CHAROTAR UNIVERSITY OF SCIENCE & TECHNOLOGY

DEVANG PATEL INSTITUE OF ADVANCE TECHNOLOGY AND RESEARCH

NAME: Parth N Patel

ID: 19DCS098

SUBJECT: Database Management System

SUBJECT CODE: CE246

SEM: 4

PRACTICAL-1

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

File System

- A file processing system(fps) is a technique of arranging the files in a storage medium like a hard disk, pen drive, DVD, etc. It helps you to organizes the data and allows easy retrieval of files when they are required. It mostly consists of different types of files like mp3, mp4, txt, doc, etc. that are grouped into directories.
- A file system enables you to handle the way of reading and writing data to the storage medium. It is directly installed into the computer with the Operating systems such as Windows and Linux.

Example:

NTFS (New Technology File System), EXT (Extended File System).

Features of a File system:

- It helps you to store data in a group of files.
- Files data are dependent on each other.
- C/C++ and COBOL languages were used to design the files.
- Shared File System Support
- Fast File System Recovery.

Advantages of File system:

- Enforcement of development and maintenance standards.
- Helps you to reduce redundancy
- Avoid inconsistency across file maintenance to get the integrity of data independence.
- Firm theoretical foundation (for the relational model).
- It is more efficient and cost less than a DBMS in certain situations.
- The design of file processing is simpler than designing Database.

Disadvantages of File Processing system:

- Each application has its data file so, the same data may have to be recorded and stored many times.
- Data dependence in the file processing system are data-dependent, but, the problem is incompatible with file format.
- The problem with security.
- Time-consuming.
- It allows you to maintain the record of the big firm having a large number of items.
- Required lots of labour work to do.

Application of File system:

- Language-specific run-time libraries
- API programs using it to make requests of the file system
- It is used for data transfer and positioning.
- Helps you to update the metadata
- Managing directories.

DBMS (Database Management System):

- Database Management System is basically a software that manages the collection of related data. It is used for storing data and retrieving the data effectively when it is needed. It also provides proper security measures for protecting the data from unauthorized access. In Database Management System the data can be fetched by SQL queries and relational algebra. It also provides mechanisms for data recovery and data backup.

Example:

Oracle, MySQL, MS SQL server.

Features of DBMS:

- A user-accessible catalog of data
- Transaction support
- Concurrency control with Recovery services
- Authorization services
- The value of data is the same at all places.

- Offers support for data communication
- Independent utility services
- Allows multiple users to share a file at the same time

Advantages of DBMS:

- DBMS offers a variety of techniques to store & retrieve data
- Uniform administration procedures for data
- Application programmers never exposed to details of data representation and Storage.
- A DBMS uses various powerful functions to store and retrieve data efficiently.
- Offers Data Integrity and Security
- Reduced Application Development Time
- Consume lesser space
- Reduction of redundancy.
- Data independence.

Disadvantages of the DBMS:

- Cost of Hardware and Software of a DBMS is quite high, which increases the budget of your organization.
- Most database management systems are often complex systems, so the training for users to use the DBMS is required.
- Data-sets begins to grow large as it provides a more predictable query response time.
- It required a processor with the high speed of data processing.
- The database can fail because or power failure or the whole system stops.

Application of the DBMS:

- Admission System Examination System Library System
- Payroll & Personnel Management System
- Accounting System Hotel Reservation System Airline Reservation System
- DBMS system also used by universities to keep call records, monthly bills, maintaining balances, etc.
- Finance for storing information about stock, sales, and purchases of financial instruments like stocks and bonds.

KEY DIFFERENCES BETWEEN FPS & DBMS:

- A file system is a software that manages and organizes the files in a storage medium, whereas DBMS is a software application that is used for accessing, creating, and managing databases.
- The file system doesn't have a crash recovery mechanism on the other hand, DBMS provides a crash recovery mechanism.
- Data inconsistency is higher in the file system. On the contrary Data inconsistency is low in a database management system.
- File system does not offer concurrency, whereas DBMS provides a concurrency facility.

RDBMS (Relational Database Management System):

- A relational database management system (RDBMS) is a program that allows you to create, update, and administer a relational database. Most relational database management systems use the SQL language to access the database.
- RDMBS adds the R of relational to the existing Database management technology. Created in the 1970s, RDBMS was designed to be a more sophisticated version of DBMS. RDBMS also adds a degree of finesse for the organization or the individuals accessing the data stored in the database.
- One key feature of RDBMS is that it can only keep the tabular form of data. Data in RDBMS is stored and sorted in the form of rows, columns (also called tuples and attribute in the DBMS language).

Example:

MySQL, PostgreSQL, Db2

Features of RDBMS:

- All data stored in the tables are provided by an RDBMS
- Ensures that all data stored are in the form of rows and columns

- Facilitates primary key, which helps in unique identification of the rows
- Facilitates a common column to be shared amid two or more tables
- Multi-user accessibility is facilitated to be controlled by individual users.

Advantages of RDBMS:

- It is secured in nature.
- The data manipulation can be done.
- It limits redundancy and replication of the data.
- It offers better data integrity.
- It provides better physical data independence.

Disadvantages of RDBMS:

- Software is expensive.
- It requires skilled human resources to implement.
- It is difficult to recover the lost data.
- Complex software refers to expensive hardware and hence increases overall cost to avail the RDBMS service.

DDBMS (Distributed Database Management System):

- Distributed Database Management System (DDBMS) is a type of DBMS which manages a number of databases hoisted at diversified locations and interconnected through a computer network. It provides mechanisms so that the distribution remains oblivious to the users, who perceive the database as a single database.

Features of DDBMS:

- It is used to create, retrieve, update and delete distributed databases.
- It synchronizes the database periodically and provides access mechanisms by the virtue of which the distribution becomes transparent to the users.
- It is used in application areas where large volumes of data are processed and accessed by numerous users simultaneously.
- It is designed for heterogeneous database platforms.
- It maintains confidentiality and data integrity of the databases.

Advantages of DDBMS:

- Reflects organizational structure
- Improved share ability
- Improved availability
- Improved reliability
- Improved performance

Disadvantages of DDBMS:

- Increased Cost
- Integrity control more difficult,
- Lack of standards,
- Database design more complex.
- Complexity of management and control. Applications must recognize data location and they must be able to stitch together data from various sites.

CONCLUSION:

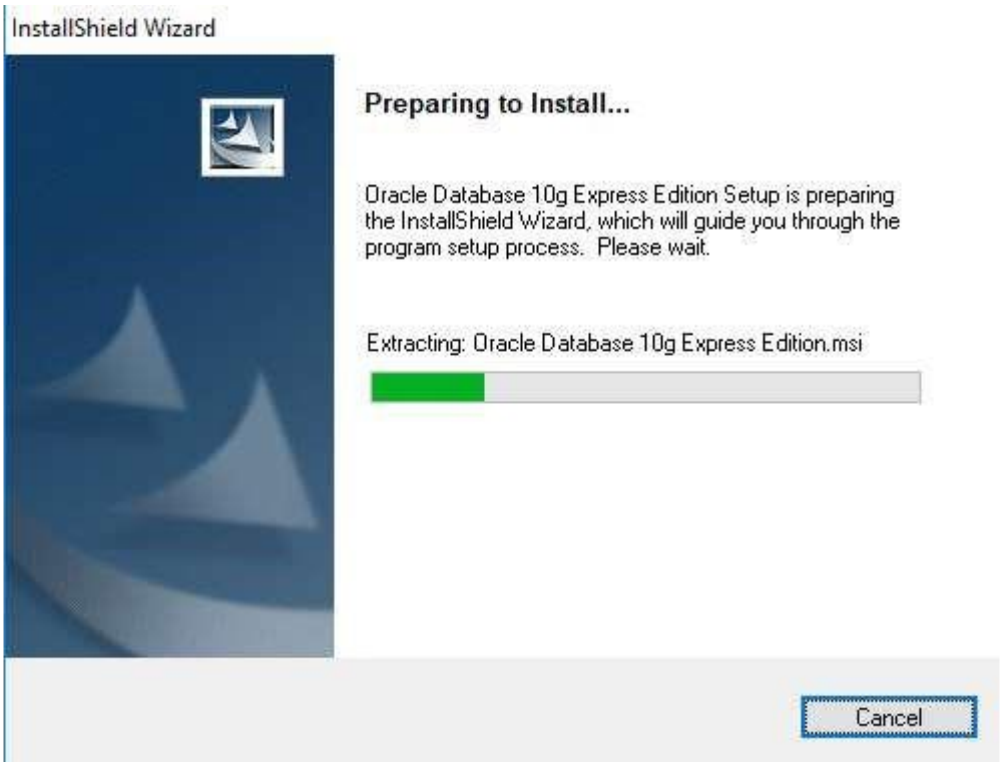
In this practical, we learned the basics of DBMS and SQL.

PRACTICAL-2

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

1) Download Oracle 10g from below link:

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

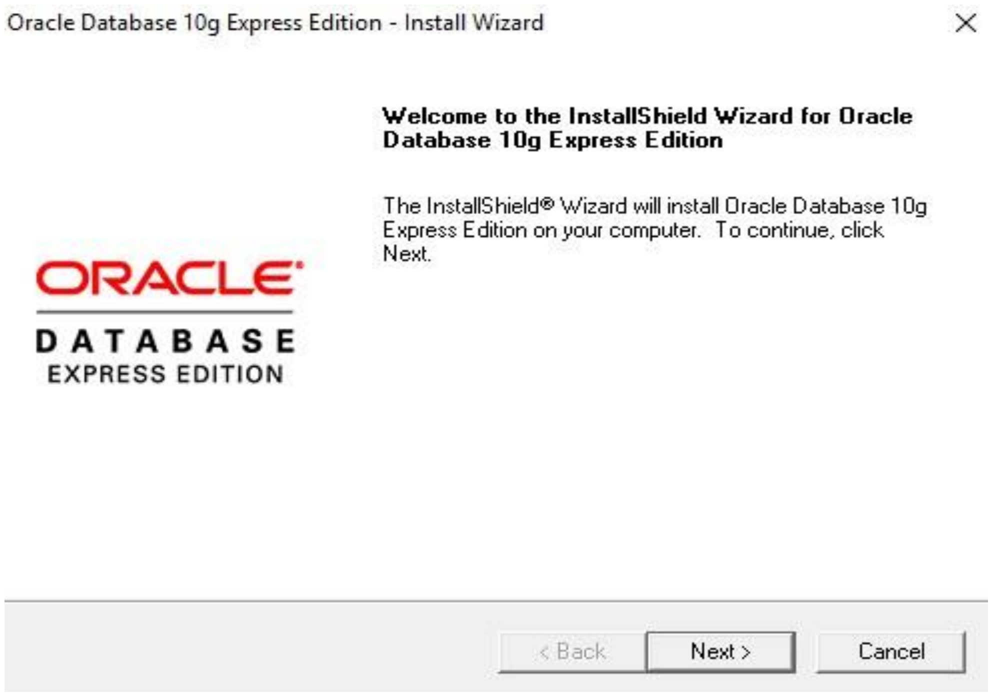


Preparing to Install...

Oracle Database 10g Express Edition Setup is preparing the InstallShield Wizard, which will guide you through the program setup process. Please wait.

Extracting: Oracle Database 10g Express Edition.msi

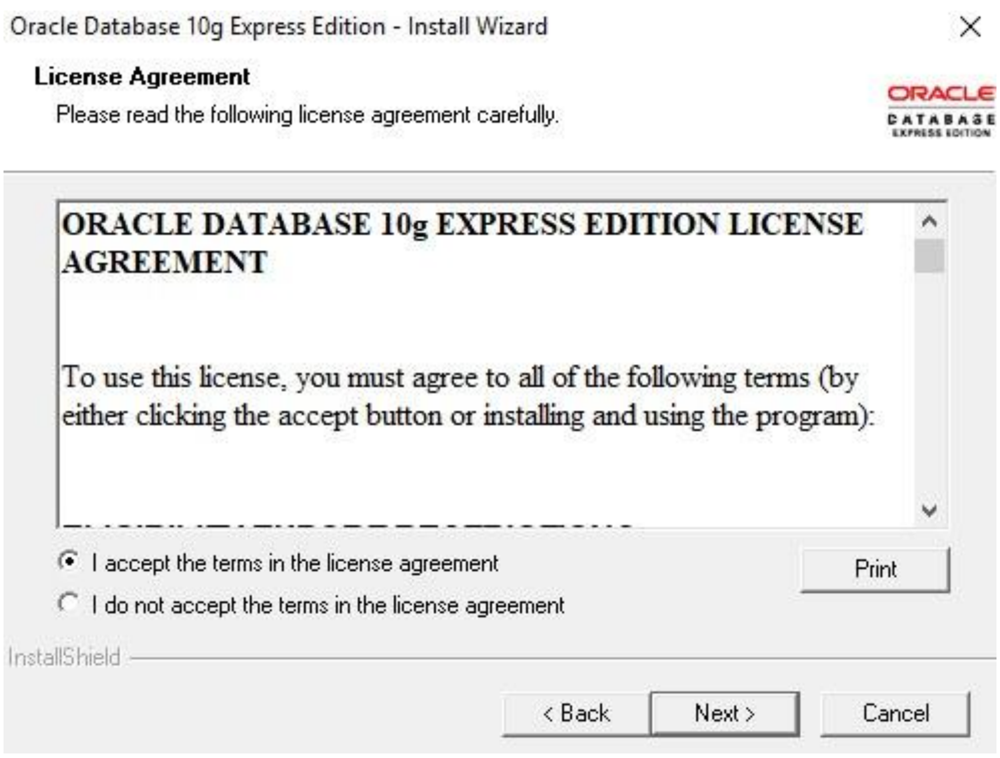
3) Click on Next button


Oracle Database 10g Express Edition - Install Wizard

Welcome to the InstallShield Wizard for Oracle Database 10g Express Edition

The InstallShield® Wizard will install Oracle Database 10g Express Edition on your computer. To continue, click Next.

