

CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY FACULTY OF TECHNOLOGY & ENGINEERING

**DEVANG PATEL INSTITUTE OF ADVANCE TECHNOLOGY
AND RESEARCH (DEPSTAR)**

Subject: CE246 -Database Management System

Semester: IV Academic Year: 2021-22

Full name: Vaishnavi Bhalodi

Id: 20DCE008

Practical-1

Aim:

Evaluation of Database (File System, DBMS, RDBMS, DDBMS)

Theory:

File System:

A file system is a process that manages how and where data on a storage disk, typically a hard disk drive (HDD), is stored, accessed and managed. It is a logical disk component that manages a disk's internal operations as it relates to a computer and is abstract to a human user.

Examples: NTFS, HFS

DBMS:

A database management system (DBMS) is system software for creating and managing databases. A DBMS makes it possible for end users to create, protect, read, update and delete data in a database. The most prevalent type of data management platform, the DBMS essentially serves as an interface between databases and end users or application programs, ensuring that data is consistently organized and remains easily accessible.

DDBMS:

This chapter introduces the concept of DDBMS. In a distributed database, there are a number of databases that may be geographically distributed all over the world. A distributed DBMS manages the distributed database in a manner so that it appears as one single database to users.

RDBMS:

The software used to store, manage, query, and retrieve data stored in a relational database is called a relational database management system (RDBMS). The RDBMS provides an interface between users and applications and the database, as well as administrative functions for managing data storage, access, and performance.

Differences:

File System	DBMS
It manages and organizes files in a storage medium.	It manages the database.
It doesn't provide backup.	It provides backup and recovery of data.
It has less consistency.	It has less consistency.
It is less complex as compared to DBMS.	It is more complex as compared to DBMS.
It provides less security.	It provides more security.
It provides less expensive.	It provides more expensive.
There is no data independence.	There is data independence.

DBMS	RDBMS
It stores data as file format.	It stores data in tabular form.
Data elements are accessed individually.	Multiple data elements can be accessed at the same time.
There is no relationship between the data.	There is relationship between the data present in the tabular form.
It does not support distributed database.	It supports distributed database.
It stores data in either a navigational or hierarchical form.	It uses a tabular structure where the headers are the column names, and the rows contain corresponding values.
It provides less expensive.	It provides more expensive.
It deals with small quantity of data	It deals with large amount of data.
It supports single user.	It supports multiple users.

Conclusion:

I learnt about the different database systems and their differences.

Practical-2

Aim:

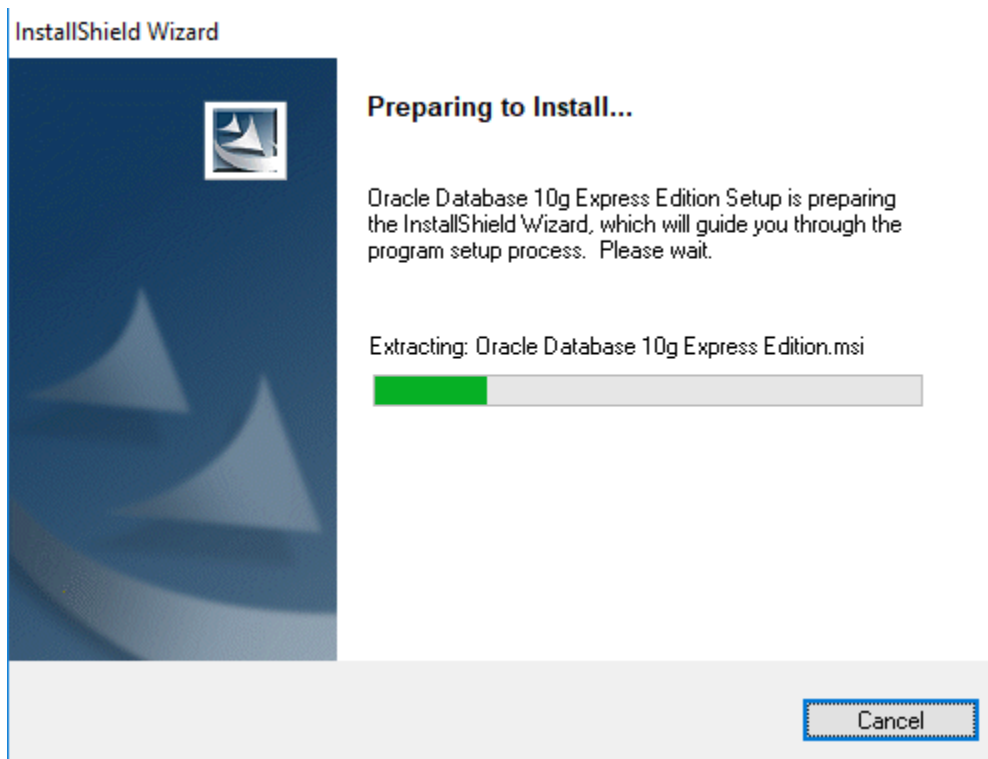
Introduction to Oracle (step by step installation, introduction of sql, plsql).

Installation:

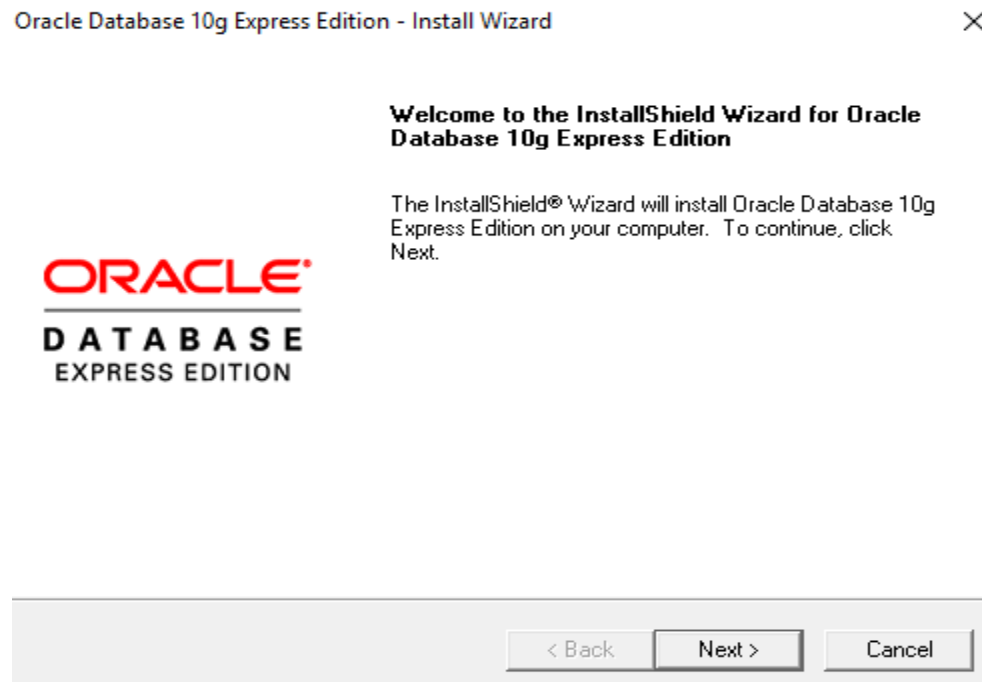
- 1) Download Oracle 10g from below link:

<https://drive.google.com/file/d/1Y6ghDOEfVorTNrzWgF1UKaENHjeGgrHG/view>

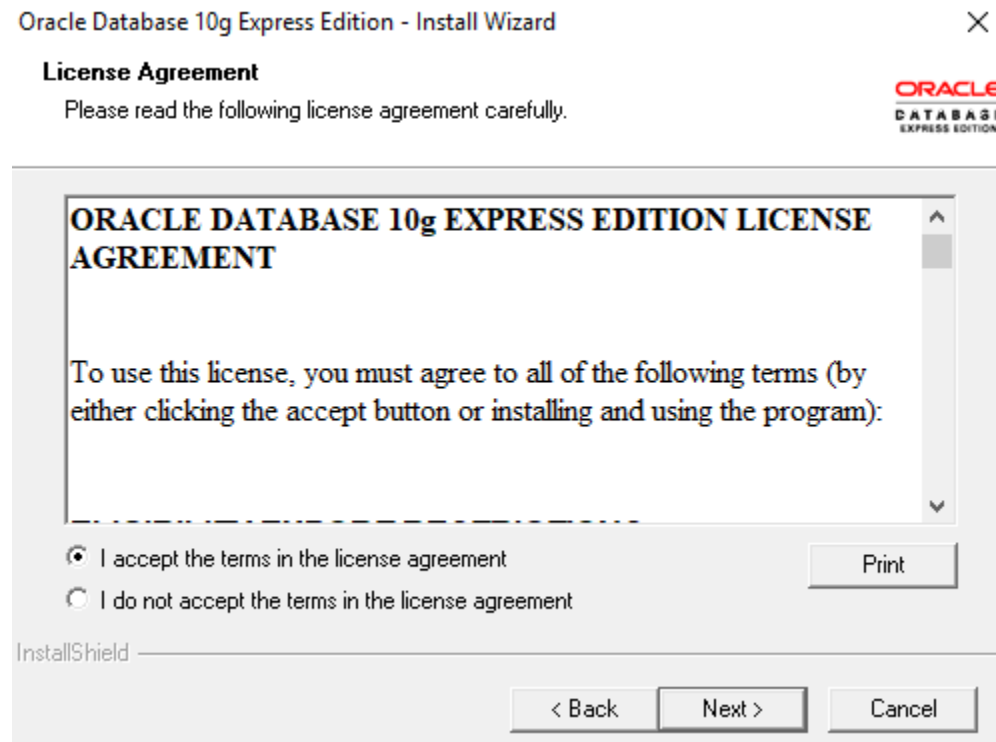
- 2) Install it by double clicking .exe which you have downloaded



