Practical – 2

Aim: To study Basic SQL commands (create database, create table, use , drop, insert) and execute the following queries using these commands:

- Create a database named ‘ Employee’.
- Use the database ‘Employee’ and create a table ‘Emp’ with attributes ‘ename’,’ecity’,’salary’,’enumber’,’eaddress’,’depttname’.
- Create another table ‘Company’ with attributes ‘cname’, ccity’,’empnumber’ in the database ‘Employee

Queries:

1) CREATE DATABASE Employee;

2) USE Employee;

```
CREATE TABLE Emp (
    ename VARCHAR(20),
    ecity VARCHAR(20),
    salary int,
    enumber INT,
    eaddress VARCHAR(20),
    depttname VARCHAR(20)
);

INSERT INTO Emp (ename, ecity, salary, enumber, eaddress, depttname)
VALUES ('Abhay', 'Delhi', 50000.00, 101, '123 Main St', 'CS'),
('Raj', 'Gurgaon', 60000.00, 102, '456 Elm St', 'HR'),
('Bob', 'Chicago', 55000.00, 103, '789 Oak St', 'IT');
```

3)

```
CREATE TABLE Company (
 cname VARCHAR(255),
 ccity VARCHAR(255),
 empnumber INT
);


INSERT INTO Company (cname, ccity, empnumber)
VALUES ('TCS', 'Delhi', 101),
       ('Google', 'Los Angeles', 102),
       ('Google', 'Gurgoan', 103),
       ('Meta', 'Chicago', 103);
```

Practical – 3

Aim: To study the viewing commands (select , update) and execute the following queries using these commands:

- Find the names of all employees who live in Delhi.
- Increase the salary of all employees by Rs. 5,000.
- Find the company names where the number of employees is greater than 10,000.
- Change the Company City to Gurgaon where the Company name is ‘TCS’

Queries:

a) SELECT ename
FROM Emp
WHERE ecity = 'Delhi';

b) UPDATE Emp
SET salary = salary + 5000;

c)SELECT cname
FROM Company
WHERE empnumber > 10000;

d)UPDATE Company
SET ccity = 'Gurgaon'
WHERE cname = 'TCS';

Practical – 4

Aim: To study the commands to modify the structure of table (alter, delete) and execute the following queries using these commands:

- Add an attribute named ‘ Designation’ to the table ‘Emp’.
- Modify the table ‘Emp’, Change the datatype of ‘salary’ attribute to float.
- Drop the attribute ‘depttname’ from the table ‘emp’.
- Delete the entries from the table ‘ Company’ where the number of employees are less than 500

Queries:

a) ALTER TABLE Emp
ADD Designation VARCHAR(20);

b) ALTER TABLE Emp
MODIFY COLUMN salary FLOAT;

c)ALTER TABLE Emp
DROP COLUMN depttname;

d)DELETE FROM Company
WHERE empnumber < 500;

Practical – 5

Aim: To study the commands that involve compound conditions (and, or, in , not in, between , not between , like , not like) and execute the following queries using these commands:

- Find the names of all employees who live in ‘ Gurgaon’ and whose salary is between Rs. 20,000 and Rs. 30,000.
- Find the names of all employees whose names begin with either letter ‘A’ or ‘B’.
- Find the company names where the company city is ‘Delhi’ and the number of employees is not between 5000 and 10,000.
- Find the names of all companies that do not end with letter ‘A’.

Queries:

a) SELECT ename
FROM Emp
WHERE ecity = 'Gurgaon' AND salary BETWEEN 20000 AND 30000;

b) SELECT ename
FROM Emp
WHERE ename LIKE 'A%' OR ename LIKE 'B%';

c) SELECT cname
FROM Company
WHERE ccity = 'Delhi' AND empnumber NOT BETWEEN 5000 AND 10000;

d) SELECT cname
FROM Company
WHERE cname NOT LIKE '%A';

Practical – 6

Aim: To study the aggregate functions (sum, count, max, min, average) and execute the following queries using these commands:

- Find the sum and average of salaries of all employees in computer science department.
- Find the number of all employees who live in Delhi.
- Find the maximum and the minimum salary in the HR department.