4) Accept license agreement and click on next button


Oracle Database 10g Express Edition - Install Wizard

License Agreement

Please read the following license agreement carefully.

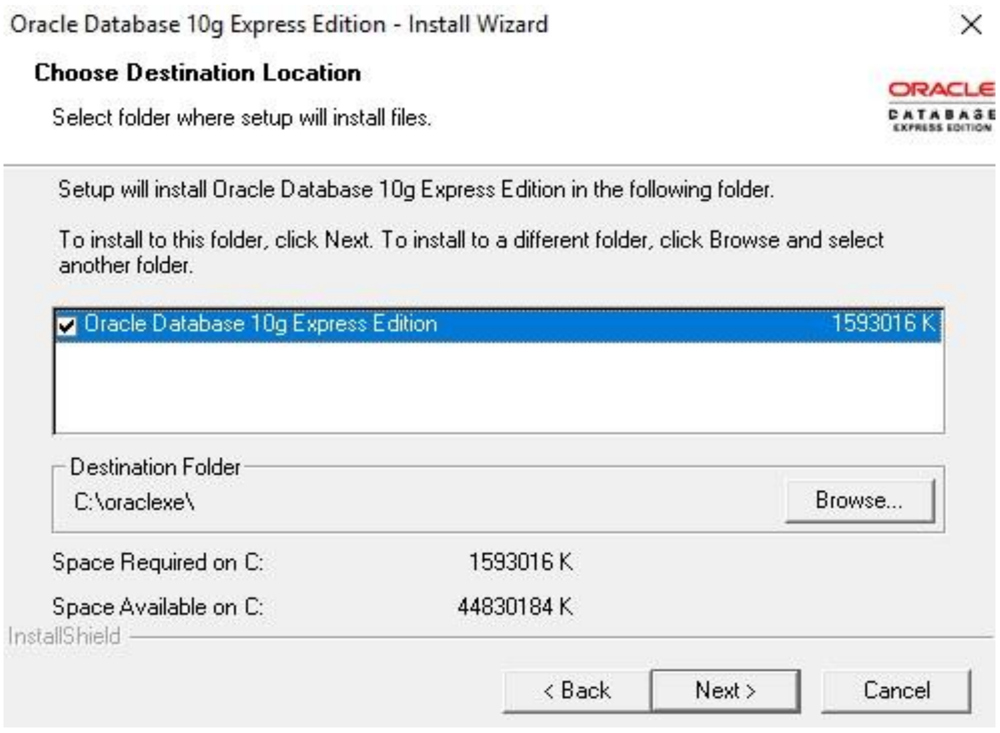
ORACLE DATABASE 10g EXPRESS EDITION LICENSE AGREEMENT

To use this license, you must agree to all of the following terms (by either clicking the accept button or installing and using the program):

[✓ I accept the terms in the license agreement]

I do not accept the terms in the license agreement

5) Click on next button


Oracle Database 10g Express Edition - Install Wizard

Choose Destination Location

Select folder where setup will install files.

Setup will install Oracle Database 10g Express Edition in the following folder.

To install to this folder, click Next. To install to a different folder, click Browse and select another folder.

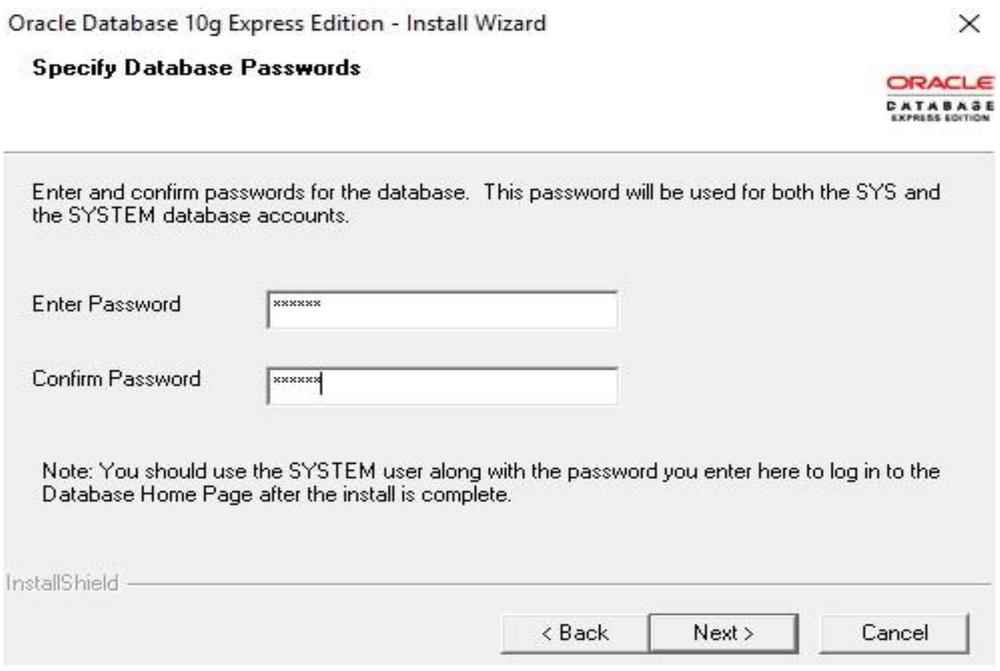
[✓ Oracle Database 10g Express Edition] 1593016 K

Destination Folder
C:\oraclexe\

Space Required on C: 1593016 K

Space Available on C: 44830184 K

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

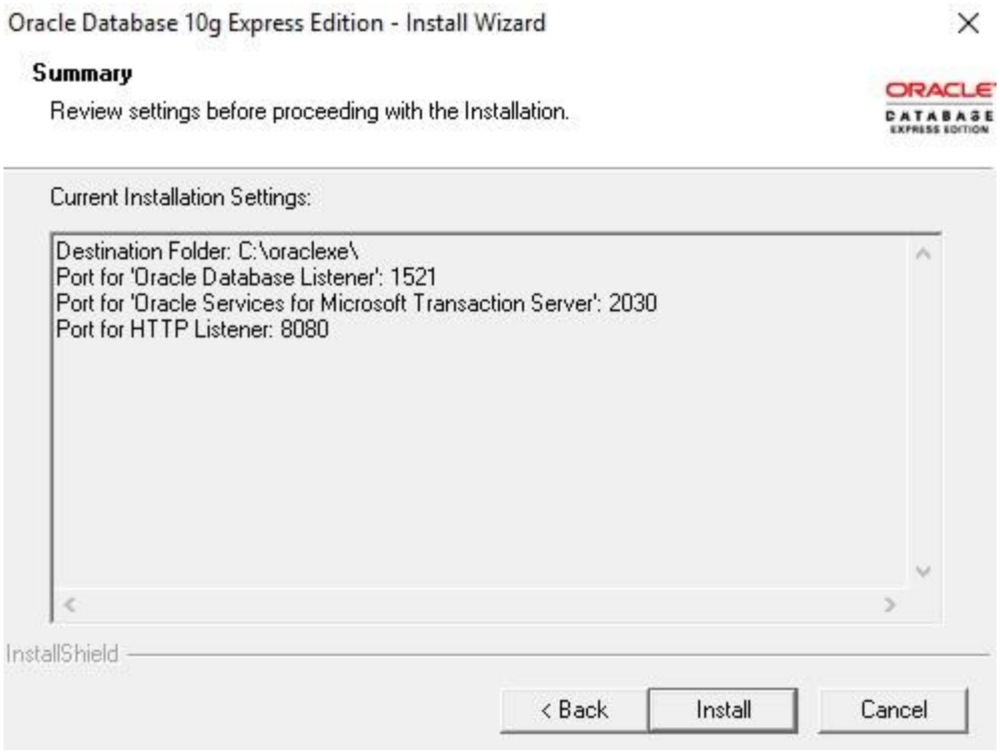
Enter and confirm passwords for the database. This password will be used for both the SYS and the SYSTEM database accounts.

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.

