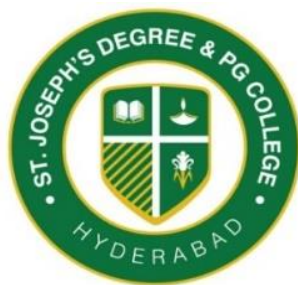


Department of Computer Science



St. Joseph's Degree & P.G College

(Autonomous), Affiliated to Osmania University
Re-accredited by NAAC with A Grade with CGPA 3.49
A Catholic Christian Minority Institution
King Koti Road, Hyderabad.

Lab Manual

Program : B.Sc [MPCs/MSCs/MECs] II Year Semester III
Course : Relational Database Management System Lab
Course code : BS.06.201.21.P

B.Sc. (Computer Science)
II Year/ III Semester
Theory Paper - III
Relational Database Management System

Scheme of Instruction	Scheme of Examination
Total durations Hrs : 60	Max. Marks : 100
Hours/Week : 06(4T+2P)	Internal Examination :30
Credits : 5	SBT : 10
Instruction Mode: Lecture +Practical	External Examination :60
Course Code : BS.06.201.13.T	Exam Duration : 3 Hrs
Course Objectives:	
To impart the students with the knowledge on the database management systems, design models, Normalization, Transaction management and Oracle in Creation and maintenance of databases.	
Course Outcomes:	
At the end of the course the student will be able to	
CO1: Understand and evaluate the database environment in an organization.	
CO 2: Design and Develop database using SQL & PL/SQL.	
CO 3: Understand and Design the ER Model utilized for developing a database.	
CO 4: Applythe Normalization techniques to evaluate and correct table structures.	
CO 5: Design real time databases using the concepts of Transaction Management, Concurrency Control& Distributed Databases.	

UNIT-I: Introduction to Database concepts & Relational Model

Basic Concepts and Definitions: Data, Information, Metadata, Database, DBMS.

The Database Environment: Traditional File Processing Systems, The Database approach, Components of Database Environment. The Three-Level Architecture, Advantages of Database Management System, Types of databases, Database Languages, Risks and costs of Database, Data Models.

Relational Model: Introduction, Keys, Relational Algebra: Unary Operations, Set Operations, Join Operations, Aggregation and Grouping Operations.

UNIT-II: SQL

Introduction to various Databases: Oracle-SQL. MySQL, SQL Server, DB2, MS Access.

SQL: Introduction, SQL Environment, The ISO SQL Data Types, Integrity Constraints, Data Definition–Creating a Database, Creating a Table, Changing a Table Definition, Removing a Table, Creating an Index, Removing an Index, Views, Granting and Revoking Privileges to Users.

Data Manipulation:Inserting, Updating & Deleting Data from database, Simple Queries, Aggregate Functions, Order by Clause, Group by Clause, Having Clause.Joins, Sub Queries, Correlated Sub Queries.

PL/SQL: Introduction to PL/SQL, Advantages of PL/SQL, The Generic PL/SQL Block, The PL/SQL Execution Environment, Declarations, Assignments, Control Statements, Exceptions, Cursors, Subprograms, Triggers , Stored Procedures, Functions, and Packages.

UNIT-III: ER Model & Normalization

Entity-Relationship Model: Entities, Attributes & Relationships. Types of Entities, Types of Attributes, Types of Relationships, Degree of a relationship, Cardinality Constraints, Structural Constraints, Problems with ER Models–Fan Traps, Chasm Traps. Specialization/Generalization.

Normalization: Definition of Normalization, Need for Normalization, Basic normal Forms: First Normal Form (1NF), Second Normal Form (2NF), Third Normal Form (3NF). Advanced Normal Forms: Boyce Codd Normal Form (BCNF), De-normalization. The Database Design Methodology for Relational Databases

UNIT-IV: Transaction Management and Concurrency Control, Distributed Database Management System

Transaction Management and Concurrency Control: What is transaction, Properties of Transactions, Database, Concurrency control, Serializability and Recoverability, Concurrency control with locking Methods, Concurrency control with time stamping methods, concurrency control with optimistic methods, Database Recovery, Database Security.

