VARDHAMAN COLLEGE OF ENGINEERING
(AUTONOMOUS)
Affiliated to JNTUH, Approved by AICTE, Accredited by NAAC with A++ Grade, ISO 9001:2015 Certified
Kacharam, Shamshabad, Hyderabad – 501218, Telangana, India

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

LAB MANUAL

DATABASE MANAGEMENT SYSTEMS

II Yr II SEM B.TECH CSE/IT

Course Title	Database Management Systems
Course Code	A6508
Course Type	Integrated
Category	Core Engineering
Regulation	VCE-R20
Academic Year	2021-2022

COURSE OVERVIEW

This course introduces the core principles and techniques required in the design and implementation of database systems. This course focus on relational database management systems, including database design theory: E-R modeling, data definition and manipulation languages, database security and administration. It also covers essential DBMS concepts such as: Transaction Processing, Concurrency Control, Recovery and various types of databases like distributed database, and intelligent database, Client/Server. Students can undertake a semester project to design, build a simple database system and demonstrate competence with the fundamental tasks involved with modeling, designing, and implementing a DBMS. It also provides students with theoretical knowledge and practical skills required for using databases in information technology applications.

COURSE OBJECTIVE

The course enables the students to model, design and implement a database supporting optimum queries for any real world scenario given.

COURSE OUTCOMES (COs)

After the completion of the course, the student will be able to:

CO#	Course Outcomes	POs	PSOs
A6508.1	Design a database for a given problem using E-R diagrams and Relational Model.	-	-
A6508.2	Construct Queries in Relational algebra, relational calculus and SQL for a case study	2,5	1,2
A6508.3	Use Normalization techniques to reduce data redundancy in data base.	1,3	1
A6508.4	Select transaction control and recovery methods to keep data base consistent.	1,4	1
A6508.5	Compare various techniques and NoSQL databases for efficient access.	2,3	1,2

BLOOM'S LEVEL OF THE COURSE OUTCOMES

CO#	Bloom's Level					
	Remember (L1)	Understand (L2)	Apply (L3)	Analyze (L4)	Evaluate (L5)	Create (L6)
A6508.1			✓			
A6508.2			✓			
A6508.3			✓			
A6508.4				✓		
A6508.5			✓			

COURSE ARTICULATION MATRIX

CO#/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
A6508.2		3			3								3	3
A6508.3	3		3										3	
A6508.4	2			3									3	
A6508.5		3	3										3	2

Note: 1-Low, 2-Medium, 3-High

LIST OF PROGRAMS FOR PRACTICE:

No	Title of the Experiment	Tools and Techniques	Expected Skills/Ability
1.	Draw an ER diagram for University Database.	Oracle SQL Plus IDE: SQL Developer OS: Windows / Linux	Able to write SQL queries by using various SQL commands
	Practice on SQL DDL commands.		
2.	Draw an ER diagram for Company Database.		
	Practice on SQL DML commands.		
3.	Practice on SQL operators and Aggregate operations.		
4.	Practice on SQL String functions, Math functions, Conversion Functions, Analytic functions and Date functions.		
5.	Practice on Group By, Having, ORDER BY clauses.		Able to write SQL Queries by using various operators
	Practice on SQL TCL commands.		
	CASE STUDY 1: EMPLOYEE AND DEPARTMENT DATABASE	Oracle SQL Plus	Able to write SQL Queries by using various operators
1.	The BlueX Company pvt.ltd has maintaining Employee information contains employee details .The company has four departments. Any employee working in the company belongs to any one of the department. An employee joined in company above 25 years only. The company may give commission for every employee if and only if more than 2 years experience. Construct the database design such that there is no redundancy. **Consider the table structure as follows:** Employee(empno,ename,job,mgr,hiredate,sal,comm,deptno) Department(deptno, dname,location) **Construct queries for the following:** 1. Write queries for creating above relations Employee and Department. 2. Write queries for inserting necessary data into above		

	relations 3. display all information of emp table 4. display unique jobs from emp table 5. list the employes in ascending order of their salaries 6. display unique job groups in descending order 7. Display all the details of all 'Mgrs' 8. List the emps who joined before 1981. 9. List the Empno, Ename, Sal, Daily sal of all emps in the asc order of Annsal 10. Display the Empno, Ename, job, Hiredate, Exp of all Mgrs 11. List the Empno, Ename, Sal, Exp of all emps working for Mgr 7369 12. Display all the details of the emps whose Comm. Is more than their Sal. 13. List the emps in the asc order of Designations of those joined after the second half of 1981. 14. List the emps along with their Exp and Daily Sal is more than Rs.100. 15. List the emps who are either 'CLERK' or 'ANALYST' in the Desc order. 16. List the emps who joined on 1-MAY-81,3-DEC-81,17-DEC-81,19-JAN-80 in asc order of seniority. 17. List the emp who are working for the Deptno 10 or20 18. List the emps who are joined in the year 81. 19. List the emps who are joined in the month of Aug 1980. 20. List the emps Who Annual sal ranging from 22000 and 45000 21. List the Enames those are having five characters in their Names. 22. List the Enames those are starting with 'S' and with five characters. 23. List the emps those are having four chars and third character must be 'r'. 24. List the emps whose Sal is four digit number ending with Zero. 25. List all the emps except 'PRESIDENT' & 'MGR" in asc order of Salaries. 26. List all the emps who joined before or after 1981. 27. List the emps whose Empno not starting with digit78 28. Display the details of SMITH. 29. Display the location of SMITH. 30. Display the total information of the emps along with Grades in the asc order. 31. List the details of the emps whose Salaries more than the employee BLAKE. 32. List the emps whose Jobs are same as ALLEN 33. List the emps who are senior to King 34. List the emps Whose Jobs are same as MILLER or Sal is	IDE: SQL Developer OS: Windows / Linux	

	more than ALLEN. 35. Find details of highest paid employee. 36. Find the highest paid employee of sales department. 37. List the employee in dept 20 whose sal is >the average sal 0f dept 10 emps. 38. List the no. of emps in each department where the no. is more than 3. 39. Display the number of employee for each job group 40. Display the number of employee for each job group deptno wise. 41. List the department, details where at least two emps are working 42. List the employees whose salary is more than 3000 after giving 20% increment. 43. List the emps name ,dept, sal and comm. For those whose salary is between 2000 and 5000 while loc is Chicago. 44. List the name ,job, dname, location for those who are working as MGRS.		
	CASE STUDY 2: SAILORS, RESERVES, BOATS DATA BASE		
2.	In Database user has to maintain sailors information with sailors sid, sailor name and every sailor age is more than 25 years and has a rating i.e (rating >=10),the sailors reserved the boats for shipment of goods. Each boat identified by bid, name, color. Every sailors may reserve more than one boat. Reservation can notice based on the date. **Answer to the following Queries** 1. Create above relations and create indexing for accessing records faster. 2. First insert data into sailors table , then insert data into Boats table and last insert data into Reserves table. Use data shown in above tables to insert. 3. display the sailors names and age 4. display the unique sailor names and age 5. Find the names of sailors who have reserved at least one boat. 6. Find all information of sailors who have reserved boat number 101 7. Find the names of sailors who have reserved a red boat 8. Find the name and the age of the youngest sailor 9. Calculate the average age of all sailors 10. Find the average age of sailors for each rating level 11. Find the sid's , names of sailors who have reserved all boats and having age greater than 30. 12. Find the sids ,names of sailors who have reserved a red or a		Able to write SQL Queries by using various operators

	green boat 13. Find the sids of sailors with age over 20 who have not reserved a red boat 14. Compute increments for the rating of sailors who have sailed two different boats on the same day 15. Find the average age of sailors who are of voting age (i.e., at least 18 years old) for each rating level that has at least two sailors. 16. Find those ratings for which the average age of sailors is the minimum overall ratings 17. Find sailors whose rating is better than some sailor called "Horatio" 18. Find sailors whose rating is better than every sailor called "Horatio 19. Find the names of sailors who are older than the oldest sailor with a rating of 10 20. Find the average age of sailors for each rating level that has at least two sailors		
6.	Practice on queries involving different types of joins.		Able to write SQL Queries by using various operators
7.	Practice on queries using Co-related sub Queries and nested queries.		Able to write SQL Queries by using various operators
8	**Practice on PL/SQL basics for writing programs using programming constructs like variables, operators and conditional, control statements.**		
	A) Write a PL/SQL program to read number from a user and find out whether it is Odd or Even.		Able to write PL/SQL programs
	B) Write a PL/SQL program to find the sum of numbers from 1-20 using for or while loop.		
	C) Write a PL/SQL program to find biggest of 3 numbers.		
	Practice on PL/SQL programs using cursors.		
9.	A) Write a cursor program to display all Employee Details.		Implement cursors
	B) Write a cursor program to display the name and salary of an employee with empno:501.		
	C) Write a cursor program to display the no.of rows updated after executing update command for changing salary of employees to 10,000.		
	Practice on PL/SQL programs using triggers.		

10.	A)Write a PL/SQL program to display the salary of an employee before updation and after updation and also display the salary difference using Triggers. B)Write a PL/SQL program for inserting record to back_up table before deletion using Triggers.		Implement Triggers
11.	**Practice on PL/SQL programs involving stored procedures and functions**		
	A) Write a PL/SQL program to create stored procedure for finding biggest of 3 numbers. B) Write a PL/SQL program using functions to find the smallest of 2 numbers and also factorial of a given number.		Procedures and Functions
12.	Write a python program to connect to Oracle Database and perform basic operations like creating table, insertion of rows, displaying all rows, updating any one row.		
13.	Case Study in developing a database following all steps in the design of databases elaborating normalization and denormalization. • Mention any 10 requirements by the user. • Draw the ER diagram based on the requirements. • Convert the ER diagram into tables. • Justify which normal form is satisfied by the database. • Normalize the tables upto minimum 3NF.		

LIST OF EXPERIMENTS WITH MAPPING:

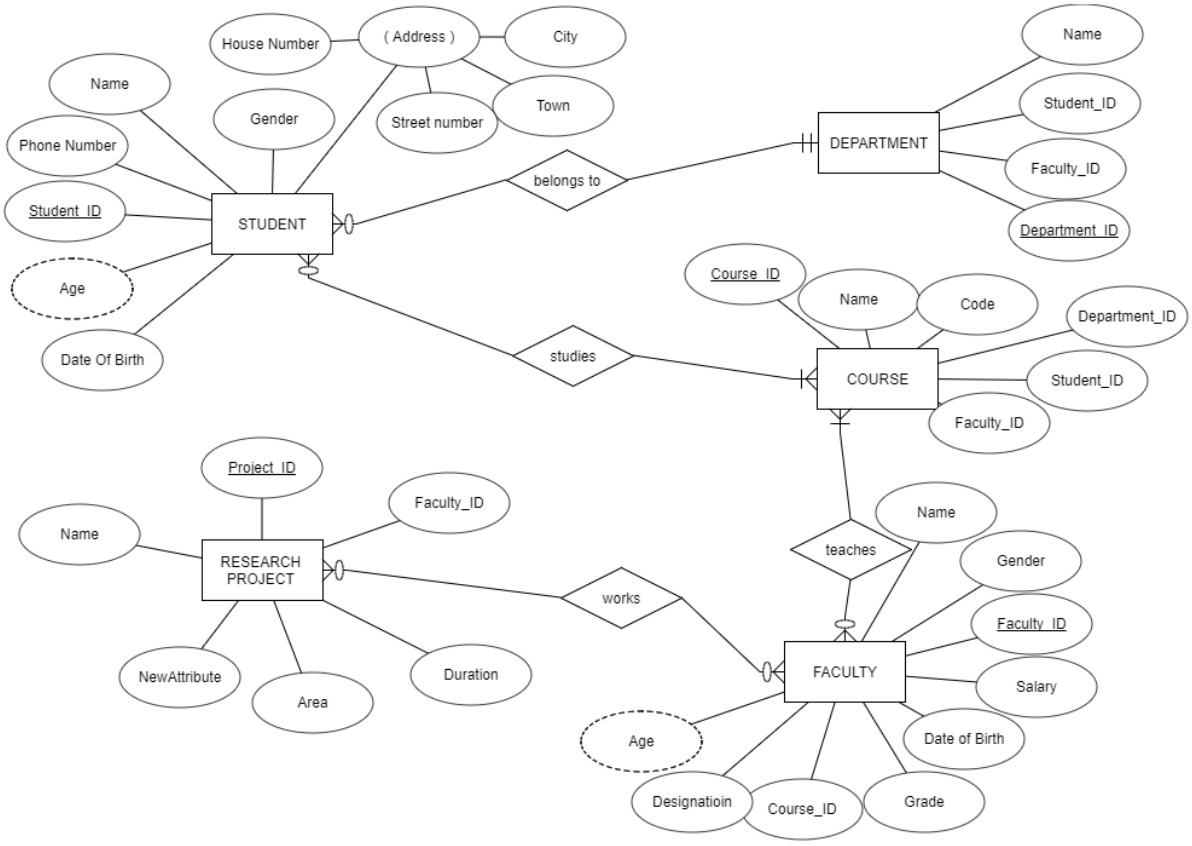
No	Title of the Experiment	CO#	BL#
1.	Draw an ER diagram for University Database.	CO1	L3
	Practice on SQL DDL commands.	CO2	L3
2.	Draw an ER diagram for Company Database.	CO1	L3
	Practice on SQL DML commands.	CO2	L3
3.	Practice on SQL operators and Aggregate operations.	CO2	L3
4.	Practice on SQL String functions, Math functions, Conversion Functions, Analytic functions and Date functions.	CO2	L3
5.	Practice on Group By, Having, ORDER BY clauses.	CO2	L3
	Practice on SQL TCL commands.	CO2	L3
6.	Practice on queries involving different types of joins.	CO2	L3
7.	Practice on queries using Co-related sub Queries and nested queries.	CO2	L3
	CASE STUDY 1: SAILORS DATABASE	CO2	L3
	CASE STUDY 2: EMPLOYEE DEPARTMENT DATABASE	CO2	L3
8	**Practice on PL/SQL basics for writing programs using programming constructs like variables, operators and conditional, control statements.**		
	A) Write a PL/SQL program to read number from a user and find out whether it is Odd or Even.	CO2	L3
	B) Write a PL/SQL program to find the sum of numbers from 1-20 using for or while loop.	CO2	L3
	C) Write a PL/SQL program to find biggest of 3 numbers.	CO2	L3
	Practice on PL/SQL programs using cursors.		
9.	A) Write a cursor program to display all Employee Details.	CO2	L3
	B) Write a cursor program to display the name and salary of an employee with empno:501.	CO2	L3
	C) Write a cursor program to display the no.of rows updated after executing update command for changing salary of employees to 10,000.	CO2	L3
	Practice on PL/SQL programs using triggers.		
10.	A)Write a PL/SQL program to display the salary of an employee before updation and after updation and also display the salary difference using Triggers.	CO2	L3

	B)Write a PL/SQL program for inserting record to back_up table before deletion using Triggers.	CO2	L3
11.	**Practice on PL/SQL programs involving stored procedures and functions**		
	A) Write a PL/SQL program to create stored procedure for finding biggest of 3 numbers.	CO2	L3
	B) Write a PL/SQL program using functions to find the smaller of 2 numbers and also factorial of a given number.	CO2	L3
12.	Write a python program to connect to Oracle Database and perform basic operations like creating table, insertion of rows, displaying all rows, updating any one row.	CO2	L3
13.	Case Study in developing a database following all steps in the design of databases elaborating normalization and denormalization.	CO3	L4

WEEK-1

1. A) DRAW AN ER DIAGRAM FOR UNIVERSITY DATABASE

ER DIAGRAM:



PRE LAB QUESTIONS:

1. What is an ER diagram?
2. What are the requirements necessary to draw an ER diagram?
3. What are Attributes?
4. What symbol is used to mention relationship between 2 attributes?
5. What is key attribute?
6. What is entity?
7. What are the components of ER diagram?

POST LAB QUESTIONS:

1. What is Cardinality of relationship between attributes?
2. Describe briefly different types of attributes?
3. Describe different types of relationships?
4. What is role and role indicator?
5. What are participation constraints in ER diagram?
6. What is Generalization in ER diagram?
7. What is Aggregation in ER diagram?

WEEK - 1

1. B) PRACTICE ON SQL DATA DEFINITION LANGUAGE (DDL) COMMANDS IN RDBMS

1.1 OBJECTIVE

Practice On SQL Data Definition Language (DDL) Commands in RDBMS By Using Below Case Study

CASE STUDY: EMPLOYEE AND DEPARTMENT DATABASE

The BlueX Company pvt. ltd has maintaining Employee information contains employee details The company has four departments. Any employee working in the company belongs to any one of the department. An employee joined in company above 25 years only. The company may give commission for every employee if and only if more than 2 years experience.

1.2 RESOURCES

Oracle 10g Express edition

1.3 DESCRIPTION/PROGRAM LOGIC

The communication with database such as create an object, alter the structure of an object or to drop the object created. The data definition language is the subset of SQL commands used to create, modify or remove oracle database structures, including tables .The commands have an immediate effect on the database, and also record information in the data dictionary

1.4 PROCEDURE

COMMAND : CREATE

This command is used to create the structure of database.

Rules to create the table

Oracle reserved words cannot be used.
Underscore, numerals, letters are allowed but not blank space.
Maximum length for the table name is 30 characters.
Two different tables should not have same name.
We should specify a unique column name.
We should specify proper data type along with width.
We can include “not null” condition when needed. By default it is ‘null’.

Syntax :

Create table <table name > (column_name1 type(size), Column_name2 type(size), …);

INPUT:

SQL> create table emp (empno number(4) not null, ename varchar2(10) not null, 4 job varchar2(9) not null, mgr number(4), hiredate date, sal number(7,2), comm number(7,2), deptno number(2) references dept(deptno));

OUTPUT:

SQL> Table created

COMMAND : DESCRIBE

This command is used to describe the structure of table

Syntax:

desc <tablename>;

INPUT:
SQL> desc emp;
OUTPUT:

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME	NOT NULL	VARCHAR2(10)
JOB	NOT NULL	VARCHAR2(7)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

COMMAND :ALTER
This command is used to add a column or to increase the size of the column

Syntax:
alter table <table name> add/modify (column data type(size));

INPUT:
SQL> alter table emp modify(job varchar2(20));

OUTPUT:
Table altered
SQL> desc emp;

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME	NOT NULL	VARCHAR2(10)
JOB	NOT NULL	VARCHAR2(20)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)
COMM		NUMBER(7,2)
DEPTNO		NUMBER(2)

INPUT:
SQL> alter table emp add (location varchar2(20));

OUTPUT:
Table altered
SQL> desc emp;

Name	Null?	Type
EMPNO	NOT NULL	NUMBER(4)
ENAME	NOT NULL	VARCHAR2(10)
JOB	NOT NULL	VARCHAR2(20)
MGR		NUMBER(4)
HIREDATE		DATE
SAL		NUMBER(7,2)

COMM	NUMBER(7,2)
DEPTNO	NUMBER(2)
LOCATION	VARCHAR2(10)

COMMAND: DROP
This command is used to delete the table from database.

Syntax:
drop table <tablename>;

INPUT:
SQL> drop table emp;

OUTPUT:
Table dropped

COMMAND :TRUNCATE
This command is used delete the rows but structure remains

Syntax:
truncate table <table name>;

INPUT:
SQL> Truncate table customer;

OUTPUT:
Table truncated

1.5 PRE LAB QUESTIONS

1. What is SQL? Is it a procedural language or nonprocedural language?
2. Differentiate Truncate and drop.
3. What is the difference between SQL, PL/SQL ?
4. Explain relational model?
5. Why do we use Alter command?
6. Elaborate DDL.
7. What is meant by view?
8. What is meant by tuple?
9. What is meant by attribute?
10. Explain different data types exist in SQL?
11. Write the applications of DBMS?

1.6 LAB ASSIGNMENT

1. Add a long column called COMMENTS to your PROJECT TABLE. Also add a number column called HOURS to the ASSIGNMENT table?
2. Customer table

Name	Data type
CUST_NAME	VARCHAR2(15)
CUST_STREET	VARCHAR2(15)
CUST_CITY	VARCHAR2(15)

   Add customer salary column to the above table?

3. Branch table

Name	Data type
BRANCH_NAME	VARCHAR2(15)
BRANCH_CITY	VARCHAR2(15)
ASSERTS	NUMBER(15)

   Increase the size of data type for asserts to the above branch table?
4. Show the structure of the customer, branch tables?
5. Add and drop a column DOJ to the emp table?

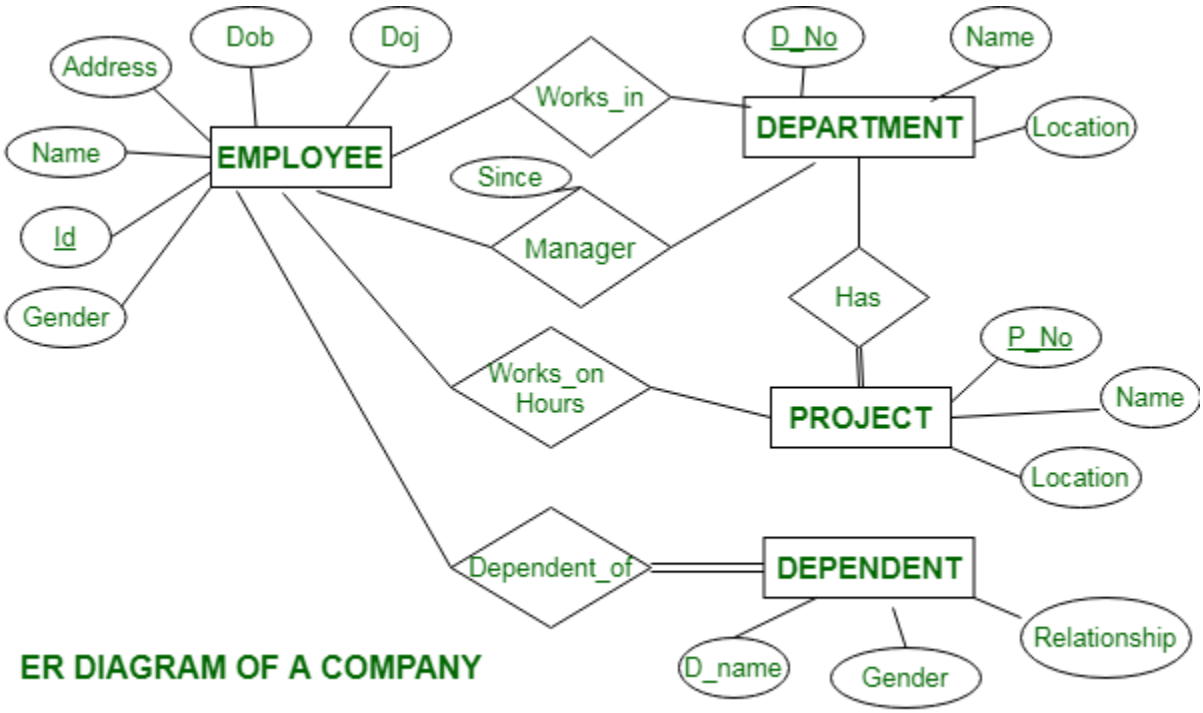
1.7 POST LAB QUESTIONS

1. What are the differences between data, database, and database management system?
2. What is the difference between the char and varchar2?
3. How many bytes occupied by date data type?
4. What is the maximum size for long data type?
5. State the syntax of adding a new column to the table.
6. Is it possible to rollback after delete operation?
7. Is it possible to change the name of column and delete the column once created?
8. What is the maximum size for char?
9. How many bytes occupied by varchar2 data types?
10. Can you give same name for more than two tables in a data base?
11. What is the difference between float and long data type?