Queries:

a) SELECT SUM(salary) AS Total_Salary, AVG(salary) AS Average_Salary
FROM Emp
WHERE depttname = 'CS';

b) SELECT COUNT(*) AS Num_Employees_In_Delhi
FROM Emp
WHERE ecity = 'Delhi';

c)SELECT MAX(salary) AS Max_Salary, MIN(salary) AS Min_Salary
FROM Emp
WHERE depttname = 'HR';

Practical – 7

Aim: To study the grouping commands (group by, order by) and execute the following queries using these commands:

- List all employee names in descending order.
- Find number of employees in each department where number of employees is greater than 5.
- List all the department names where average salary of a department is Rs.10,000.

Queries:

a) SELECT ename
FROM Emp
ORDER BY ename DESC;

b) SELECT depttname, COUNT(*) AS Num_Employees
FROM Emp
GROUP BY depttname
HAVING COUNT(*) > 5;

c) SELECT depttname
FROM Emp
GROUP BY depttname
avg(salary)> 10000;

Practical – 8

Aim: To study the commands involving data constraints and execute the following queries using these commands:

- Alter table ‘Emp’ and make ‘enumber’ as the primary key.
- Alter table ‘Company’ and add the foreign key constraint.
- Add a check constraint in the table ‘Emp’ such that salary has the value between 0 and Rs.1,00,000.
- Alter table ‘Company’ and add unique constraint to column cname.
- Add a default constraint to column ccity of table company with the value ‘Delhi’.

Queries:

a) ALTER TABLE Emp
ADD CONSTRAINT PRIMARY KEY (enumber);

b) ALTER TABLE Company
ADD CONSTRAINT FOREIGN KEY (empnumber) REFERENCES Emp(enumber);

c) ALTER TABLE Emp
ADD CONSTRAINT CHECK (salary >= 0 AND salary <= 100000);

d) ALTER TABLE Company
ADD CONSTRAINT UNIQUE (cname);

e) ALTER TABLE Company
ALTER COLUMN ccity SET DEFAULT 'Delhi';

Practical – 9

Aim: To study the commands for aliasing and renaming and execute the following queries using these commands:

- Rename the name of database to ‘Employee1’.
- Rename the name of table ‘Emp’ to ‘Emp1’.
- Change the name of the attribute ‘ename’ to ‘empname’.

Queries:

b)ALTER TABLE Emp RENAME TO Emp1;

c)ALTER TABLE Emp1 CHANGE COLUMN ename empname VARCHAR(255);

Practical – 10

Aim : To study the commands for joins (cross join, inner join, outer join) and execute the following queries using these commands:

- Retrieve the complete record of an employee and its company from both the table using joins.
- List all the employees working in the company ‘TCS’.

Queries:

a) SELECT Emp.*, Company.*
FROM Emp
INNER JOIN Company ON Emp.ecity = Company.ccity;

b) SELECT *FROM Emp JOIN Company ON Emp.enumber = Company.empnumber WHERE Company.cname = 'TCS';

Practical – 11

Aim: To study the various set operations and execute the following queries using these commands:

- List the enumber of all employees who live in Delhi and whose company is in Gurgaon or if both conditions are true.
- List the enumber of all employees who live in Delhi but whose company is not in Gurgaon.

Queries:

a) `SELECT enumber`

```
FROM Emp
JOIN Company ON Emp.ecity = 'Delhi' AND Company.ccity = 'Gurgaon'
UNION
SELECT enumber
FROM Emp
JOIN Company ON Emp.ecity = 'Delhi'
WHERE Company.ccity != 'Gurgaon';
```

b) `SELECT enumber`

```
FROM Emp
INNER JOIN Company ON Emp.ecity = 'Delhi'
WHERE Company.ccity != 'Gurgaon';
```

Practical – 12

Aim: To study the various scalar functions and string functions (power, square, substring, reverse, upper, lower, concatenation) and execute the following queries using these commands:

- Reverse the names of all employees.
- Change the names of company cities to uppercase.
- Concatenate name and city of the employee

Queries:

a) SELECT REVERSE(ename)
FROM Emp;

b) UPDATE Company
SET ccity = UPPER(ccity);

c) SELECT CONCAT(ename, ', ', ecity)
FROM Emp;

Practical – 13

Aim: To study the commands for views and execute the following queries using these commands:

- Create a view having ename and ecity.
- In the above view change the ecity to ‘Delhi’ where ename is ‘John’.
- Create a view having attributes from both the tables.
- Update the above view and increase the salary of all employees of IT department by Rs.1000.