Distributed Database Management System: Evolution of Distributed Database Management System (DDBMS), DDBMS Advantages and Disadvantages, Characteristics of DDBMS, DDBMS Components, And Distributed Concurrency Control.

TextBook:

Thomas M. Connolly, Carolyn E. Begg, Database Systems–A Practical Approach to Design, Implementation, and Management.

References:

1. Modern Database Management: Fred R. McFadden
2. Database Systems: Design, Implementation, and Management: C Coronel, S Morris, Peter Rob
3. SQL, PL/SQL: The Programming Language of Oracle: Ivan Bayross

B.Sc. (Computer Science)

II Year / III Semester

PRACTICAL PAPER - III

SQL Lab (Oracle 10g XE)

Course Objectives:
To impart the students with the knowledge about the process of creation and maintenance of databases.
Course Outcomes:
On successful completion of this course, the students should have understood <ul style="list-style-type: none">➤ Students can implement Orderby and Group by clauses.➤ Students can gain knowledge on How to use SQL for Creating, Modifying and Accessing tables in Database.

1. (Exercise on retrieving records from the table)

EMPLOYEES (Employee_Id, First_Name, Last_Name, Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, Manager_Id, Department_Id)

- (a) Find out the employee id, names, salaries of all the employees
- (b) List out the employees who works under manager 100
- (c) Find the names of the employees who have a salary greater than or equal to 4800
- (d) List out the employees whose last name is 'AUSTIN'
- (e) Find the names of the employees who works in departments 60,70 and 80
- (f) Display the unique Manager_Id.

2. (Exercise on updating records in table)

Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due)

- (a) Insert five records
- (b) Find the names of clients whose bal_due > 5000 .
- (c) Change the bal_due of ClientNO " C123" to Rs. 5100
- (d) Change the name of Client_master to Client12 .
- (e) Display the bal_due heading as "BALANCE"

3. Rollback and Commit commands

Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary)

- (a) Insert five records
- (b) Give Increment of 25% salary for Mathematics Department .
- (c) Perform Rollback command
- (d) Give Increment of 15% salary for Commerce Department
- (e) Perform commit command

4 . (Exercise on order by and group by clauses)

Create Sales table with the following fields(Sales No, Salesname, Branch, Salesamount, DOB)

- (a) Insert five records
- (b) Calculate total salesamount in each branch
- (c) Calculate average salesamount in each branch .
- (d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09
- (e) Display the name and DOB of salesman in alphabetical order of the month.

5. Create an Emp table with the following fields:

(EmpNo, EmpName, Job,Basic, DA, HRA,PF, GrossPay, NetPay)

(Calculate DA as 30% of Basic and HRA as 40% of Basic)

- (a) Insert Five Records and calculate GrossPay and NetPay.
- (b) Display the employees whose Basic is lowest in each department .
- (c) If NetPay is less than <Rs. 10,000 add Rs. 1200 as special allowances .

- (d) Display the employees whose GrossPay lies between 10,000 & 20,000
- (e) Display all the employees who earn maximum salary .

6. Employee Database

An Enterprise wishes to maintain a database to automate its operations. Enterprise is divided into certain departments and each department consists of employees. The following two tables describes the automation schemas

Dept (deptno, dname, loc)

Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

- a) Update the employee salary by 15%, whose experience is greater than 10 years.
- b) Delete the employees, who completed 30 years of service.
- c) Display the manager who is having maximum number of employees working under him?
- d) Create a view, which contain employee names and their manager

7. Using Employee Database perform the following queries

- a) Determine the names of employee, who earn more than their managers.
- b) Determine the names of employees, who take highest salary in their departments.
- c) Determine the employees, who are located at the same place.
- d) Determine the employees, whose total salary is like the minimum Salary of any department.
- e) Determine the department which does not contain any employees.

