DATABASE TECHNOLOGY LABORATORY

(V semester of B.Tech)

As per the curriculam and syllabus

Of

Bharath Institute of Higher Education & Research

PREPARED BY

Ms.D.Sharmila

NEW EDITION

Department of information technology

Bharath

INSTITUTE OF HIGHER EDUCATION AND RESEARCH

(Declared as Deemed - to - be - University under section 3 of UGC Act 1956)

ACCREDITED WITH 'A' GRADE BY NAAC

DATABASE TECHNOLOGY

(V semester of B.Tech)

As per the curricullam and syllabus

Of

Bharath Institute of Higher Education & Research

PREPARED BY

Ms.D.Sharmila

Department of information technology

SCHOOL OF COMPUTING

DEPARTMENT OF INFORMATION TECHNOLOGY

LAB MANUAL

SUBJECT NAME: DATABASE TECHNOLOGY LABORATORY

SUBJECT CODE: U20ITCJ05

Regulation - 2020

VISION AND MISSION OF THE INSTITUTE

VISION

"Bharath Institute of Higher Education & Research (BIHER) envisions and constantly strives to provide an excellent academic and research ambience for students and members of the faculties to inherit professional competence along with human dignity and transformation of community to keep pace with the global challenges so as to achieve holistic development."

MISSION

- To develop as a Premier University for Teaching, Learning, Research and Innovation on par with leading global universities.
- To impart education and training to students for creating a better society with ethics and morals.
- To foster an interdisciplinary approach in education, research and innovation by supporting lifelong professional development, enriching knowledge banks through scientific research, promoting best practices and innovation, industry driven and institute-oriented cooperation, globalization and international initiatives.
- To develop as a multi-dimensional institution contributing immensely to the cause of societal advancement through spread of literacy, an ambience that provides the best of international exposures, provide health care, enrich rural development and most importantly impart value-based education.
- To establish benchmark standards in professional practice in the fields of innovative and emerging areas in engineering, management, medicine, dentistry, nursing, physiotherapy and allied sciences.
- To imbibe human dignity and values through personality development and social service activities.

VISION AND MISSION OF THE DEPARTMENT

VISION

To be an excellence in education and research in Information Technology producing global scholars for improvement of the society

MISSION

- To provide sound fundamentals, and advances in Information Technology, Software Engineering, data Communications and Computer Applications by offering world class curriculum.
- To create ethically strong leaders and expert for next generation IT.
- To nurture the desire among faculty and students from across the globe to perform outstanding and impactful research for the benefit of humanity and, to achieve meritorious and significant growth.

PROGRAM EDUCATIONAL OBJECTIVES (PEO)

The Program Educational Objectives (PEOs) of Information technology are listed below: The graduate after 3-5 years of programme completion will

PEO1: PREPARATION
To provide students with sound fundamental in Mathematical, Scientific and Engineering fundamentals necessary to formulate, analyse, and comprehend the fundamental concepts essential to articulate, solve and assess engineering problems and to prepare them for research & development and higher learning.

PEO2: CORE COMPETENCE
To apply critical reasoning, quantitative, qualitative, designing and programming skills, to identify, solve problems and to analyze the experimental evaluations, and finally making appropriate decisions along with knowledge of

computing principles and applications and be able to integrate this knowledge in a variety of industry and inter-disciplinary setting.

PEO3: PROFESSIONALISM
To broaden knowledge to establish themselves as creative practicing professionals, locally and globally, in fields such as design, development, problem solving to production support in software industries and R&D sectors.

PEO4: SKILL
To provide better opportunity to become a future researchers / scientist with good communication skills so that they may be both good team-members and leaders with innovative ideas for a sustainable development.

PEO5: ETHICS
To be ethically and socially responsible solution providers and entrepreneurs in Computer Science and other engineering discipline.