7) Click on install button


Oracle Database 10g Express Edition - Install Wizard

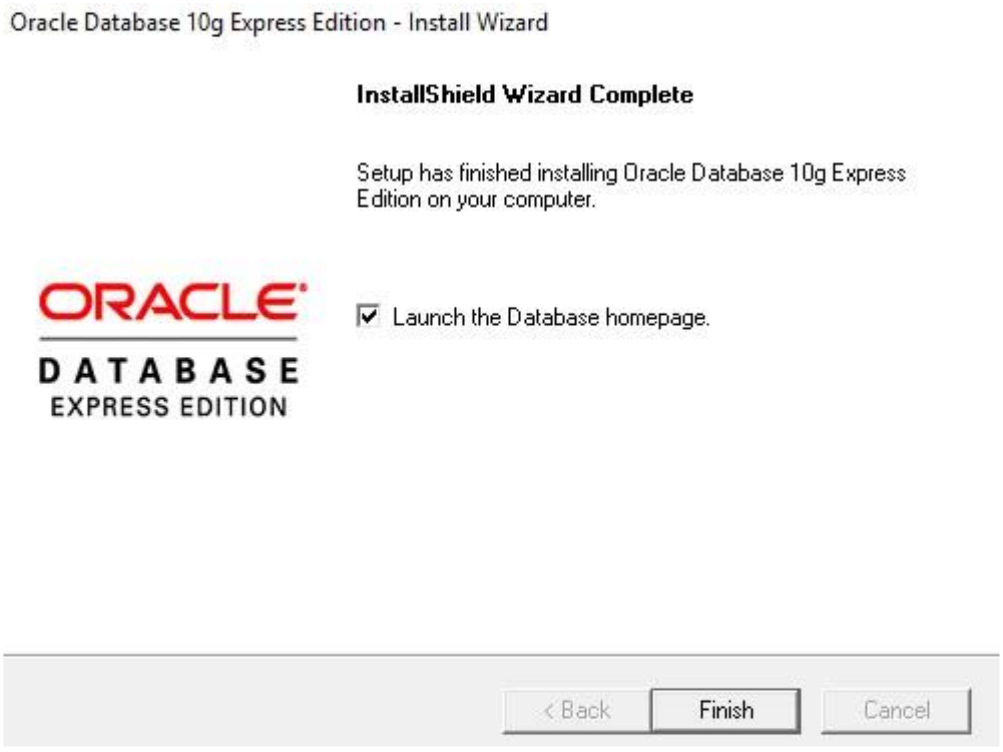
Summary

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
```

8) Click on finish button.

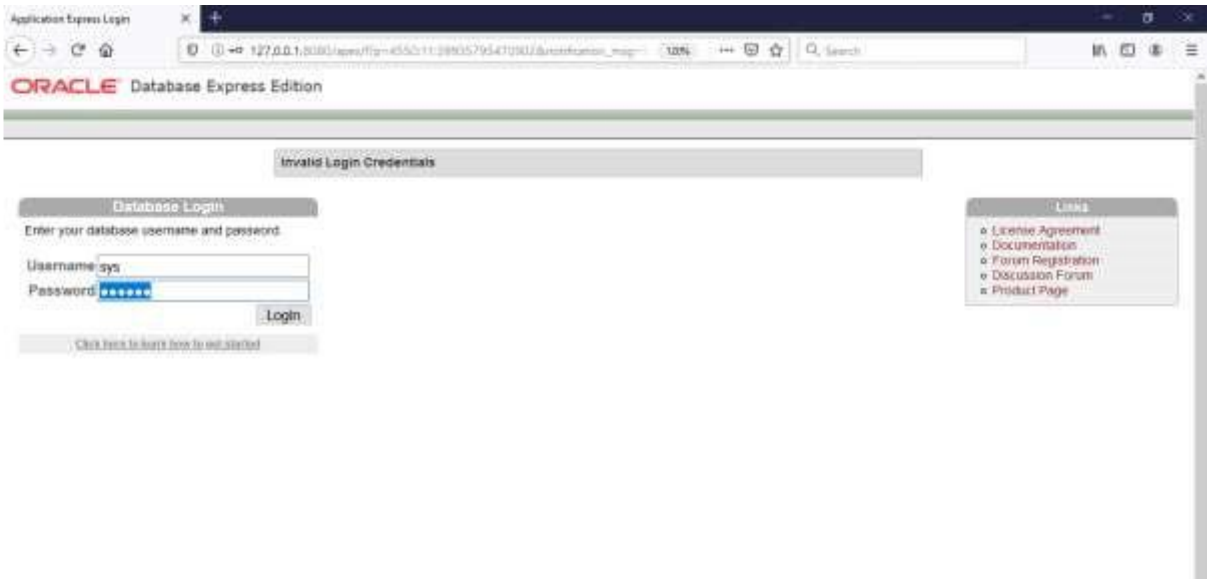

Oracle Database 10g Express Edition - Install Wizard

InstallShield Wizard Complete

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

[✓] Launch the Database homepage.

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


Invalid Login Credentials

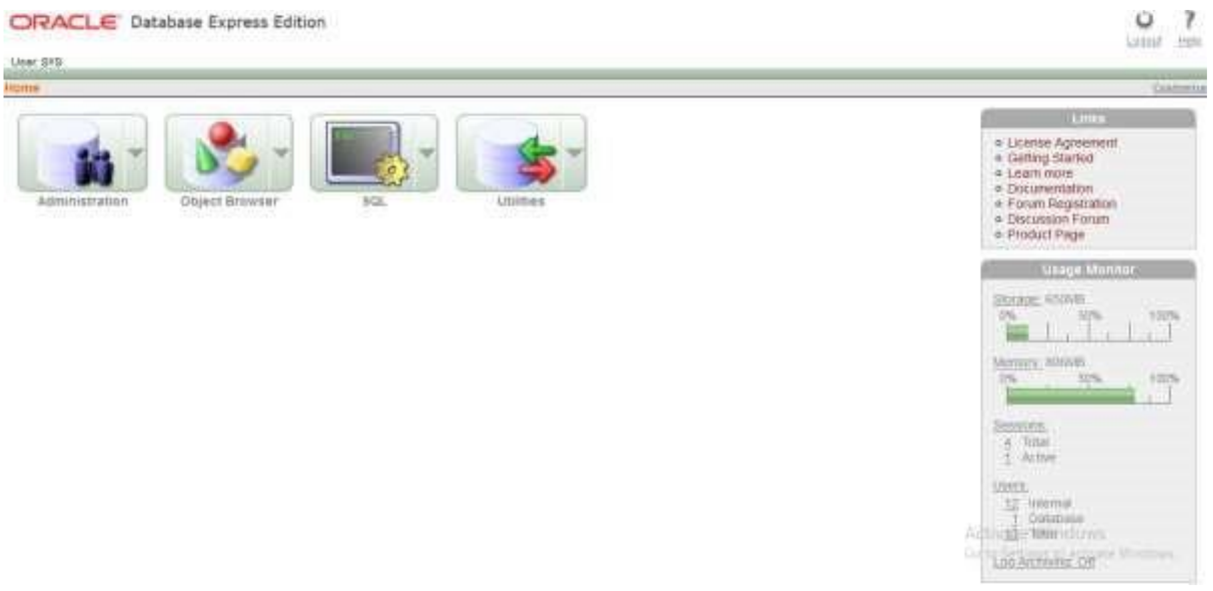
Database Login

Enter your database username and password

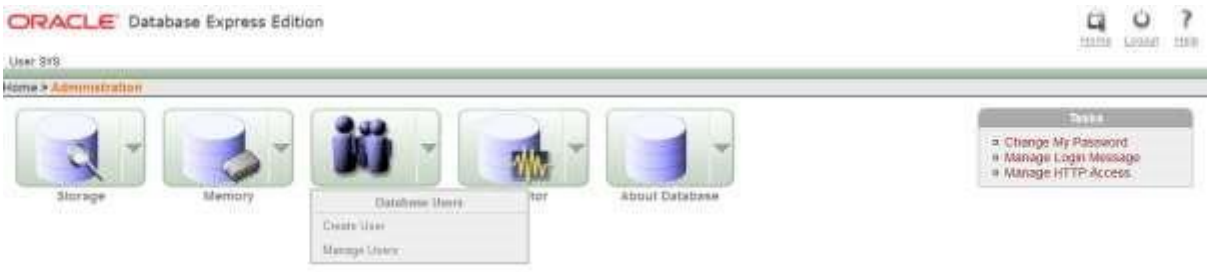
Username: sys

Password: ••••••

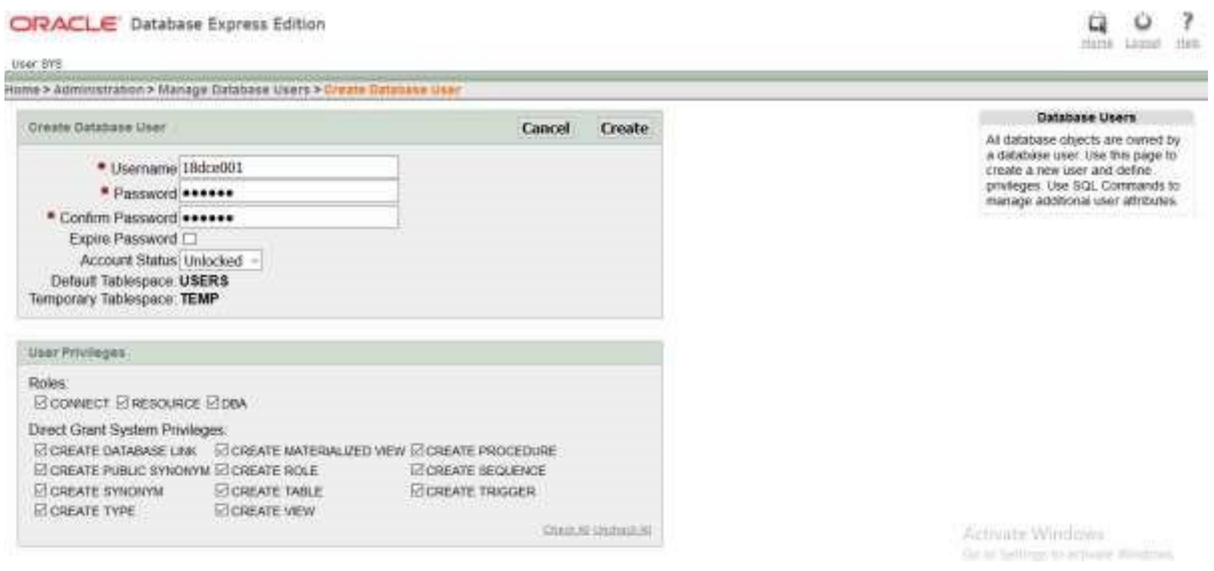
10) Click on Administration



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

User Privileges

Roles:
☑ CONNECT ☑ RESOURCE ☑ DBA

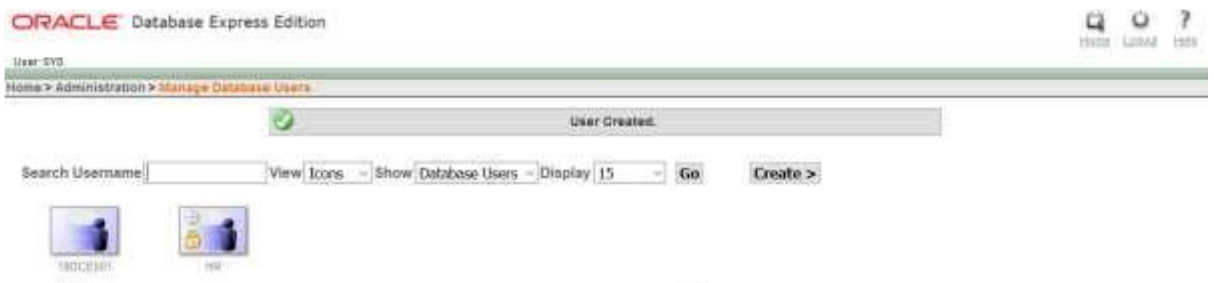
Direct Grant System Privileges:

☑ CREATE DATABASE LINK	☑ CREATE MATERIALIZED VIEW	☑ CREATE PROCEDURE
☑ CREATE PUBLIC SYNONYM	☑ CREATE ROLE	☑ CREATE SEQUENCE
☑ CREATE SYNONYM	☑ CREATE TABLE	☑ CREATE TRIGGER
☑ CREATE TYPE	☑ CREATE VIEW	

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.

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



ORACLE Database Express Edition

User: SYS

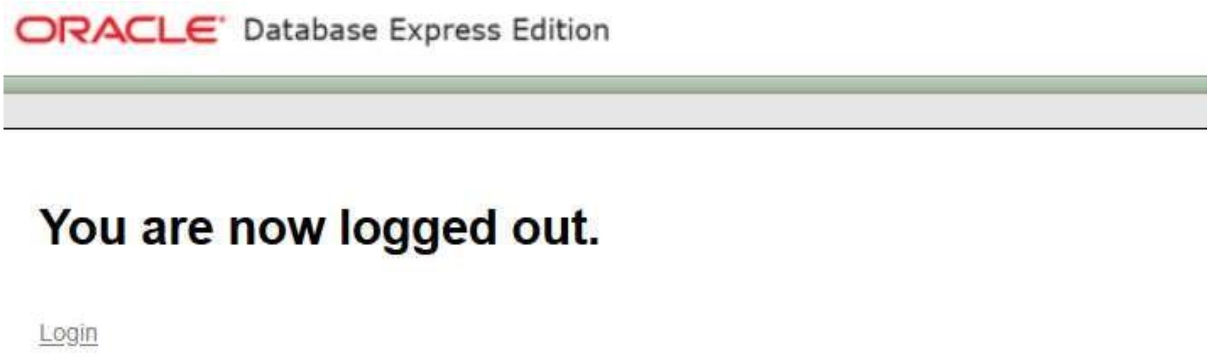
Home > Administration > Manage Database Users

User Created.

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

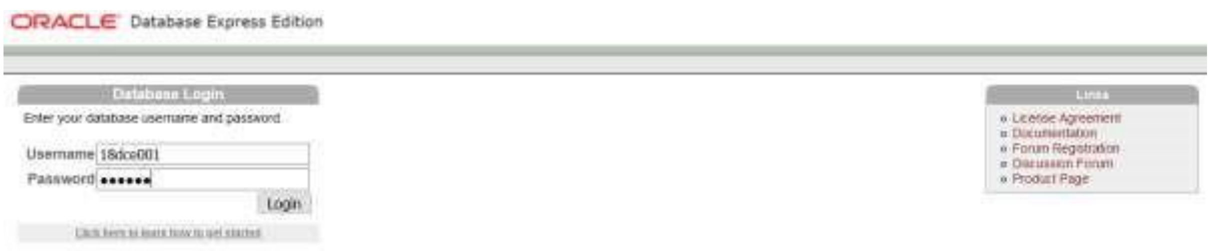
1-2

14) Click on login



You are now logged out.

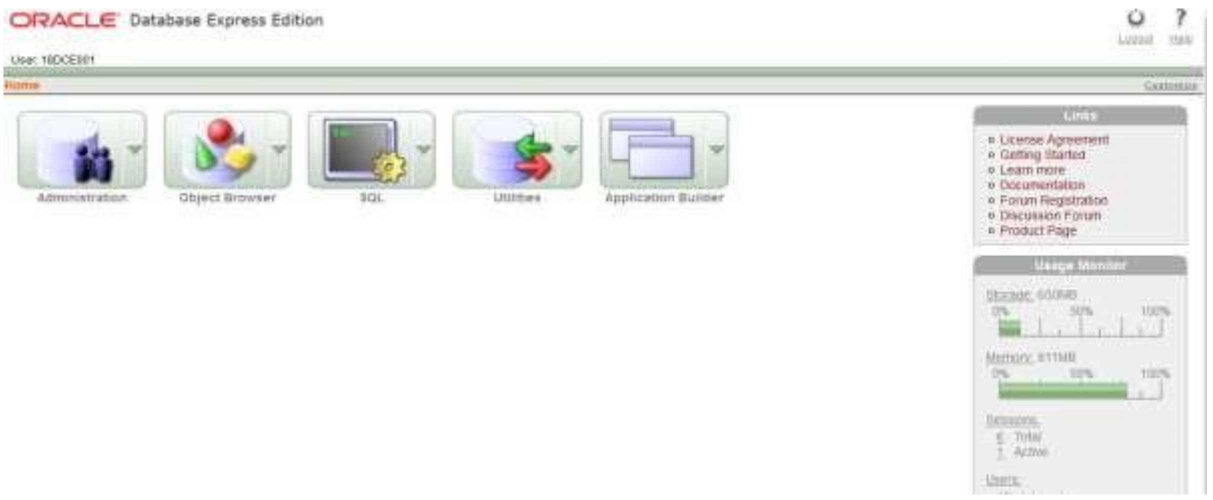
15)Enter username and password that you just created and click on "login" button



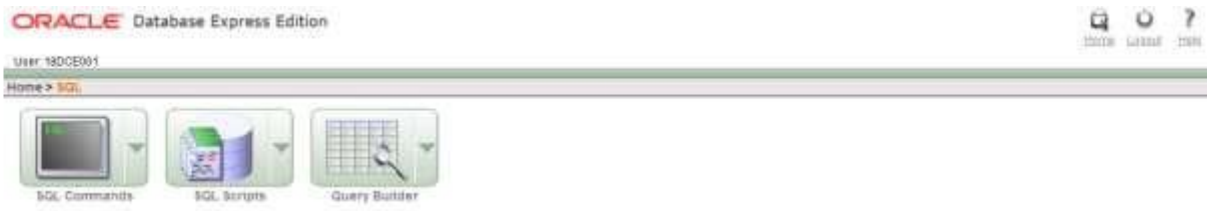
Enter your database username and password.

Username 18dce001
Password ••••••

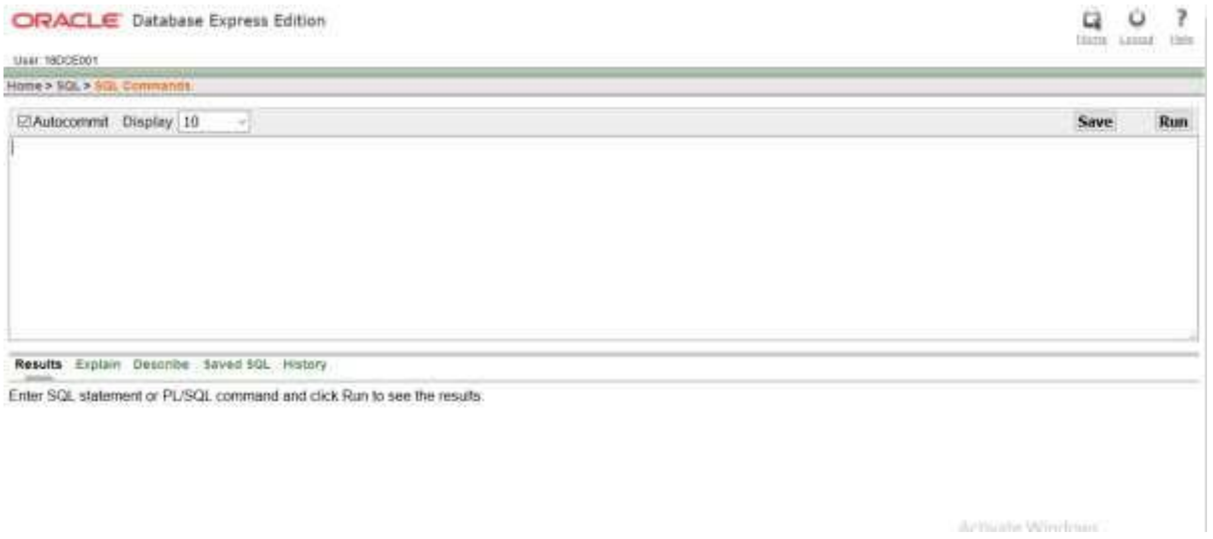
16) Click on SQL



17) Click on SQL Commands



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



Enter SQL statement or PL/SQL command and click Run to see the results.

What is SQL ?

• SQL is a standard language for accessing and manipulating databases.

• SQL is a standard language for accessing and manipulating databases. SQL stands for

Structured Query Language

• SQL lets you access and manipulate databases

• SQL became a standard of the American National Standards Institute (ANSI) in 1986,

and of the International Organization for Standardization (ISO) in 1987.

• SQL can retrieve data, update, insert, create tables, create new database, can set

permissions on tables, etc.

What is PLSQL?

• PL/SQL stands for "Procedural Language extensions to the Structured Query

Language".

• SQL is a popular language for both querying and updating data in the relational

database management systems (RDBMS).

• PL/SQL adds many procedural constructs to SQL language to overcome some

limitations of SQL.

• Besides, PL/SQL provides a more comprehensive programming language solution for

building mission-critical applications on Oracle Databases.

• PL/SQL is a highly structured and readable language. Its constructs express the intent

of the code clearly. Also, PL/SQL is a straightforward language to learn.

• PL/SQL is a standard and portable language for Oracle Database development. If you

develop a program that executes on an Oracle Database, you can quickly move it to

another compatible Oracle Database without any changes.

CONCLUSION:

In this practical, we learned the basics of oracle database and SQL and PL/SQL.

PRACTICAL-3

To study DDL-create and DML-insert commands.

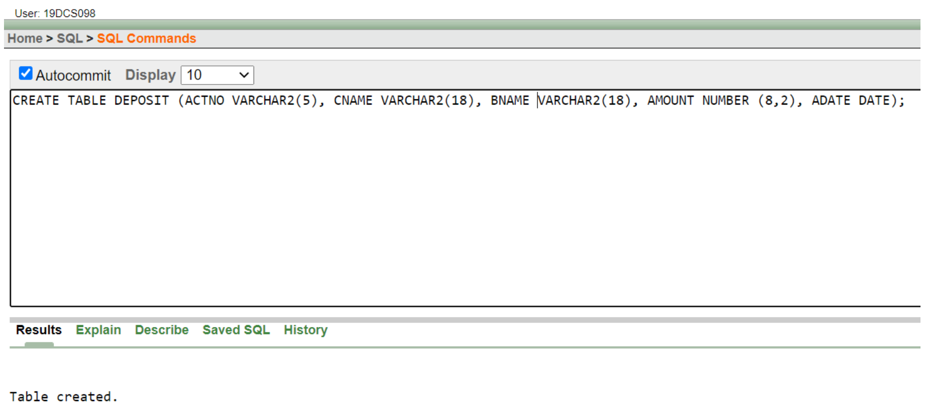
(i) Create tables according to the following definition.

• CREATE TABLE D

EPOSIT (ACTNO VARCHAR2(5), CNAME

VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2),

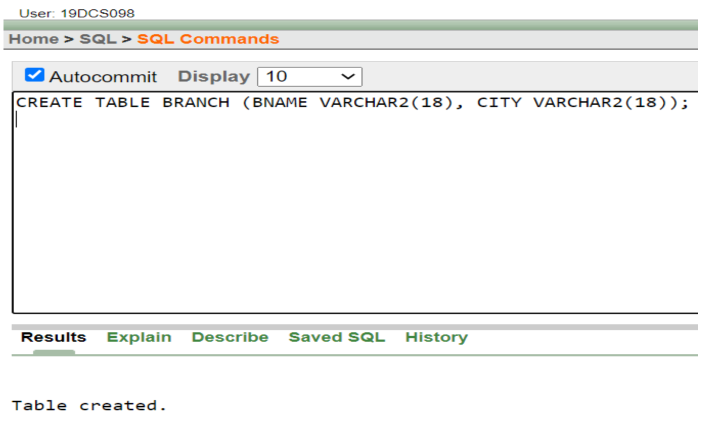
ADATE DATE);


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

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