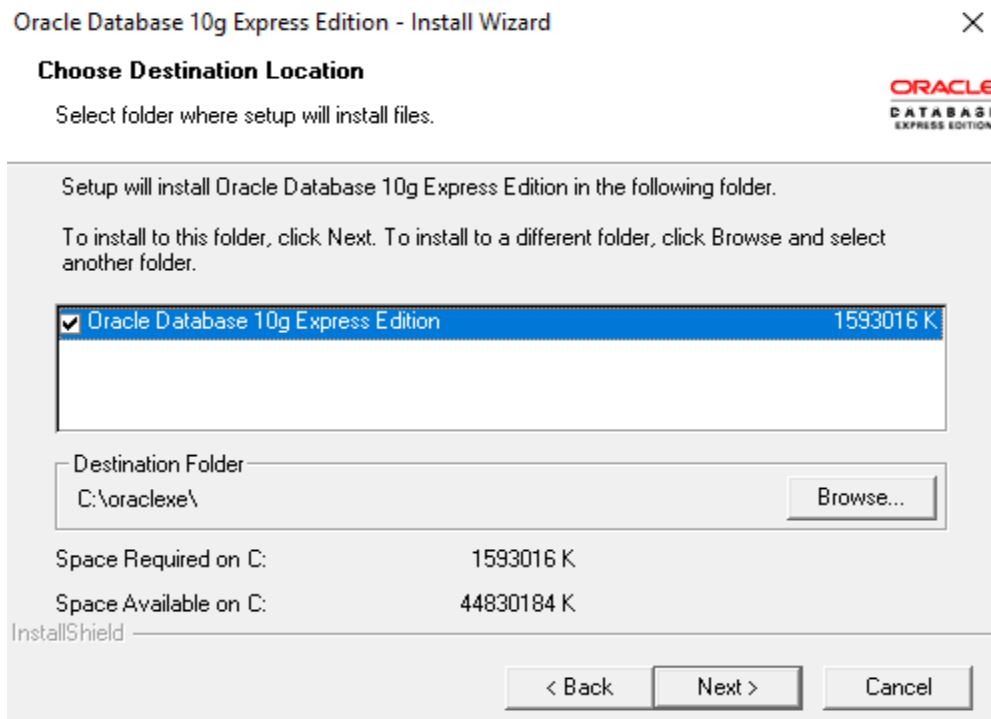
3) Click on Next button



4) Accept license agreement and click on next button



5) Click on next button



- 6) Enter password and confirm password for SYS and SYSTEM user. Please remember it because once installation will be over you have to enter it. To make it easy to remember give password as : “oracle”

Oracle Database 10g Express Edition - Install Wizard

**Specify Database Passwords**

Enter and confirm passwords for the database. This password will be used for both the SYS and the SYSTEM database accounts.

Enter Password

Confirm Password

Note: You should use the SYSTEM user along with the password you enter here to log in to the Database Home Page after the install is complete.

InstallShield

< Back

Next >

Cancel

7) Click on install button

Oracle Database 10g Express Edition - Install Wizard

**Summary**

Review settings before proceeding with the Installation.



Current Installation Settings:

Destination Folder: C:\oraclexe\
Port for 'Oracle Database Listener': 1521
Port for 'Oracle Services for Microsoft Transaction Server': 2030
Port for HTTP Listener: 8080

InstallShield

< Back

Install

Cancel

8) Click on finish button.

Oracle Database 10g Express Edition - Install Wizard

InstallShield Wizard Complete

Setup has finished installing Oracle Database 10g Express Edition on your computer.



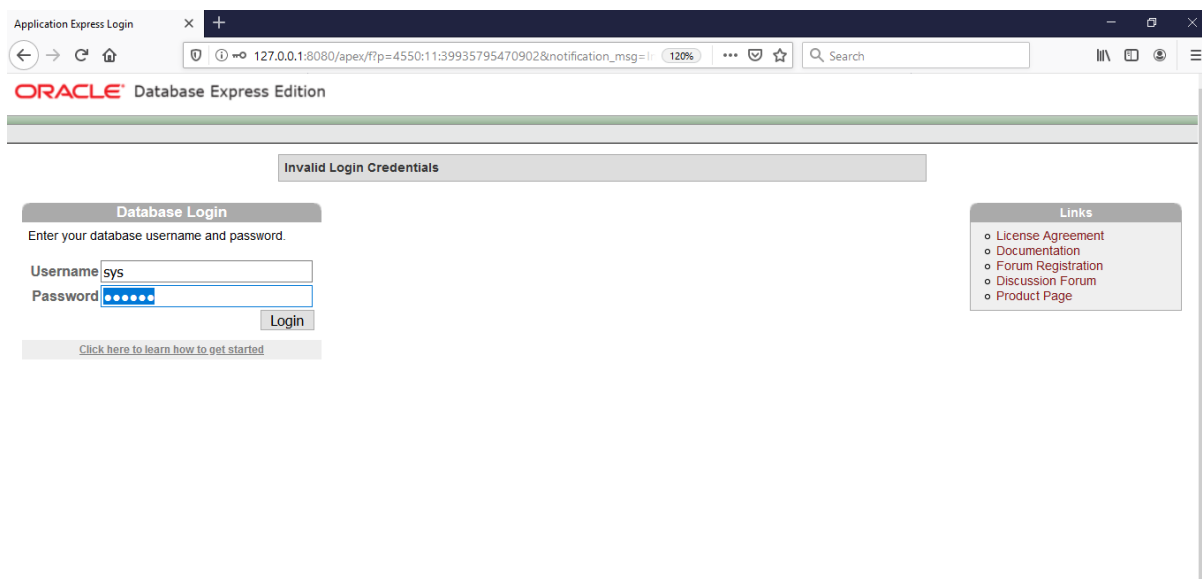
☒ Launch the Database homepage.

< Back

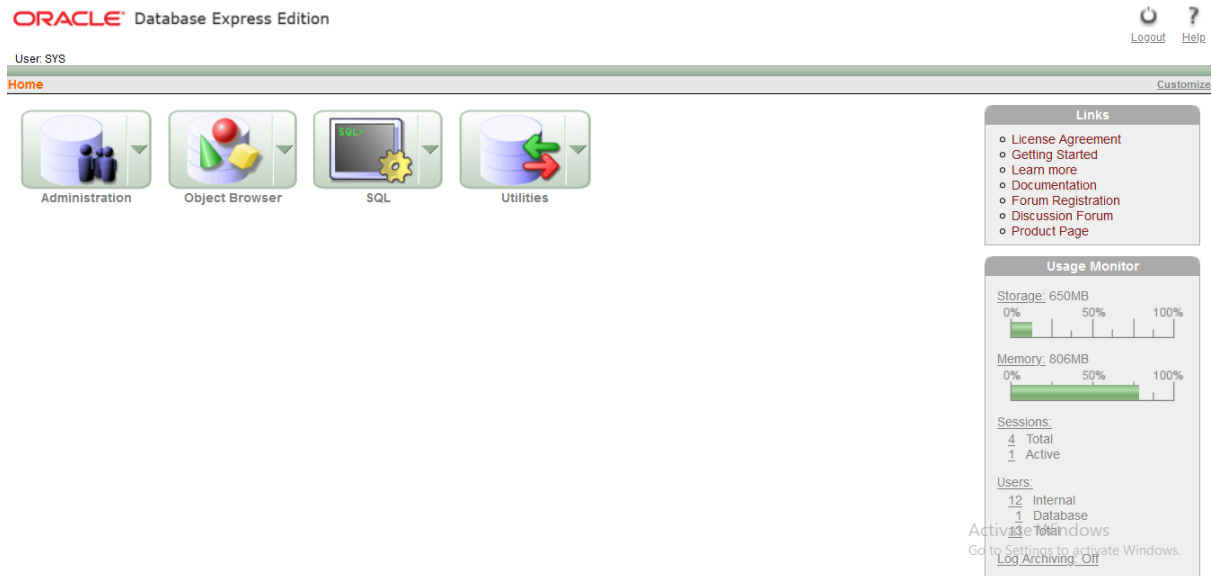
Finish

Cancel

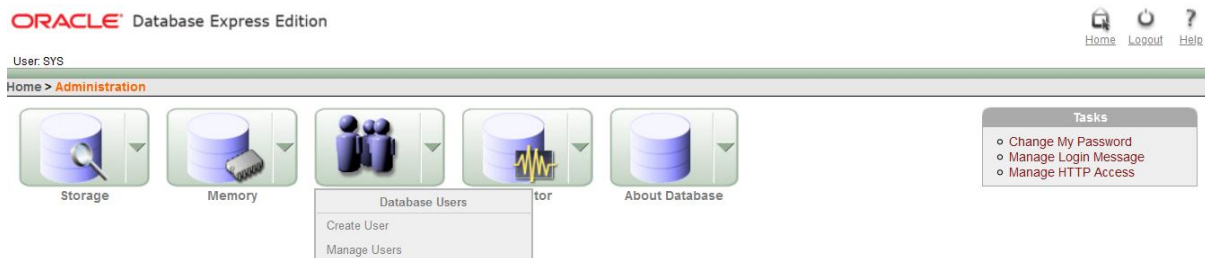
9) Enter username as SYS OR SYSTEM and enter your password (Entered in step: 6)



10) Click on Administration



- 11) Now click on “database user drop down button”. From that click on “create user”.