PROGRAMME OUTCOMES

PO 1	**gineering Knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
PO2	**oblem Analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences and engineering sciences.
PO 3	**sign/Development of Solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
PO 4	**nduct Investigations of Complex Problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions for complex problems.
PO 5	**odern Tool Usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
PO 6	**e Engineer and Society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
PO 7	**vironment and Sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
PO 8	**hics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
PO 9	**dividual and Team Work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
PO 10	**mmunication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
PO 11	**oject Management and Finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
PO 12	**fe-long Learning:** Recognize the need for, and have the preparation and ability to engage in independent and lifelong learning in the broadest context of technological change.

PROGRAMME SPECIFIC OUTCOME

PSO 1	**Programming Design :** Design and develop algorithm for real life problems using latest technologies and solve it by using computer programming languages and database technologies .
PSO 2	**IT Business Scalable Design :** Analyze and recommend computing infrastructures and operations requirements and Simulate and implement information networks using configurations, algorithms, suitable protocol and security for valid and optimal connectivity.
PSO 3	**Intelligent Agents Design :** Design and execute projects for the development of data modeling, data analytics and knowledge representation in various domain.

U20ITCJ05- DATABASE TECHNOLOGY

PART-A INTRODUCTION OF THE COURSE

Databases form the backbone of all major applications today – tightly or loosely coupled, intranet or internet based, financial, social, administrative, and so on. Structured Database Management Systems (DBMS) based on relational and other models have long formed the basis for such databases. Consequently, Oracle, Microsoft SQL Server, Sybase etc. have emerged as leading commercial systems while MySQL, PostgreSQL etc. lead in open source and free domain.

Course Code	Course Category	Course Title	L 3	T 0	P 2	C 4
U20ITCJ05	C	DATABASE TECHNOLOGY	Pre- requisite: U20CSEJ01 or Diploma			
Name Of the Course Coordinator:		Ms.D.Sharmila	Contact Hrs: 75			
Course Offering Department/School:		Department of IT	Total Marks :100			

Course Objective and Summary

- To understand the concept of DBMS and ER Modeling.
- To explain the normalization, Query optimization and relational algebra.
- To apply the concurrency control, recovery, security and indexing for the real time data.

Course Outcomes (COs)

Course Outcomes (COs)		BT Level
CO1	Explain the basic concept and role of DBMS in an organization.	2
CO2	Illustrate the design principles for database design, ER model and normalization.	2
CO3	Demonstrate the basics of query evaluation and heuristic query optimization techniques.	2
CO4	Apply Concurrency control and recovery mechanisms for the desirable database problem.	3
CO5	Compare the basic database storage structure and access techniques	4
CO6	Implement the database system with the fundamental concepts of DBMS.	B

Mapping / Alignment of COs with PO & PSO

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO1	3												3		
CO2	3	2											3		
CO3	3	2		2									3		
CO4		2	3										3		
CO5		3	2	2									3		
CO6		2	3		2								3		
(Tick mark or level of correlation: 3-High, 2-Medium, 1-Low)															

PART-B CONTENT OF THE COURSE

1.COURSE CONTENT

UNIT I Data Models **(9)**

History and motivation for database systems - characteristics of database approach - Data Definition Commands, Data Manipulation Commands for inserting, deleting, updating and retrieving Tables and Transaction Control statements - Workers behind the scene Advantages of using DBMS approach - Database Querying – Simple queries, Nested queries, Sub queries and Joins - Data Models, Schemas, and Instances - Three-Schema Architecture and Data Independence - Views, Sequences, Synonyms.

UNIT II Relational Database Design And Query Languages **(9)**

The Database System Environment - Centralized and Client/Server Architectures for DBMSs - Centralized and Client/Server Architectures for DBMSs - Database Programming: Implicit and Explicit Cursors - Classification of database management systems - Entity Relationship Model: Types of Attributes, Relationship, Structural Constraints - Relational Model, Relational model Constraints - Mapping ER model to a relational schema - Database Triggers.

UNIT III Query Processing And Optimization **(9)**

Integrity constraints - Guidelines for Relational Schema - Exception Handling - Functional dependency; Normalization - Boyce Code Normal Form - Multi-valued dependency and Fourth Normal form - Join dependency and Fifth Normal form - Database Connectivity with Front End Tools.