Queries:

a) CREATE VIEW EmpView AS
SELECT ename, ecity
FROM Emp;

b) UPDATE EmpView
SET ecity = 'Delhi'
WHERE ename = 'John';

c) CREATE VIEW EmpCompanyView AS
SELECT Emp.*, Company.cname, Company.ccity
FROM Emp
JOIN Company ON Emp.ecity = Company.ccity;

d) UPDATE EmpCompanyView
SET salary = salary + 1000
WHERE depttname = 'IT';

Practical – 14

Aim: To study the commands involving indexes and execute the following queries:

- Create an index with attribute ename on the table employee.
- Create a composite index with attributes cname and ccity on table company.
- Drop all indexes created on table company

Queries:

a) CREATE INDEX idx_ename ON emp (ename);

b) CREATE INDEX idx_company_cc ON company (cname, ccity);

c) DROP INDEX idx_company_cc ON company;

Practical – 15

AIM- Introduction to PL-SQL.

THEORY

PL/SQL is a block structured language that enables developers to combine the power of SQL with procedural statements. All the statements of a block are passed to oracle engine all at once which increases processing speed and decreases the traffic.

DISADVANTAGES OF SQL:

- SQL doesn't provide the programmers with a technique of condition checking, looping and branching.
- SQL statements are passed to Oracle engine one at a time which increases traffic and decreases speed.
- SQL has no facility of error checking during manipulation of data.

FEATURES OF PL/SQL:

- PL/SQL is basically a procedural language, which provides the functionality of decision making, iteration and many more features of procedural programming languages.
- PL/SQL can execute a number of queries in one block using single command.
- One can create a PL/SQL unit such as procedures, functions, packages, triggers, and types, which are stored in the database for reuse by applications.
- PL/SQL provides a feature to handle the exception which occurs in PL/SQL block known as exception handling block.
- Applications written in PL/SQL are portable to computer hardware or operating system where Oracle is operational.

  PL/SQL Offers extensive error checking.

DIFFERENCES BETWEEN SQL AND PL/SQL: SQL-

- SQL is a single query that is used to perform DML and DDL operations.
- It is declarative, that defines what needs to be done, rather than how things need to be done.
- Execute as a single statement.
- Mainly used to manipulate data.
- Cannot contain PL/SQL code in it.

PL/SQL-

- PL/SQL is a block of codes that used to write the entire program blocks/procedure/ function, etc.
- PL/SQL is procedural that defines how the things needs to be done.

- Execute as a whole block.
- Mainly used to create an application.
- It is an extension of SQL, so it can contain SQL inside it

Basic Commands

Here are some basic PL/SQL commands and constructs that you'll commonly use:

1. DECLARE: Used to declare variables, constants, and cursors.

2. BEGIN...END: Encloses the executable statements within a block.

3. IF...THEN...ELSE: Conditional statement for branching logic.

4. LOOP: Used for iterative processing.

5. FOR LOOP: Executes a set of statements a specified number of times.

6. WHILE LOOP: Executes a set of statements as long as a condition is true.

7. EXIT WHEN: Used to exit a loop based on a condition.

8. CASE: Evaluates a condition and executes statements based on different cases.

9. EXECUTE IMMEDIATE: Executes a dynamic SQL statement.

10. FUNCTION: Defines a function that returns a single value.

11. PROCEDURE: Defines a stored procedure that performs operations.

12. EXCEPTION: Handles errors and exceptions in PL/SQL code.

Practical – 16

Aim: To study the conditional controls and case statement in PL-SQL and execute the following queries:

- Calculate the average salary from table ‘Emp’ and print increase the salary if the average salary is less that 10,000.
- Print the deptno from the employee table using the case statement if the deptname is ‘Technical’ then deptno is 1, if the deptname is ‘HR’ then the deptno is 2 else deptno is 3.