WEEK 2

2. A) DRAW AN ER DIAGRAM FOR COMPANY DATABASE

ER DIAGRAM:



ER DIAGRAM OF A COMPANY

PRE LAB QUESTIONS:

1. What is the main purpose of ER diagram?
2. What is derived attribute?
3. What are the steps involved in drawing a ER diagram?
4. What is many-one relationship?
5. What is Composite attributes?
6. Explain difference between derived and multi-valued attribute.

POST LAB QUESTIONS:

1. What is Descriptive attribute?
2. What is Recursive relationship?
3. What is Entity set and Relationship set?
4. What is the difference between weak and strong entities?
5. Draw the symbols for representing 1-1,1-M,M-1,M-N relationship?

WEEK - 2

2. B) PRACTICE ON SQL DATA MANIPULATION LANGUAGE (DML) COMMANDS IN RDBMS

2.1 OBJECTIVE

To do manipulations on the database developed for the given case study by using DML commands
CASE STUDY: EMPLOYEE AND DEPARTMENT DATABASE
The BlueX Company pvt. ltd has maintaining Employee information contains employee details The company has four departments. Any employee working in the company belongs to any one of the department. An employee joined in company above 25 years only. The company may give commission for every employee if and only if more than 2 years experience.

2.2 RESOURCES

Oracle 10g Express edition.

2.3 DESCRIPTION /PROGRAM LOGIC

The insert command is used to add one or more rows to a table. The values are separated by commas and the data types char and date are enclosed in apostrophes. The values must be entered in the same order as they are defined. The select command used to retrieve information from the table. It is generally referred to as querying the table. We can either display all columns in a table or only specify columns from the table. The update command is used to alter the column values in a table. A single column may be updated or more than one column could be updated. The delete command is used to after inserting row in a table we can also delete them if required. The delete command consists of a from clause followed by an optional where clause

2.4 PROCEDURE

COMMAND: INSERT
Inserting a single row into a table

Syntax:
insert into <table name> values (value list);
SQL> insert into s values ('s3','sup3','blore', 10)

Inserting more than one record using a single insert commands:

INPUT

```
SQL>  insert into emp
Values(&empno,'&ename','&job','&mgr','&hirdate',&sal,'&comm',&deptno,'&empgen');
Enter value for empno: 23
Enter value for ename: aaa
Enter value for job: clerk
Enter value for mgr: 45
Enter value for hirdate: 12-dec-89
Enter value for sal: 5000
Enter value for comm: 20
Enter value for deptno: 30
Enter value for empgen: m
```

old 1: insert into emp
values(&empno,'&ename','&job','&mgr','&hirdate',&sal,'&comm',&deptno,'&emp
new 1: insert into emp values(23,'aaa','clerk','45','12-dec-89',5000,'20',30,'m')

1 row created.

SQL> /
Enter value for empno: 24
Enter value for ename: bbb
Enter value for job: manager
Enter value for mgr: 46
Enter value for hirdate: 13-jan-89
Enter value for sal: 6000
Enter value for comm: 60
Enter value for deptno: 20
Enter value for empgen: female
old 1: insert into emp
values(&empno,'&ename','&job','&mgr','&hirdate',&sal,'&comm',&deptno,'&emp
new 1: insert into emp values(24,'bbb','manager','46','13-jan-89',6000,'60',20,'female')

1 row created.

OUTPUT
emp table

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM.	DEPTNO
7369	SMITH	CLERK	7902	17 – DEC – 80	800		20
7499	ALLEN	SALESMAN	7698	20 – FEB – 81	1600	300	30
7521	WARD	SALESMAN	7698	22 – FEB – 81	1250	500	30
7566	JONES	MANAGER	7839	02 – APR – 81	2975		20
7654	MARTIN	SALESMAN	7698	28 – SEP – 81	1250	1400	30
7698	BLAKE	MANAGER	7839	01 – MAY – 81	2850		30
7782	CLARK	MANAGER	7839	09 – JUN – 81	2450		10
7788	SCOTT	ANALYST	7566	19 – APR – 87	3000		20
7839	KING	PRESIDENT		17 – NOV – 81	5000		10
7844	TURNER	SALESMAN	7698	08 – SEP – 81	1500	0	30
7876	ADAMS	CLERK	7788	23 - MAY – 87	1100		20
7900	JAMES	CLERK	7698	03 – DEC – 81	950		30
7902	FORD	ANALYST	7566	03 – DEC – 81	3000		10
7934	MILLER	CLERK	7782	23 – JAN – 82	1300		10

Skipping the fields while inserting
Syntax:
insert into <tablename>(coln names to which data to be inserted)> values (list of values);
Other way is to give null while passing the values

COMMAND : SELECT
To retrieve all rows from a table

Syntax:
SQL>Select * from table name;

INPUT:
SQL>Select * from emp;
OUTPUT:

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM.	DEPTNO
7369	SMITH	CLERK	7902	17 – DEC – 80	800		20
7499	ALLEN	SALESMAN	7698	20 – FEB – 81	1600	300	30
7521	WARD	SALESMAN	7698	22 – FEB – 81	1250	500	30
7566	JONES	MANAGER	7839	02 – APR – 81	2975		20
7654	MARTIN	SALESMAN	7698	28 – SEP – 81	1250	1400	30
7698	BLAKE	MANAGER	7839	01 – MAY – 81	2850		30
7782	CLARK	MANAGER	7839	09 – JUN – 81	2450		10
7788	SCOTT	ANALYST	7566	19 – APR – 87	3000		20
7839	KING	PRESIDENT		17 – NOV – 81	5000		10
7844	TURNER	SALESMAN	7698	08 – SEP – 81	1500	0	30
7876	ADAMS	CLERK	7788	23 - MAY – 87	1100		20

To retrieve specific columns from a table
Syntax:
Select column_name1, …..,column_namen from table name;

INPUT:
SQL> select ename, deptno, sal from emp;

OUTPUT:

ENAME	DEPTNO	SAL
adams	20	1100
allen	30	1600
blake	30	2850
clark	10	2450
ford	20	3000
james	30	950
jones	20	2975

KEYWORD : DISTINCT
Elimination of duplicates from the select clause: It prevents retrieving the duplicated values. Distinct keyword is to be used.

Syntax:
SQL>Select DISTINCT col1, col2 from table name;

INPUT:
SQL>select distinct job from emp;
OUTPUT:
JOB

CLERK

SALESMAN
MANAGER
ANALYST
PRESIDENT

To select specific rows from a table we include 'where' clause in the select command. It can appear only after the 'from' clause

Syntax:
Select column_name1, …..,column_name from table name where condition;

INPUT:
SQL> select ename, job from emp where job = 'manager';
OUTPUT:

ENAME	**JOB**
jones	manager
blake	manager
clark	manager

COMMAND: UPDATE
To modify column or group of columns
Syntax:
update <tablename> set field=values where condition;

INPUT:
SQL>update std set branch='mca' where sno=03;
1 row updated.
sql> select * from std;

OUTPUT:

S No	**SNAME**	**BRANCH**
1	Ram	B.Tech
2	Ravi	B.Tech
3	Krish	MCA
4	Rfde	B.Tech

COMMAND :DELETE
The command is used to delete the rows in a table
Syntax:
Delete from <table_name> where conditions;
INPUT:
SQL> delete from emp where empno=7369;

2.5 PRE LAB QUESTIONS

1. What is meant by column alias?
2. Elaborate DML.
3. What are literals? Give an example?
4. Which command is used to display the details of a table?
5. What is meant by null value in the DBMS? How to handle these values?

6. What is the purpose of update command?
7. How delete is different from truncate and drop?

2.6 LAB ASSIGNMENT

1. Consider the dept table

NAME	DATATYPE
DEPTNAME	NUMBER(2)
DNAME	VARCHAR2(15)
LOCATION	VARCHAR2(15)

   Insert 5 records into dept table
2. Add to dept one row at a time using the insert into syntax
3. Update the emp table to set the default commission of all employees to Rs1000/- who are working as managers
4. Create a pseudo table employee with the same structure as the table emp and insert rows into the table using select clauses
5. Delete the employee whose empno is 7599.
6. List the records in the emp table order by salary in ascending order
7. Display only those employees whose deptno is 30.

2.7 POST LAB QUESTIONS:

1. State 2 different syntaxes for insert statement.
2. Can a delete operation be rollbacked ?If yes,how?
3. Write command to select all rows in a table.
4. What is the purpose of order by clause? What is the default ordering of the data?
5. How to display the data in reverse order?
6. Why do we use WHERE CLAUSE?
7. Write the difference between truncate and delete statement.
8. What is the difference between having and where clause?

WEEK 3: PRACTICE ON RELATIONAL OPERATORS, SQL OPERATORS, SET OPERATORS

3.1 OBJECTIVE

To get the results in specified condition for a given queries using relational operators, SQL operators, set operators

3.2 RESOURCES

Oracle 10g Express edition.

3.3 DESCRIPTION/PROGRAM LOGIC

RELATIONAL OPERATORS

In complex search conditions which form the requirement of the end user cannot be implemented without the usage of the relational, logical and special operators available in oracle. The usage of these operators is not limited to DML select statements but can be used in almost all of the various types of SQL commands.

SQL OPERATORS: There are four SQL operators, which operate with all datatypes.

SET OPERATORS: The operators are used to combine the two or more select statements

3.4 PROCEDURE

OPERATOR: RELATIONAL

Syntax:

select column name
From <table name>
Where condition**;**
Example:
Select ename, empno, job, deptno
From emp
Where job='CLERK';
Example:
Select ename, sal, comm.
From emp
Where comm>sal;

OPERATOR: BETWEEN AND

Test for values between , and inclusive low and high range

Syntax:

select column list from table name where column name between low range and high range;

INPUT:

SQL> select empno,ename,sal from emp where sal between 1000 and 2000 and job = 'clerk';

OUTPUT:

EMPNO ENAME SAL
--------- ---------- ----------

7876 adams 1100
7934 miller 1300

OPERATOR: IN
Test for the values in specified list

Syntax:
select column list from table name Where column name in(list of values);

INPUT:
SQL> select ename,mgr from emp where mgr in(7702,7566,7788);

OUTPUT:
ENAME MGR
---------- ----------
scott 7566
adams 7788
ford 7566

OPERATOR: LIKE
select rows that match a character pattern

Syntax:
select column list from table name Where column name like 'char%';

INPUT:
SQL> select age from sailor where sname like 'b___%';

OUTPUT:
AGE

33

OPERATOR: NULL
Test for values that are null

Syntax:
select column list from table name where column in null;

Example:
select ename , mgr from emp where mgr is null;

NEGATION OPERATORS

OPERATOR: NOT BETWEEN AND
Test for values not between and inclusive low and high range

Syntax:

Select column names from <table-name> where column name not between low range and high range;

INPUT:
SQL> select ename,sal from emp where sal not between 1000 and 3000;

OUTPUT:

ENAME	SAL
Smith	800
King	5000
james	950

OPERATOR:NOT LIKE
select rows that can match a character pattern

Syntax:
Select column names from <table-name> where column name not like 'character%';

INPUT:
SQL> select ename,job from emp where job not like 'm%';

OUTPUT:

ENAME	JOB
smith	clerk
allen	salesman
ward	salesman
martin	salesman
scott	analyst
king	president
turner	salesman
adams	clerk
james	clerk
ford	analyst
miller	clerk

OPERATOR: NOT IN
Test for the values in not in specified list

Syntax:
select column list from table name Where column name not in(list of values);

INPUT:
SQL> select ename,mgr from emp where mgr not in(7702,7566,7788);

OUTPUT:

ENAME	MGR
smith	7902
allen	7698

ward	7698
jones	7839
martin	7698
blake	7839
clark	7839
turner	7698
james	7698
miller	7782

OPERATOR : NOT NULL

Test for values that are not null

Syntax:

select column list from table name where column in not null;

INPUT:

SQL> select ename,mgr from emp where mgr is not null;

OUTPUT:

ENAME	MGR
smith	7902
allen	7698
ward	7698
jones	7839
martin	7698
blake	7839
clark	7839
scott	7566

SET OPERATORS

OPERATOR: UNION

return all distinct rows retrieved by either of the queries.