UNIT IV Transaction Processing **(9)**

Translating SQL Queries into Relational Algebra - heuristic query optimization - Introduction to Transaction Processing - Transaction and System concepts - Desirable properties of Transactions - Characterizing schedules based on recoverability and serializability - Two-Phase Locking Techniques for Concurrency Control - Concurrency Control based on timestamp.

UNIT V Database Security **(9)**

Recovery Concepts - Recovery based on deferred update - Recovery techniques based on immediate update - Shadow Paging - Indexing: Single level indexing - multi-level indexing - dynamic multilevel Indexing - Need of NoSQL. **45 HOURS**

2.LIST OF EXPERIMENTS **30 Hours**

1. Data Definition Commands, Data Manipulation Commands for inserting, deleting, updating and retrieving Tables and Transaction Control statements.
2. Database Querying – Simple queries, Nested queries, Sub queries and Joins.
3. Views, Sequences, Synonyms.
4. Database Programming: Implicit and Explicit Cursors.
5. Procedures and Functions.
6. Database Triggers.
7. Exception Handling.
8. Database Design using ER modeling, normalization and Implementation for any application.
9. Database Connectivity with Front End Tools.
10. Case Study using real life database applications.

STUDY ON RELATIONAL DATABASE MANAGEMENT SYSTEM (RDBMS)

INTRODUCTION TO SQL:

SQL (structured query language) is a database computer language designed for the retrieval and management of data in relational database systems (RDBMS), database scheme creation and modification, and database object access control management.

FEATURES OF SQL

SQL is both an easy –to-understand language and a comprehensive tool for managing data.

- High-level. English-like structure
- Interactive, ad hoc queries
- Programmatic database access
- Multiple views of data
- Complete database language
- Dynamic data definition
- Client/server architecture

TYPES OF SQL COMMANDS

SQL is used to control all of the functions that a DBMS provides for its users, including:

Data definition, SQL lets a user define the structure and organization of the stored data and relationship among the stored data items.

Data retrieval, SQL allows a user or an application program to retrieve stored data from the database and use it.

Data manipulation, SQL allows a user or an application program to update the database by adding new data, removing old data, and modifying previously stored data.

Access control, SQL can be used to restrict a user’s ability to retrieve, add, and modify data, protecting stored data against unauthorized access.

Data sharing, SQL is used to coordinate data sharing by concurrent users, ensuring that they do not interface with one another.

Data integrity, SQL defines integrity constraints in the database, protecting it from corruption due to inconsistent updates or system failures.

EX.NO:1.A

DATE:

DATA DEFINITION LANGUAGES (DDL) COMMANDS IN RDBMS

A Data Definition Language (DDL) statement are used to define the database structure or schema.

Aim:

To study and execute the DDL commands in RDBMS.

DDL commands:

- ✴ CREATE
- ✴ ALTER
- ✴ DROP
- ✴ RENAME
- ✴ TRUNCATE

SYNTAX'S OF COMMANDS

CREATE TABLE:

To make a new database, table, index, or stored query. A create statement in SQL creates an object inside of a relational database management system (RDBMS).

```
CREATE TABLE <table_name>
(
Column_name1 data_type ([size]),
Column_name2 data_type ([size]),
.
.
.
Column_name-n data_type ([size])
);
```

ALTER A TABLE:

To modify an existing database object. Alter the structure of the database.

To add a column in a table

ALTER TABLE table_name ADD column_name datatype;

To delete a column in a table

ALTER TABLE table_name DROP column column_name;

DROP TABLE:

Delete Objects from the Database

DROP TABLE table_name;

TRUNCATE TABLE:

Remove all records from a table, including all spaces allocated for the records are removed.