8. Consider the following tables namely “DEPARTMENTS” and “EMPLOYEES” Their schemas are as follows,

Departments (dept_no , dept_name , dept_location);

Employees (emp_id , emp_name , emp_salary,dept_no);

- a) Develop a query to grant all privileges of employees table into departments table
- b) Develop a query to grant some privileges of employees table into departments table
- c) Develop a query to revoke all privileges of employees table from departments table
- d) Develop a query to revoke some privileges of employees table from departments table
- e) Write a query to implement the save point.

9. Using the tables “DEPARTMENTS” and “EMPLOYEES” perform the following queries

- a) Display the employee details, departments that the departments are same in both the emp and dept.
- b) Display the employee name and Department name by implementing a left outer join.
- c) Display the employee name and Department name by implementing a right outer join.
- d) Display the details of those who draw the salary greater than the average salary.

10. Write a PL/SQL program to demonstrate Exceptions.

11. Write a PL/SQL program to demonstrate Cursors.

- 12.** Write a PL/SQL program to demonstrate Functions.
- 13.** Write a PL/SQL program to demonstrate Packages.
- 14.** Write PL/SQL queries to create Procedures.
- 15.** Write PL/SQL queries to create Triggers.

B.Sc. (Computer Science)
II Year / III Semester
PRACTICAL PAPER - III
Course Name: SQL Lab (Oracle 10g XE)
Course Code : BS.06.201.13.T

Lab Practical No.1

Question:

1. (Exercise on retrieving records from the table) EMPLOYEES (Employee_Id, First_Name, Last_Name, Email, Phone_Number, Hire_Date, Job_Id, Salary, Commission_Pct, Manager_Id, Department_Id)
 - (a) Find out the employee id, names, salaries of all the employees
 - (b) List out the employees who works under manager 100
 - (c) Find the names of the employees who have a salary greater than or equal to 4800
 - (d) List out the employees whose last name is 'AUSTIN'
 - (e) Find the names of the employees who works in departments 60,70 and 80
 - (f) Display the unique Manager_Id

Employees table:

1. create an employee's table with the following fields:
(Emp_id,First_name,Last_name,Phone_No,Hire_date,Job_id,Emp_Salary,Comission_Pct,manager_id,Department_id)

Query:

create table Employees (Emp_id NUMBER(6),First_name CHAR(25),Last_name CHAR(20),Phone_No NUMBER(12),Hire_date DATE,Job_Id NUMBER(5),Emp_Salary NUMBER(7),Comission_Pct NUMBER(5),manager_id NUMBER(5),Department_id NUMBER(5));

2. Insert five records into the table employees:

Query:

SQL> insert into employees values(47401,'Rama','Rao',8965324170,'28-Jan-2003',301,60000,601,100,60);

1 row created.

SQL> insert into employees values(47402,'Ranga','Reddy',7020321450,'23-Jun-2004',302,56464,602,101,70);

1 row created.

```
SQL> insert into employees values(47403,'Raja','Shekhar',9848002255,'12-aug-2004',303,58451,603,103,80);
```

1 row created.

```
SQL> insert into employees values(47404,'Ravi','AUSTIN ',9701811356,'30-sep-2006',304,36520,604,100,90);
```

1 row created.

```
SQL> insert into employees values(47405,'Ranga','Raju',9032553262,'17-May-2014',305,2568,605,105,60);
```

1 row created.

```
SQL> Select * from Employees;
```

3. Display the table Employees :

Query:

```
sql>select_* from employees;
```

a) Find out the employee id, names, salaries of all the employees

Query:

```
sql>select_Emp_id,First_Name,Last_Name,Emp_Salary from employees;
```

b) List out the employees who works under manager 100

Query:

```
sql>select_* from employees where manager_id=100;
```

c) Find the names of the employees who have a salary greater than or equal to 4800

Query:

```
sql>select_* from employees where EMP_SALARY>=4800;
```

d) List out the employees whose last name is 'AUSTIN'

Query:

```
sql>select_* from employees where Last_Name='AUSTIN ';
```

e) Find the names of the employees who works in departments 60,70 and 80

Query: sql>select_* from employees where DEPARTMENT_ID IN(60,70,80);

f) Display the unique Manager_Id from employees table

Query:

```
sql>select_DISTINCT(MANAGER_ID) from employees;
```


Lab Practical No: 2

Question:

(Exercise on updating records in table) Create Client_master with the following fields(ClientNO, Name, Address, City, State, bal_due)

(a) Insert five records

(b) Find the names of clients whose bal_due > 5000 .