```
Table created.
```

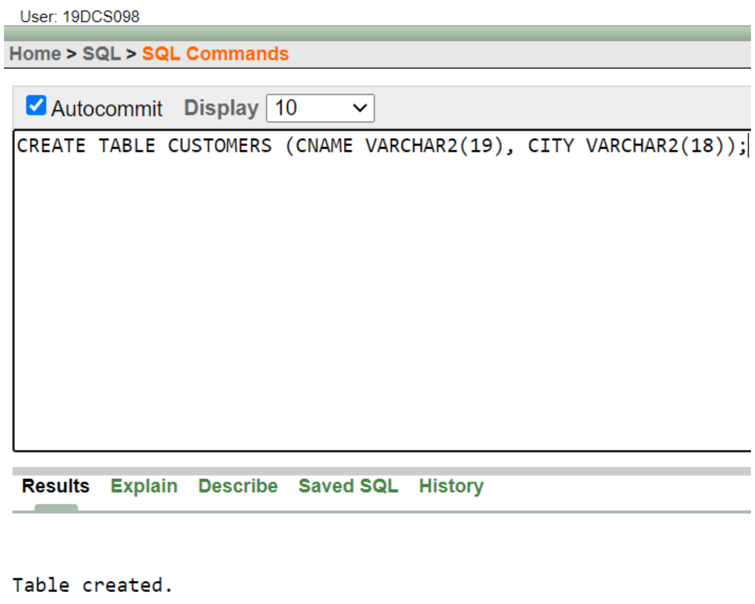
• CREATE TABLE BRANCH (BNAME VARCHAR2(18), CITY VARCHAR2(18));


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE TABLE BRANCH (BNAME VARCHAR2(18), CITY VARCHAR2(18));
```

```
Table created.
```

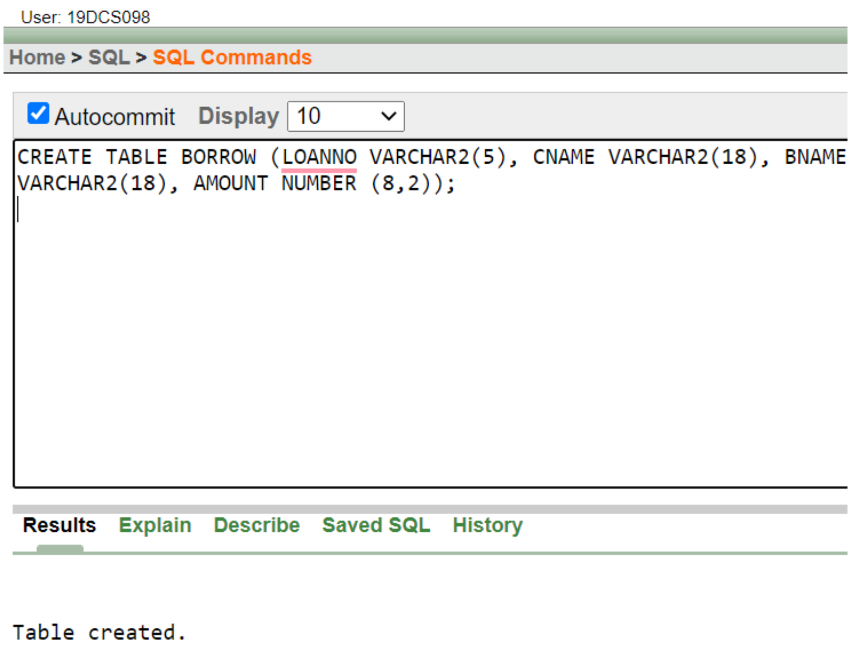
• CREATE TABLE CUSTOMERS (CNAME VARCHAR2(19), CITY VARCHAR2(18));


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE TABLE CUSTOMERS (CNAME VARCHAR2(19), CITY VARCHAR2(18));
```

```
Table created.
```

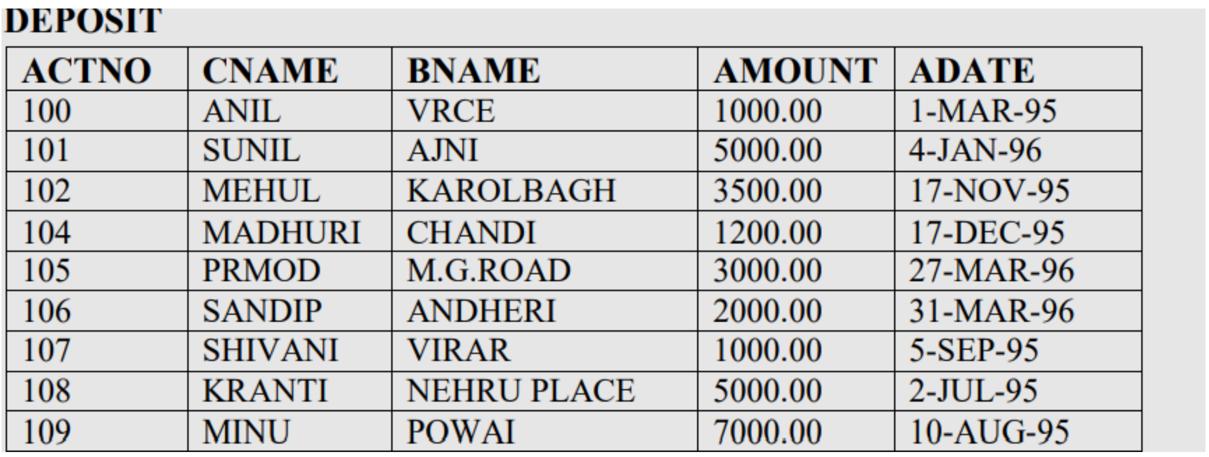
• CREATE TABLE BORROW (LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME VARCHAR2(18), AMOUNT NUMBER (8,2));


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE TABLE BORROW (LOANNO VARCHAR2(5), CNAME VARCHAR2(18), BNAME
VARCHAR2(18), AMOUNT NUMBER (8,2));
```

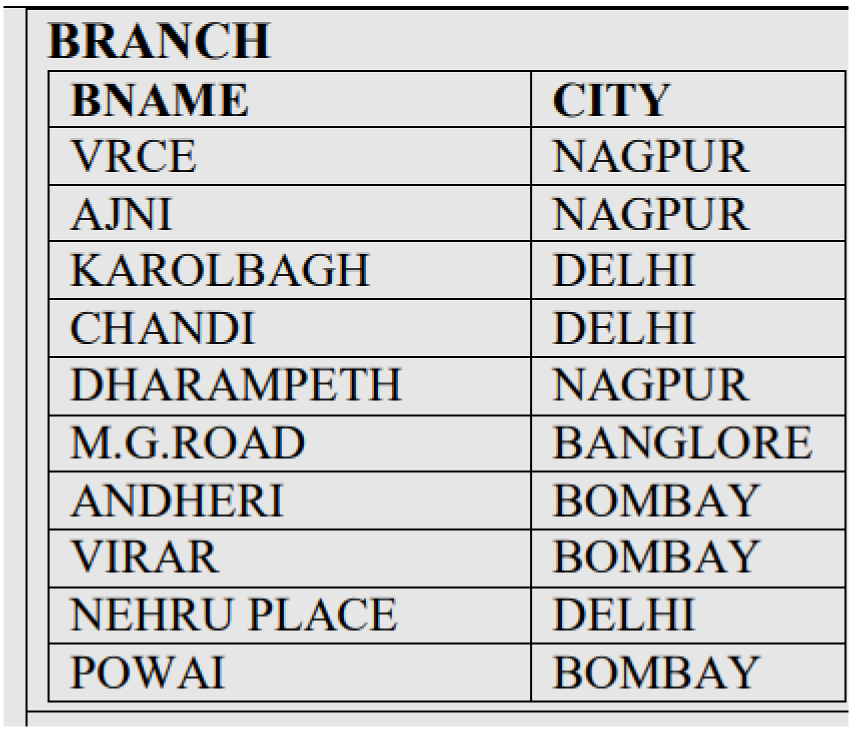
```
Table created.
```

(ii) Insert the data as shown below.