TRUNCATE TABLE table_name;

EXERCISE:

Create Table

```
SQL> create table employee
  2 (
  3 empid varchar(10) primary key,
  4 empname varchar2(20) not null,
  5 gender varchar2(7) not null,
  6 age number(3) not null,
  7 dept varchar2(15) not null,
  8 dob date not null,
  9 doj date not null
 10);
```

Table created.

```
SQL> create table salary
  2  (
  3  empid varchar(10) references employee(empid),
  4  salary number(10) not null,
  5  dept varchar(15) not null,
  6  branch varchar2(20) not null
  7  );
Table created.

SQL> create table branchtable
  2  (
  3  branch varchar2(20) not null,
  4  city varchar2(20) not null
  5  );
Table created.
```

DESCRIBE TABLE

```
SQL> desc employee;
 Name                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                     NOT NULL VARCHAR2(10)
 EMPNAME                   NOT NULL VARCHAR2(20)
 GENDER                    NOT NULL VARCHAR2(7)
 AGE                       NOT NULL NUMBER(3)
 DEPT                      NOT NULL VARCHAR2(15)
 DOB                       NOT NULL DATE
 DOJ                       NOT NULL DATE
```

SQL> desc salary;

Name	Null?	Type
EMPID		VARCHAR2 (10)
SALARY	NOT NULL	NUMBER (10)
DEPT	NOT NULL	VARCHAR2 (15)
BRANCH	NOT NULL	VARCHAR2 (20)

SQL> desc branchtable;

Name	Null?	Type
BRANCH	NOT NULL	VARCHAR2 (20)
CITY	NOT NULL	VARCHAR2 (20)

ALTER TABLE

I. ADD:

SQL> alter table employee add(designation varchar2(15));

Table altered.

SQL> alter table salary add(constraint nithi unique(empid));

Table altered.

II.MODIFY

SQL> alter table employee modify (designation varchar2(20));

Table altered.

RENAME TABLE

```
SQL> create table emp
  2  (
  3  empid varchar2(10),
  4  empname varchar2(20),
  5  age number(3),
  6  sex char
  7  );
```

Table created.

```
SQL> rename emp to empl;
```

Table renamed.

```
SQL> desc empl;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------------
 EMPID                                              VARCHAR2(10)
 EMPNAME                                            VARCHAR2(20)
 AGE                                                NUMBER(3)
 SEX                                                CHAR(1)
```

```
SQL> desc emp;
ERROR:
ORA-04043: object emp does not exist
```

Table altered.

TRUNCATE TABLE DATA

```
SQL> insert into emp values(&no,'&name','&dept',&age,'&sex');
Enter value for no: 1
Enter value for name: arun
```

Enter value for dept: it

Enter value for age: 22

Enter value for sex: m

old 1: insert into emp values(&no,'&name','&dept',&age,'&sex')

new 1: insert into emp values(1,'arun','it',22,'m')

1 row created.

SQL> insert into emp values(&no,'&name','&dept',&age,'&sex');

Enter value for no: 2

Enter value for name: bala

Enter value for dept: service

Enter value for age: 26

Enter value for sex: m

old 1: insert into emp values(&no,'&name','&dept',&age,'&sex')

new 1: insert into emp values(2,'bala','service',26,'m')

1 row created.

SQL> insert into emp values(&no,'&name','&dept',&age,'&sex');

Enter value for no: 3

Enter value for name: chitra

Enter value for dept: sales

Enter value for age: 25

Enter value for sex: f

old 1: insert into emp values(&no,'&name','&dept',&age,'&sex')

new 1: insert into emp values(3,'chitra','sales',25,'f')

1 row created.

SQL> select * from emp;

EMPID	EMPNAME	DEPT	AGE	SEX
1	arun	it	22	m

2	bala	service	26	m
3	chitra	sales	25	f

SQL> commit;

Commit complete.

SQL> truncate table emp;

Table truncated.

SQL> select * from emp;

no rows selected

SQL> commit;

Commit complete.

DROP TABLE

SQL> drop table empl;

Table dropped.