- 12) Enter your college roll no in username and give password (NEW) and confirm password. Don't check expire password, make account status

unblocked if it is not. Give all privileges to your user. Finally click on “create” button.

ORACLE Database Express Edition

User: SYS

Home > Administration > Manage Database Users > **Create Database User**

Create Database User Cancel Create

Username: 18dce001
 Password:
 Confirm Password:
 Expire Password: ☐
 Account Status: Unlocked
 Default Tablespace: USERS
 Temporary Tablespace: TEMP

Database Users

All database objects are owned by a database user. Use this page to create a new user and define privileges. Use SQL Commands to manage additional user attributes.

User Privileges

Roles:
☒ CONNECT ☒ RESOURCE ☒ DBA

Direct Grant System Privileges:

<input checked="" type="checkbox"/> CREATE DATABASE LINK	<input checked="" type="checkbox"/> CREATE MATERIALIZED VIEW	<input checked="" type="checkbox"/> CREATE PROCEDURE
<input checked="" type="checkbox"/> CREATE PUBLIC SYNONYM	<input checked="" type="checkbox"/> CREATE ROLE	<input checked="" type="checkbox"/> CREATE SEQUENCE
<input checked="" type="checkbox"/> CREATE SYNONYM	<input checked="" type="checkbox"/> CREATE TABLE	<input checked="" type="checkbox"/> CREATE TRIGGER
<input checked="" type="checkbox"/> CREATE TYPE	<input checked="" type="checkbox"/> CREATE VIEW	

[Check All](#) [Uncheck All](#)

Activate Windows
Go to Settings to activate Windows.

13) This page will be shown to you. Now click on “logout” button.

ORACLE Database Express Edition

User: SYS

Home > Administration > **Manage Database Users**

✓ **User Created.**

Search Username: View: Icons Show: Database Users Display: 15 Go Create >

18DCE001

HR

14) Click on login

ORACLE Database Express Edition

You are now logged out.

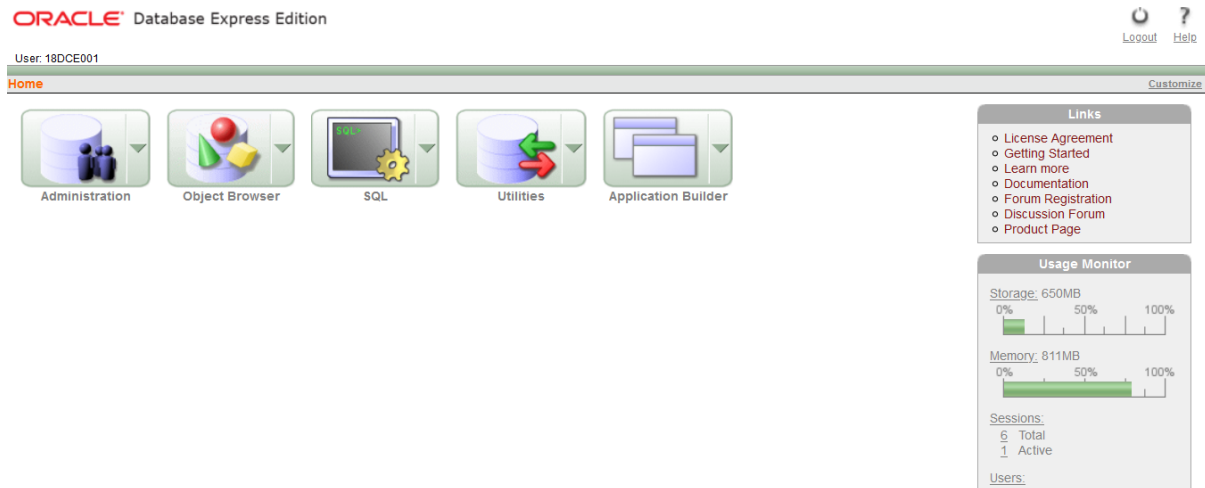
[Login](#)

15) Enter username and password that you just created and click on “login” button

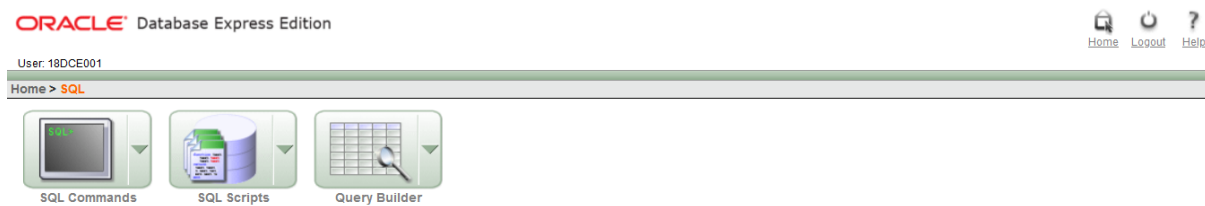
ORACLE Database Express Edition

Database Login	Links
Enter your database username and password.	<ul style="list-style-type: none">◦ License Agreement◦ Documentation◦ Forum Registration◦ Discussion Forum◦ Product Page
Username <input type="text" value="18dce001"/>	
Password <input type="password" value="•••••"/>	
<input type="button" value="Login"/>	
Click here to learn how to get started	

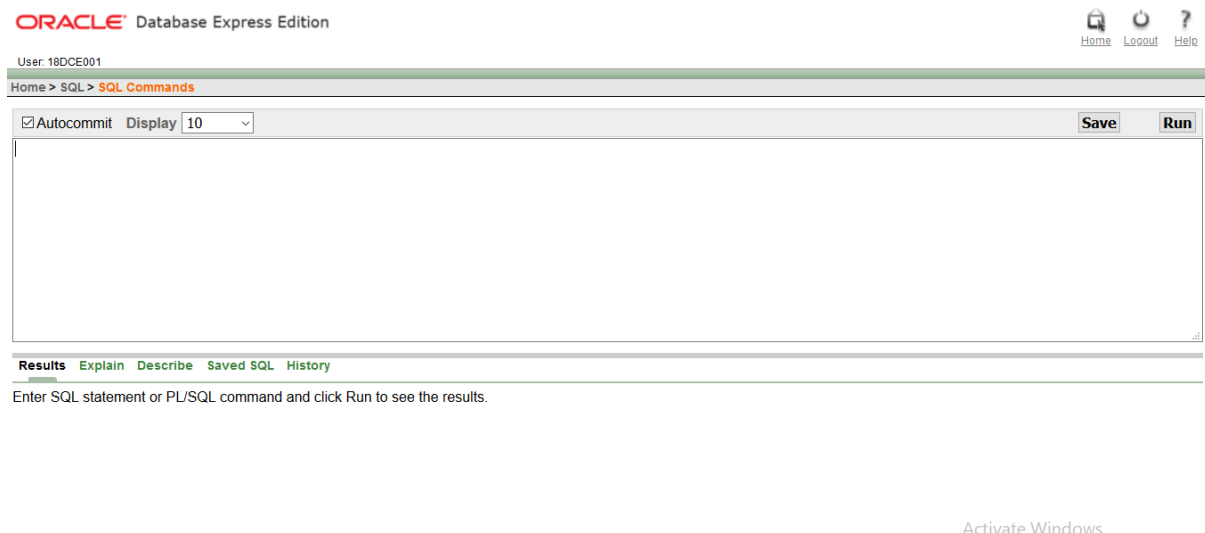
16) Click on SQL



17) Click on SQL Commands



18) Congratulation!!! Now you are ready to code SQL and PLSQL.



Conclusion: We learnt how to install oracle 10g and how to use it.

Practical-3

Aim:

(i) Create tables according to the following definition.

- CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2), ADATE DATE);
- CREATE TABLE BRANCH (BNAME VARCHAR2(18), CITY VARCHAR2(18));
- CREATE TABLE CUSTOMERS (CNAME VARCHAR2(19), CITY VARCHAR2(18));
- CREATE TABLE BORROW (LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));

(ii) Insert the data as shown below.

DEPOSIT

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000.00	1-MAR-95
101	SUNIL	AJNI	5000.00	4-JAN-96
102	MEHUL	KAROLBAGH	3500.00	17-NOV-95
104	MADHURI	CHANDI	1200.00	17-DEC-95
105	PRMOD	M.G.ROAD	3000.00	27-MAR-96
106	SANDIP	ANDHERI	2000.00	31-MAR-96
107	SHIVANI	VIRAR	1000.00	5-SEP-95
108	KRANTI	NEHRU PLACE	5000.00	2-JUL-95
109	MINU	POWAI	7000.00	10-AUG-95

BRANCH		CUSTOMERS	
BNAME	CITY	CNAME	CITY
VRCE	NAGPUR	ANIL	CALCUTTA
AJNI	NAGPUR	SUNIL	DELHI
KAROLBAGH	DELHI	MEHUL	BARODA
CHANDI	DELHI	MANDAR	PATNA
DHARAMPETH	NAGPUR	MADHURI	NAGPUR
M.G.ROAD	BANGLORE	PRAMOD	NAGPUR
ANDHERI	BOMBAY	SANDIP	SURAT
VIRAR	BOMBAY	SHIVANI	BOMBAY
NEHRU PLACE	DELHI	KRANTI	BOMBAY
POWAI	BOMBAY	NAREN	BOMBAY

BORROW			
LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

Theory:

DDL: A data definition language (DDL) is a computer language used to create and modify the structure of database objects in a database.