(c) Change the bal_due of ClientNO " C123" to Rs. 5100

(d) Change the name of Client_master to Client12 .

(e) Display the bal_due heading as "BALANCE" Client master table:

1. create a client master table with attributes

(Client_no, Client_Name, Client_Address, Client_City, Client_State, Balance_Due)

Query:

```
create table Client_Master(Client_no varchar(6), Client_Name char(25), Client_Address  
varchar(25), Client_City varchar(20), Client_State varchar(20), Balance_Due number(20));
```

a) insert five records into the Client_Master

Query:

```
sql>insert INTO CLIENT_MASTER Values('C123','Ramesh','L B Nagar', 'Hyderabad',  
'Telangana', 7000);  
sql>insert INTO CLIENT_MASTER Values('C124', 'Suresh', 'Dilsuknagar', 'Hyderabad',  
'Telangana', 6000);  
sql>insert INTO CLIENT_MASTER Values('C125', 'Vignesh', 'Saroor nagar', 'Hyderabad',  
'Telangana', 3500);  
sql>insert INTO CLIENT_MASTER Values('C126', 'Rajiv', 'A S Rao Nagar', 'Hyderabad',  
'Telangana', 4500);  
sql>insert INTO CLIENT_MASTER Values('C127', 'Ranga', 'Vanasthalipuram', 'Hyderabad',  
'Telangana', 5478);
```

1. Display Client Master Table

Query: sql>select * from Client_Master;

b) Find the name of Clients whose balance_due > 5000

Query: sql>select Client_Name from Client_Master where Balance_Due > 5000;

c) Change the bal_due of ClientNO " C123" to Rs. 5100

Query: sql>update Client_Master set Balance_Due=5100 where Client_No='C123';

d) Change the name of Client_master to Client12 .

Query: sql>rename Client_Master to Client12;

e) Display the bal_due heading as “BALANCE”_Client master table:

Query: sql> select Client_No, Balance_Due Balance from Client_Master

Lab Practical No: 3

Question:

Rollback and Commit commands

Create Teacher table with the following fields(Name, DeptNo, Date of joining, DeptName, Location, Salary)

(a) Insert five records

(b) Give Increment of 25% salary for Mathematics Department .

(c) Perform Rollback command

(d) Give Increment of 15% salary for Commerce Department

(e) Perform commit command

Create Teacher table with the following fields(Id,Name, DeptNo, Date of joining, DeptName, Location, Salary)

Query :

SQL> create table teacher(Id number(2) primary key, name varchar2(20) not null, Deptno number(2) not null, Deptname varchar2(20) not null, joinDate date not null, location varchar2(20) not null, salary number(10,2) not null);

(a) Insert five records

SQL> insert into teacher values(10,'kiran',4,'computer science', '03-Jun-2003', 'hyderabad', 50000);

SQL> insert into teacher values(11,'ramu',5,'mathematics', '13-Jul-2008', 'hyderabad', 40000);

SQL> insert into teacher values(12,'raju',6,'commerce', '23-Dec-2005', 'abids', 30000);

SQL> insert into teacher values(13,'manirathnam',6,'commerce', '18-Dec-2006', 'abids', 30000);

SQL> insert into teacher values(14,'sita',6,'commerce', '28-Aug-2016', 'kingkoti', 23000);

(b) Give Increment of 25% salary for Mathematics Department .