SQL> desc empl;

ERROR:

ORA-04043: object empl does not exist

CRITERIA	MAX.MARKS	MARKSOBTAINED
AIM& ALGORITHM	**5**	
EXECUTION&OUTPUT	**10**	
VIVA	**5**	
TOTAL	**20**	

RESULT:

EX.NO:1.B

DATE:

DATA MANIPULATION LANGUAGE (DML) IN RDBMS

Data manipulation language allows the users to query and manipulate data in existing schema in object. It allows following data to insert, delete, update and recovery data in schema object.

Aim:

To study DML commands in RDBMS.

DML COMMANDS:

- INSERT
- UPDATE
- DELETE
- SELECT

QUERY:

Query is a statement in the DML that request the retrieval of data from database.

- The portion of the DML used in a Query is called Query language. The SELECT statement is used to query a database

SYNTAX OF COMMANDS

INSERT:

Values can be inserted into table using insert commands. There are two types of insert commands. They are multiple value insert commands (using '&' symbol) single value insert command(without using '&'symbol)

Syntax:

INSERT INTO table_name VALUES (value1, value2, value3,…..);

(OR)

INSERT INTO table_name (column1, column2, column3,….) VALUES (value1,value2,value3,…..);

UPDATE:

This allows the user to update the particular column value using the where clause condition.

Syntax:

UPDATE <table_name> SET <col1=value> WHERE <column=value>;

DELETE:

This allows you to delete the particular column values using where clause condition.

Syntax:

DELETE FROM <table_name> WHERE <condition>;

SELECT:

The select statement is used to query a database .This statement is used to retrieve the information from the database. The SELECT statement can be used in many ways. They are:

1. **Selecting some columns** :

   To select specified number of columns from the table the

   Following command is used.

   Syntax**:**

   SELECT column_name FROM table_name;

2. **Query All Columns**:

   To select all columns from the table * is used instead of column names.

   Syntax**:**

   SELECT * FROM table_name;

3. Select using DISTINCT:

The DISTINCT keyword is used to return only different values (i.e.) this command does not select the duplicate values from the table.

Syntax**:**

SELECT DISTINCT column name(s) FROM table_name;

4. Select using IN:

If you want to get the rows which contain certain values, the best way to do it is to use the IN conditional expression.

Syntax**:**

SELECT column name(s) FROM table_name WHERE

Column name IN (value1, value2,……,value-n);

5. **Select using BETWEEN**:

BETWEEN can be used to get those items that fall within a range.

Syntax**:**

SELECT column name FROM table_name WHERE

Column name BETWEEN value1 AND value2;

6. Renaming:

The select statement can be used to rename either a column or the entire table.

Syntax**:**

Renaming a column:

SELECT column name AS new name FROM table_name;

Renaming a table:

SELECT column name FROM table_name AS newname;

7. **Sorting**:

The select statement with the **order by Clause** is used to sort the contents

Table either in ascending or descending order.

Syntax**:**

SELECT column name FROM table_name WHERE

Condition ORDER BY column name ASC/DESC;

8. **To Select by matching some patterns**:

The select statement along with **like clause** I is used to match strings. The **like** condition is used to specify a search pattern in a column.

Syntax**:**

SELECT column name FROM table_name WHERE Column name LIKE "% or-";

%: Matches any sub string.

\- : Matches a single character.

9. SELECT INTO statement:

The SELECT INTO statement is most often used to create backup copies of tables or for archiving records.

Syntax**:**

SELECT Column_name(s) INTO variable_name(s) FROM table_name

WHERE condition.

10. **To Select NULL values**:

We can use the SELECT statement to select the 'null' values also.
For retrieving roes where some of the columns have been defined as NULLs there is a special comparison operator of the form IS [NOT]NULL.

Syntax:

SELECT column name FROM table_name WHERE Column name IS NULL;

11. **Select using AND, OR, NOT**:

We can combine one or more conditions in a SELECT statement using the logical operators AND, OR, NOT.