DDL-CREATE command: This command builds a new table and has a predefined syntax. The create command syntax is:

CREATE TABLE [table name] ([column definitions]) [table parameters];

For example:

CREATE TABLE DEPOSIT (ACTNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2), ADATE DATE);

DML: A data manipulation language (DML) is a family of computer languages including commands permitting users to manipulate data in a database. This manipulation involves inserting data into database tables, retrieving existing data, deleting data from existing tables and modifying existing data. DML is mostly incorporated in SQL databases.

DML_INSERT command: This command adds one or more records to a database table. The insert command syntax is:

INSERT INTO [table name] [column(s)] VALUES [value(s)].

For Example:

```
INSERT INTO DEPOSIT (ACTNO, CNAME, BNAME, AMOUNT, ADATE)
VALUES('101','SUNIL','AJNI',5000.00,'4-JAN-96');
```

Program:

1. Describe deposit, branch.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▾

desc deposit;

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **DEPOSIT**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
<u>DEPOSIT</u>	<u>ACTNO</u>	Varchar2	5	-	-	-	✓	-	-
	<u>CNAME</u>	Varchar2	18	-	-	-	✓	-	-
	<u>BNAME</u>	Varchar2	18	-	-	-	✓	-	-
	<u>AMOUNT</u>	Number	-	8	2	-	✓	-	-
	<u>ADATE</u>	Date	7	-	-	-	✓	-	-

1 - 5

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

desc branch;

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **BRANCH**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRANCH	BNAME	Varchar2	18	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-
1 - 2									

2. Describe borrow, customers.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

DESC BORROW

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **BORROW**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BORROW	LOANNO	Varchar2	5	-	-	-	✓	-	-
	CNAME	Varchar2	18	-	-	-	✓	-	-
	BNAME	Varchar2	18	-	-	-	✓	-	-
	AMOUNT	Number	-	8	2	-	✓	-	-
1 - 4									

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

DESC customers

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **CUSTOMERS**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CNAME	Varchar2	19	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-

1 - 2

3. List all data from table DEPOSIT.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

select *from deposit;

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000	01-MAR-95
101	SUNIL	AJNI	5000	04-JAN-96
102	MEHUL	KAROLBAGH	3500	17-NOV-95
104	MADHURI	CHANDI	1200	17-DEC-95
105	PRMOD	M.G.ROAD	3000	27-MAR-96
106	SANDIP	ANDHERI	2000	31-MAR-96
107	SHIVANI	VIRAR	1000	05-SEP-95
108	KRANTI	NEHRUPLACE	5000	02-JUL-95
109	MINU	POWAI	7000	10-AUG-95

9 rows returned in 0.01 seconds [CSV Export](#)

4. List all data from table BORROW.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select *from borrow;
```

Results Explain Describe Saved SQL History

LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000
206	MEHUL	AJNI	5000
311	SUNIL	DHARAMPETH	3000
321	MADHURI	ANDHERI	2000
375	PRMOD	VIRAR	8000
481	KRANTI	NEHRU PALACE	3000

6 rows returned in 0.00 seconds [CSV Export](#)

5. List all data from table CUSTOMERS.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select *from customers;
```

Results Explain Describe Saved SQL History

CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

10 rows returned in 0.00 seconds [CSV Export](#)

6. List all data from table BRANCH.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select *from branch;
```

Results Explain Describe Saved SQL History

BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAUGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

10 rows returned in 0.00 seconds [CSV Export](#)

7. Give account no and amount of depositors.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select actno,amount from deposit;
```

Results Explain Describe Saved SQL History

ACTNO	AMOUNT
100	1000
101	5000
102	3500
104	1200
105	3000
106	2000
107	1000
108	5000
109	7000

9 rows returned in 0.02 seconds [CSV Export](#)

8. Give name of depositors having amount greater than 4000.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select cname from deposit where amount>4000;
```

Results Explain Describe Saved SQL History

CNAME

SUNIL

KRANTI

MINU

3 rows returned in 0.00 seconds

[CSV Export](#)

9. Give name of customers who opened account after date '1-12-96'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select cname from deposit where ADATE >'1-dec-96'
```

Results Explain Describe Saved SQL History

no data found

10. Give name of city where branch karolbagh is located.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select city from branch where BNAME ='KAROLBAGH'
```

Results Explain Describe Saved SQL History

no data found

11. Give account no and amount of customer having account opened between date 1-12-96 and 1-6-96.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select actno,amount from deposit where adate between '1-dec-96' and '1-jun-96'
```

Results Explain Describe Saved SQL History

no data found

12. Give names of depositors having account at VRCE.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select CNAME from DEPOSIT where BNAME='VRCE'
```

Results Explain Describe Saved SQL History

CNAME
ANIL

1 rows returned in 0.00 seconds [CSV Export](#)

Conclusion:

We learnt some basic functions like creating tables, entering values into it and performing some operations on it.

Practical-4

Aim: Create the below given table and insert the data accordingly.

Create Table Job (job_id, job_title, min_sal, max_sal)

COLUMN NAME	DATA TYPE
job_id	Varchar2(15)
job_title	Varchar2(30)
min_sal	Number(7,2)
max_sal	Number(7,2)

Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no, l_name, dept_name, job_id, location, manager_id, hiredate)

COLUMN NAME	DATA TYPE
emp_no	Number(3)
emp_name	Varchar2(30)
emp_sal	Number(8,2)
emp_comm	Number(6,1)
dept_no	Number(3)
l_name	Varchar2(30)
dept_name	Varchar2(30)
job_id	Varchar2(15)
location	Varchar2(15)
manager_id	Number(5)
hiredate	Date

Create table deposit(a_no,cname,bname,amount,a_date).

COLUMN NAME	DATA TYPE
a_no	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Number(7,2)
a_date	Date

Create table borrow (loanno, cname, bname, amount)

COLUMN NAME	DATA TYPE
loanno	Varchar2(5)
cname	Varchar2(15)
bname	Varchar2(10)
amount	Varchar2(7,2)

Insert following values in the table Employee.

emp_id	emp_name	emp_sal	emp_code	dept_id	l_name	dept_name	job_id	location	manager_id	hiredate
101	Smith	800		20	shah	machine learning	fi_mgr	toronto	105	09-aug-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas		14-mar-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-nov-95
104	Aman	3000		15	sharma	virtual reality	comp_op	mexico	12	02-oct-97
105	Anita	5000	50,000	10	patel	big data analytics	comp_op	germany	107	01-jan-98
106	Sneha	2450	24,500	10	joseph	big data analytics	fi_acc	melbourne	105	26-sep-97
107	Anamika	2975		30	jha	artificial intelligence	it_prog	new york		15jul-97

Insert following values in the table Job.

job_id	job_name	min_sal	max_sal
it_prog	Programmer	4000	10000
mk_mgr	Marketing manager	9000	15000
fi_mgr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000
comp_op	Computer Operator	1500	3000

Insert following values in the table deposit

A_no	cname	Bname	Amount	date
101	Anil	andheri	7000	01-jan-06
102	sunil	virar	5000	15-jul-06
103	jay	villeparle	6500	12-mar-06
104	vijay	andheri	8000	17-sep-06
105	keyur	dadar	7500	19-nov-06
106	mayur	borivali	5500	21-dec-06

Program:

Perform following queries :

(1) Retrieve all data from employee, jobs and deposit.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

select * from Employee

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

7 rows returned in 0.02 seconds [CSV Export](#)

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select * from deposit_1
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

A_NO	CNAME	BNAME	AMOUNT	A_DATE
101	ANIL	ANDHERI	7000	01-JAN-06
102	SUNIL	VIRAR	5000	15-JUL-06
103	JAY	VILLEPARLE	6500	12-MAR-06
104	VIJAY	ANDHERI	8000	17-SEP-06
105	KEYUR	DADAR	7500	19-NOV-06
106	MAYUR	BORIVALI	5500	21-DEC-06

6 rows returned in 0.00 seconds

[CSV Export](#)

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select * from job
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
IT_PROG	PROGRAMMER	4000	10000
MK_MGR	MARKETING MANAGER	9000	15000
FI_MGR	FINANCE MANAGER	8200	12000
FI_ACC	ACCOUNT	4200	9000
LEC	LECTURER	6000	17000
COMP_OP	COMPUTER OPERATOR	1500	3000

6 rows returned in 0.00 seconds

[CSV Export](#)

- (2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display

```
select A_NO, AMOUNT from deposit_1
WHERE A_DATE BETWEEN '01-JAN-06' AND '25-JUL-06';
```

Results Explain Describe Saved SQL History

A_NO	AMOUNT
101	7000
102	5000
103	6500

3 rows returned in 0.01 seconds [CSV Export](#)

- (3) Display all jobs with minimum salary is greater than 4000.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display

```
select job_title , min_sal from job
where min_sal>4000;
```

Results Explain Describe Saved SQL History

JOB_TITLE	MIN_SAL
MARKETING MANAGER	9000
FINANCE MANAGER	8200
ACCOUNT	4200
LECTURER	6000

4 rows returned in 0.00 seconds [CSV Export](#)

- (4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select emp_name as ename,emp_sal from employee
where dept_no=20;
```

Results Explain Describe Saved SQL History

ENAME	EMP_SAL
Smith	800
Adama	1100

2 rows returned in 0.00 seconds [CSV Export](#)

- (5) Display employee no, name and department details of those employee whose department lies in (10,20).