Queries:

a)

```
DELIMITER $$ CREATE PROCEDURE update() BEGIN if ((select
avg(salary) from emp1)<10000) then select ‘increment salary’
update emp1 set salary = salary + 10000 else select ‘increment’end
if; end; // delimiter;

call update_sal();
```

b)

```
select designation,
case
when designation = ‘Technical’ then 1
when designation = ‘HR’ then 2
else 3
end as designation from emp1;
```

Practical – 17

Aim: To study procedures and triggers in PL-SQL and execute the following queries:

- Create a procedure on table employee to display the details of employee to display the details of employees by providing them value of salaries during execution.
- Create a trigger on table company for deletion where the whole table is displayed when delete operation is performed

Queries:

a)

delimiter // create procedure sol (num int) begin select* from emp1 where salary = num; end; // delimiter; call sol (21000);

B)
DELIMITER $$ CREATE TRIGGER displayCompany AFTER DELETE ON COMPANY FOR EACH ROW BEGIN SELECT * FROM COMPANY; END $$DELIMITER ;

Assignment PSLP

1. Select the detail of the employee whose name starts with P.

```
SELECT *
FROM Employee
WHERE EmpName LIKE 'P%';
```

2. How many permanent candidates take a salary more than 5000?

```
SELECT COUNT(*)
FROM EmpSalary
WHERE IsPermanent = 'Yes' AND Salary > 5000;
```

3. Select the detail of the employee whose emailId is in Gmail.

```
SELECT *
FROM Employee
WHERE EmailId LIKE '%@gmail.com';
```

4. Select the details of the employee who works either for department E-104 or E-102.

```
SELECT *
FROM Employee
WHERE Department IN ('E-104', 'E-102');
```

5. What is the department name for DeptID E-102?

```
SELECT DeptName
FROM EmpDept
WHERE DeptId = 'E-102';
```

6. What is the total salary that is paid to permanent employees?

```
SELECT SUM(Salary)
FROM EmpSalary
WHERE IsPermanent = 'Yes';
```

7. List the names of all employees whose name ends with 'a'.

```
SELECT EmpName
FROM Employee
WHERE EmpName LIKE '%a';
```

8. List the number of departments of employees in each project.

```
SELECT ProjectId, COUNT(DISTINCT Department) AS NumDepartments
FROM Employee  ,EmpProject
GROUP BY ProjectId;
```

9. How many projects started in the year 2010?

```
SELECT COUNT(*)
FROM Project
WHERE YEAR(StartYear) = 2010;
```

10. How many projects started and finished in the same year?

```
SELECT COUNT(*)
FROM Project
WHERE YEAR(StartYear) = YEAR(EndYear);
```

11. Select the name of the employee whose name's 3rd character is 'h'.

```
SELECT EmpName
FROM Employee
WHERE SUBSTRING(EmpName, 3, 1) = 'h';
```

12. Select the department name of the company assigned to the employee whose employee ID is greater than 103.

```
SELECT DeptName
FROM EmpDept
WHERE DeptHead > 103;
```

13. Select the name of the employee who is working under Abhishek.

```
SELECT EmpName
FROM Employee
```

WHERE EmpHeadId = (SELECT EmpId FROM Employee WHERE EmpName = 'Abhishek');

14. Select the name of the employee who is the department head of HR.

```
SELECT EmpName
FROM Employee
WHERE Empid = (SELECT DeptHead FROM EmpDept WHERE DeptName = 'HR');
```

15. Select the name of the employee head who is permanent.

```
SELECT e.EmpName
FROM Employee e
JOIN EmpSalary es ON e.EmpHeadId = es.EmpId
WHERE es.IsPermanent = 'Yes';
```

```
SELECT EmpName
FROM employee
WHERE EmpId = (SELECT EmpHeadId FROM Employee, empsalary WHERE IsPermanent =
'Yes');

SELECT EmpName
FROM Employee
WHERE EmpId = (
   SELECT EmpHeadId
   FROM Employee
   WHERE EmpId IN (
      SELECT EmpId
      FROM EmpSalary
      WHERE IsPermanent = 'Yes'
      LIMIT 1
   )
   LIMIT 1
);
```

