p\_title IN OUT VARCHAR2,

p\_author OUT VARCHAR2,

p\_genre OUT VARCHAR2,

p\_publication\_year OUT NUMBER

) IS

BEGIN

SELECT title, author, genre, publication\_year

INTO p\_title, p\_author, p\_genre, p\_publication\_year

FROM books

WHERE book\_id = p\_book\_id;

EXCEPTION

WHEN NO\_DATA\_FOUND THEN

DBMS\_OUTPUT.PUT\_LINE('No book found with ID: ' || p\_book\_id);

END get\_book\_info; /

DECLARE

book\_id NUMBER := 3;

title VARCHAR2(100) := 'Default Title';

author VARCHAR2(100);

genre VARCHAR2(50);

publication\_year NUMBER;

BEGIN

get\_book\_info(book\_id, title, author, genre, publication\_year);

DBMS\_OUTPUT.PUT\_LINE('Title: ' || title);

DBMS\_OUTPUT.PUT\_LINE('Author: ' || author);

DBMS\_OUTPUT.PUT\_LINE('Genre: ' || genre);

DBMS\_OUTPUT.PUT\_LINE('Publication Year: ' || publication\_year);

END; /

**Ex. No.: 13**

**WORKING WITH TRIGGER**

**Initial:**

CREATE TABLE orders (

order\_id NUMBER PRIMARY KEY,

item\_id NUMBER,

quantity NUMBER,

order\_date DATE,

running\_total NUMBER,

user\_id NUMBER,

FOREIGN KEY (item\_id) REFERENCES items(item\_id)

);

INSERT INTO orders (order\_id, item\_id, quantity, order\_date, running\_total, user\_id)

VALUES (1, 1, 20, SYSDATE, 20, 101);

INSERT INTO orders (order\_id, item\_id, quantity, order\_date, running\_total, user\_id)

VALUES (2, 2, 30, SYSDATE, 50, 102);

CREATE TABLE items (

item\_id NUMBER PRIMARY KEY,

item\_name VARCHAR2(50),

stock\_level NUMBER,

pending\_orders NUMBER DEFAULT 0

);

INSERT INTO items (item\_id, item\_name, stock\_level, pending\_orders)

VALUES (1, 'Item A', 100, 0);

INSERT INTO items (item\_id, item\_name, stock\_level, pending\_orders)

VALUES (2, 'Item B', 50, 0);

INSERT INTO items (item\_id, item\_name, stock\_level, pending\_orders)

VALUES (3, 'Item C', 150, 0);

CREATE TABLE audit\_log (

log\_id NUMBER PRIMARY KEY,

table\_name VARCHAR2(50),

operation VARCHAR2(10),

change\_time TIMESTAMP DEFAULT CURRENT\_TIMESTAMP,

user\_id NUMBER,

details VARCHAR2(200)

);

CREATE SEQUENCE audit\_log\_seq

START WITH 1

INCREMENT BY 1;

**1.**

CREATE OR REPLACE TRIGGER prevent\_parent\_delete

BEFORE DELETE ON items

FOR EACH ROW

DECLARE

child\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO child\_count FROM orders

WHERE item\_id = :OLD.item\_id;

IF child\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20001, 'Cannot delete item; dependent orders exist.');

END IF;

END; /

**2.**

CREATE OR REPLACE TRIGGER check\_for\_duplicates

BEFORE INSERT OR UPDATE ON orders

FOR EACH ROW

DECLARE

duplicate\_count NUMBER;

BEGIN

SELECT COUNT(\*) INTO duplicate\_count FROM orders

WHERE item\_id = :NEW.item\_id AND order\_id != :NEW.order\_id;

IF duplicate\_count > 0 THEN

RAISE\_APPLICATION\_ERROR(-20002, 'Duplicate item entry found in orders.');

END IF;

END; /

**3.**

CREATE OR REPLACE TRIGGER restrict\_insertion

BEFORE INSERT ON orders

FOR EACH ROW

DECLARE

total\_quantity NUMBER;

BEGIN

SELECT SUM(quantity) INTO total\_quantity FROM orders;

IF (total\_quantity + :NEW.quantity) > 500 THEN

RAISE\_APPLICATION\_ERROR(-20003, 'Cannot insert order; total quantity exceeds threshold.');

END IF;

END; /

**4.**

CREATE OR REPLACE TRIGGER log\_changes

AFTER UPDATE ON orders

FOR EACH ROW

BEGIN

INSERT INTO audit\_log (log\_id, table\_name, operation, user\_id, details) VALUES (audit\_log\_seq.NEXTVAL, 'orders', 'UPDATE', :NEW.user\_id, 'Order ' || :NEW.order\_id || ' changed from ' || :OLD.quantity || ' to ' || :NEW.quantity );

END; /

**5.**

CREATE OR REPLACE TRIGGER log\_user\_activity

AFTER INSERT OR DELETE OR UPDATE ON orders

FOR EACH ROW

BEGIN

INSERT INTO audit\_log (log\_id, table\_name, operation, user\_id, details) VALUES (audit\_log\_seq.NEXTVAL, 'orders',

CASE

WHEN INSERTING THEN 'INSERT'

WHEN UPDATING THEN 'UPDATE'

WHEN DELETING THEN 'DELETE'

END,

NVL(:NEW.user\_id, :OLD.user\_id), 'User action recorded on order ' || NVL(:NEW.order\_id, :OLD.order\_id));

END; /

**7.**

CREATE OR REPLACE TRIGGER update\_running\_total

AFTER INSERT ON orders

FOR EACH ROW

BEGIN

UPDATE orders SET running\_total = (SELECT SUM(quantity) FROM orders) WHERE order\_id = :NEW.order\_id;

END; /

**8.**

CREATE OR REPLACE TRIGGER validate\_item\_availability

BEFORE INSERT ON orders

FOR EACH ROW

DECLARE

available\_stock NUMBER;

BEGIN

SELECT stock\_level - pending\_orders INTO available\_stock FROM items

WHERE item\_id = :NEW.item\_id;

IF :NEW.quantity > available\_stock THEN

RAISE\_APPLICATION\_ERROR(-20004, 'Insufficient stock available for the order.');

END IF;

UPDATE items SET pending\_orders = pending\_orders + :NEW.quantity

WHERE item\_id = :NEW.item\_id;

END; /