Syntax:

SELECT column name FROM table_name WHERE Condition1 LOGICAL OPERATOR condition2;

EXERCISE:

INSERT COMMAND

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: it9001

Enter value for empname: arunkumar

Enter value for gender: male

Enter value for age: 22

Enter value for dept: it

Enter value for dob: 12-jan-1988

Enter value for doj: 23-oct-2006

Enter value for desig: manager

old 1: insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi

new 1: insert into employee values('it9001','arunkumar','male',22,'it','12-jan-1988','23-oct-2006'

1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: it9001

Enter value for empname: arunkumar

Enter value for gender: male

Enter value for age: 22

Enter value for dept: it

Enter value for dob: 12-jan-1988

Enter value for doj: 23-oct-2006

Enter value for desig: manager

old 1: insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi

new 1: insert into employee values('it9001','arunkumar','male',22,'it','12-jan-1988','23-oct-2006'

1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: it9002

Enter value for empname: balakrishnan

Enter value for gender: male

Enter value for age: 27

Enter value for dept: it

Enter value for dob: 27-mar-1983

Enter value for doj: 02-dec-2008

Enter value for desig: coordinator

old 1: insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi

new 1: insert into employee values('it9002','balakrishnan','male',27,'it','27-mar-1983','02-dec-20

1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: acc9001

Enter value for empname: kannan

Enter value for gender: male

Enter value for age: 35

Enter value for dept: accounts

Enter value for dob: 28-dec-1975

Enter value for doj: 01-jan-1995

Enter value for desig: manager

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi

new 1: insert into employee values('acc9001','kannan','male',35,'accounts','28-dec-1975','01-jan-1

1 row created.

SQL> insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: acc9002

Enter value for empname: magudeshwaran

Enter value for gender: male

Enter value for age: 27

Enter value for dept: accounts

Enter value for dob: 25-aug-1983

Enter value for doj: 12-apr-2000

Enter value for desig: asst manager

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi

new 1: insert into employee values('acc9002','magudeshwaran','male',27,'accounts','25-aug-1983','1

1 row created.

SQL> insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: ser9001

Enter value for empname: jagadheesh

Enter value for gender: male

Enter value for age: 33

Enter value for dept: service

Enter value for dob: 31-mar-1877

Enter value for doj: 3-jun-1999

Enter value for desig: manager

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi
new 1: insert into employee values('ser9001','jagadheesh','male',33,'service','31-mar-1877','3-jun
1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: ser9006

Enter value for empname: muruganandam

Enter value for gender: male

Enter value for age: 35

Enter value for dept: service

Enter value for dob: 09-aug-1975

Enter value for doj: 02-mar-2000

Enter value for desig: painter

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi
new 1: insert into employee values('ser9006','muruganandam','male',35,'service','09-aug-1975','02-
1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

SQL> /

Enter value for empid: sal9001

Enter value for empname: suresh

Enter value for gender: male

Enter value for age: 40

Enter value for dept: sales

Enter value for dob: 12-jul-1970

Enter value for doj: 01-apr-1996

Enter value for desig: manager

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi
new 1: insert into employee values('sal9001','suresh','male',40,'sales','12-jul-1970','01-apr-1996
1 row created.

SQL> insert into employee

values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desig');

Enter value for empid: sal9006

Enter value for empname: sharmila

Enter value for gender: female

Enter value for age: 27

Enter value for dept: sales

Enter value for dob: 12-jan-1983

Enter value for doj: 09-aug-2007

Enter value for desig: executive

old 1: insert into employee
values('&empid','&empname','&gender',&age,'&dept','&dob','&doj','&desi
new 1: insert into employee values('sal9006','sharmila','female',27,'sales','12-jan-1983','09-aug-
1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: it9002

Enter value for salary: 18000

Enter value for dept: it

Enter value for branch: abt maruthi

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('it9002',18000,'it','abt maruthi')

1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: acc9001

Enter value for salary: 35000

Enter value for dept: accounts

Enter value for branch: cars india

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('acc9001',35000,'accounts','cars india')

1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: acc9002

Enter value for salary: 26000

Enter value for dept: accounts

Enter value for branch: cars india

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('acc9002',26000,'accounts','cars india')

1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: ser9001

Enter value for salary: 35000

Enter value for dept: service

Enter value for branch: chennai cars

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('ser9001',35000,'service','chennai cars')

1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: ser9006

Enter value for salary: 12000

Enter value for dept: service

Enter value for branch: greenland cars

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('ser9006',12000,'service','greenland cars')

1 row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: sal9001

Enter value for salary: 40000

Enter value for dept: sales

Enter value for branch: abt maruthi

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('sal9001',40000,'sales','abt maruthi')1

row created.