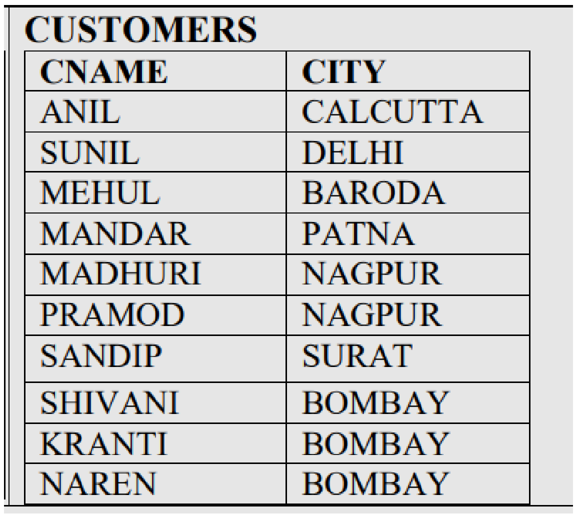
DEPOSIT

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



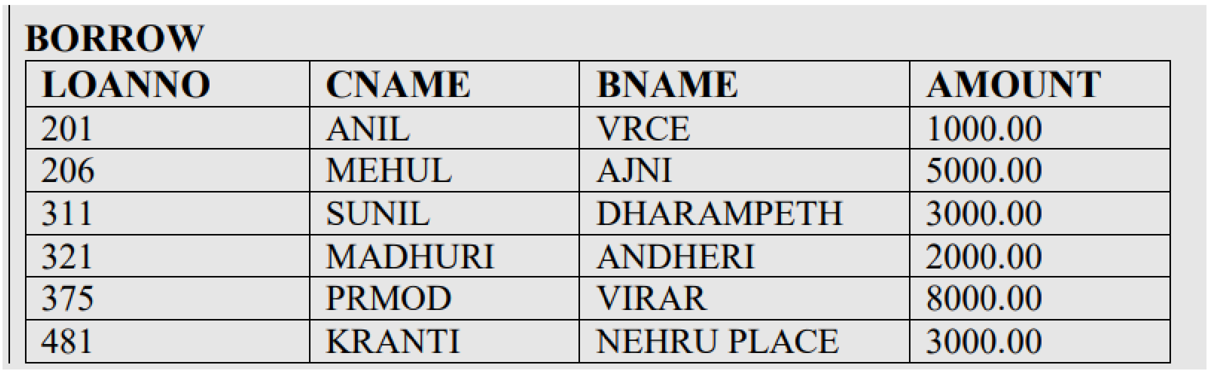
BRANCH

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



CUSTOMERS

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



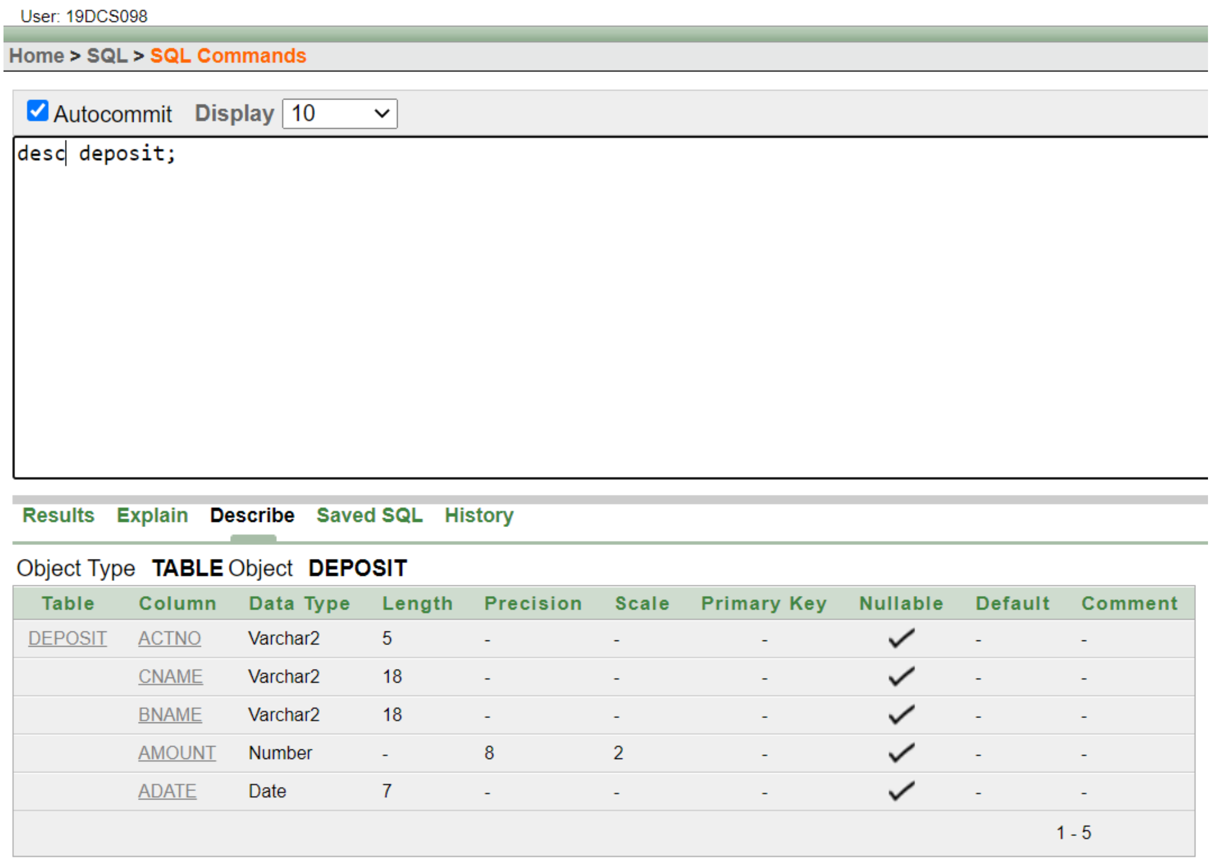
BORROW

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

- **From the above given tables perform the following queries:**

(1) Describe deposit, branch.

e


User: 19DCS098

Home > SQL > SQL Commands

☑ Autocommit Display 10

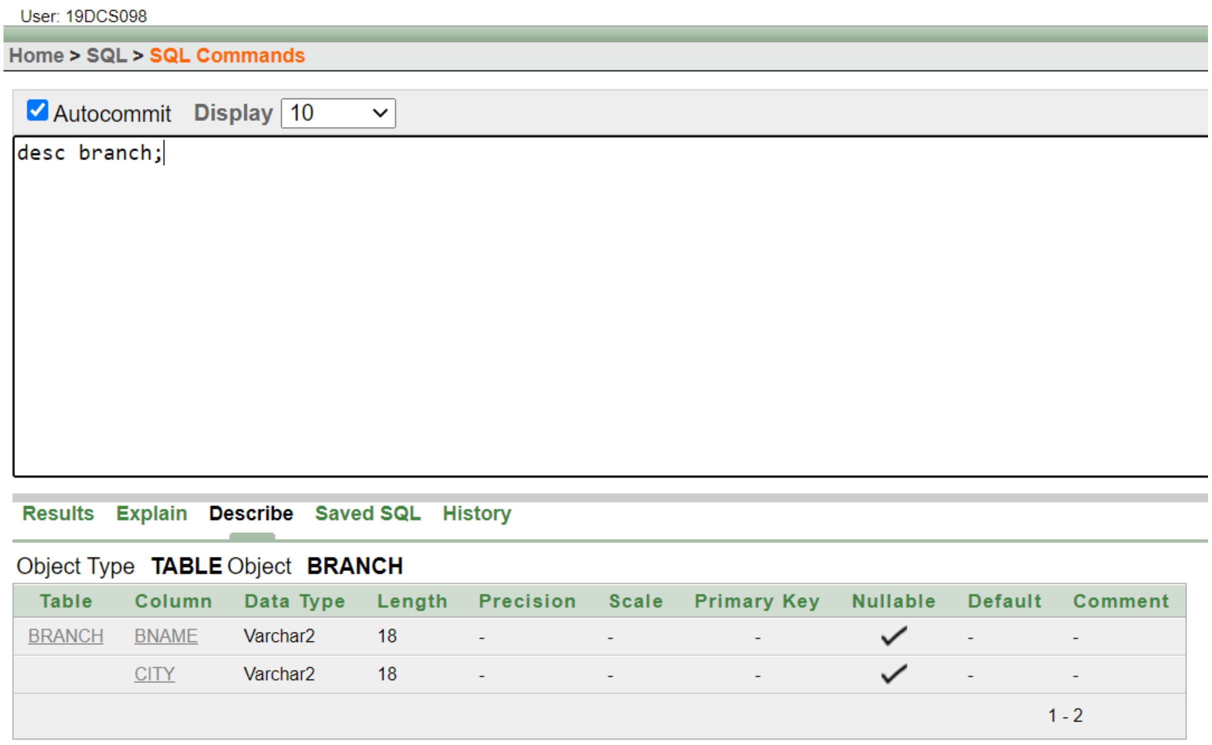
```
desc deposit;
```

Results Explain Describe Saved SQL History

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

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

1 - 5


User: 19DCS098

Home > SQL > SQL Commands

☑ Autocommit Display 10

```
desc branch;
```

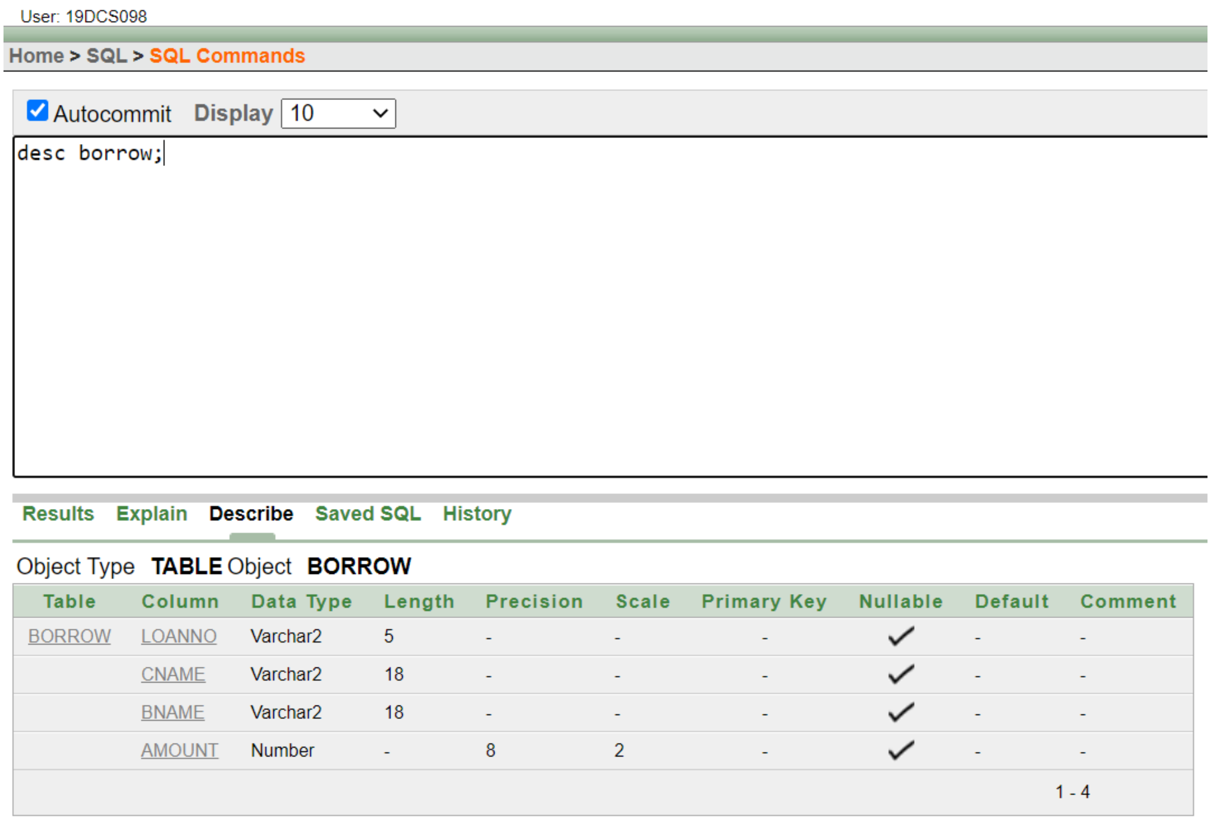
Results Explain Describe Saved SQL History

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.