Example:

```
Select job from emp where deptno=10
  Union
select job from emp where deptno=30;
```

OPERATOR :UNION ALL

Returns all rows including duplicates retrieved by either of the query

Example:

```
Select job from emp where deptno=10
  Union all
select job from emp where deptno=30;
```

OPERATOR: INTERSECT

Return only rows retrieved by both of the queries.

Example:

```
Select job from emp where deptno=10
 intersect
select job from emp where deptno=30
```

OPERATOR: MINUS

Return all rows retrieved by first query that are not in second

Example:

```
Select job from emp where deptno=10
 minus
select job from emp where deptno=30;
```

3.5 PRE-LAB QUESTIONS

1. What are SQL operators?
2. Why do we use operators in DBMS?
3. What is the difference between union and union all?
4. What are Relational operators?
5. What are set operators?

3.6 LAB ASSIGNMENT

1. Select all employees from department numbers 7369,7499
2. Display all the details of the records whose employee name starts with 'S'.
3. Display all the details of the records whose employee name does not starts with 'S'.
4. Display the rows whose empno ranges from 7500 to 7600.
5. Display the rows whose empno not in range from 7500 to 7600.
6. Calculate the square root of the salary of all employees
7. List all employee names and their manager whose manager is 77499or 7566 or 7611**.**

3.7 POST-LAB QUESTIONS

1. What are SQL bitwise ,comparison operators?
2. State some logical operators in SQL.
3. State some syntaxes of set operatos.
4. What are the different types of SQL operators?
5. What is the use of SQL compound operators?
6. What is the use of LIKE operator and mention it's syntax.

4. PRACTICE ON BUILT IN FUNCTIONS/AGGREGATE FUNCTIONS IN RDBMS

4.1 OBJECTIVE

To retrieve the values from single rows and group of rows

4.2 RESOURCES

Oracle 10g Express edition

4.3 DESCRIPTION/PROGRAM LOGIC

Function accept zero or more arguments and both return one or more results. Both are used to manipulate individual data items. Operators differ from functional in that they follow the format of function_name(arg..). An argument is a user defined variables or constants. Most operators accept at most 2 arguments while the structure of functions permit to accept 3 or more arguments.
Function can be classifies into single row function and group functions.

SINGLE ROW FUNCTIONS

A single row function or scalar function returns only one value for every row queries in table. Single row function can appear in a select command and can also be included in a where clause. The single row function can be broadly classified as,

Date Function : They operate on date values and produce outputs, which also belong to date data type except for months, between, date function returns a number.

Numeric Function : Numeric function accept numeric input and return numeric values

Character Function : Single row character function accept character input and can return both character and number values.

Conversion Function: A conversion function convert a value from one data type to another.

The example that follows mostly uses the symbol table “dual”. It is a table, which is automatically created by oracle along with the data dictionary

GROUP FUNCTION: The group functions which will return results based on group of rows

4.4 PROCEDURE

DATE FUNCTION
FUNCTION: SYSDATE
Display current date.

Syntax:
Select Sysdate from dual;

INPUT:
SQL> select sysdate from dual;

OUTPUT:
SYSDATE

-- ---------
15-DEC-11

FUNCTION: ADD_MONTH

This function returns a date after adding a specified date with specified number of months.

Syntax:

Add_months(d,n); where d-date n-number of months

INPUT:

SQL> select add_months(sysdate,2) from dual;

OUTPUT:

ADD_MONTH

15-FEB-12

SQL> select add_months('13-mar-81',10) from dual;

ADD_MONTH

13-JAN-82

FUNCTION: LAST_DAY

It displays the last date of that month.

Syntax:

last_day (d); where d-date

INPUT:

SQL> select last_day(sysdate) from dual;

OUTPUT:

LAST_DAY()

31-DEC-11

FUNCTION: MONTHS_BETWEEN

It gives the difference in number of months

Syntax:

month_between (d1,d2); where d1 & d2 –dates

INPUT:

SQL> select months_between(sysdate,'30-oct-1988') from dual;

OUTPUT:
MONTHS_BETWEEN (SYSDATE,'30-OCT-1988')

277.536088

NUMERICAL FUNCTIONS:

FUNCTION :**ABSOLUTE**
Return absolute value

Syntax:
abs(n)

INPUT:
Select abs(-15) from dual;

OUTPUT:
ABS(-15)

15

FUNCTION: CEIL
Return smallest integer greater than or equal to n

Syntax:
ceil(n)
INPUT:
SQL> select ceil(15.7) "ceiling" from dual;

OUTPUT:
ceiling

16

FUNCTION: FLOOR
Returns largest integer equal to or less than n

Syntax:
floor(n)

INPUT:
select floor(100.2) from dual;

OUTPUT:
FLOOR

100

FUNCTION:POWER

Returns m raised to the nth power

Syntax:
power(m,n)

INPUT:
Select power(4,2) from dual;
OUTPUT:
POWER

16

FUNCTION :MOD
Returns remainder of m divided by n

Syntax:
mod(m,n)

INPUT:
SQL> Select mod(10,3) from dual;

OUTPUT:
MOD()

1

FUNCTION: ROUND
Returns to n rounded to m places

Syntax:
round(m,n)

INPUT:
Select round(100.256,2) from dual

OUTPUT:
ROUND

(100.25)

FUNCTION: LENGTH
Length of the string
syntax:
length(n)

INPUT:
SQL> select length('vardhaman') " length" from dual;

OUTPUT:

```
length
----------
   9
```

FUNCTION: LOGARITHM

Returns the logarithm , base m of n

Syntax:

LOG(A1,A2)

INPUT:

```
SQL>Select Log(10,100) from dual;
```

OUTPUT:

```
LOG(10,100)
-----------
    2
```

CONVERSION FUNCTIONS

FUNCTION:TO_CHAR()

This function converts date to a value of varchar type in a form specified by date format. If format is neglected then it converts date to varchar2 in the default date format

Syntax:

to_char(d,[format]);

Example: select to_char (sysdate, 'dd-mm-yy') from dual;

FUNCTION: TO_DATE()

This function converts character to date data format specified in the form character

Syntax:

to_date(d,[format]);

Example: select to_date('aug 15 2009','mm-dd-yy') from dual;

GROUP FUNCTIONS OR AGGREGATE FUNCTIONS:

FUNCTION: SUM

return sum of values of n

Syntax:

Sum(column name)

INPUT:

```
SQL> select sum(sal) from emp1;
```

OUTPUT:

SUM(SAL)

29025

FUNCTION: AVERAGE
Returns average value of n

Syntax:
avg([distict/all] n)

INPUT:
SQL> select avg(sal) from emp1;

OUTPUT:
AVG(SAL)

2073.21429

FUNCTION:COUNT
Returns the number of rows in the query

Syntax:
count(*/ [distinct/all])

INPUT:
SQL> select count(*) from emp1;

OUTPUT:
COUNT(*)

14

FUNCTION: MAXIMUM
Returns maximum value

Syntax:
max([distinct/all]expr)

INPUT:
SQL> select max(sal) from emp1;

OUTPUT:
MAX(SAL)

5000

FUNCTION; MINIMUM
Returns minimum value

Syntax:
min([distinct/all]expr);

INPUT:
SQL> select min(sal) from emp1;

OUTPUT :
MIN(SAL)

800

FUNCTION: STANDARD DEVIATION
Returns standard deviation of x, a number.
Syntax:
stddev(colname)

INPUT:
SQL> select stddev(sal) from emp1;

OUTPUT:
STDDEV(SAL)

1182.50322
FUNCTION: VARIANCE
Returns variance of x, anumber.
Syntax:
Variance([distinct/all] x)
INPUT:
SQL> select variance(sal) from emp1;

OUTPUT:
VARIANCE(SAL)

1398313.87

4.5 PRE-LAB QUESTIONS

1 What is the syntax for current date?
2 How to convert lower case letters in upper case?
3 How to calculate square root function?
4 What are aggregate operators?
5 State some built-in functions in SQL.

4.6 LAB ASSIGNMENT

1. Display number of employees working in each department and their department name?
2. Show that value returned by sign(n) function?
3. How many days between day of birth to current date?
4. Show that two substring as single string?

5. List all employee names, salary and 15% rise in salary.
6. List all employees which start with either B or C.
7. Display lowest paid employee details under each manager

4.7 POST-LAB QUESTIONS

1. Which function is used to find the length of word?
2. Give the example for replace function?
3. What is the function “trunk”
4. What is the difference between LN and LOG function?
5. What is the basic condition required to use aggregate operator in SQL query?
6. State syntax of using aggregate operator in SQl query.
7. Display the rounded date in the year format, month format, day format in the employees**.**
8. Where we can use group by clause and having clause?
9. Why built-in functions are used and what is its main purpose.
10. State some built-in functions in SQl along with their syntaxes.

WEEK 5

5. A) PRACTICE ON GROUP BY CLAUSE, HAVING CLAUSE, ORDER BY CLAUSE

5.1 OBJECTIVE

To calculate the summary of information for each group and to specify which groups are to displayed

5.2 RESOURCES

Oracle 10g Express edition.

5.3 DESCRIPTION/PROGRAM LOGIC

The order by clause used to retrieve the values sorted order group functions may be used to return summary information for each group having clause if you wish to specify which are to displayed

5.4 PROCEDURE

COMMAND: ORDER BY CLAUSE.

The order by clause used to retrieve the values sorted either in ascending order or descending order. The default is ascending order

INPUT:

SQL> select ename,deptno,sal from emp order by ename;

OUTPUT:

ENAME	DEPTNO	SAL
Adams	20	1100
Allen	30	1600
Blake	30	2850
Clark	10	2450
Ford	20	3000
James	30	950
Jones	20	2975
King	10	5000
Martin	30	1250
Miller	10	1300
Scott	20	3000

11 rows selected.

COMMAND: GROUP BY CLAUSE

The group by clause used to divide the rows in a table in to smaller groups.

INPUT:

SQL> select rating,avg(age) from sailors group by rating having count(*) >=2;

OUTPUT:

RATING AVG(AGE)

------- --------

8 40.5

7 40
3 44.5
10 25.5

COMMAND :HAVING CLAUSE

Restrict the groups that you return

INPUT:

SQL> select job,avg(sal) from emp group by job having count(*)>2;

OUTPUT:

JOB	AVG(SAL)
salesman	1400
clerk	1037.5
manager	2758.3

5.5 PRE-LAB QUESTIONS

1 Can we use where clause to restrict the groups that are returned.
2 Which clause is used to exclude /include groups ?
3 Can we use group by clause with in the groups?
4 When we use order by clause?
5 When we use having clause ?

5.6 LAB ASSIGNMENT

a) Write a query to display the total count of customer.
b) Write a query to display the minimum cost of product.
c) Write a query to display average value of product cost rounded to 2nd decimal places.
d) Write a query to display product name with total sale detail in descending order.
e) Write a query to display product name, sale date and total amount collected for the product.
f) Write a query to display sale date and total sale date wise which was sold after "14-jul-08".
g) Write a query to display the customer name who belongs to those places whose name is having I or P.
h) Write a query to display customer name who belongs to a city whose name contains characters 'C' and whose name contains character 'A'.
i) Write a query to display the customer name who does not belong to PUNE.

5.7 POST-LAB QUESTIONS:

1. Why having clause may precede the group by clause?
2. Which clause is used to restrict which group of rows defined by the GROUP BY clause are return by the query?
3. Which clause locks the rows selected by the query?
4. You can substitute a value for NULL using which function?
5. Write the order of using different clauses in SQL.

5.b) PRACTICE ON TRANSACTION CONTROL LANGUAGE COMMANDS IN RDBMS

5.1 OBJECTIVE

To implement basic DTL Commands such as commit, rollback, save point etc.

5.2 RESOURCES

Oracle 10g Express edition

5.3 DESCRIPTION/PROGRAM LOGIC

Commitoperation

A commit operation makes permanent all changes made under commitment control since the previous commit or rollback operation. The system also releases all locks related to the transaction.

Rollbackoperation

A rollback operation removes all changes made since the previous commit or rollback operation. The system also releases all locks related to the transaction.