SQL> insert into salary values(‘&empid’,&salary,’&dept’,’&branch’);

Enter value for empid: sal9006

Enter value for salary: 17000

Enter value for dept: sales

Enter value for branch: abt maruthi

old 1: insert into salary values('&empid',&salary,'&dept','&branch')

new 1: insert into salary values('sal9006',17000,'sales ','abt maruthi')

1 row created.

SQL> select * from salary;

EMPID	SALARY	DEPT	BRANCH
it9001	35000	it	abt maruthi
it9002	18000	it	abt maruthi
acc9001	35000	accounts	cars india
acc9002	26000	accounts	cars india
ser9001	35000	service	chennai cars
ser9006	12000	service	greenland cars
sal9001	40000	sales	abt maruthi
sal9006	17000	sales	abt maruthi

8 rows selected.

SQL> select * from employee;

EMPID	EMPNAME	GENDER	AGE	DEPT	DOB	DOJ	DESIGNATION
it9001	arunkumar	male	22	it	12-JAN-88	23-OCT-06	manager
it9002	balakrishnan	male	27	it	27-MAR-83	02-DEC-08	coordinator
acc9001	kannan	male	35	accounts	28-DEC-75	01-JAN-95	manager

EMPID EMPNAME GENDER AGE DEPT DOB

DOJ DESIGNATION

---

acc9002 magudeshwaran male 27 accounts 25-AUG-83

12-APR-00 asst manager

ser9001 jagadheesh male 33 service 31-MAR-77

03-JUN-99 manager

ser9006 muruganandam male 35 service 09-AUG-75

02-MAR-00 painter

EMPID EMPNAME GENDER AGE DEPT DOB

DOJ DESIGNATION

---

sal9001 suresh male 40 sales 12-JUL-70

01-APR-96 manager

sal9006 sharmila female 27 sales 12-JAN-83

09-AUG-07 executive

8 rows selected.

SQL> insert into branchtable values('&branch','&city');

Enter value for branch: abt maruthi

Enter value for city: chennai

old 1: insert into branchtable values('&branch','&city')

new 1: insert into branchtable values('abt maruthi','chennai')

1 row created.

SQL> select * from salary;

EMPID	SALARY	DEPT	BRANCH
it9001	35000	it	abt maruthi
it9002	18000	it	abt maruthi
acc9001	35000	accounts	cars india
acc9002	26000	accounts	cars india
ser9001	35000	service	chennai cars
ser9006	12000	service	greenland cars
sal9001	40000	sales	abt maruthi
sal9006	17000	sales	abt maruthi

8 rows selected.

SQL> insert into branchtable values('&branch','&city');
Enter value for branch: cars india
Enter value for city: vellore
old 1: insert into branchtable values('&branch','&city')
new 1: insert into branchtable values('cars india','vellore')
1 row created.

SQL> insert into branchtable values('&branch','&city');
Enter value for branch: chennai cars
Enter value for city: thambaram
old 1: insert into branchtable values('&branch','&city')
new 1: insert into branchtable values('chennai cars','thambaram')
1 row created.

SQL> insert into branchtable values('&branch','&city');
Enter value for branch: greenland cars

Enter value for city: kanchipuram

old 1: insert into branchtable values('&branch','&city')

new 1: insert into branchtable values('greenland cars','kanchipuram')

1 row created.

SQL> select * from branchtable;

BRANCH	CITY
abt maruthi	chennai
cars india	vellore
chennai cars	thambaram
greenland cars	kanchipuram

UPDATE COMMAND

SQL> update employee set empname = 'arunprasanth' where empid='it9001';

1 row updated.