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select emp_no,emp_name,dept_no,dept_name from employee
where dept_no in(10,20);
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	DEPT_NO	DEPT_NAME
101	Smith	20	machine learning
103	Adama	20	machine learning
105	Anita	10	big data analytics
106	Sneha	10	big data analytics

4 rows returned in 0.01 seconds [CSV Export](#)

(6) Display the non-null values of employees.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select * from employee
where emp_comm is NOT NULL AND MANAGER_ID IS NOT NULL;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

3 rows returned in 0.01 seconds [CSV Export](#)

(7) Display name of customer along with its account no (both column should be displayed as one) whose amount is not equal to 8000 Rs.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select cname || a.no as DETAILS from deposit_1
where amount!=8000;
```

Results Explain Describe Saved SQL History

DETAILS
ANIL101
SUNIL102
JAY103
KEYUR105
MAYUR106

5 rows returned in 0.00 seconds [CSV Export](#)

(8) Display the content of job details with minimum salary either 2000 or 4000.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select * from job
where min_sal in(2000,4000);
```

Results Explain Describe Saved SQL History

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
IT_PROG	PROGRAMMER	4000	10000

1 rows returned in 0.00 seconds [CSV Export](#)

To study various options of LIKE predicate

(1) Display all employee whose name start with 'A' and third character is 'a'.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select * from employee
where emp_name like 'A_a%';
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

3 rows returned in 0.01 seconds [CSV Export](#)

- (2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select emp_name,emp_no,emp_sal from employee
where emp_name like 'Ani__';
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_NO	EMP_SAL
Anita	105	5000

1 rows returned in 0.00 seconds [CSV Export](#)

- (3) Display all information of employee whose second character of name is either 'M' or 'N'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select * from employee
where emp_name like '_M%' or emp_name like '_N%';
```

Results Explain Describe Saved SQL History

no data found

- (4) Find the list of all customer name whose branch is in 'andheri' or 'dadar' or 'virar'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select cname from deposit_1
where bname='ANDHERI' or bname='DADAR' or bname='VIRAR';
```

Results Explain Describe Saved SQL History

CNAME
ANIL
SUNIL
VIJAY
KEYUR

4 rows returned in 0.00 seconds [CSV Export](#)

- (5) Display the job name whose first three character in job id field is 'FI_'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select job_title from job
where job_id like 'FI\_%' escape '\';
```

Results Explain Describe Saved SQL History

JOB_TITLE
FINANCE MANAGER
ACCOUNT

2 rows returned in 0.00 seconds [CSV Export](#)

- (6) Display the title/name of job who's last three character are '_MGR' and their maximum salary is greater than Rs 12000.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select job_title from job
where job_id like '%\_MGR' escape '\' and max_sal>12000;
```

Results Explain Describe Saved SQL History

JOB_TITLE
MARKETING MANAGER

1 rows returned in 0.00 seconds [CSV Export](#)

- (7) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5-character long.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select emp_name,emp_comm,manager_id from employee
where emp_name like '_n____%' and emp_comm is not NULL and manager_id is not NULL;
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_COMM	MANAGER_ID
Anita	50000	107
Sneha	24500	105

2 rows returned in 0.00 seconds [CSV Export](#)

- (8) Display the null values of employee and also employee name's third character should be 'a'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp name, emp comm, manager id from employee
where emp name like '__a%' and (emp comm is NULL or manager id is NULL);
```

Results Explain Describe Saved SQL History

EMP_NAME	EMP_COMM	MANAGER_ID
Aman	-	12
Anamika	-	-

2 rows returned in 0.00 seconds [CSV Export](#)

- (9) What will be output if you are giving LIKE predicate as '%_%' ESCAPE '\'

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select * from job
where job_id like '%\_%' ESCAPE '\';
```

Results Explain Describe Saved SQL History

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
IT_PROG	PROGRAMMER	4000	10000
MK_MGR	MARKETING MANAGER	9000	15000
FI_MGR	FINANCE MANAGER	8200	12000
FI_ACC	ACCOUNT	4200	9000
COMP_OP	COMPUTER OPERATOR	1500	3000

5 rows returned in 0.00 seconds [CSV Export](#)

Conclusion:

Through this practical we learnt about the Like predicate.

Practical-5

Aim: To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

Program:

(1) List total deposit from deposit.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select sum(amount) from deposit;
```

Results Explain Describe Saved SQL History

SUM(AMOUNT)
28700

1 rows returned in 0.00 seconds [CSV Export](#)

(2) List total loan from karolbagh branch

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select sum(amount) from deposit
WHERE bname='KAROLBAGH';
```

Results Explain Describe Saved SQL History

SUM(AMOUNT)
3500

1 rows returned in 0.00 seconds [CSV Export](#)

(3) Give maximum loan from branch vrce.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select max(amount) from deposit  
WHERE BNAME='VRCE';
```

Results Explain Describe Saved SQL History

MAX(AMOUNT)
1000

1 rows returned in 0.00 seconds [CSV Export](#)

(4) Count total number of customers

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select count(cname) from customers;
```

Results Explain Describe Saved SQL History

COUNT(CNAME)
10

1 rows returned in 0.00 seconds [CSV Export](#)

(5) Count total number of customer's cities.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select count(city) from customers;
```

Results Explain Describe Saved SQL History

COUNT(CITY)
10

1 rows returned in 0.00 seconds [CSV Export](#)

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select count(city) from customers;  
select count(distinct city) from customers;
```

Results Explain Describe Saved SQL History

COUNT(DISTINCTCITY)
7

1 rows returned in 0.01 seconds [CSV Export](#)

(6) Create table supplier from employee with all the columns.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
CREATE TABLE supplier as(select * from employee);
select * from supplier;
```

Results Explain Describe Saved SQL History

Table created.

0.06 seconds

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
CREATE TABLE supplier as(select * from employee);
select * from supplier;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

7 rows returned in 0.00 seconds [CSV Export](#)

(7) Create table sup1 from employee with first two columns.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
CREATE TABLE sup1 as(select emp_no,emp_name from employee);  
select * from sup1;
```

Results Explain Describe Saved SQL History

Table created.

0.01 seconds

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
CREATE TABLE sup1 as(select emp_no,emp_name from employee);  
select * from sup1;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME
101	Smith
102	Snehal
103	Adama
104	Aman
105	Anita
106	Sneha
107	Anamika

7 rows returned in 0.02 seconds

[CSV Export](#)

(8) Create table sup2 from employee with no data

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
CREATE TABLE sup2 as select * from employee where 1=2;
select * from sup2;
```

Results Explain Describe Saved SQL History

Table created.

0.00 seconds

(9) Insert the data into sup2 from employee whose second character should be 'n' and string should be 5 characters long in employee name field.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
insert into sup2 (select * from employee where emp_name like 'n____' and length(emp_name)=5 );
select * from sup2;
```

Results Explain Describe Saved SQL History

2 row(s) inserted.

0.01 seconds

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
insert into sup2 (select * from employee where emp_name like 'n__' and length(emp_name)=5 );
select * from sup2;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds [CSV Export](#)

(10) Delete all the rows from sup1.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
delete from sup1;
```

~~delete from supplier where emp_no='103';~~

Results Explain Describe Saved SQL History

7 row(s) deleted.

(11) Delete the detail of supplier whose sup_no is 103.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
delete from supplier where emp_no='103';
select * from supplier;
```

Results Explain Describe Saved SQL History

1 row(s) deleted.

0.00 seconds

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
delete from supplier where emp_no='103';
select * from supplier;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	Joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

6 rows returned in 0.00 seconds [CSV Export](#)

(12) Rename the table sup2.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
rename sup2 to supp2;
desc supp2;
```

Results Explain Describe Saved SQL History

Statement processed.

0.00 seconds

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
rename sup2 to supp2;
desc supp2;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)
Object Type **TABLE** Object **SUPP2**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
SUPP2	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-
	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	LOCATION	Varchar2	15	-	-	-	✓	-	-
	MANAGER_ID	Number	-	5	0	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-

1 - 11

(13) Destroy table sup1 with all the data.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
Drop table sup1;
Desc sup1;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

Table dropped.

0.02 seconds

- (14) Update the value dept_no to 10 where second character of emp. name is 'm'.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
update employee set dept_no='10' where emp_name='_m%';  
select * from employee;
```

Results Explain Describe Saved SQL History

0 row(s) updated.

0.00 seconds

- (15) Update the value of employee name whose employee number is 103.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
update employee set emp_name='Jack' where emp_no=103;  
select * from employee;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.00 seconds

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
update employee set emp_name='Jack' where emp_no=103;
select * from employee;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
103	Jack	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97

7 rows returned in 0.00 seconds

[CSV Export](#)

(16) Add one column phone to employee with size of column is 10.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
alter table employee add PHONE number(10);
desc employee;
```

Results Explain Describe Saved SQL History

Table altered.

0.00 seconds

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
alter table employee add PHONE number(10);
desc employee;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-
	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	LOCATION	Varchar2	15	-	-	-	✓	-	-
	MANAGER_ID	Number	-	5	0	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
	PHONE	Number	-	10	0	-	✓	-	-

1 - 12

(17) Modify the column emp_name to hold maximum of 30 characters.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
alter table employee modify emp_name varchar2(30);
desc employee;
```

Results Explain Describe Saved SQL History

Table altered.