5.4 PROCEDURE:

COMMAND : COMMIT

The COMMIT command is the transactional command used to save changes made by a transaction to the database. The COMMIT command will save all changes to the database since the last COMMIT or ROLLBACK command. Frequent commits in the case of transaction involving large amount of data is recommended. But too many commits can affect performance. In many implementations, an implicit commit of all the changes is done, if user logs off from the database

Syntax:

commit [work] [comment '*comment_text*']
commit [work] [force '*force_text*' [,*int*]]

INPUT:

SQL>commit;

OUTPUT:

Commit complete;

COMMAND : ROLLBACK

The ROLLBACK command is the transactional control command to undo the transactions that have not already been committed to the database. The ROLLBACK command can be issued to undo the changes since the last COMMIT or ROLLBACK.

Syntax:

ROLLBACK [WORK] [TO [SAVEPOINT]'savepoint_text_identifier'];
ROLLBACK [WORK] [FORCE 'force_text'];

INPUT:

SQL>rollback;

OUTPUT:

Rollback complete;

COMMAND : SAVEPOINT

Syntax:

SAVEPOINT <text identifier>;

INPUT:

```
update employees
  set salary = 95000
  where last_name = 'smith';
  savepoint justsmith;
  update employees
  set salary = 1000000;
  savepoint everyone;
  select sum(salary) from employees;
  rollback to savepoint justsmith;
  commit;
```

OUTPUT:

Commit complete;

5.5 PRE LAB QUESTIONS:

a. What is meant by data control language?
b. What is the purpose of DCL?
c. Why we need these commands?
d. What are TCL commands?
e. What is the main purpose of using TCL commands?

5.6 LAB ASSIGNMENT

1. Insert the any three record in the existing table and use rollback? Check the result?
2. Insert the any three record in the existing table and use commit ? Check the result?
3. Insert the any three record in the existing table and use save point? Check the result?

5.7 POST LAB QUESTIONS

1. What is the difference between rollback and save point?
2. what is the difference between commit and save point?
3. what is meant by transaction?
4. Write syntax of using savepoint.
5. Why do we use savepoint in SQL?
6. What is the main purpose of using rollback command?

CASE STUDY 1: EMPLOYEE AND DEPARTMENT DATABASE:

A. OBJECTIVE

To study about employee database.

CASE STUDY : EMPLOYEE AND DEPARTMENT DATABASE

The BlueX Company pvt. ltd has maintaining Employee information contains employee details .The company has four departments. Any employee working in the company belongs to any one of the department. An employee joined in company above 25 years only. The company may give commission for every employee if and only if more than 2 years experience. Answer for the following queries

B. RESOURCES

Oracle 10g Express edition

C. DESCRIPTION/ PROGRAM LOGIC

SCHEMA OF DATABASE:

emp	Details of employees Primary key : empno Foreign key : deptno refers to deptno of dept table. Ename and job not null
dept	Details of department primary key : deptno dname and loc not null

D. PROCEDURE

QUERY:

Display employee details in the order of their names?

INPUT:

```
SQL> select ename,deptno,sal from emp order by ename;
```

OUTPUT:

ENAME	DEPTNO	SAL
adams	20	1100
Allen	30	1600
Blake	30	2850
Clark	10	2450
Ford	20	3000
james	30	950
jones	20	2975
King	10	5000
martin	30	1250
miller	10	1300
Scott	20	3000

11 rows selected.

QUERY :

Find all clerks who earn between 1000 and 2000.

INPUT:
SQL> select empno,ename,sal from emp where sal between 1000 and 2000 and job = 'clerk';

OUTPUT:

EMPNO	ENAME	SAL
7876	adams	1100
7934	miller	1300

QUERY:
Find the employee who have no manager.

INPUT:
SQL > select ename,mgr from emp where mgr is null;

OUTPUT:

ENAME	MGR
king	

QUERY :
Find the ages of sailor whose name begin with b and at least 3 characters.

INPUT:
SQL> select age from sailor where sname like 'b___%';

OUTPUT:

AGE
33

QUERY : Find all employees who have one of the three mgr no`s.

INPUT:
SQL> select ename,mgr from emp where mgr in(7702,7566,7788);

OUTPUT:

ENAME	MGR
Scott	7566
adams	7788
Ford	7566

E. PRE-LAB QUESTIONS

1. Find employee details?
2. Display department details?
3. List employee details whose name start with “s”?
4. List employee details whose job is clerk?
5. Display employee details in the order of their deptno?

F. LAB ASSIGNMENT

1. Update emp table and change employee name ,ADAMS to ADAM?
2. Select deptno, dname ,of deptno>10 and located in 'NEWYORK'
3. List all employee details who belongs to deptno=10 and whose job is clerk
4. List all employee hired during 1981?
5. List all empno, ename of all employee in format "empno ename"
6. Find the total number of clerks in department 10
7. Find the average salary of employees?
8. List all employee of their average salaries
9. Find minimum salary paid employee and employee details with that salaries?
10. Find the name of employee which starts with 'A' and end with 'N'?
11. List all employees who have a salary greater than 15000 in the order of department number?

G. POST-LAB QUESTIONS

1. List all employees who have a salary greater than 15000 in the order of department number?
2. Find the name of employee which starts with 'A' and end with 'N'?
3. List deptno , dname ,min(sal) for all departments?
4. List all employees dept-wise and job –wise?
5. Display all employee names, number, deptname & location of all employees?
6. Find the employees belongs to the research department?
7. Find employee name employee number, their salary who were hired after 01/02/97
8. Find the second maximum salary of employee table?
9. Find employee name from employee table whose manager is nil?

CASE STUDY 2: SAILORS RESERVES THE BOATS

A. OBJECTIVE

To retrieve the rows from more than one table and sub query can useful to answer multiple part of questions for the given case study's database.

CASE STUDY: SAILORS RESERVING THE BOATS DATABASE

In Database user has to maintain sailors information with sailors identity., and every sailor age is more than 25 years and has a rating i.e (rating >=10),the sailors re reserved the boats for shipment of goods. Each boat identified by Id, name, color Every sailors may reserve more than one boat. Reservation can noticed based on the date.

B. RESOURCES

Oracle 10g Express edition

C. DESCRIPTION/PROGRAM LOGIC

The purpose of a join concept is to combine data spread across tables. A join is actually performed by the 'where' clause which combines specified rows of tables.

Types of Joins

1. Simple Join: select columns from table1, table2 where logical expression;
2. Self Join : Joining of a table to itself is known as self-join. It joins one row in a table to another. It can compare each row of the table to itself and also with other rows of the same table.
3. Outer Join: It extends the result of a simple join. An outer join returns all the rows returned by simple join as well as those rows from one table that do not match any row from the table. The symbol(+) represents outer join.

NESTED QUERIES:

The query within another is known as a subquery. A statement containing subquery is called parent statement. The rows returned by subquery are used by the parent statement

D. PROCEDURE

SIMPLE JOIN

Syntax:

select columns from table1, table2 where logical expression;

EQUI-JOIN:

A join, which is based on equalities, is called equi-join.

Example**:**

select * from item, cust where item.id=cust.id;

NON EQUI-JOIN :

It specifies the relationship between columns belonging to different tables by making use of relational operators other than'='.

Example**:**

select * from item, cust where item.id<cust.id;

OUTER JOIN:

Example:
select ename, job, dname from emp, dept where emp.deptno (+) = dept.deptno;

QUERY:
Find the sid's,names of sailors who have reserved boat number 103?

INPUT:
SQL> select s.sid,s.sname from sailors s,reserves r where s.sid=r.sid and r.bid=103;

OUTPUT :

SID	SNAME
22	DustiN
31	Lubber
74	Horatio

QUERY:
Find the sid's,names of sailors who have reserved a red color boat?

INPUT:
SQL> select distinct(s.sid),sname from sailors s,reserves r,boats b where s.sid=r.sid and r.bid=b.bi
d and b.color='Red';

OUTPUT:

SID	SNAME
22	Dustin
31	lubber
64	Horatio

QUERY:
Find the color of boats reserved by lubber?

INPUT:
SQL> select B.color from sailors s,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and s.sname=
'lubber';

OUTPUT:

COLOR
Red
Green
Red

QUERY:
Find the sid's,names of sailors who have reserved atleast one boat?

INPUT:

SQL> select distinct(s.sid),s.sname from sailors s,reserves r where s.sid=r.sid;

OUTPUT:

SID	SNAME
22	Dustin
31	lubber
64	Horatio
74	Horatio

QUERY:

Find the sid's,names of sailors who have reserved a Red or Green boat?

INPUT:

SQL> select distinct(s.sid),s.sname from sailors s,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and (b.color='Red' or b.color='Green');

OUTPUT:

SID	SNAME
22	Dustin
31	lubber
64	Horatio
74	Horatio

QUERY:

Find the sid's, names of sailors who have reserved a Red and Green boat?

INPUT:

SQL> select distinct(s.sname),s.sid from sailors s,reserves r1,boats b1,reserves r2,boats b2 where s.sid=r1.sid and r1.bid=b1.bid and s.sid=r2.sid and r2.bid=b2.bid and b1.color='Red' and b2.color='Green';

OUTPUT:

SNAME	SID
dustin	22
lubber	31

QUERY:

Find the sid's,names of sailors who have reserved all boats called "interlink"?

INPUT:

SQL> select s.sid,s.sname from sailors s ,reserves r, boats b where s.sid=r.sid and r.bid=b.bid and b.bname='Interlink';

OUTPUT:

SID	SNAME
22	Dustin
22	Dustin
31	lubber
64	Horatio
64	Horatio

QUERY:

Find all sailors with their rating above 7?

INPUT:

```
SQL> select * from sailors where rating >=7;
```

OUTPUT:

SID	SNAME	RATING	AGE
22	Dustin	7	45
31	lubber	8	55.5
32	Andy	8	25.5
58	Rusty	10	35
64	Horatio	7	35
71	Zorba	10	16
74	Horatio	9	35

7 rows selected.

QUERY:

Find the names and bid and reservation date for each reservation?

INPUT:

```
SQL> select s.sname,r.bid,r.day from sailors s ,reserves r where s.sid=r.sid;
```

OUTPUT:

SNAME	BID	DAY
Dustin	101	10-OCT-98
Dustin	102	10-OCT-98
Dustin	103	08-OCT-98
Dustin	104	07-OCT-98
lubber	102	10-NOV-98
lubber	103	06-NOV-98
lubber	104	12-OCT-98
Horatio	101	05-SEP-98
Horatio	102	08-SEP-98
Horatio	103	08-SEP-98

10 rows selected.

QUERY:

Find the sid's of sailors who have reserved a Red color boat?

INPUT:
SQL> select s.sid from sailors s ,reserves r,boats b where s.sid=r.sid and r.bid=b.bid and b.color='
red';
OUTPUT:
SID

22
22
31
31
64
QUERY:
Find the age of sailors whose names begins with "B" and atleast 3 characters?

INPUT:
SQL> select s.sname,s.age from sailors s where s.sname like 'b__%';

OUTPUT:
SNAME AGE
---
brustus 33
bob 63.5

QUERY:
Find the sid's of sailors who have reserved a Red boat but not a Green boat ?

INPUT:
SQL> select r.sid from reserves r,boats b where r.bid=b.bid and b.color='red' minus select r.sid fro
m reserves r,boats b where r.bid=b.bid and b.color='green';

OUTPUT:
SID

64

QUERY:
Find the sid's of sailors who have reserved a red and green boat but not a green boat ?

INPUT:
SQL> select distinct(s.sname),s.sid,b1.color from sailors s,reserves r1,boats b1,reserves r2,boats b
where s.sid=r1.sid and r1.bid=b1.bid and s.sid=r2.sid and r2.bid=b2.bid and b1.color='red' and b2.
color='green';

OUTPUT:
SNAME SID COLOR
---
lubber 31 red
dustin 22 red

QUERY:
Find the average age of all sailors?

INPUT:
SQL> select avg(age) from sailors;
AVG(AGE)

37.85

QUERY:
Find the average age of sailors with a rating of 10?
INPUT:
SQL> select rating,avg(age) from sailors where rating=10;

OUTPUT:
AVG(AGE)

35

QUERY:
Count the number of sailors?