SQL> update employee set designation='&designation' where empname='&empname';

Enter value for designation: supervisor

Enter value for empname: muruganandam

old 1: update employee set designation='&designation' where empname='&empname'

new 1: update employee set designation='supervisor' where empname='muruganandam'

1 row updated.

SQL> select empname,designation from employee;

EMPNAME	DESIGNATION
arunprasanth	manager
balakrishnan	coordinator
kannan	manager
magudeshwaran	asst manager

jagadheesh manager

muruganandam supervisor

suresh manager

sharmila executive

8 rows selected.

SELECT COMMAND

To retrieve particular column

SQL> select empname from emp;

EMPNAME

---

arun

bala

bakyaraj

chitra

To retrieve all columns

SQL> select * from emp;

EMPID	EMPNAME	DEPT	AGE	S
1	arun	it	22	m
2	bala	accounts	26	m
3	bakyaraj	stores	30	m
4	chitra	sales	24	f

DELETE COMMAND

To delete particular record

SQL> delete emp where empid=1;

1 row deleted.

SQL> select * from emp;

EMPID	EMPNAME	DEPT	AGE	S
2	bala	accounts	26	m
3	bakyaraj	stores	30	m
4	chitra	sales	24	f

To delete all records

SQL> delete from emp;

3 rows deleted.

SQL> create table student (idno number, name varchar(10),branch varchar(4));

Table created.

SQL> desc student;

NAME	NULL?	TYPE
IDNO		NUMBER
NAME		VARCHAR2(10)
BRANCH		VARCHAR2(4)

SQL> alter table student add degree varchar(10);

Table altered.

SQL> desc student;

NAME	NULL?	TYPE
IDNO		NUMBER
NAME		VARCHAR2 (10)
BRANCH		VARCHAR2 (4)
DEGREE		VARCHAR2 (10)

SQL> alter table student modify degree varchar(6);

Table altered.

SQL> desc student;

```
NAME                      NULL?  TYPE
----------------------------------------------------
IDNO                              NUMBER
NAME                              VARCHAR2 (10)
BRANCH                            VARCHAR2 (4)
DEGREE                            VARCHAR2 (6)
SQL> insert into student (name, degree, branch, idno) values('ASHOK','BE','CSE',01);
1 row created.
SQL> insert into student values(02,'BHAVANA','CSE','BE');
1 row created.
SQL> insert into student values(&idno, &name, &branch, &degree);
Enter value for idno: 03
Enter value for name: 'CAVIN'
Enter value for branch: 'CSE'
Enter value for degree: 'BE'
old  1: insert into student values(&idno,&name,&branch,&degree)
new  1: insert into student values(03,'CAVIN','CSE','BE')
1 row created.
SQL> /
Enter value for idno: 04
Enter value for name: 'DANNY'
Enter value for branch: 'IT'
Enter value for degree: 'BE'
old  1: insert into student values(&idno,&name,&branch,&degree)
new  1: insert into student values(04,'DANNY','IT','BE')
1 row created.
SQL> /
Enter value for idno: 05
```

Enter

value

for

name:

'HARR

Y'Enter

value

for

branch:

'IT'

Enter

value

for

degree:

'BE'

old 1: insert into student

values(&idno,&name,&branch,&degree)new

1: insert into student

values(05,'HARRY','IT','BE')

1 row created.

SQL> select * from student;

IDNO	NAME	BRAN	DEGREE
1	ASHOK	CSE	BE
2	BHAVANA	CSE	BE

3 CAVIN CSE BE

4 DANNY IT BE

5 HARRY IT BE

```
SQL> update student set
degree='B.TECH' where branch='IT';2
rows updated.
SQL> select * from student;
```

IDNO	NAME	BRAN	DEGREE
1	ASHOK	CSE	BE
2	BHAVANA	CSE	BE
3	CAVIN	CSE	BE
4	DANNY	IT	B.TECH
5	HARRY	IT	B.TECH

```
SQL> delete
from student
where idno=5;1
row deleted.
```