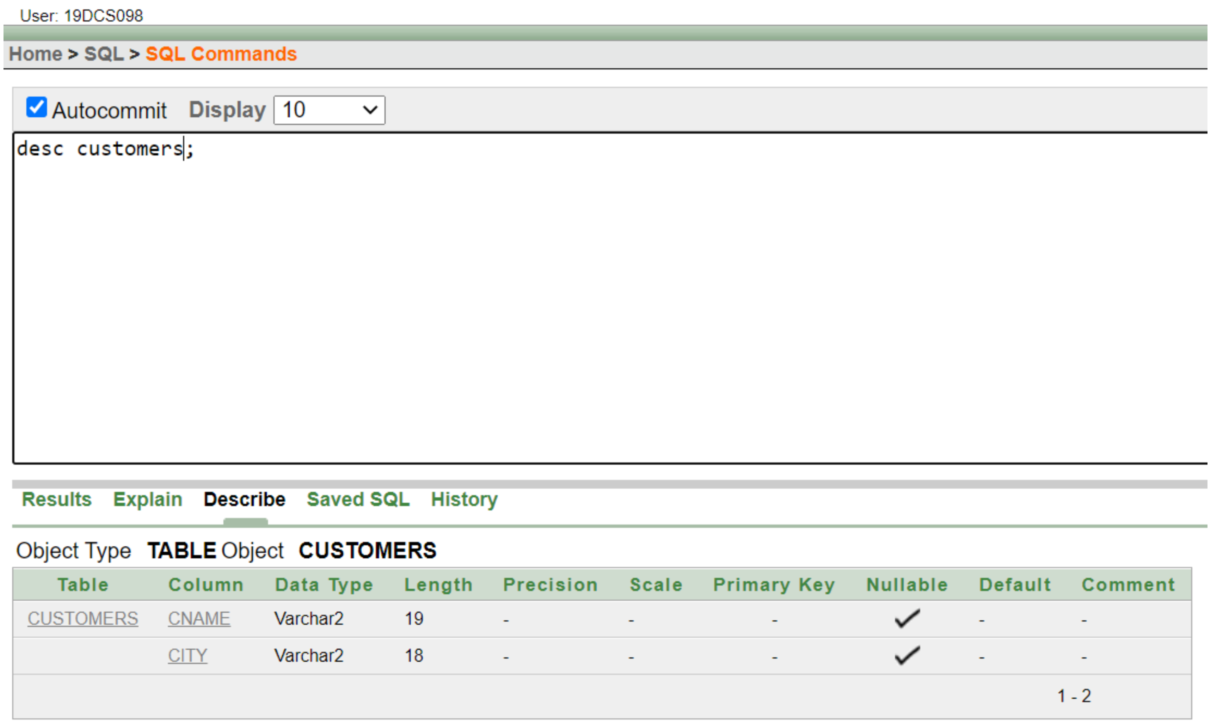

User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
desc borrow;
```

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: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

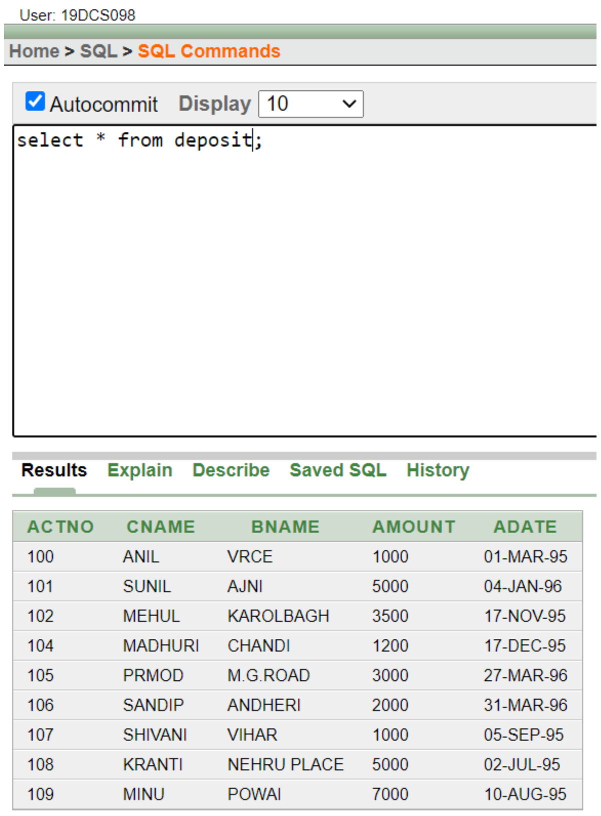
```
desc customers;
```

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: 19DCS098

Home > SQL > SQL Commands

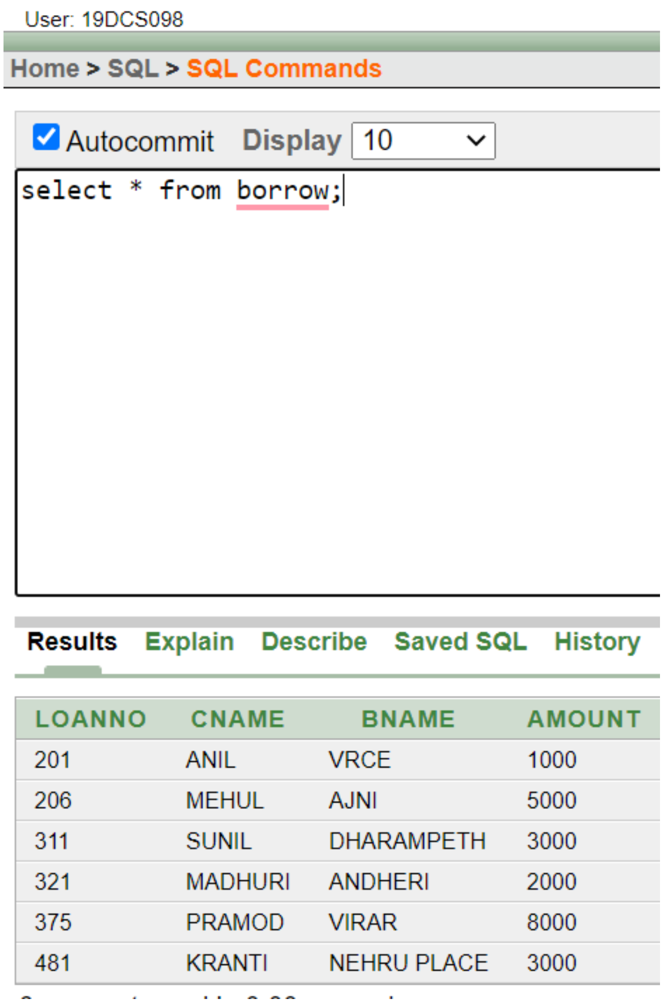
Autocommit Display 10

```
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	VIHAR	1000	05-SEP-95
108	KRANTI	NEHRU PLACE	5000	02-JUL-95
109	MINU	POWAI	7000	10-AUG-95

(4) List all data from table BORROW.


User: 19DCS098

Home > SQL > SQL Commands

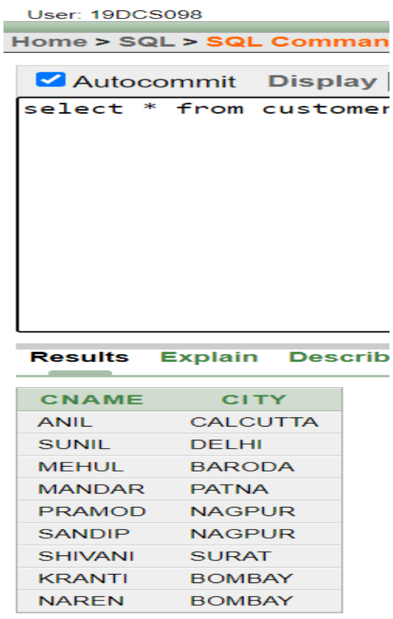
Autocommit Display 10

```
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	PRAMOD	VIRAR	8000
481	KRANTI	NEHRU PLACE	3000

(5) List all data from table CUSTOMERS.


User: 19DCS098

Home > SQL > SQL Comman

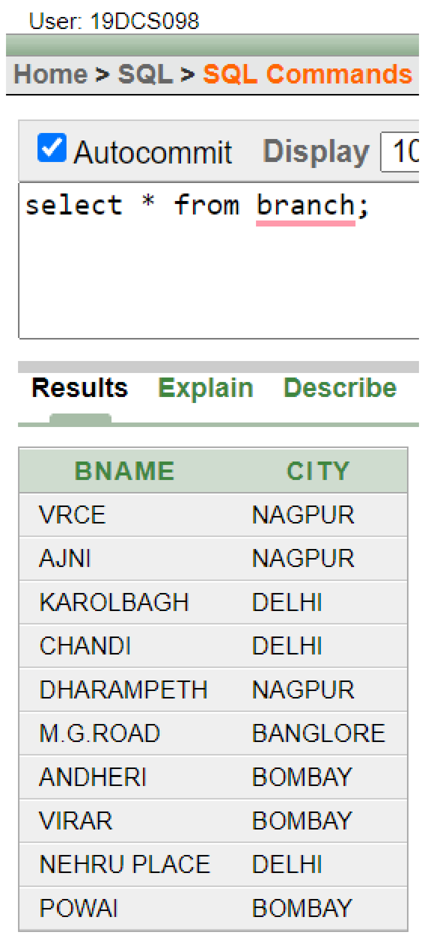
Autocommit Display

```
select * from customer
```

Results Explain Describ

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

(6) List all data from table BRANCH.


User: 19DCS098

Home > SQL > SQL Commands

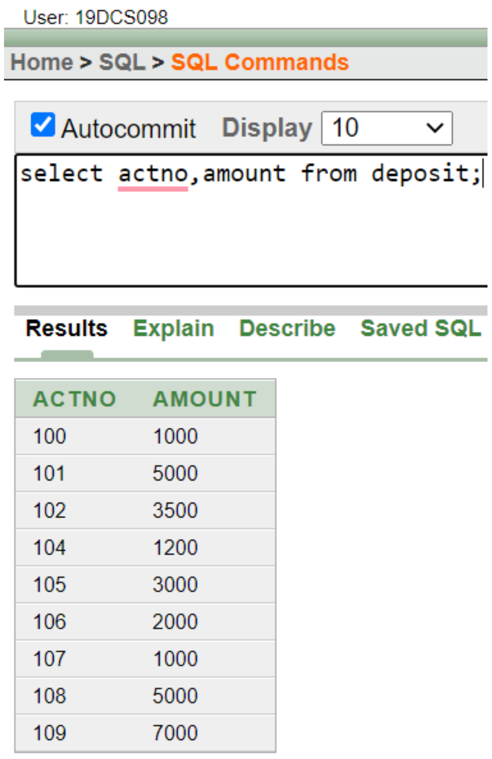
Autocommit Display 10

```
select * from branch;
```

Results Explain Describe

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

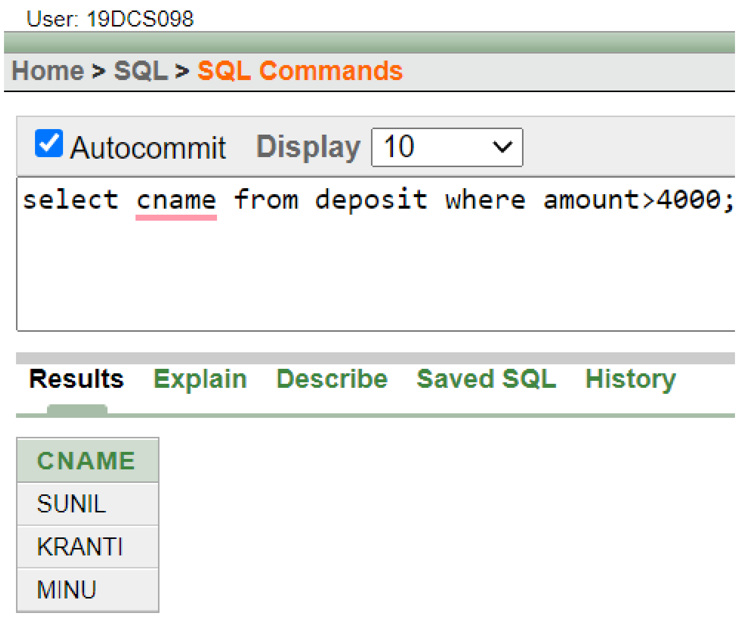
(7) Give account no and amount of depositors.



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

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

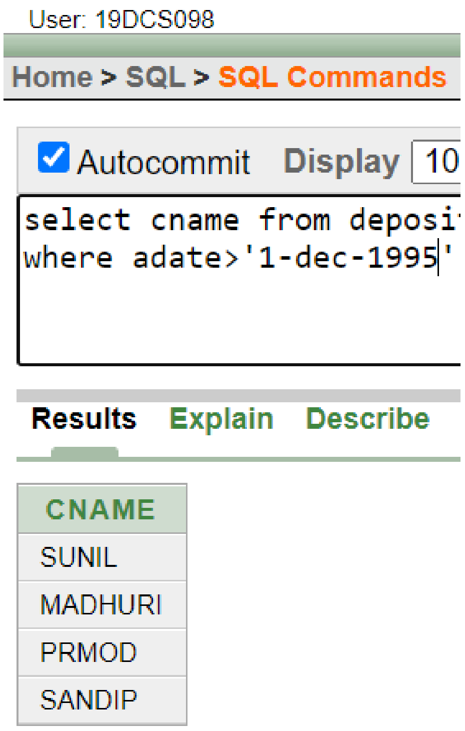
(8) Give name of depositors having amount greater than 4000.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

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

CNAME
SUNIL
KRANTI
MINU

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



User: 19DCS098

Home > SQL > SQL Commands

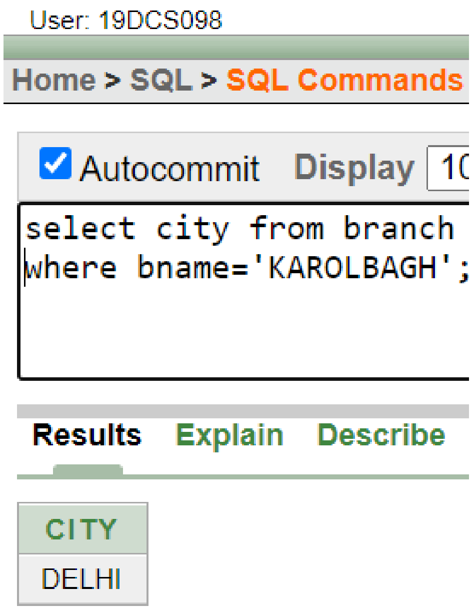
☑ Autocommit Display 10

```
select cname from deposi
where adate>'1-dec-1995'
```

Results Explain Describe

CNAME
SUNIL
MADHURI
PRMOD
SANDIP

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



User: 19DCS098

Home > SQL > SQL Commands

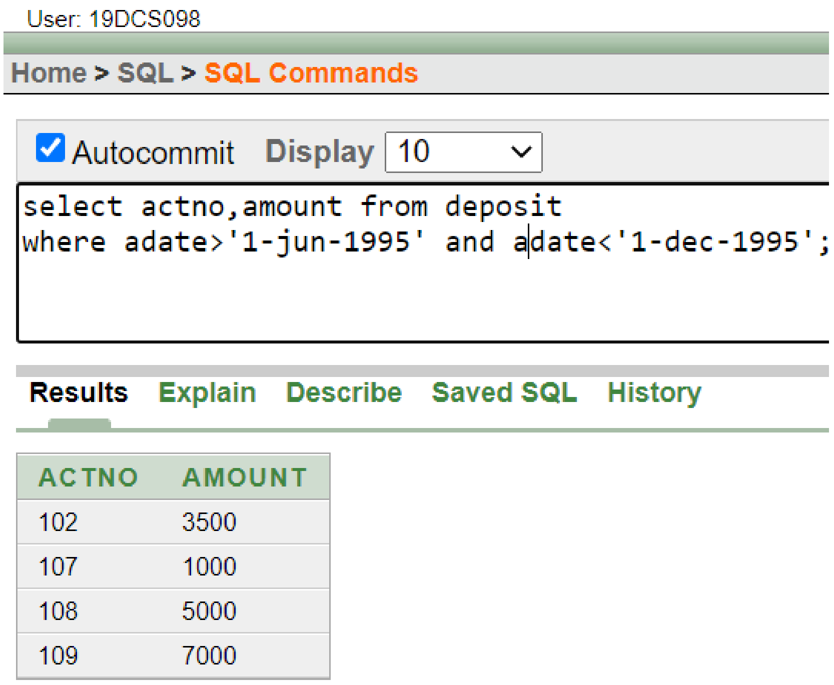
☑ Autocommit Display 10

```
select city from branch
where bname='KAROLBAGH';
```

Results Explain Describe

CITY
DELHI

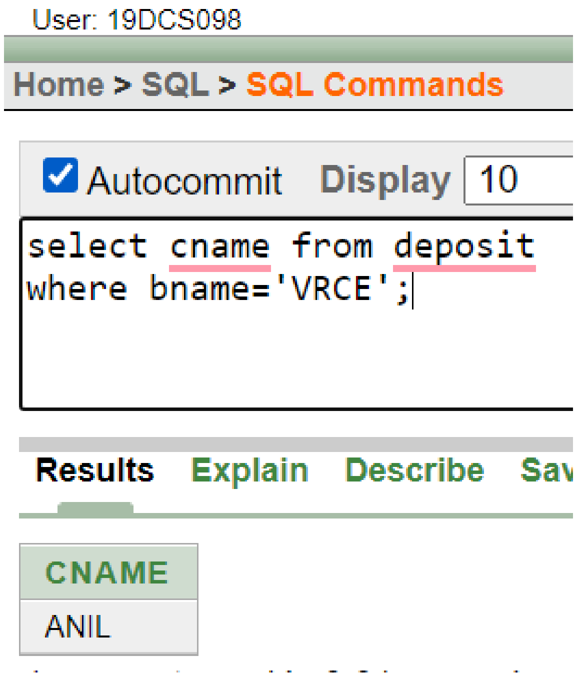
(11) Give account no and amount of customer having account opened between date 1-12-96 and 1-6-96.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
select actno,amount from deposit
where adate>'1-jun-1995' and adate<'1-dec-1995';
```