INPUT:
SQL> select count(*) from sailors;

OUTPUT:
COUNT(*)

10

QUERY:
Count the number of different sailors name?

INPUT:
SQL> select count(distinct sname) from sailors;

OUTPUT:
COUNT(DISTINCTSNAME)

9

QUERY:
Find the average age of sailors for each rating level that has atleast two sailors?

INPUT:
SQL> select rating,avg(age) from sailors group by rating having count(*)>=2;

OUTPUT:

```
RATING AVG(AGE)
------ --------
     8     40.5
     7      40
     3     44.5
    10      35
```

NESTED QUERIES

QUERY:
Find the sid's,names of sailors who have reserved atleast two boats?

INPUT:

```
SQL> select s.sid,s.sname from sailors s where s.sid in (select sid from reserves group by sid having
count(sid)>=2);
```

OUTPUT:

```
SID SNAME
------ ---------
    22 Dustin
    31 lubber
    64 Horatio
```

QUERY:
Find the sid's,names of sailors with age over 20 who have not reserved a red boat?

INPUT:

```
SQL> select s.sid,s.sname from sailors s where s.age>20 and s.sid not in (select r.sid from reserves
 r,boats b where r.bid=b.bid and b.color='Red');
```

OUTPUT:

```
 SID SNAME
------ -----------
    29 Brutus
    32 Andy
    58 Rusty
    74 Horatio
    85 Art
    95 Bob

6 rows selected.
```

QUERY:
Find the sid's, names of sailors who have reserved all boats?

INPUT:
SQL> select s.sid,s.sname from sailors s where not exists((select b.bid from boats b) MINUS (select r.bid from reserves r where r.sid=s.sid));

OUTPUT:
SID SNAME
------ ------------
22 Dustin

QUERY:
Find the sid's of sailors who have reserved all boats?
INPUT:
SQL> select s.sid from sailors s where not exists(select b.bid from boats b where not exists(select r.sid from reserves r where r.bid=b.bid and r.sid=s.sid));
OUTPUT:
SID

22
QUERY:
Compute increment for the ratings of sailors who have sailed two different boats on the same day ?

INPUT:
select s.rating+1 from sailors s where s.sid in(select sid from reserves group by sid having count(sid)>=2);

OUTPUT:
RATING+1

8
9
8

QUERY:
Find the sid's of sailors who have a rating of 10 or who have reserved a boat 104?

INPUT:
SQL> select s.sid,s.rating from sailors s where s.rating>=10 or s.sid in(select sid from reserves wh ere bid=104);

OUTPUT:
SID RATING
---------- ----------
22 7
31 8
58 10
71 10

QUERY:
Find the names of sailors who have not reserved a red boat?

INPUT:
select distinct(s.sid),s.sname from sailors s,reserves r where s.sid not in(select r.sid from boats b,reserves r where r.bid=b.bid and b.color='red');

OUTPUT:

SID	SNAME
32	andy
58	rusty
29	brustus
74	horatio
85	art
95	bob
71	zorba

QUERY:
Find sailors whose rating is better than 'Hoartio'?

INPUT:
SQL> select * from sailors where rating > any(select rating from sailors where sname='horatio');

OUTPUT:

SID	SNAME	RATING	AGE
58	rusty	10	35
71	zorba	10	35
74	horatio	9	25.5
31	lubber	8	55.5
32	andy	8	25.5

QUERY:
Find sailors whose rating is better than every sailor called "Horatio"?
INPUT:
SQL> select * from sailors where rating > all(select rating from sailors where sname='horatio');

OUTPUT:

SID	SNAME	RATING	AGE
71	zorba	10	35
58	rusty	10	35

QUERY:
Find the sailors with highest rating?

INPUT:

```
SQL>  select s.sid,s.sname from sailors s where s.rating=(select max(rating) from sailors);
```

OUTPUT:

```
 SID SNAME
------ -----------
  58 rusty
  71 zorba
```

E. PRE-LAB QUESTIONS

1. What is the need of join?
2. What is the difference between left outer join and right outer join?
3. Why we go for nested queries?
4. What is the difference simple join and equi-join?

F. LAB ASSIGNMENT

1. Issue a query to find all the employees who work in the same job as jones
2. Issue a query to display information about employees who earn more than any employee in dept 30.
3. Display the employees who have the same job as jones and whose salary >= fords.
4. Write a query to display the name and job of all employees in dept 20 who have a job that someone in the Management dept as well as write a query that would display the empname, job where each employee works and the name of their dept
5. Write a query to list the employees having the same job as employees located in ' main block'.(use multiple sub query)

G. POST-LAB QUESTIONS

1. Write a query to list all depts. with at least 2 salesmen.
2. Write a query to list the employees with the same job and salary as 'ford'.
3. Write a query to list the employees in dept 20 with the same job as anyone in dept 30.
4. List out the employee names who get the salary greater than the maximum salaries of dept with dept no 20,30
5. Display the maximum salaries of the departments whose maximum salary is greater than 9000
6. Display the maximum salaries of the departments whose minimum salary is greater than 1000 and lesser than 5000.

WEEK 6
PRACTICE ON JOINS

TASK: To practice queries using joins in SQL

Consider below student and course table for queries on different types of joins.

Student

ROLL_NO	NAME	ADDRESS	PHONE	Age
1	HARSH	DELHI	XXXXXXXXXX	18
2	PRATIK	BIHAR	XXXXXXXXXX	19
3	RIYANKA	SILIGURI	XXXXXXXXXX	20
4	DEEP	RAMNAGAR	XXXXXXXXXX	18
5	SAPTARHI	KOLKATA	XXXXXXXXXX	19
6	DHANRAJ	BARABAJAR	XXXXXXXXXX	20
7	ROHIT	BALURGHAT	XXXXXXXXXX	18
8	NIRAJ	ALIPUR	XXXXXXXXXX	19

Course

COURSE_ID	ROLL_NO
1	1
2	2
2	3
3	4
1	5
4	9
5	10
4	11

INNER JOIN :

```
SELECT Course.COURSE_ID, Student.NAME, Student.AGE FROM Student
INNER JOIN Course
ON Student.ROLL_NO = Course.ROLL_NO;
```

OUTPUT :

COURSE_ID	NAME	Age
1	HARSH	18
2	PRATIK	19
2	RIYANKA	20
3	DEEP	18
1	SAPTARHI	19

LEFT JOIN :

```
SELECT Student.NAME,Course.COURSE_ID
FROM Student
LEFT JOIN Course
ON Course.ROLL_NO = Student.ROLL_NO;
```

OUTPUT :

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
DHANRAJ	*NULL*
ROHIT	*NULL*
NIRAJ	*NULL*

FULL JOIN :

```
SELECT Student.NAME,Course.COURSE_ID
FROM Student
RIGHT JOIN Course
ON Course.ROLL_NO = Student.ROLL_NO;
```

OUTPUT :

NAME	COURSE_ID
HARSH	1
PRATIK	2
RIYANKA	2
DEEP	3
SAPTARHI	1
NULL	4
NULL	5
NULL	4

PRE LAB QUESTIONS:

1. What is a join in SQL?
2. Why we use join in SQL?
3. What is left outer join?
4. What is Right outer join?
5. What is Full outer join?
6. What is Inner join?

POST LAB QUESTIONS:

1. Write syntax of Right outer join.
2. State different types of SQL joins.
3. Why we use Full outer join in SQL?
4. Write syntax of Left outer join.
5. If the user want to select all the rows in both the tables even with null values the which join is preferrable?

WEEK – 7

TASK : PRACTICE ON CO-RELATED SUB QUERIES AND NESTED QUERIES.

NESTED QUERIES:
The query within another is known as a subquery. A statement containing subquery is called parent statement. The rows returned by subquery are used by the parent statement.

NESTED QUERIES and CORELATED NESTED QUERIES

QUERY:
Find the sid's,names of sailors who have reserved atleast two boats?

INPUT:
```
SQL> select s.sid,s.sname from sailors s where s.sid in (select sid from reserves group by sid having
count(sid)>=2);
```

OUTPUT:
```
SID SNAME
------ ---------
   22 Dustin
   31 lubber
   64 Horatio
```

QUERY:
Find the sid's,names of sailors with age over 20 who have not reserved a red boat?

INPUT:
```
SQL> select s.sid,s.sname from sailors s where s.age>20 and s.sid not in (select r.sid from reserves
 r,boats b where r.bid=b.bid and b.color='Red');
```

OUTPUT:
```
 SID SNAME
------ -----------
   29 Brutus
   32 Andy
   58 Rusty
   74 Horatio
   85 Art
   95 Bob

6 rows selected.
```

QUERY:
Find the sid's, names of sailors who have reserved all boats?
INPUT:
```
SQL> select s.sid,s.sname from sailors s where not exists((select b.bid from boats b) MINUS (select
```

r.bid from reserves r where r.sid=s.sid));

OUTPUT:
SID SNAME
------ ------------
22 Dustin

QUERY:
Find the sid's of sailors who have reserved all boats?
INPUT:
SQL> select s.sid from sailors s where not exists(select b.bid from boats b where not exists(select r.sid from reserves r where r.bid=b.bid and r.sid=s.sid));
OUTPUT:
SID

22
QUERY:
Compute increment for the ratings of sailors who have sailed two different boats on the same day ?

INPUT:
select s.rating+1 from sailors s where s.sid in(select sid from reserves group by sid having count(sid)>=2);

OUTPUT:
RATING+1

8
9
8

QUERY:
Find the sid's of sailors who have a rating of 10 or who have reserved a boat 104?

INPUT:
SQL> select s.sid,s.rating from sailors s where s.rating>=10 or s.sid in(select sid from reserves where bid=104);

OUTPUT:
SID RATING
---------- ----------
22 7
31 8
58 10
71 10

QUERY:
Find the names of sailors who have not reserved a red boat?

INPUT:

```
select distinct(s.sid),s.sname from sailors s,reserves r where s.sid not in( select r.sid from
boats b,reserves r where r.bid=b.bid and b.color='red');
```

OUTPUT:

SID	SNAME
32	andy
58	rusty
29	brustus
74	horatio
85	art
95	bob
71	zorba

PRE LAB QUESTIONS:

1. What is meant by nested queries?
2. Why do we use nested queries?
3. What are Co-related nested queries?
4. What is meant by Co-related subquery?
5. What is co-relation in nested queries?

POST LAB QUESTIONS :

1. What is the difference between co-related query and subquery?
2. Where does the sub query appear with in SQL query?
3. Which is faster subquery or co-related subquery?
4. What is the difference between co-related subquery and nested query?
5. What are the keywords used in co-related nested queries?

WEEK 8 - PL/SQL PROGRAMS

8.1)Aim: Write a PL/SQL program to read number from a user and find out whether it is Odd or Even.

PROGRAM:

```
declare
        n number:=&n;

begin
        if mod(n,2)=0
        then
            dbms_output.put_line('number is even');
        else
            dbms_output.put_line('number is odd');
        end if;
end;
/
```

Output:

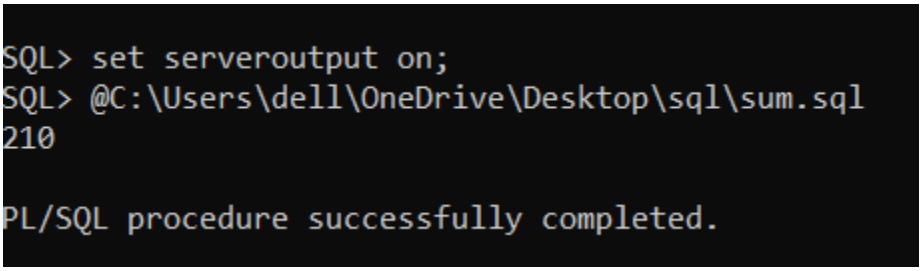
```
SQL> @E:\r18dbmsstarts\plsql\evenodd.sql
Enter value for n: 4
old   2:        n number:=&n;
new   2:        n number:=4;
number is even

PL/SQL procedure successfully completed.
```

8.2) Write a PL/SQL program to find sum of numbers from 1 to 20 using for or while loop.