0.02 seconds

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
alter table employee modify emp_name varchar2(30);
desc employee;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **EMPLOYEE**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-
	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	LOCATION	Varchar2	15	-	-	-	✓	-	-
	MANAGER_ID	Number	-	5	0	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
	PHONE	Number	-	10	0	-	✓	-	-

1 - 12

- (18) Count the total no as well as distinct rows in dept_no column with a condition of salary greater than 1000 of employee

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select count(*) from employee where emp_sal>1000;
select count(distinct " " dept no) from employee where emp_sal>1000;
```

Results Explain Describe Saved SQL History

COUNT(*)
6

1 rows returned in 0.00 seconds [CSV Export](#)

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select count(*) from employee where emp_sal>1000;
select count(distinct dept_name) from employee where emp_sal>1000;
```

Results Explain Describe Saved SQL History

COUNT(DISTINCTDEPT_NAME)
5

1 rows returned in 0.00 seconds [CSV Export](#)

- (19) Display the detail of all employees in ascending order, descending order of their name and no.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select * from employee order by emp_name asc,emp_no desc
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
103	Jack	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-

7 rows returned in 0.00 seconds [CSV Export](#)

- (20) Display the dept_no in ascending order and accordingly display emp_comm in descending order.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select dept_no,emp_comm from employee order by emp_comm desc,dept_no asc;
select dept_no,emp_comm from employee order by emp_comm desc;
```

Results Explain Describe Saved SQL History

DEPT_NO	EMP_COMM
15	-
20	-
30	-
10	50000
10	24500
25	300
20	0

7 rows returned in 0.00 seconds

[CSV Export](#)

21) Update the value of emp_comm to 500 where dept_no is 20.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
update employee set emp_comm=500 where dept_no=20 ;
select * from employee;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE
101	Smith	800	500	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	Jack	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
106	Sneha	24500	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-

7 rows returned in 0.00 seconds

[CSV Export](#)

(22) Display the emp_comm in ascending order with null value first and accordingly sort employee salary in descending order.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_comm,emp_sal from employee order by emp_comm asc nulls first,emp_sal desc;
```

Results Explain Describe Saved SQL History

EMP_COMM	EMP_SAL
-	3000
-	2975
300	1600
500	1100
500	800
24500	24500
50000	5000

7 rows returned in 0.00 seconds [CSV Export](#)

(23) Display the emp_comm in ascending order with null value last and accordingly sort emp_no in descending order.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_comm,emp_no from employee order by emp_comm asc nulls last,emp_no desc;
```

Results Explain Describe Saved SQL History

EMP_COMM	EMP_NO
300	102
500	103
500	101
24500	106
50000	105
-	107
-	104

7 rows returned in 0.00 seconds [CSV Export](#)

Conclusion:

In this practical we learnt about various data manipulation commands, aggregate functions and sorting concept on all created tables.

Practical-6

Aim: To study Single-row functions.

Program:

1. Write a query to display the current date. Label the column Date.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select sysdate as "date" from dual
```

Results Explain Describe Saved SQL History

Date
12-FEB-22

1 rows returned in 0.00 seconds [CSV Export](#)

2. For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select job_id,l_name,emp_sal,round(emp_sal+(emp_sal*15/100)) "new salary" from employee;
```

Results Explain Describe Saved SQL History

JOB_ID	L_NAME	EMP_SAL	New Salary
fi_mgr	shah	800	920
lec	gupta	1600	1840
mk_mgr	wales	1100	1265
comp_op	sharma	3000	3450
comp_op	patel	5000	5750
fi_acc	joseph	24500	28175
it_prog	jha	2975	3421

7 rows returned in 0.00 seconds [CSV Export](#)

3. Modify your query no (2) to add a column that subtracts the old salary from the new salary. Label the column Increase.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select job_id,l_name,emp_sal,round(emp_sal+(emp_sal*15/100)) "new salary",round(emp_sal+(emp_sal*15/100))-emp_sal "increase" from employee
```

Results Explain Describe Saved SQL History

JOB_ID	L_NAME	EMP_SAL	New Salary	Increase
fi_mgr	shah	800	920	120
lec	gupta	1600	1840	240
mk_mgr	wales	1100	1265	165
comp_op	sharma	3000	3450	450
comp_op	patel	5000	5750	750
fi_acc	joseph	24500	28175	3675
it_prog	jha	2975	3421	446

7 rows returned in 0.00 seconds [CSV Export](#)

4. Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save

```
select initcap(emp_name) "NAME", length(emp_name) "LENGTH" from employee where emp_name like 'J%' or emp_name like 'A%' or emp_name like 'M%' order by l_name;
```

Results Explain Describe Saved SQL History

NAME	LENGTH
Anamika	7
Anita	5
Aman	4
Jack	4

4 rows returned in 0.00 seconds [CSV Export](#)

5. Write a query that produces the following for each employee:
<employee last name> earns <salary> monthly

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select emp_name || ' EARNs ' || emp_sal || ' MONTHLY ' as "SENTENCE" from employee
```

Results Explain Describe Saved SQL History

SENTENCE
Smith EARNs 800 MONTHLY
Snehal EARNs 1600 MONTHLY
Jack EARNs 1100 MONTHLY
Aman EARNs 3000 MONTHLY
Anita EARNs 5000 MONTHLY
Sneha EARNs 24500 MONTHLY
Anamika EARNs 2975 MONTHLY

7 rows returned in 0.00 seconds [CSV Export](#)

6. Display the name, date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼ Save Run

```
select emp_name,hiredate,round(months_between(sysdate,hiredate)) as months, to_char(hiredate,'DAY') as "DAY" from employee order by to_char(hiredate,'DAY') desc;
```

Results Explain Describe Saved SQL History

EMP_NAME	HIREDATE	MONTHS	DAY
Anamika	15-JUL-97	295	TUESDAY
Jack	30-NOV-95	314	THURSDAY
Snehal	14-MAR-96	311	THURSDAY
Aman	02-OCT-97	292	THURSDAY
Anita	01-JAN-98	289	THURSDAY
Sneha	26-SEP-97	293	FRIDAY
Smith	09-AUG-96	306	FRIDAY

7 rows returned in 0.00 seconds [CSV Export](#)

7. Display the date of emp in a format that appears as Seventh of June 1994 12:00:00 AM.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select l_name,hiredate,to_char(hiredate,'DDSPTH "OF" MONTH YYYY HH:MM:SS AM') "DATE OF EMP" from employee;
```

Results Explain Describe Saved SQL History

L_NAME	HIREDATE	DATE OF EMP
shah	09-AUG-96	NINTH OF AUGUST 1996 12:08:00 AM
gupta	14-MAR-96	FOURTEENTH OF MARCH 1996 12:03:00 AM
wales	30-NOV-95	THIRTIETH OF NOVEMBER 1995 12:11:00 AM
sharma	02-OCT-97	SECOND OF OCTOBER 1997 12:10:00 AM
patel	01-JAN-98	FIRST OF JANUARY 1998 12:01:00 AM
joseph	26-SEP-97	TWENTY-SIXTH OF SEPTEMBER 1997 12:09:00 AM
jha	15-JUL-97	FIFTEENTH OF JULY 1997 12:07:00 AM

7 rows returned in 0.00 seconds

[CSV Export](#)

8. Write a query to calculate the annual compensation of all employees (sal+comm.).

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select sum(emp_sal+ emp_comm) as "COMPOSITION" FROM EMPLOYEE
```

Results Explain Describe Saved SQL History

COMPOSITION
108800

1 rows returned in 0.00 seconds

[CSV Export](#)

Conclusion:

We learnt some single row function in this practical.

Practical-7

Aim: Displaying data from Multiple Tables (join)

Program:

1. Give details of customers ANIL.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select * from customers full join borrow on customers.cname=borrow.cname where customers.cname='ANIL';
```

Results Explain Describe Saved SQL History

CNAME	CITY	LOANNO	CNAME	BNAME	AMOUNT
ANIL	CALCUTTA	201	ANIL	VRCE	1000

1 rows returned in 0.01 seconds [CSV Export](#)

2. Give name of customer who are borrowers and depositors and having living city Nagpur

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select * from deposit full join borrow on borrow.cname=deposit.cname where deposit.bname='NAGPUR';
```

Results Explain Describe Saved SQL History

no data found

3. Give city as their city name of customers having same living branch.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select branch.city as "CITY" from branch join deposit on deposit.bname=branch.bname join borrow on borrow.bname=branch.bname
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

CITY
NAGPUR
NAGPUR
BOMBAY
BOMBAY

4 rows returned in 0.00 seconds

[CSV Export](#)

4. Write a query to display the last name, department number, and department name for all employees.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select l name,dept no,dept name from employee
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)

L_NAME	DEPT_NO	DEPT_NAME
shah	20	machine learning
gupta	25	data science
wales	20	machine learning
sharma	15	virtual reality
patel	10	big data analytics
joseph	10	big data analytics
jha	30	artificial intelligence

7 rows returned in 0.00 seconds

[CSV Export](#)

5. Create a unique listing of all jobs that are in department 30. Include the location of the department in the output

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select job.job_title,employee.location from job full join employee on job.job_id=employee.job_id where dept no=30;
```

Results Explain Describe Saved SQL History

JOB_TITLE	LOCATION
-	new york

1 rows returned in 0.00 seconds [CSV Export](#)

6. Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select emp name,dept no,dept name from employee where location='new york'
```

Results Explain Describe Saved SQL History

EMP_NAME	DEPT_NO	DEPT_NAME
Anamika	30	artificial intelligence

1 rows returned in 0.00 seconds [CSV Export](#)

7. Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select l_name "Employee", emp_no "Emp#", manager_id "Mgr#" from employee;
```

Results Explain Describe Saved SQL History

Employee	Emp#	Mgr#
shah	101	105
gupta	102	-
wales	103	105
sharma	104	12
patel	105	107
joseph	106	105
jha	107	-

7 rows returned in 0.00 seconds

[CSV Export](#)

8. Create a query to display the name and hire date of any employee hired after employee “smith”.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
select emp_name, hiredate from employee where hiredate > '09-AUG-96'
```

Results Explain Describe Saved SQL History

EMP_NAME	HIREDATE
Aman	02-OCT-97
Anita	01-JAN-98
Sneha	26-SEP-97
Anamika	15-JUL-97

4 rows returned in 0.00 seconds

[CSV Export](#)

Conclusion:

We learnt some operations that can be performed on table using different types of join query.

Practical-8

Aim: To apply the concept of Aggregating Data using Group functions.

Program:

- (1) List total deposit of customer having account date after 1-jan-96.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select sum(amount) as "TOTAL AFTER 1-JAN-96" from deposit where adate>'1-jan-96'
```

Results Explain Describe Saved SQL History

TOTAL AFTER 1-JAN-96
10000

1 rows returned in 0.00 seconds [CSV Export](#)

- (2) List total deposit of customers living in city Nagpur.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼ Save

```
select sum(amount) as "TOTAL OF CUSTOMERS IN NAGPUR" from deposit left join customers on customers.cname=deposit.cname where customers.city='NAGPUR'
```

Results Explain Describe Saved SQL History

TOTAL OF CUSTOMERS IN NAGPUR
1200

1 rows returned in 0.00 seconds [CSV Export](#)

- (3) List maximum deposit of customers living in bombay.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save

```
select max(amount) as "MAX OF CUSTOMERS IN BOMBAY" from deposit left join customers on customers.cname=deposit.cname where customers.city='BOMBAY'
```

Results Explain Describe Saved SQL History

MAX OF CUSTOMERS IN BOMBAY
5000

1 rows returned in 0.00 seconds [CSV Export](#)

- (4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save

```
select round(max(emp_sal)) as "MAXIMUM", round(min(emp_sal)) as "MINIMUM", round(sum(emp_sal)) as "SUM", round(avg(emp_sal)) as "AVERAGE" from employee;
```

Results Explain Describe Saved SQL History

MAXIMUM	MINIMUM	SUM	AVERAGE
24500	800	38975	5568

1 rows returned in 0.00 seconds [CSV Export](#)

- (5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save

```
select max(emp_sal)-min(emp_sal) as "DIFFERENCE" from employee
```

Results Explain Describe Saved SQL History

DIFFERENCE
23700

1 rows returned in 0.01 seconds [CSV Export](#)

- (6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
select count(emp_name), to_char(hiredate,'YY') as "YEAR" from employee where hiredate between '1-jan-95' and '31-dec-98' group by to_char(hiredate,'YY') order by to_char(hiredate,'YY') Asc;
```

Results Explain Describe Saved SQL History

COUNT(EMP_NAME)	YEAR
1	95
2	96
3	97
1	98

4 rows returned in 0.00 seconds [CSV Export](#)

- (7) Find the average salaries for each department without displaying the respective department numbers.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
select dept_name,avg(emp_sal) as "AVG SAL" from employee group by dept_name;
```

Results Explain Describe Saved SQL History

DEPT_NAME	AVG SAL
big data analytics	14750
artificial intelligence	2975
machine learning	950
virtual reality	3000
data science	1600

5 rows returned in 0.00 seconds [CSV Export](#)

- (8) Write a query to display the total salary being paid to each job title, within each department.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select dept_name,sum(emp_sal) as "TOTAL SALARY" from employee group by dept_name;
```

Results Explain Describe Saved SQL History

DEPT_NAME	TOTAL SALARY
big data analytics	29500
artificial intelligence	2975
machine learning	1900
virtual reality	3000
data science	1600

5 rows returned in 0.00 seconds

[CSV Export](#)

- (9) Find the average salaries > 2000 for each department without displaying the respective department numbers.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
select dept_name,avg(emp_sal) from employee group by dept_name having avg(emp_sal)>2000;
```

Results Explain Describe Saved SQL History

DEPT_NAME	AVG(EMP_SAL)
big data analytics	14750
artificial intelligence	2975
virtual reality	3000

3 rows returned in 0.00 seconds

[CSV Export](#)

- (10) Display the job and total salary for each job with a total salary amount exceeding 3000 and sorts the list by the total salary.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select job_id,sum(emp_sal) "Total Salary" from employee group by job_id having sum(emp_sal)>=3000;
```

Results Explain Describe Saved SQL History

JOB_ID	Total Salary
comp_op	8000

1 rows returned in 0.00 seconds [CSV Export](#)

- (11) List the branches having sum of deposit more than 5000 and located in city bombay.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
select bname from deposit natural join branch where branch.city='BOMBAY' group by bname having sum(amount)>5000;
```

Results Explain Describe Saved SQL History

BNAME
POWAI

1 rows returned in 0.00 seconds [CSV Export](#)

Conclusion: We learnt how to apply the concept of aggregating data using group function.

Practical-9

Aim: To solve queries using the concept of sub query.

Program:

- (1) Write a query to display the last name and hire date of any employee in the same department as smith. Exclude smith

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
SELECT L_NAME , HIREDATE FROM EMPLOYEE WHERE DEPT_NAME = (SELECT DEPT_NAME FROM EMPLOYEE WHERE EMP_NAME = 'Smith') AND EMP_NAME <> 'Smith';
```

Results Explain Describe Saved SQL History

L_NAME	HIREDATE
wales	30-NOV-95

1 rows returned in 0.02 seconds [CSV Export](#)

- (2) Give name of customers who are depositors having same branch city of mr. sunil.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN BRANCH ON BRANCH.BNAME=DEPOSIT.BNAME WHERE BRANCH.CITY='NAGPUR';
```

Results Explain Describe Saved SQL History

CNAME
ANIL
SUNIL

2 rows returned in 0.00 seconds [CSV Export](#)

- (3) Give deposit details and loan details of customer in same city where pramod is living.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN CUSTOMERS ON CUSTOMERS.CNAME=DEPOSIT.CNAME WHERE CUSTOMERS.CITY=(SELECT CITY FROM CUSTOMERS WHERE CNAME='PRAMOD') AND DEPOSIT.CNAME<>'PRMOD';
```

Results Explain Describe Saved SQL History

CNAME
MADHURI

1 rows returned in 0.01 seconds [CSV Export](#)

- (4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
SELECT EMP_NO , L_NAME FROM EMPLOYEE WHERE EMP_SAL > (SELECT AVG(EMP_SAL) FROM EMPLOYEE);
```

Results Explain Describe Saved SQL History

EMP_NO	L_NAME
104	sharma
105	patel
106	joseph
107	jha

4 rows returned in 0.00 seconds [CSV Export](#)

- (5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 Save Run

```
SELECT DEPOSIT.CNAME FROM DEPOSIT JOIN CUSTOMERS ON CUSTOMERS.CNAME=DEPOSIT.CNAME WHERE CUSTOMERS.CITY=(SELECT CITY FROM CUSTOMERS WHERE CNAME='ANIL') AND DEPOSIT.AMOUNT>2000;
```

Results Explain Describe Saved SQL History

no data found

(6) Display the last name and salary of every employee who reports to ford.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT L_NAME,EMP_SAL FROM EMPLOYEE WHERE EMP_NAME='FORD';
```

Results Explain Describe Saved SQL History

no data found

(7) Display the department number, name, and job for every employee in the Accounting department.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT DEPT_NO,EMP_NAME,JOB_ID FROM EMPLOYEE WHERE DEPT_NAME='ACCOUNTING';
```

Results Explain Describe Saved SQL History

no data found

(8) List the name of branch having highest number of depositors.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
SELECT BNAME,COUNT(CNAME) AS "DEPOSITORS" FROM DEPOSIT GROUP BY BNAME HAVING COUNT(CNAME)>=COUNT(CNAME);
```

Results Explain Describe Saved SQL History

BNAME	DEPOSITORS
VRCE	1
AJNI	1
KAROLBAGH	1
M.G.ROAD	1
VIRAR	1
POWAI	1
CHANDI	1
ANDHERI	1
NEHRUPACE	1

9 rows returned in 0.00 seconds [CSV Export](#)

(9) Give the name of cities where in which the maximum numbers of branches are located.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
SELECT CITY FROM BRANCH WHERE BNAME=(SELECT MAX(BNAME) FROM BRANCH);
```

Results Explain Describe Saved SQL History

CITY
NAGPUR

1 rows returned in 0.00 seconds [CSV Export](#)