ACTNO	AMOUNT
102	3500
107	1000
108	5000
109	7000

(12) **Give names of depositors having account at VRCE.**



```
select cname from deposit
where bname='VRCE';
```

CNAME
ANIL

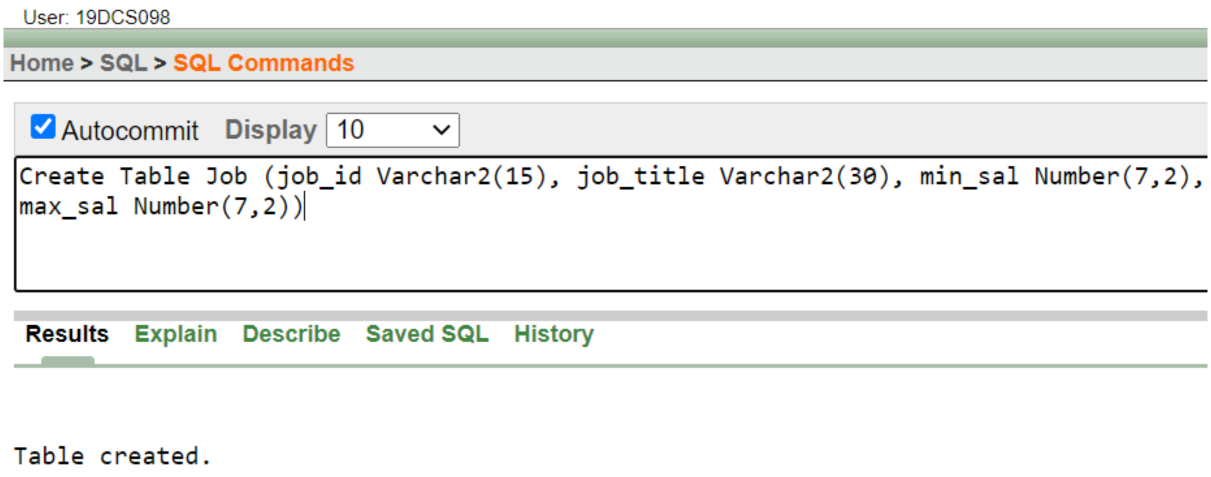
CONCLUSION:

In the above practicals, we learned the basics of DDL and DML.

PRACTICAL-4

Create the below given table and insert the data accordingly

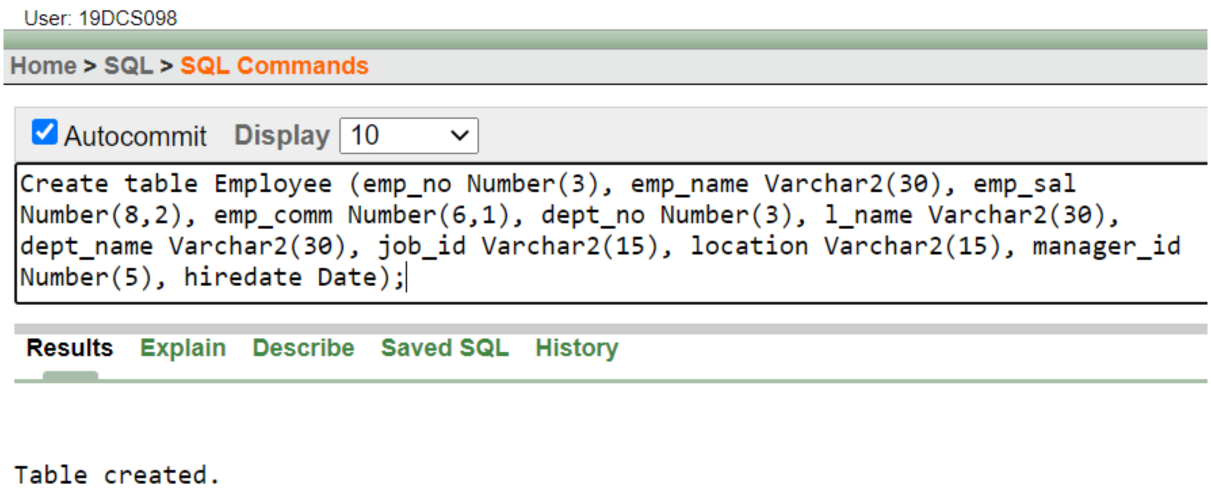
• **Create Table Job (job_id, job_title, min_sal, max_sal)**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
Create Table Job (job_id Varchar2(15), job_title Varchar2(30), min_sal Number(7,2),
max_sal Number(7,2))
```

```
Table created.
```

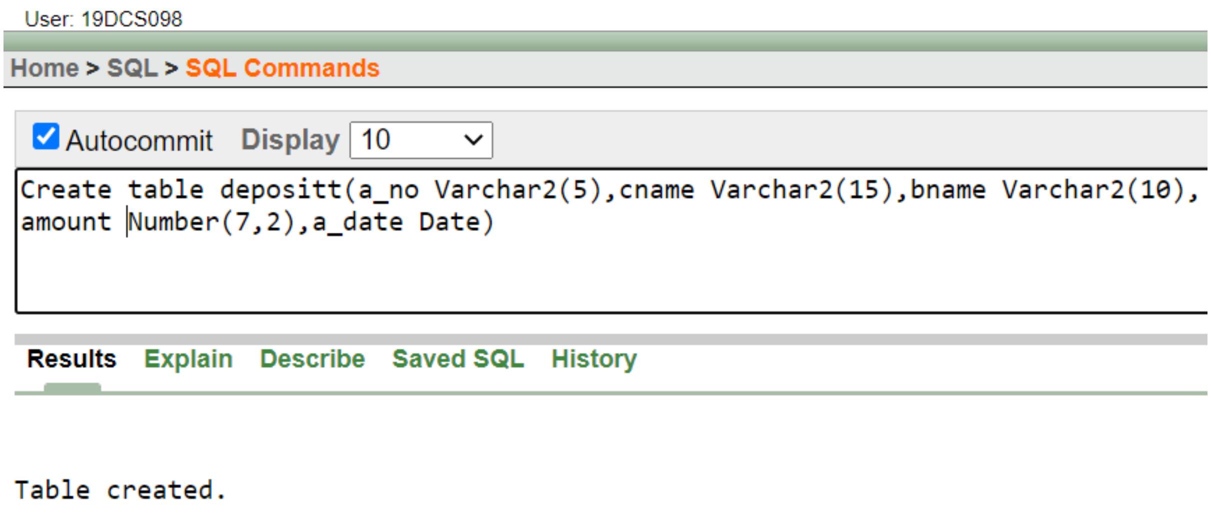
Create table Employee (emp_no, emp_name, emp_sal, emp_comm, dept_no, l_name, dept_name,job_id, location, manager_id, hiredate)


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
Create table Employee (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);
```

```
Table created.
```

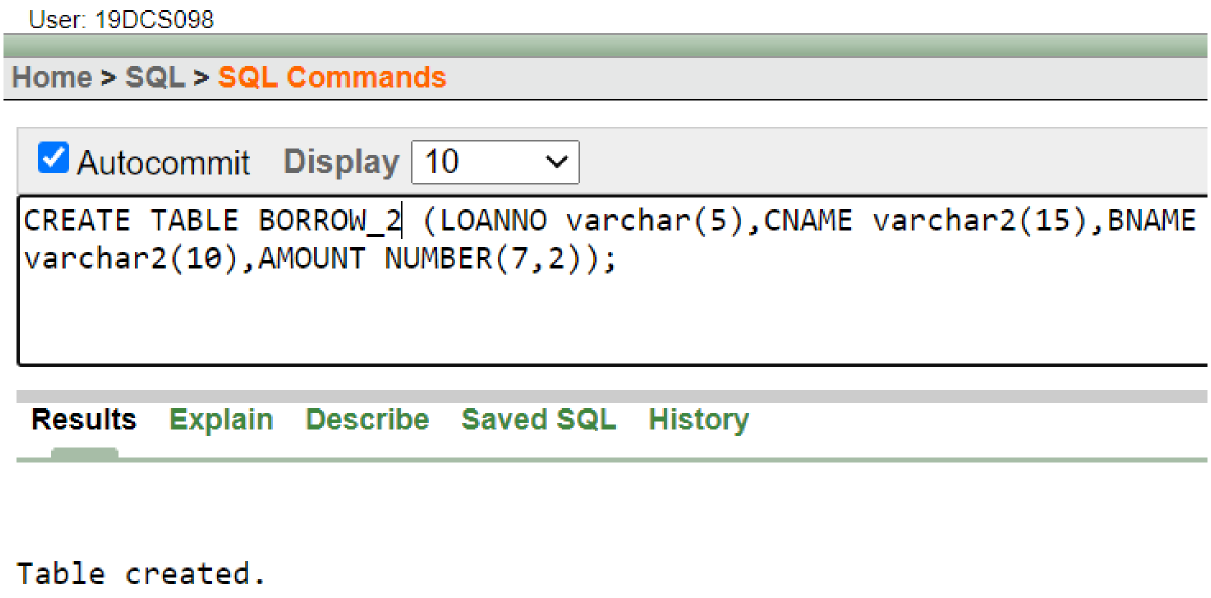
Create table deposit(a_no,cname,bname,amount,a_date).


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
Create table depositt(a_no Varchar2(5),cname Varchar2(15),bname Varchar2(10),
amount Number(7,2),a_date Date)
```

```
Table created.
```

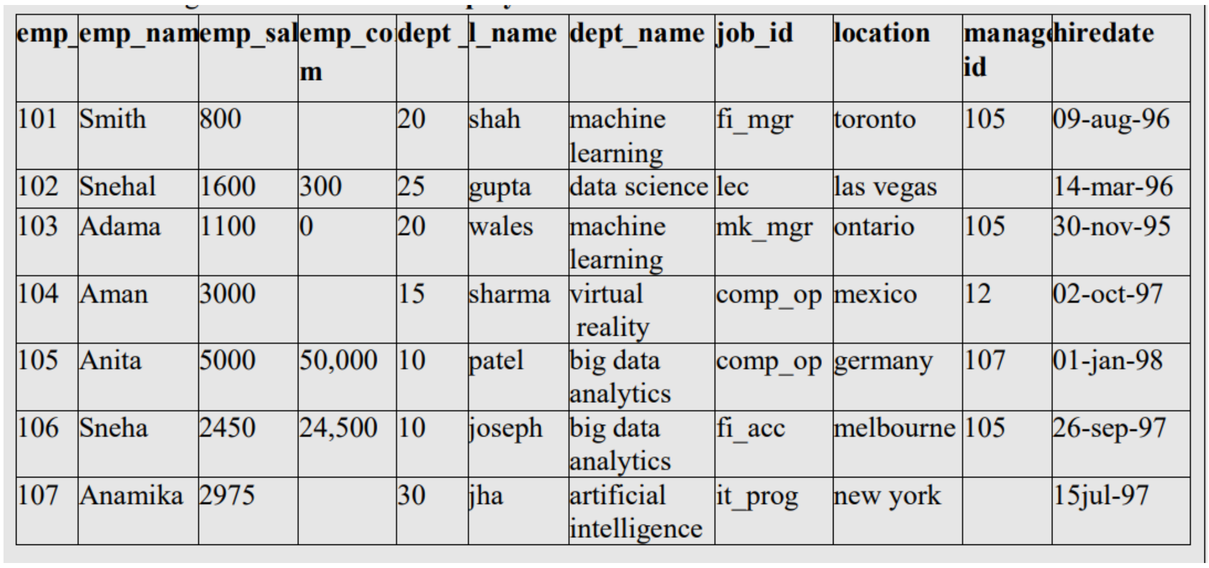
Create table borrow (loanno, cname, bname, amount).


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE TABLE BORROW_2 (LOANNO varchar(5),CNAME varchar2(15),BNAME
varchar2(10),AMOUNT NUMBER(7,2));
```