16. Select the name and email of the department head who is not Permanent.

```
SELECT E.EmpName, E.EmailId
FROM Employee E
JOIN Employee EH ON E.EmpId = EH.EmpHeadId
JOIN EmpSalary ES ON EH.EmpId = ES.EmpId
WHERE ES.IsPermanent <> 'Yes';

select empname, emailid from employee where empid in(select
depthead from empdept ) and empid in(select empid from empsalary
where ispermanent='no')
```

17. Select the employee whose department off is Monday.

```
SELECT *
FROM Employee
WHERE Empid IN (SELECT Empid FROM EmpDept WHERE Dept_off = 'Monday');
```

18. Select the Indian clients' details.

```
SELECT *
FROM ClientTable
WHERE cid IN (SELECT cid FROM Country WHERE cname = 'India');
```

19. Show the details of the second-highest salaried person.

```
select *from empsalary
group by salary
order by  salary desc limit 1,1;
```

Set B:
Create execute the next set of questions
Supplier(Sid , Sname, Address)
Parts(Pid, Pname, Color)
Catalog(Pid,Sid,Cost)
1. Find the name of the supplier who supplies some green part.

```
SELECT DISTINCT S.Sname
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON C.Pid = P.Pid
WHERE P.Color = 'green';
```

2. Find the name of the supplier who supplies some red or green part.

```
SELECT DISTINCT S.Sname
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON C.Pid = P.Pid
WHERE P.Color IN ('red', 'green');
```

3. Find the sid of the supplier who supplies some red or green part or is at '21 Rohini Delhi'.

```
SELECT DISTINCT S.Sid
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON C.Pid = P.Pid
WHERE P.Color IN ('red', 'green')
OR S.Address = '21 Rohini Delhi';
```

4. Find the name of the supplier who supplies some red or some green parts.

```
SELECT DISTINCT S.Sname
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON C.Pid = P.Pid
WHERE P.Color = 'red'
OR P.Color = 'green';
```

5. Find sid's of suppliers who supply every green part.

```
SELECT DISTINCT S.Sid
FROM Supplier S
JOIN Catalog C ON S.Sid = C.Sid
JOIN Parts P ON C.Pid = P.Pid
WHERE P.Color = 'green'
GROUP BY S.Sid
HAVING COUNT(DISTINCT P.Pid) = (SELECT COUNT(*) FROM Parts WHERE Color =
'green');
```

6. Find pid's of parts supplied by at least two different suppliers.

```
SELECT P.Pid
FROM Parts P
JOIN Catalog C ON P.Pid = C.Pid
GROUP BY P.Pid
HAVING COUNT(DISTINCT C.Sid) >= 2;
```