- (10) Give name of customers living in same city where maximum depositors are located.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
SELECT CNAME FROM CUSTOMERS WHERE CITY=(SELECT MAX(CITY) FROM BRANCH);
```

Results Explain Describe Saved SQL History

CNAME
MADHURI
PRAMOD

2 rows returned in 0.00 seconds [CSV Export](#)

Conclusion:

We learnt to solve queries using concept of sub queries.

Practical-10

Aim: Manipulating Data

Program:

(1) Give 10% interest to all depositors.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
alter table deposit add interest number(10);
update deposit set interest=amount+(amount*(0.1))
select * from deposit
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	1000	01-MAR-95	1100
101	SUNIL	AJNI	5000	04-JAN-96	5500
102	MEHUL	KAROLBAGH	3500	17-NOV-95	3850
104	MADHURI	CHANDI	1200	17-DEC-95	1320
105	PRMOD	M.G.ROAD	3000	27-MAR-96	3300
106	SANDIP	ANDHERI	2000	31-MAR-96	2200
107	SHIVANI	VIRAR	1000	05-SEP-95	1100
108	KRANTI	NEHRUPLACE	5000	02-JUL-95	5500
109	MINU	POWAI	7000	10-AUG-95	7700

9 rows returned in 0.00 seconds

[CSV Export](#)

(2) Give 10% interest to all depositors having branch vrce

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
alter table deposit add interest number(10);
update deposit set interest=amount+(amount*(0.1)) where bname='VRCE'
select * from deposit
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	1000	01-MAR-95	1100
101	SUNIL	AJNI	5000	04-JAN-96	5500
102	MEHUL	KAROLBAGH	3500	17-NOV-95	3850
104	MADHURI	CHANDI	1200	17-DEC-95	1320
105	PRMOD	M.G.ROAD	3000	27-MAR-96	3300
106	SANDIP	ANDHERI	2000	31-MAR-96	2200
107	SHIVANI	VIRAR	1000	05-SEP-95	1100
108	KRANTI	NEHRUPLACE	5000	02-JUL-95	5500
109	MINU	POWAI	7000	10-AUG-95	7700

9 rows returned in 0.00 seconds

[CSV Export](#)

(3) Give 10% interest to all depositors living in nagpur and having branch city bombay.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

Save

Run

```
update deposit set interest=amount+(amount*10)/100 where cname in (select cname from customers where city='NAGPUR') and bname in (select bname from branch where city='BOMBAY')
```

Results Explain Describe Saved SQL History

0 row(s) updated.

0.00 seconds

- (4) Write a query which changes the department number of all employees with empno 7788's job to employee 7844's current department number.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
update employee set dept_no=(select dept_no from employee where emp_no='7844') where emp_no='7788';
```

Results Explain Describe Saved SQL History

0 row(s) updated.

0.02 seconds

- (5) Transfer 10 Rs from account of anil to sunil if both are having same branch.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
update deposit_1 set amount=amount-10 where cname='ANIL' and bname=(select bname from deposit_1 where cname='SUNIL');  
update deposit_1 set amount=amount+10 where cname='SUNIL' and bname=(select bname from deposit_1 where cname='ANIL');
```

Results Explain Describe Saved SQL History

0 row(s) updated.

0.00 seconds

(6) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
update deposit set amount=amount+100 where amount in (select max(amount) from deposit group by bname);
select * from deposit
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	1100	01-MAR-95	1100
101	SUNIL	AJNI	5100	04-JAN-96	5500
102	MEHUL	KAROLBAGH	3600	17-NOV-95	3850
104	MADHURI	CHANDI	1300	17-DEC-95	1320
105	PRMOD	M.G.ROAD	3100	27-MAR-96	3300
106	SANDIP	ANDHERI	2100	31-MAR-96	2200
107	SHIVANI	VIRAR	1100	05-SEP-95	1100
108	KRANTI	NEHRUPLACE	5100	02-JUL-95	5500
109	MINU	POWAI	7100	10-AUG-95	7700

9 rows returned in 0.02 seconds [CSV Export](#)

(7) Delete depositors of branches having number of customers between 1 to 3.

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10 ▼

```
delete deposit where bname in (select bname from deposit group by bname having count(cname) between '1' and '3')
```

Results Explain Describe Saved SQL History

9 row(s) deleted.

0.02 seconds

(8) Delete deposit of vijay.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
delete deposit where cname='VIJAY'
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

0.02 seconds

(9) Delete borrower of branches having average loan less than 1000.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
delete borrow where bname in (select bname from borrow group by bname having avg(amount)<1000)
```

Results Explain Describe Saved SQL History

0 row(s) deleted.

0.01 seconds

Conclusion:

We learnt manipulation of data using different techniques.

Practical-11

Aim: Add and remove constraint

Program:

(1) Add primary key constraint on job_id in job table.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
ALTER TABLE JOB1 ADD CONSTRAINT PK_JOB1 PRIMARY KEY(JOB_ID);
```

```
DESC JOB1;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **JOB1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOB1	JOB_ID	Varchar2	15	-	-	1	-	-	-
	JOB_TITLE	Varchar2	30	-	-	-	✓	-	-
	MIN_SAL	Number	-	7	2	-	✓	-	-
	MAX_SAL	Number	-	7	2	-	✓	-	-
1 - 4									

(2) Add foreign key constraint on employee table referencing job table.

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10 ▼

```
ALTER TABLE EMPLOYEE1 ADD CONSTRAINT FK_EMP FOREIGN KEY(JOB_ID) REFERENCES JOB(JOB_ID);
```

```
DESC EMPLOYEE1;
```

Results Explain Describe Saved SQL History

Object Type **TABLE** Object **EMPLOYEE1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE1	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-
	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	LOCATION	Varchar2	15	-	-	-	✓	-	-
	MANAGER_ID	Number	-	5	0	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
1 - 11									

(3) Add composite primary key on lock table (lock table does not exist, while creating table add composite key)

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
CREATE TABLE LOCK1(L_ID NUMBER, L_NAME VARCHAR(30), L_ADDRESS VARCHAR(50), AGE NUMBER, CONSTRAINT PK_LOCK1 PRIMARY KEY(L_ID,L_NAME));
DESC LOCK1;
```

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **LOCK1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
LOCK1	L_ID	Number	-	-	-	1	-	-	-
	L_NAME	Varchar2	30	-	-	2	-	-	-
	L_ADDRESS	Varchar2	50	-	-	-	✓	-	-
	AGE	Number	-	-	-	-	✓	-	-
1 - 4									

(4) Remove primary key constraint on job_id

User: 20DCE008

Home > SQL > **SQL Commands**

☒ Autocommit Display 10

```
DESC LOCK1;
```

```
ALTER TABLE JOB1 DROP CONSTRAINT PK_JOB1;
DESC JOB1;
```

Results Explain **Describe** Saved SQL History

Object Type **TABLE** Object **JOB1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOB1	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	JOB_TITLE	Varchar2	30	-	-	-	✓	-	-
	MIN_SAL	Number	-	7	2	-	✓	-	-
	MAX_SAL	Number	-	7	2	-	✓	-	-
1 - 4									

(5) Remove foreign key constraint on employee table

User: 20DCE008

Home > SQL > SQL Commands

☒ Autocommit Display 10

```
ALTER TABLE EMPLOYEE1 DROP CONSTRAINT FK_EMP;
DESC EMPLOYEE1;
```

[Results](#) [Explain](#) [Describe](#) [Saved SQL](#) [History](#)
Object Type **TABLE** Object **EMPLOYEE1**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE1	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-	-	-	✓	-	-
	JOB_ID	Varchar2	15	-	-	-	✓	-	-
	LOCATION	Varchar2	15	-	-	-	✓	-	-
	MANAGER_ID	Number	-	5	0	-	✓	-	-
	HIREDATE	Date	7	-	-	-	✓	-	-
1 - 11									

Conclusion:

We learnt how to add and remove constraint .

Practical-12

Aim:

Data Dictionary and E-R Diagram

Suppose that as the database administrator (DBA) in a hotel, you have to set up a database to capture all the following information that the hotel needs to maintain.

- The hotel offers three types of ROOMS, including single room, double room, and triple room.
- Every room is Identified by its unique number.
- Every employee at the hotel is either a receptionist, a cleaning staff, or a kitchen staff. Each RECEPTIONIST is identified with her/his name, employee number and years of experience. Receptionists are responsible for ensuring the room is clean before the room is assigned to the guest. Thus, they assign a single CLEANING STAFF to clean each room every morning and/or whenever it is required. Note that the same room may need to be cleaned several times on the same day, before it gets reassigned. For each cleaning assignment, the date and the status need to be provided. The KITCHEN STAFF is characterized by their specific responsibilities, e.g. being a cook or a waiter. The cleaning staff and the kitchen staff are also uniquely identified by their employee number.
- Receptionists welcome GUESTS and upon presentation of their valid traveling documents, they allocate a unique room to each guest and specify one group of facilities which is accessible to the guest during his stay. Guests are uniquely identified with their passport number but other necessary information are also recorded about the guests, including: name, phone numbers, arrival date, departure date, and credit card number. Each FACILITY GROUP contains specific set of facilities, e.g. the bar or gym, in order to be used by the guests. The arrival and departure dates of a guest will in turn determine the occupation of a specific room.
- A guest can be accompanied with one person to have a double room or at most two people for a triple room. Each ACCOMPANYING person is identified by his/her name.

Ans:

1. Design Data Dictionary for above problem.

No.	Field Name	Data Type	Size	Detail	Example	Constraints
1						
2						

2. Considering the descriptions given above, draw an ER diagram for the database, representing entities, attributes, and relationships. Hint: Pay attention to clear identification of different kinds of attributes (e.g. multi-valued, derived, and Primary key), the total participation for the relationship sets and generalization (or specialization) of entities.