Sql> update teacher set salary= salary+(salary * 0.25) where Deptname= 'mathematics';

(c) Perform Rollback command

Sql>rollback;

(d) Give Increment of 15% salary for Commerce Department

Sql> update teacher set salary= salary+(salary * 0.15) where Deptname= 'commerce';

(e) Perform commit command

Sql>commit;

Lab Practical No: 4

Question:

4 . (Exercise on order by and group by clauses) Create Sales table with the following fields(Sales No, Salesname, Branch, Salesamount, DOB)

(a) Insert five records

(b) Calculate total salesamount in each branch

(c) Calculate average salesamount in each branch .

(d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09

(e) Display the name and DOB of salesman in alphabetical order of the month.

Sales Table:

Create a Sales Table with the following fields

(Sales_No,Sales_Name,Branch,Sales_Amount,DOB)

Query:

Sql> Create Table Sales(Sales_No number(5),Sales_Name char(25),Branch char(25), Sales_Amount number(10), DOB Date);

(a) Insert five records

(b) Calculate total salesamount in each branch

(c) Calculate average salesamount in each branch .

(d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09

(e) Display the name and DOB of salesman in alphabetical order of the month.

Query:

Sql> insert into Sales VALUES(1020,'AutoMobiles','Hyderabad',68452,'28-JUL-1985');

Sql> insert into Sales VALUES(1021,'Electronics','Secunderabad',47850,'22-DEC-1995');

Sql> insert into Sales VALUES(1022,'Electronics','Secunderabad',44500,'03-JUN-1986');

Sql> insert into Sales VALUES(1023,'AutoMobiles','Hyderabad',74200,'28-SEP-1996');

Sql> insert into Sales VALUES(1024,'AutoMobiles','Hyderabad',54500,'28-OCT-1984');

Display the Sales table:

Query: sql>select * from Sales;

(b) Calculate total salesamount in each branch

Query: sql>select branch, sum(Sales_Amount) from sales group by Branch;

(c) Calculate average salesamount in each branch .

Query:

selectbranch,avg(Sales_Amount) from sales group by Branch;

(d) Display all the salesmen, DOB who are born in the month of December as day in character format i.e. 21-Dec-09

Query:

SQL> select Sales_Name, DOB from sales where SUBSTR(DOB,4,3)='DEC';

(e) Display the name and DOB of salesman in alphabetical order of the month.

Query:

sql>select Sales_Name,to_char(DOB,'MONTH') from sales Order by to_Char(DOB,'Day');

Lab Practical No: 5

Question:

5. Create an Emp table with the following fields:(EmpNo, EmpName, Job,Basic, DA, HRA,PF, GrossPay, NetPay)

(Calculate DA as 30% of Basic and HRA as 40% of Basic)

(a) Insert Five Records and calculate GrossPay and NetPay.

(b) Display the employees whose Basic is lowest in each department .

(c) If NetPay is less than <Rs. 10,000 add Rs. 1200 as special allowances .

(d) Display the employees whose GrossPay lies between 10,000 & 20,000

(e) Display all the employees who earn maximum salary .

Employee Table

create an employee table with the following fields:

(Emp_No,Emp_ Name, Designation, basic, DA, HRA, PF, Gross pay, Net pay);

Query:

Sql> create table Employee (Emp_No number(6) primary key, Emp_Name char(25) not null , Designation varchar(25),Emp_Basic number(10,2));

(a) Insert Five Records and calculate GrossPay and NetPay.

Query:

```
insert into Employee values(4019,'Talatam Venkatesh','Director',10000);
insert into Employee values(4039,'Gumaa','Technical Engineer',15000);
insert into Employee values(4015,'Anudeep Varma','Technical Engineer',12000);
insert into Employee values(4016,'I Vasudeva Varma','Technical Engineer',13000);
insert into Employee values(4027,'Uday Reddy','Director',16000);
insert into Employee values(4006,'Sai Mani','Director',10000);
```

#Updating Attributes DA, HRA, PF, Gross pay, Net Pay ?