TABLES CREATED

```
create database assignment_file;

use assignment_file

CREATE TABLE Employee (
    EmpId INT PRIMARY KEY,
    EmpName VARCHAR(255),
    Department VARCHAR(255),
    ContactNo VARCHAR(20),
    EmailId VARCHAR(255),
    EmpHeadId INT
);

INSERT INTO Employee (EmpId, EmpName, Department, ContactNo, EmailId, EmpHeadId)
VALUES
(101, 'Isha', 'E-101', '1234567890', 'isha@gmail.com', 105),
(102, 'Priya', 'E-104', '1234567890', 'priya@yahoo.com', 103),
(103, 'Neha', 'E-101', '1234567890', 'neha@gmail.com', 101),
(104, 'Rahul', 'E-102', '1234567890', 'rahul@yahoo.com', 105),
(105, 'Abhishek', 'E-101', '1234567890', 'abhishek@gmail.com', 102);

CREATE TABLE EmpDept (
    DeptID INT PRIMARY KEY,
    DeptName VARCHAR(255),
    Dept_off VARCHAR(255),
    DeptHead INT,
    FOREIGN KEY (DeptHead) REFERENCES Employee(EmpID)
);


INSERT INTO EmpDept (DeptID, DeptName, Dept_off, DeptHead)
VALUES
('E-101', 'HR', 'Monday', 105),
('E-102', 'Development', 'Tuesday', 101),
('E-103', 'House Keeping', 'Saturday', 103),
('E-104', 'Sales', 'Sunday', 104),
('E-105', 'Purchase', 'Tuesday', 104);


CREATE TABLE EmpSalary (
    EmpId INT PRIMARY KEY,
    Salary INT,
    IsPermanent VARCHAR(3)
);
```

```
INSERT INTO EmpSalary (EmpId, Salary, IsPermanent)
VALUES
(101, 2000, 'Yes'),
(102, 10000, 'Yes'),
(103, 5000, 'No'),
(104, 1900, 'Yes'),
(105, 2300, 'No');

CREATE TABLE Project (
    ProjectId VARCHAR(255) PRIMARY KEY,
    Duration INT
);

INSERT INTO Project (ProjectId, Duration)
VALUES
('p-1', 23),
('p-2', 15),
('p-3', 45),
('p-4', 2),
('p-5', 30);

CREATE TABLE Country (
    cid VARCHAR(255) PRIMARY KEY,
    cname VARCHAR(255)
);

INSERT INTO Country (cid, cname)
VALUES
('c-1', 'India'),
('c-2', 'USA'),
('c-3', 'China'),
('c-4', 'Pakistan'),
('c-5', 'Russia');

CREATE TABLE ClientTable (
    ClientId VARCHAR(255) PRIMARY KEY,
    ClientName VARCHAR(255),
    cid VARCHAR(255),
    FOREIGN KEY (cid) REFERENCES Country(cid)
);

INSERT INTO ClientTable (ClientId, ClientName, cid)
VALUES
('cl-1', 'ABC Group', 'c-1'),
('cl-2', 'PQR', 'c-1'),
('cl-3', 'XYZ', 'c-2'),
('cl-4', 'tech altum', 'c-3'),
('cl-5', 'mnp', 'c-5');
```

```
CREATE TABLE EmpProject (
    EmpId INT,
    ProjectId VARCHAR(255),
    ClientID VARCHAR(255),
    StartYear INT,
    EndYear INT,
    PRIMARY KEY (EmpId, ProjectId),
    FOREIGN KEY (EmpId) REFERENCES Employee(Empid),
    FOREIGN KEY (ProjectId) REFERENCES Project(ProjectId),
    FOREIGN KEY (ClientID) REFERENCES ClientTable(ClientId)
);

INSERT INTO EmpProject (EmpId, ProjectId, ClientID, StartYear, EndYear)
VALUES
(101, 'p-1', 'Cl-1', 2010, 2010),
(102, 'p-2', 'Cl-2', 2010, 2012),
(103, 'p-1', 'Cl-3', 2013, NULL),
(104, 'p-4', 'Cl-1', 2014, 2015),
(105, 'p-4', 'Cl-5', 2015, NULL);


CREATE TABLE Supplier (
    Sid INT PRIMARY KEY,
    Sname VARCHAR(255),
    Address VARCHAR(255)
);

CREATE TABLE Parts (
    Pid INT PRIMARY KEY,
    Pname VARCHAR(255),
    Color VARCHAR(50)
);

CREATE TABLE Catalog (
    Pid INT,
    Sid INT,
    Cost DECIMAL(10, 2),
    FOREIGN KEY (Pid) REFERENCES Parts(Pid),
    FOREIGN KEY (Sid) REFERENCES Supplier(Sid)
);

INSERT INTO Supplier (Sid, Sname, Address)
VALUES
(101, 'Supplier A', '123 Main St'),
(102, 'Supplier B', '456 Elm St'),
(103, 'Supplier C', '789 Oak St'),
(104, 'Supplier D', '101 Pine St'),
(105, 'Supplier E', '202 Maple St');
```

```
INSERT INTO Parts (Pid, Pname, Color)
VALUES
(201, 'Part X', 'Red'),
(202, 'Part Y', 'Blue'),
(203, 'Part Z', 'Green'),
(204, 'Part W', 'Yellow'),
(205, 'Part V', 'Purple');

INSERT INTO Catalog (Pid, Sid, Cost)
VALUES
(201, 101, 10.50),
(202, 101, 12.75),
(203, 102, 8.99),
(204, 103, 11.20),
(205, 104, 14.50);
```