PROGRAM :

```
declare
                n integer := 20;
                s integer :=0;
begin
                for i in 1..n loop
                        s  : = s + i;
                end loop;
                dbms_output.put_line(s);
end;
/
```

OUTPUT :



8.3) Write a PL/SQL program to find biggest of 3 numbers.

PROGRAM :

```
declare
        a integer;
        b integer;
        c integer;
begin
        a := &a;
        b := &b;
        c := &c;
        if((a>b) and (a>c)) then
                dbms_output.put_line('BIG IS  '||a);
        elsif((b>a) and (b>c)) then
                dbms_output.put_line('BIG IS  '||b);
        else
                dbms_output.put_line('BIG IS   '||c);
        end if;
end;
/
```

OUTPUT :

```
SQL> set serveroutput on;
SQL> @C:\Users\dell\OneDrive\Desktop\sql\big_among_3.sql
Enter value for a: 58
old   6:        a := &a;
new   6:        a := 58;
Enter value for b: 96
old   7:        b := &b;
new   7:        b := 96;
Enter value for c: 88
old   8:        c := &c;
new   8:        c := 88;
BIG IS   96
```

PRE LAB QUESTIONS:

1) What are conditional statements in PLSQL?

2) What are looping statements/blocks in PLSQL?

3) What are the differences between for loop and while loop in PLSQL?

4) Write the syntaxes of conditional statements in PLSQL?

5) Write the syntax of looping statements in PLSQL?

POST LAB QUESTIONS:

1) What is the structure of the PLSQL program?

2) When is a “declare” statement required in the PLSQL program?

3) What are the data types available in PLSQL?

4) How to read input dynamically in PLSQL program?

5) Write steps to follow while using PLSQL while loop.

WEEK – 9: CURSORS

9.1) write a cursor program to display all employee details.

PROGRAM :

```
declare
                eno  emp11.empno%type;
                name  emp11.ename%type;
                salary  emp11.sal%type;
                cursor display is select * from emp11;
begin
                open display;
                loop
                fetch display into eno,name,salary;
                        EXIT when(display%notfound);
                        dbms_output.put_line(eno||name||salary);
                end loop;
                close display;
end;
/
```

OUTPUT :

```
SQL> select * from emp11;

     EMPNO ENAME                          SAL
---------- -------------------- ----------
       501 Virat                       10000
       502 Rohit                       10000
       503 Dhawan                      10000

SQL> _
```

```
SQL> @C:\Users\dell\OneDrive\Desktop\sql\records_emp.sql
501  Virat  70000
502  Rohit  50000
503  Dhawan  40000

PL/SQL procedure successfully completed.
```

9.2) Write a PL/SQL program to get the name and salary of employee whose eno is 501.(use %type)

DESCRIPTION:

Data Selection

Data projection/fetching means to retrieve the required data from the database table. This can be achieved by using the command 'SELECT' with 'INTO' clause. The 'SELECT' command will fetch the values from the database, and 'INTO' clause will assign these values to the local variable of the PL/SQL block.

Below are the points that need to be considered in 'SELECT' statement.

- 'SELECT' statement should return only one record while using 'INTO' clause as one variable can hold only one value. If the 'SELECT' statement returns more than one value than 'TOO_MANY_ROWS' exception will be raised.
- 'SELECT' statement will assign the value to the variable in the 'INTO' clause, so it needs to get at least one record from the table to populate the value. If it didn't get any record, then the exception 'NO_DATA_FOUND' is raised.
- The number of columns and their datatype in 'SELECT' clause should match with the number of variables and their datatypes in the 'INTO' clause.
- The values are fetched and populated in the same order as mentioned in the statement.
- 'WHERE' clause is optional that allows to having more restriction on the records that are going to be fetched.
- 'SELECT' statement can be used in the 'WHERE' condition of other DML statements to define the values of the conditions.
- The 'SELECT' statement when using 'INSERT', 'UPDATE', 'DELETE' statements should not have 'INTO' clause as it will not populate any variable in these cases.

Syntax:

```
BEGIN
  SELECT <columnl>,..<column_n> INTO <vanable 1 >,. .<variable_n>
   FROM <table_name>
   WHERE <condition to fetch the required records>;
END;
```

- The above syntax shows the SELECT-INTO command. The keyword 'FROM' is mandatory that identifies the table name from which the data needs to be fetched.
- 'WHERE' clause is optional. If this clause is not given, then the data from the entire table will be fetched.

PROGRAM:

```
declare
eno   emp.empno%type;
ename    emp.empname%type;
sal    emp.salary%type;

begin
                    select empname,salary into ename,sal from emp where empno=501;
                    Dbms_output.put_line('Employe name is:'||ename);
                    Dbms_output.put_line('Employe salary is:'||sal);

end;
/
```

Output:

```
SQL> @E:\r18dbmsstarts\plsql\fetch.sql
Employe name is:federer
Employe salary is:100000

PL/SQL procedure successfully completed.
```

9.3) Write a cursor program to display no.of rows updated after executing update command for changing salary of employees to 10,000.

PROGRAM :

```
declare
        count1 int;
begin
        update emp11 set sal=10000;
        count1 := SQL%Rowcount;
        dbms_output.put_line(count1);
end;
/
```

OUTPUT :

```
SQL> @C:\Users\dell\OneDrive\Desktop\sql\count_emp.sql
3

PL/SQL procedure successfully completed.
```

PRE LAB QUESTIONS:

1) What are the cursors in PSQL?

2) Write the syntax of cursor creation?

3) What are the types of cursors available?

4) Explain explicit cursors?

5) Explain implicit cursors?

POST LAB QUESTIONS:

1) What are the types of cursors available?

2) What are the attributes of the cursors?

3) Write steps to follow while working with an explicit cursor?

4) Difference between implicit and explicit cursors?

5) What is the keyword used for fetching information from a cursor?

WEEK – 10 (TRIGGERS)

10.1) write a trigger program to display the salary of an employee before updation and after updation and also display the salary diff.

PROGRAM :

```
create or replace trigger up
after update on employ
for each row
declare
                diff int;
begin
                diff := :new.salary - :old.salary;
                dbms_output.put_line('old salary '||:old.salary);
                dbms_output.put_line('new salary '||:new.salary);
                dbms_output.put_line('diff is '|| diff);
end;
/
```

OUTPUT :

```
SQL> select * from employ;

       ENO ENAME                    SALARY     ANNUAL DOJ
---------- -------------------- ---------- ---------- ---------
       501 sara                      20000     120000 18-JUN-22

SQL> set serveroutput on;
SQL> @C:\Users\dell\OneDrive\Desktop\sql\diff_of_sal.sql

Trigger created.

SQL> update employ set salary=10000;
old salary 20000
new salary 10000
diff is -10000

1 row updated.
```

10.2) Write a program to create trigger for inserting record to back_up table before deletion.

PROGRAM :

```
create or replace trigger del
before delete on employ
for each row
declare
begin
                insert into backup_emp values(:old.eno,:old.ename,:old.salary,:old.annual,:old.doj);
end;
/
```

OUTPUT :

```
SQL> @C:\Users\dell\OneDrive\Desktop\sql\backup.sql

Trigger created.

SQL> delete from employ where eno=501;

1 row deleted.

SQL> select * from backup_emp;

        NO NAME                          SAL       ASAL D_O_J
---------- -------------------- ---------- ---------- ---------
       501 sara                      10000     120000 18-JUN-22

SQL> select * from employ;

no rows selected
```

PRE LAB QUESTIONS :

1. What is trigger?
2. Why do we use triggers in PL/SQL?
3. Write syntax for creating a trigger.
4. When a trigger is fired?
5. What are the 3 types of SQL triggers?
6. What are the different types of triggers?

POST LAB QUESTIONS:

1. Can we commit inside a trigger?
2. What are the benefits of using triggers in PL/SQL?
3. Can anyone commit or rollback within a trigger?
4. What is disadvantage of using triggers?
5. What happens if Trigger fails to create?
6. DML Trigger can be disabled or enabled using which command.
7. What happens if after UPDATE trigger fails?
8. Which command removes trigger?

WEEK – 11

11.a) Write a PL/SQL program to create stored procedure for finding biggest of 3 numbers.

PROGRAM :

```
declare
        a integer;
        b integer;
        c integer;
        big integer;
        procedure big1(a1 in integer,b1 in integer,c1 in integer,bigger out integer)
        is
        begin
            if((a>b) and(a>c)) then
                bigger := a;
            elsif((b>a) and (b>c)) then
                bigger := b;
            else
                bigger := c;
            end if;
        end big1;
begin
        a := &a;
        b := &b;
        c := &c;
        big1(a,b,c,big);
        dbms_output.put_line(big);
end;
/
```

OUTPUT :

```
SQL> @C:\Users\dell\OneDrive\Desktop\sql\proce_big.sql
Enter value for a: 59
old  18:        a := &a;
new  18:        a := 59;
Enter value for b: 45
old  19:        b := &b;
new  19:        b := 45;
Enter value for c: 99
old  20:        c := &c;
new  20:        c := 99;
99

PL/SQL procedure successfully completed.
```

11.b) Write a PL/SQL program using functions to find smallest of 2 numbers and also factorial of a given number.

PROGRAM :

```
declare
        a integer;
        b integer;
        res integer;
        res_fact integer;
        function smallest(a1 in integer,b1 in integer)
        return int
        is
                s integer;
        begin
                if(a1<b1) then
                        s := a1;
                else
                        s := b1;
                end if;
                return s;
        end smallest;
        function factorial(a1 in integer)
        return int
        is
                fact integer;
        begin
                if(a1=1) then
                        fact :=1;
                else
                        fact := a1*factorial(a1-1);
                end if;
                return fact;
        end factorial;
begin
        a := &a;
        b := &b;
        res := smallest(a,b);
        res_fact := factorial(a);
        dbms_output.put_line(res);
        dbms_output.put_line(res_fact);
end;
/
```

OUTPUT :

```
SQL> @C:\Users\dell\OneDrive\Desktop\sql\s_f.sql
Enter value for a: 10
old  31:        a := &a;
new  31:        a := 10;
Enter value for b: 20
old  32:        b := &b;
new  32:        b := 20;
10
3628800

PL/SQL procedure successfully completed.
```

PRE LAB QUESTIONS :

1. What is a Procedure?
2. How a Procedure is executed?
3. Write syntax of using a Procedure.
4. What is a function?
5. What are different types of functions used?
6. Write syntax of using a Function.

POST LAB QUESTIONS:

1. What is the difference between Procedure and Function in SQL?
2. Which is better Function or Stored Procedure?
3. How many types of functions are there in SQL?
4. Can we write a function inside a procedure?
5. What is the difference between Stored Procedure and Trigger?
6. How a procedure can be called and how it is executed?

WEEK – 12

12) Write a python program to connect to oracle database and perform basic operations like creating table , insertion of rows , displaying all rows , updating any one row.

PROGRAM :

```
# importing module
import cx_Oracle

# Create a table in Oracle database
try:

            con = cx_Oracle.connect('tiger/scott@localhost:1521/xe')
            print(con.version)

            # Now execute the sqlquery
            cursor = con.cursor()

            # Creating a table employee
            cursor.execute("create table employee(empid integer primary key, name varchar2(30), salary
          number(10, 2))")

            print("Table Created successfully")
            data = [[10007, 'Vikram', 48000.0], [10008, 'Sunil', 65000.1], [10009, 'Sameer', 75000.0]]

            cur = con.cursor()
            cur.executemany('insert into employee values(:1,:2,:3)', data)
           con.commit()
            print('Multiple records are inserted successfully')
            cur.execute('select * from employee')
            rows = cur.fetchall()
            print(rows)
            cur.execute('update employee set salary=20000 where empno=501')
            print('one record updated successfully')
except cx_Oracle.DatabaseError as e:
            print("There is a problem with Oracle", e)

# by writing finally if any error occurs
# then also we can close the all database operation
finally:
            if cursor:
                  cursor.close()
            if con:
                  con.close()
```

OUTPUT :

```
C:\Users\dell\OneDrive\Desktop\python programs>test.py
11.2.0.2.0
Table Created successfully
Multiple records are inserted successfully
[(10007, 'Vikram', 48000.0), (10008, 'Sunil', 65000.1), (10009, 'Sameer', 75000.0)]
one record updated successfully
```

PRE LAB QUESTIONS:

1. Which module is used to connect to Oracle database using python?
2. What is python pip used for?
3. What is the full form of pip in python?
4. Which method in python is used to establish a connection with Oracle database?
5. Which method in python is used to close a connection?
6. Write syntax for exception in python.