Adding coloumn to table and Updating Attributes DA

```
Sql>alter table employeeadd(Emp_DA number(6));
```

```
Sql>update Employeeset Emp_DA=(30/100)*Emp_Basic;
```

Adding coloumn to table and Updating Attributes HRA

```
Sql>alter table employee add(Emp_HRA number(6));
```

```
update Employee set Emp_HRA=(40/100)*Emp_Basic;
```

Adding coloumn to table and Updating Attributes PF

```
Sql>alter table employee add (Emp_PF number(6));
```

```
Sql> Employee set Emp_pf= Emp_basic*(12/100);
```

Adding coloumn to table and Updating Attributes Gross Pay

```
Sql>alter table employee add (Emp_Grosspay number(6));
```

```
Sql>update Employee set Emp_grosspay= EMp_hra+emp_da+emp_basic;
```

Adding coloumn to table and Updating Attributes Net Pay

```
alter table employee add (Emp_netpay number(6));
```

```
update Employee set Emp_netpay=emp_grosspay-emp_pf;
```

Display the employeetable:

Query: sql>select_* from employee;

(b) Display the employees whose Basic is lowest in each department .

Query: sql>select_min(emp_basic) from employee group by designation;

(c) If NetPay is less than <Rs. 10,000 add Rs. 1200 as special allowances .

Query:

Sql> update employee set emp_netpay=emp_netpay+1200 where emp_netpay<30000;

(d) Display the employees whose GrossPay lies between 10,000 & 20,000

Query: sql>select_* from employee where emp_grosspay between 10000 and 20000;

(e) Display all the employees who earn maximum salary .

Query:

sql>select_* from employee where emp_grosspay = (select max(emp_grosspay) from employee);

Lab Practical No:6

Question:

6. Employee Database An Enterprise wishes to maintain a database to automate its operations. Enterprise is divided into certain departments and each department consists of employees. The following two tables describes the automation schemas

Dept (deptno, dname, loc)

Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

a)Update the employee salary by 15%, whose experience is greater than 10 years.

b)Delete the employees, who completed 30 years of service.

c)Display the manager who is having maximum number of employees working under him?

d)Create a view, which contain employee names and their manager

Creating Dept and Emp table

Create Dept table : Dept (deptno, dname, loc)

Sql> create table dept(deptno number(3) primary key, dname varchar2(30) not null, loc varchar2(30) not null);

Create Dept table : Emp (empno, ename, job, mgr, hiredate, sal, comm, deptno)

Sql>create table emp(empno number(3) primary key, ename varchar2(20) not null, job varchar2(20) not null, mgr number(3) references emp(empno), hiredate date not null, sal number(10,2) not null, comm Number(10,2), deptno number(3));

Inserting data int Dept and Emp tables

Inserting data into Dept table

Sql>insert into dept values(101, 'FINANCE',' SYDNEY');

Sql>insert into dept values(102, 'AUDIT',' MELBOURNE');

```
Sql>insert into dept values(103, 'MARKETING',' PERTH');
Sql>insert into dept values(104, 'PRODUCTION',' BRISBANE');
Sql>insert into dept values(105, 'Humanresource',' hyderabad');
```

Inserting data into Emp table

```
Sql> insert into emp (empno, ename ,job, hiredate, sal, deptno) values
(68319,'KAYLING','PRESIDENT','18-Nov-1991',6000.00,101);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(66928,' BLAZE',' MANAGER',68319,'09 -Jun-1991', 2750.00,103);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(67832,' CLARE',' MANAGER',68319,'18-Nov-1991', 2550.00,101);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(65646,' JONAS',' MANAGER',68319,'02-Apr-1991', 2957.00,102);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(67858,' SCARLET',' ANALYST', 65646,'19-Apr-1997', 3100.00,102);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(69062,' FRANK',' ANALYST', 65646,'03-Dec-1991', 3100.00,102);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(63679,' SANDRINE',' CLERK', 69062,'18-Dec-1990', 900.00,102);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal,comm, deptno) values
(64989,' ADELYN',' SALESMAN', 66928,'20-Feb-1991', 1700.00,400, 103);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal,comm, deptno) values
(65271,' WADE',' SALESMAN', 66928,'22-Feb-1991', 1350.00,600, 103);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal,comm, deptno) values
(66564,' MADDEN',' SALESMAN', 66928,'28-Sep-1991', 1350.00,1500, 103);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal,comm, deptno) values
(68454,' TUCKER',' SALESMAN', 66928,'08-Sep-1991', 1600.00, 0, 103);
```

```
Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values
(68736,' ADNRES',' CLERK', 67858,'23-May-1997', 1200.00,102);
```

Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values (69000,' JULIUS', ' CLERK', 66928,'03-Dec-1991', 1050.00,103);

Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values (69324,' MARKER', ' CLERK', 67832,'23-Jan-1992', 1400.00,101);

Sql> insert into emp (empno, ename ,job,mgr, hiredate, sal, deptno) values (69924,' MARKER', ' CLERK', 67832,'23-Jan-1992', 1400.00,104);

a)Update the employee salary by 15%, whose experience is greater than 30 years.

Query

Sql> update emp set sal=sal+(sal*0.15) where (sysdate-hiredate)/365>30;

b>Delete the employees, who completed 30 years of service.

Sql> delete from emp where (sysdate-hiredate)/365>30;

c)Display the manager who is having maximum number of employees working under him?

Create view

Sql> create view mgrcount as select mgr, count(empno) total from emp group by mgr;

Sql> select mgr from mgrcount where total in (select max(total) from mgrcount);

d)Create a view, which contain employee names and their manager

Sql> create view employee_manager as select e1.ename ,e2.ename from emp e1, emp e2 where e1.mgr=e2.empno;

Lab Practical No:7

Question:

7. Using Employee Database perform the following queries

- a)Determine the names of employee, who earn more than their managers.
- b)Determine the names of employees, who take highest salary in their departments.
- c)Determine the employees, who are located at the same place.
- d)Determine the employees, whose total salary is like the minimum Salary of any department.
- e)Determine the department which does not contain any employees.

a)Determine the names of employee, who earn more than their managers.

sql>select e1.ename from emp e1, emp e2 where e1.mgr=e2.empno and e1.sal>e2.sal;

b)Determine the names of employees, who take highest salary in their departments.

Create a view to store maximum salaries of each department

SQL> create view maxsalaries as select max(sal) maxsalary, deptno from emp group by deptno;

Displaythe names of employees, who take highest salary in their departments.

Sql>select ename, sal from emp, maxsalaries where emp.deptno= maxsalaries.deptno and sal=maxsalary;

c)Determine the employees, who are located at the same place.

SQL> select ename,dname from emp , dept where emp.deptno=dept.deptno order by dname;

d)Determine the employees, whose total salary is like the minimum Salary of any department.

SQL> select empno, ename, sal from emp where sal in(select max(sal) from emp group by deptno);

e)Determine the department which does not contain any employees.

SQL> select dname from dept where deptno not in(select deptno from emp);

Lab Practical No:8

Question:

9. Usingthe tables “DEPARTMENTS” and “EMPLOYEEES” perform the following queries

a)Display the employee details, departments that the departments are same in both the emp and dept.

b)Display the employee name and Department name by implementing a left outer join.

c)Display the employee name and Department name by implementing a right outer join.

d)Display the details of those who draw the salary greater than the average salary.

a)Display the employee details, departments that the departments are same in both the emp and dept.

sql > select ename, dname from dept,emp where emp.deptno=dept.deptno;

b)Display the employee name and Department name by implementing a left outer join.

sql > select ename, dname from dept,emp where emp.deptno(+)=dept.deptno;

c)Display the employee name and Department name by implementing a right outer join.

sql > select ename, dname from dept,emp where emp.deptno=+dept.deptno;

d)Display the details of those who draw the salary greater than the average salary.

SQL> Select empno,sal from emp where sal> (select avg(sal) from emp);

Lab Practical No:9

Question:

9.Write a PL/SQL program to demonstrate Exceptions.

```
DECLARE
  Eid emp.empno%type:= 69000;
  name emp.ename%type;
BEGIN
  SELECT empno,ename INTO eid,name
  FROM emp
  WHERE empno = eid;
  DBMS_OUTPUT.PUT_LINE ('empno: ' || eid);
  DBMS_OUTPUT.PUT_LINE ('Name: '|| name);
```

```
EXCEPTION
  WHEN no_data_found THEN
    dbms_output.put_line('No such employee!');
  WHEN others THEN
    dbms_output.put_line('Error!');
END;
/
```

Output

```
empno: 69000
Name: JULIUS
```

PL/SQL procedure successfully completed.

Lab Practical No:10

Question:

10.Write a PL/SQL program to demonstrate Cursors.

```
SQL>declare
  cursor c11 is
  select * from emp order by sal,job;
  cval emp%rowtype;
begin
  dbms_output.put_line('empno  ' || 'ename');
  open c11;
  loop
  fetch c11 into cval;
  exit when c11%rowcount=10;
```

```
dbms_output.put_line(cval.empno||' '||cval.ename);  
end loop;  
close c11;  
end;  
/
```

Output

empno	ename
69000	JULIUS
63679	SANDRINE
68736	ADNRES
69324	MARKER
69924	MARKER
67832	CLARE
66928	BLAZE
65646	JONAS
67858	SCARLET

PL/SQL procedure successfully completed.

Lab Practical No:11

Question:

11. Write a PL/SQL program to demonstrate Functions.

SQL>

```
create or replace function fname(a in number,b in out number)  
return number is  
begin  
b:=a;  
return b;  
end;
```

SQL> /

Function created.

Program to invoke a function

SQL>

```
declare  
x number;  
begin  
x:=fname(23,x);  
dbms_output.put_line(x);  
end;
```

Output

```
SQL> /  
23
```

PL/SQL procedure successfully completed.

Lab Practical No:12

Question:

12. Write PL/SQL queries to create Procedures.

```
SQL>  
create or replace procedure emp_proc1  
IS  
BEGIN  
    update emp  
    set sal=sal+sal*0.10  
    where comm<>sal*0.09;  
END;  
SQL> /  
Procedure created.
```

Executing Procedure:

```
SQL> execute emp_proc1;
```

Output

PL/SQL procedure successfully completed.

Lab Practical No:13

Question:

13. Write a PL/SQL program to demonstrate Packages.

Step1: Creating package specification

```
CREATE or replace PACKAGE emppackage AS  
procedure emp_proc2;  
function fname2(a in number,b in out number)  
return number;  
end;
```

Step2: Creating package definition or body

CREATE or replace PACKAGE body emppackage AS

```
procedure emp_proc2
is
begin
    update emp
    set sal=sal+sal*0.10
where comm<>sal*0.09;
        DBMS_OUTPUT.PUT_LINE ('I am a procedure ');
end emp_proc2;
```

```
function fname2(a in number,b in out number)
return number is
begin
b:=a;
return b;
end fname2;
END;
```

Step3: Calling function / procedure of a package in a program

```
declare
    a number:=10;
    b number:=10;
begin
a:=emppackage.fname2(20,b);
dbms_output.put_line(a || b);
emppackage.emp_proc2;
end;
/
```

Output

2020

I am a procedure

PL/SQL procedure successfully completed.

Lab Practical No:14

Question:

14. Write PL/SQL queries to create Triggers.

SQL>create or replace trigger t11

before update on emp

for each row

begin

```
if :new.sal<1000
then
dbms_output.put_line('trigger fired');
end if;
end;
```

Trigger created.

Trigger gets fired when update is performed on EMP table

```
SQL> update emp
      set sal=500
      where empno=69000;
```

Output

trigger fired

1 row updated.