POST LAB QUESTIONS :

1. Why cursor is used in above program?
2. Which method is used to execute SQL query in python?
3. Which method is used to insert multiple values in Oracle database table?
4. Which method is used to retrieve all the records of a table?
5. Write syntax for establishing a connection with Oracle database using python.
6. What are the different types of fetch statements?
7. Do DDL statements required to be committed . If yes, why?

WEEK – 13

TASK: Case Study in developing a database following all steps in the design of databases elaborating normalization and denormalization.

Description:
Consider any project with database of minimum 3 tables along with key constraints and do the following:

- Mention any 10 requirements by the user for particular database requirements.
- Draw the ER diagram based on the Requirements.
- Convert the ER diagram into tables.
- Justify which NF is satisfied by the database tables.
- Normalize the tables up to 3NF.

Normalization and De-normalization of tables:

Normalization and **Denormalization** are the processes which are being used to optimize the performance of the database. The difference between them is explained with the help of examples.

Normalization:

Normalization is a process which is carried out to minimize/remove the anomalies in order to maintain the database's consistency. Example of normalization is given below: –

Company ID	Company Name	Product ID	Product Name	Product Quantity
1	Iqra Enterprises	101	Beauty Soap	120
		102	Tooth Brush	100
2	Shahid Enterprises	103	Polish	140
3	Khalid Enterprises	103	Polish	250

Normalization is carried out in following forms:

- First Normal Form (1NF)
- Second Normal Form(2NF)
- Third Normal Form(3NF)
- Boyce & Codd Normal Form(BCNF)

First Normal Form:

In first normal form, the duplicate columns are removed.

Company ID	Company Name	Product ID	Product Name	Product Quantity
1	Iqra Enterprises	101	Beauty Soap	120
1	Iqra Enterprises	102	Tooth Brush	100
2	Shahid Enterprises	103	Polish	140
3	Khalid Enterprises	103	Polish	250

Second Normal Form:

In case of second normal form, it contains step of first normal form in addition to removal of duplicate data which is placed in a child table.
Company Table:

Company ID	Company Name
1	Iqra Enterprises
1	Iqra Enterprises
2	Shahid Enterprises
3	Khalid Enterprises

Product Table:

Product ID	Product Name	Product Quantity
101	Beauty Soap	120
102	Tooth Brush	100
103	Polish	140
103	Polish	250

Company – Product Table

Company ID	Product ID
1	101
1	102
2	103
3	103

Third Normal Form:

The third normal form include 2nd normal form and further steps are carried out. In this form the columns are removed which are not dependent on primary key columns.
Company Table:

Company ID	Company Name
1	Iqra Enterprises
2	Shahid Enterprises

3	Khalid Enterprises

Product Table:

Product ID	Product Name	Product Quantity
101	Beauty Soap	120
102	Tooth Brush	100
103	Polish	140
103	Polish	250

Company – Product Table

Company ID	Product ID
1	101
1	102
2	103
3	103

Denormalization:

De-normalization is a reverse process of Normalization. It reduces the work load and optimizes the performance. Example of de-normalization is given below: –
Employee Table:

Employee ID	Employee Name
1	Shahid
2	Rizwan
3	Khalid

Course Table:

Employee ID	Course Qualified
1	MS Office
2	Web Designing
3	C++ programming

After De-normalization,

Employee ID	Employee Name	Course Qualified
1	Shahid	MS Office
2	Rizwan	Web Designing
3	Khalid	C++ programming

PRE LAB QUESTIONS:

1. What are the different types of key constraints?
2. How do we convert ER diagram into Relational model?
3. What is primary key and foreign key?
4. What is Normalization?
5. How many types of NF exist?
6. What is de-normalization?
7. What is transitive dependency?
8. What is BCNF?

POST LAB QUESTIONS:

1. State 2NF.
2. Why do we use normal forms?
3. What is Domain key normal form?
4. Explain different types of functional dependencies.
5. What is the difference between candidate and super key and define prime and non-prime attributes?
6. What are the different types of Decomposition?
7. State 4NF and 5NF.

VIVA QUESTIONS

1 Define DBMS?

The collection of programs that enables user to create and maintain a database. In other words it is general-purpose software that provides the users with the processes of defining, constructing and manipulating the database for various applications

2 What are the advantages of DBMS?

Redundancy is controlled.
Un authorized access is restricted.
Providing multiple user interfaces.
Enforcing integrity constraints.
Enforcing integrity constraints.
Providing backup and recovery

3 Describe the three levels of data abstraction?

The are three levels of abstraction:
Physical level: The lowest level of abstraction describes how data are stored.
Logical level: The next higher level of abstraction, describes what data are stored in database and what relationship among those data.
View level: The highest level of abstraction describes only part of entire database.

4 Define the "integrity rules"?

There are two Integrity rules.
Entity Integrity: States that "Primary key cannot have NULL value"
Referential Integrity: States that "Foreign Key can be either a NULL value or should be Primary Key value of other relation.

5 What is Data Independence?

Data independence means that "the application is independent of the storage structure and access strategy of data". In other words, The ability to modify the schema definition in one level should not affect the schema definition in the next higher level.
Two types of Data Independence:
Physical Data Independence: Modification in physical level should not affect the logical level.
Logical Data Independence: Modification in logical level should affect the view level.

6 What is a view? How it is related to data independence?

A view may be thought of as a virtual table, that is, a table that does not really exist in its own right but is instead derived from one or more underlying base table. In other words, there is no stored file that direct represents the view instead a definition of view is stored in data dictionary

7 What is E-R model?

This data model is based on real world that consists of basic objects called entities and of relationship among these objects. Entities are described in a database by a set of attributes.

8 What is Object Oriented model?

This model is based on collection of objects. An object contains values stored in instance variables with in the object. An object also contains bodies of code that operate on the object. These bodies of code are called methods. Objects that contain same types of values and the same methods are grouped together into classes.

9 What is Weak Entity set?

An entity set may not have sufficient attributes to form a primary key, and its primary key compromises of its partial key and primary key of its parent entity, then it is said to be Weak Entity set.

10 What is an attribute?

It is a particular property, which describes the entity.

11 What is degree of a Relation?

It is the number of attribute of its relation schema.

12 What is normalization?

It is a process of analyzing the given relation schemas based on their Functional Dependencies (FDs) and primary key to achieve the properties
(1).Minimizing redundancy, (2). Minimizing insertion, deletion and update anomalies.

13 What is Functional Dependency?

A Functional dependency is denoted by X Y between two sets of attributes X and Y that are subsets of R specifies a constraint on the possible tuple that can form a relation state r of R. The constraint is for any two tuples t1 and t2 in r if t1[X] = t2[X] then they have t1[Y] = t2[Y]. This means the value of X component of a tuple uniquely determines the value of component Y.

14 What is Fully Functional dependency?

It is based on concept of full functional dependency. A functional dependency X Y is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

15 What are the unary operations in Relational Algebra?

PROJECTION and SELECTION.

16 Define SQL and state the differences between SQL and other conventional programming Languages.?

SQL is a nonprocedural language that is designed specifically for data access operations on normalized relational database structures. The primary difference between SQL and other conventional programming languages is that SQL statements specify what data operations should be performed rather than how to perform them

17 What is database Trigger?

A database trigger is a PL/SQL block that can defined to automatically execute for insert, update, and delete statements against a table. The trigger can e defined to execute once for the entire statement or once for every row that is inserted, updated, or deleted. For any one table, there are twelve events for which you can define database triggers. A database trigger can call database procedures that are also written in PL/SQL.

18 What are stored-procedures? And what are the advantages of using them?

Stored procedures are database objects that perform a user defined operation. A stored procedure can have a set of compound SQL statements. A stored procedure executes the SQL commands and returns the result to the client. Stored procedures are used to reduce network traffic.

19 What is Storage Manager?

It is a program module that provides the interface between the low-level data stored in database, application programs and queries submitted to the system.

20 What is Fully Functional dependency?

A functional dependency X Y is full functional dependency if removal of any attribute A from X means that the dependency does not hold any more.

21 What is 2NF?

A relation schema R is in 2NF if it is in 1NF and every non-prime attribute A in R is fully functionally dependent on primary key

22 What is 3NF?

A relation is in third normal form if it is in Second Normal Form and there are no functional (transitive) dependencies between two (or more) non-primary key attributes.

23 What is BCNF (Boyce-Codd Normal Form)?

A table is in Boyce-Codd normal form (BCNF) if and only if it is in 3NF and every determinant is a candidate key.

24 What is an order By Clause?

ORDER BY clause helps to sort the data in either ascending order to descending order.
Ascending order sort query
SELECT name,age FROM pcdsEmployee ORDER BY age ASC
Descending order sort query

25 What is Relationship set?

The collection (or set) of similar relationships.

26 What is Relationship type?

Relationship type defines a set of associations or a relationship set among a given set of entity types.

27 What is degree of Relationship type?

It is the number of entity type participating.

28 What is DML Compiler?

It translates DML statements in a query language into low-level instruction that the query evaluation engine can understand

29 What is Query evaluation engine?

It executes low-level instruction generated by compiler.

30 What is DDL Interpreter?

It interprets DDL statements and record them in tables containing metadata.

31 What are the different phases of transaction?
Different phases are
1.) Analysis phase,
2.) Redo Phase,
3.) Undo phase.

32 What is a checkpoint and When does it occur?
A Checkpoint is like a snapshot of the DBMS state. By taking checkpoints, the DBMS can reduce the amount of work to be done during restart in the event of subsequent crashes

33 What is Buffer Manager?
It is a program module, which is responsible for fetching data from disk storage into main memory and deciding what data to be cache in memory

34 What is Transaction Manager?
It is a program module,
which ensures that database, remains in a consistent state despite
system failures and concurrent transaction execution proceeds without conflicting.

35 What is File Manager?
It is a program module, which manages the allocation of space on disk storage an d data structure used to represent information stored on a disk

36 What are cursors give different types of cursors?
PL/SQL uses cursors for all database information accesses statements. The language supports the use two types of cursors
1.) Implicit
2.) Explicit

37 **What is the SQL " IN " clause?**
SQL IN operator is used to see if the value exists in a group of values. For instance the below SQL checks if the Name is either 'rohit' or 'Anuradha'
SELECT * FROM pcds Employee WHERE name IN ('Rohit','Anuradha')
Also you can specify a not clause with the same. SELECT * FROM pcdsEmployee WHERE age NOT IN (17,16)

38 **What's the difference between "UNION" and "UNION ALL" ?**
UNION SQL syntax is used to select information from two tables. But it selects only distinct records from both the table. , while UNION ALL selects all records from both the tables.

39 **What is a self-join?**
If we want to join two instances of the same table we can use self-join.

40 **What is the default "-SORT " order for a SQL?**
ASCENDING

41 **What are cursors and what are the situations you will use them?**

SQL statements are good for set at a time operation. So it is good at handling set of data. But there are scenarios where we want to update row depending on certain criteria. we will loop through all rows and update data accordingly. There's where cursors come in to picture.

42 What is " Group by " clause?

"Group by" clause group similar data so that aggregate values can be derived.

43 What is the difference between "HAVING" and "WHERE" clause?

HAVING" clause is used to specify filtering criteria for "GROUP BY", while "WHERE" clause applies on normal SQL.

44 What is a View?

View is a virtual table which is created on the basis of the result set returned by the select statement.

CREATE VIEW [MyView] AS SELECT * from pcdsEmployee where LastName = 'singh'

In order to query the view

SELECT * FROM [MyView]

WORKSHEET

WORKSHEET

WORKSHEET

WORKSHEET

WORKSHEET