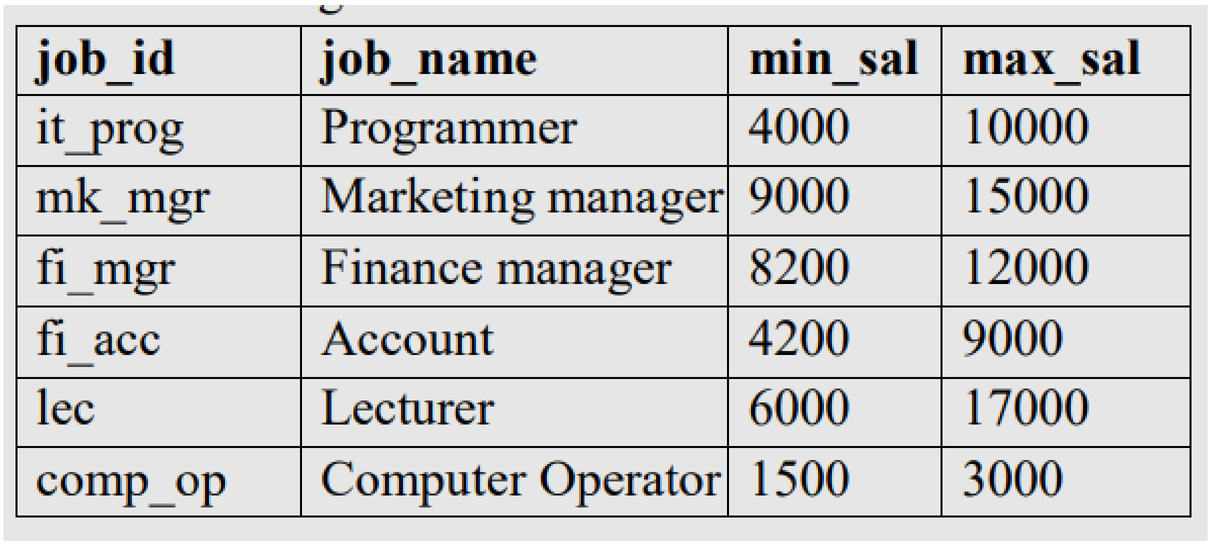
```
Table created.
```

• **Insert following values in the table Employee.**

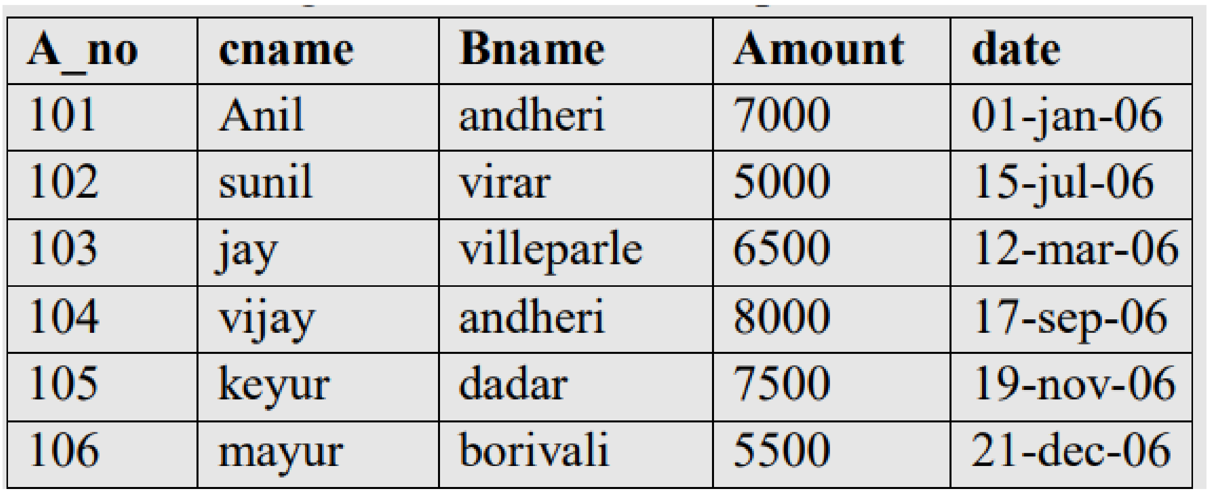


emp_	emp_nam	emp_sal	emp_co m	dept_	l_name	dept_name	job_id	location	manag id	hiredate
101	Smith	800		20	shah	machine learning	fi_mgr	toronto	105	09-aug-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas		14-mar-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-nov-95
104	Aman	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.**



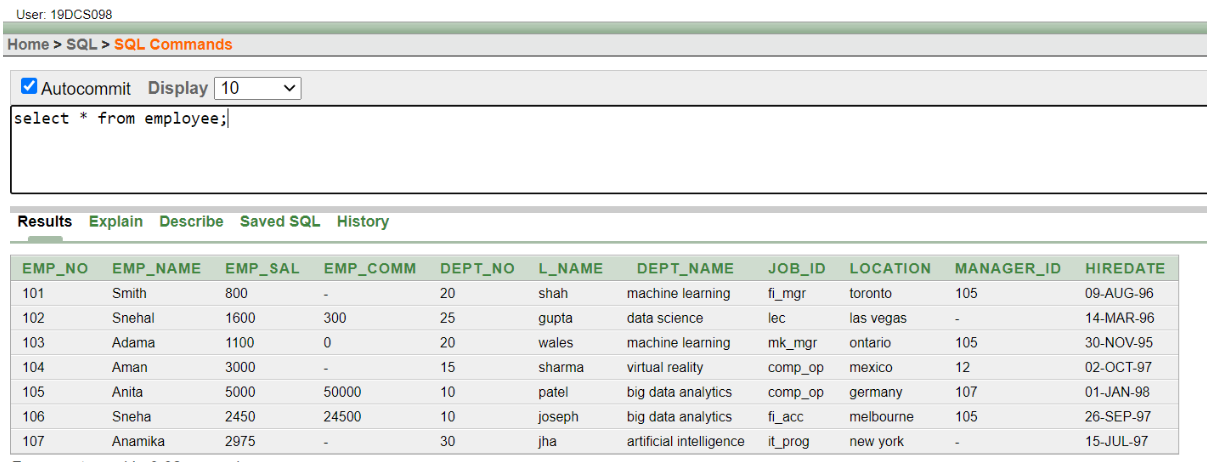
job_id	job_name	min_sal	max_sal
it_prog	Programmer	4000	10000
mk_mgr	Marketing manager	9000	15000
fi_mgr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000
comp_op	Computer Operator	1500	3000



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

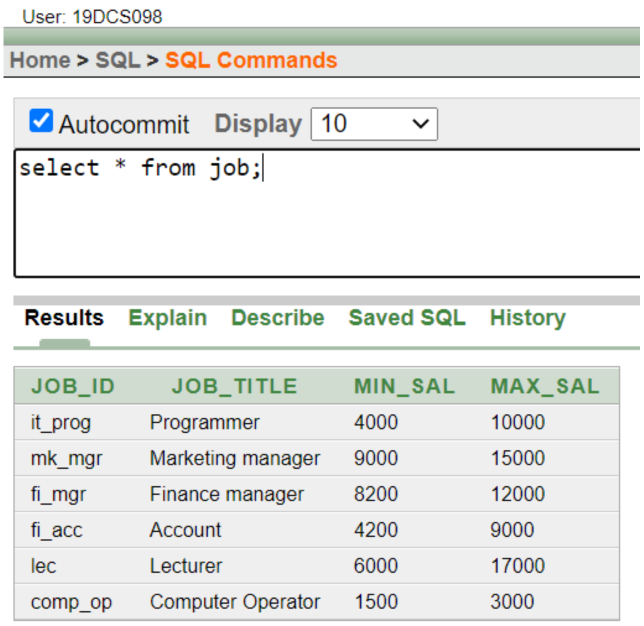
Perform following queries

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


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

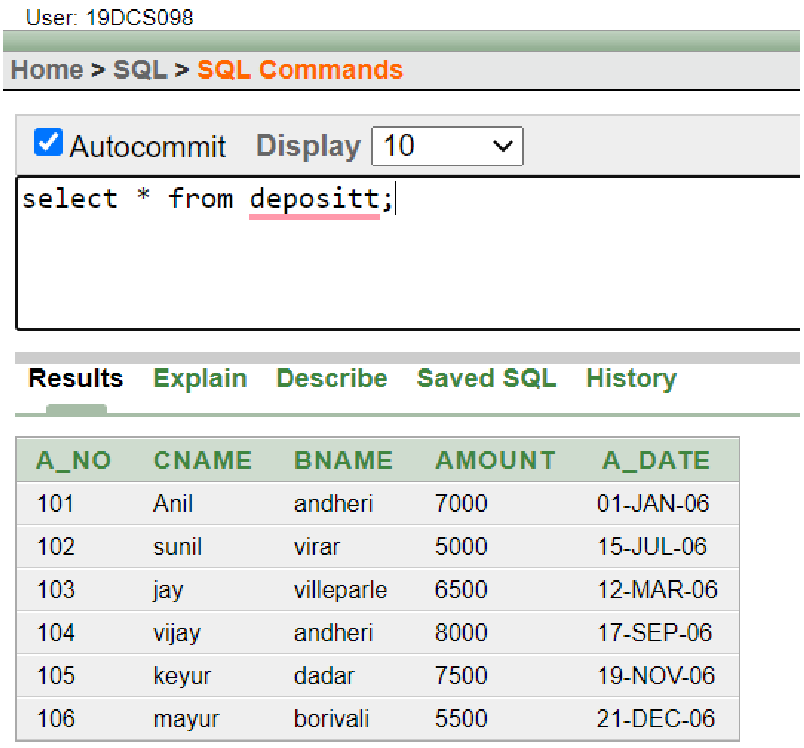
```
select * from employee;
```

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	2450	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



```
select * from job;
```

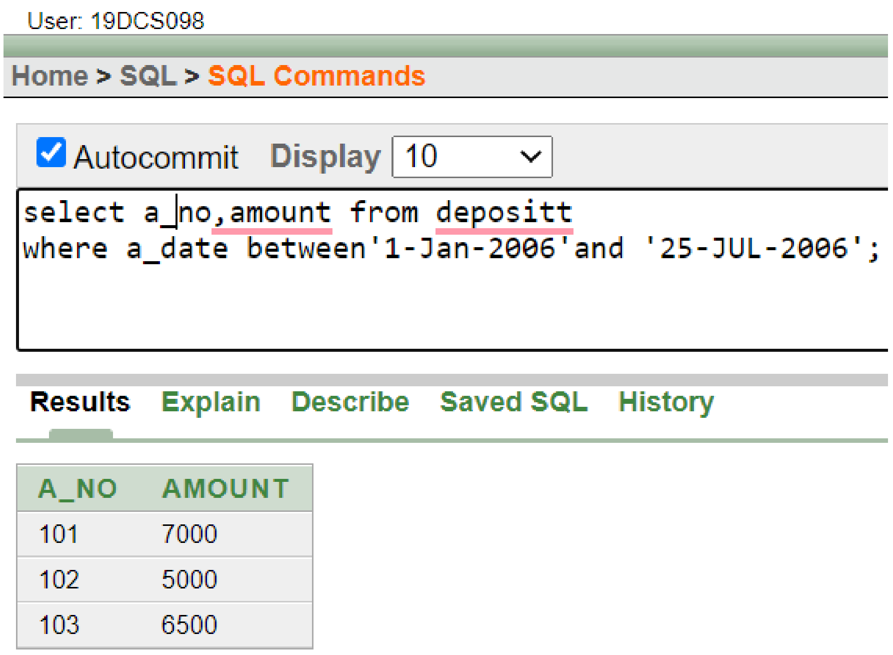
JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
it_prog	Programmer	4000	10000
mk_mgr	Marketing manager	9000	15000
fi_mgr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000
comp_op	Computer Operator	1500	3000



```
select * from depositt;
```

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

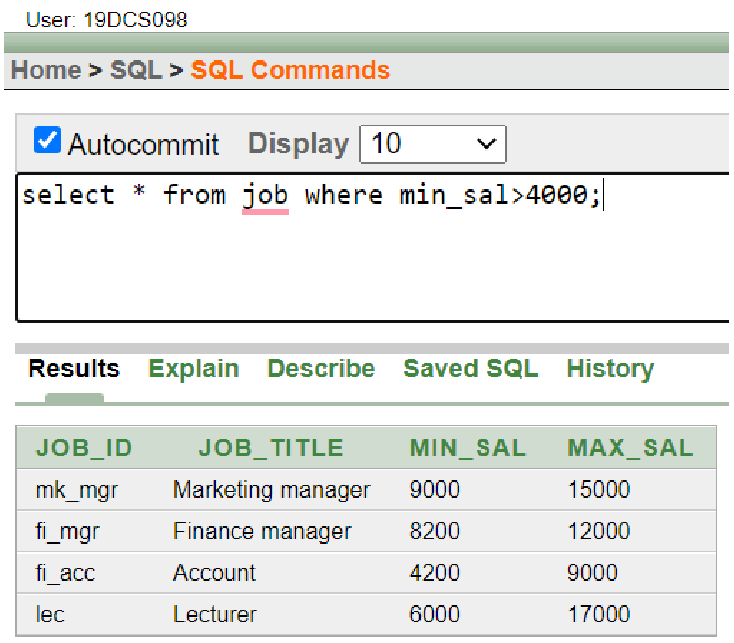
(2) Give details of account no. and deposited rupees of customers having account opened between dates 01-01-06 and 25-07-06.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
select a_no,amount from depositt
where a_date between'1-Jan-2006'and '25-JUL-2006';
```

A_NO	AMOUNT
101	7000
102	5000
103	6500

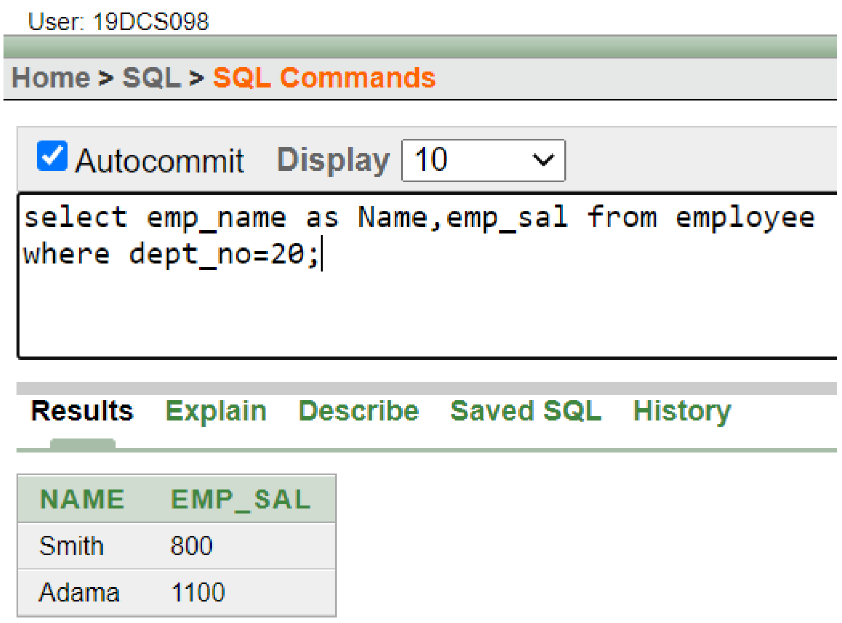
(3) Display all jobs with minimum salary is greater than 4000.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
select * from job where min_sal>4000;
```

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
mk_mgr	Marketing manager	9000	15000
fi_mgr	Finance manager	8200	12000
fi_acc	Account	4200	9000
lec	Lecturer	6000	17000

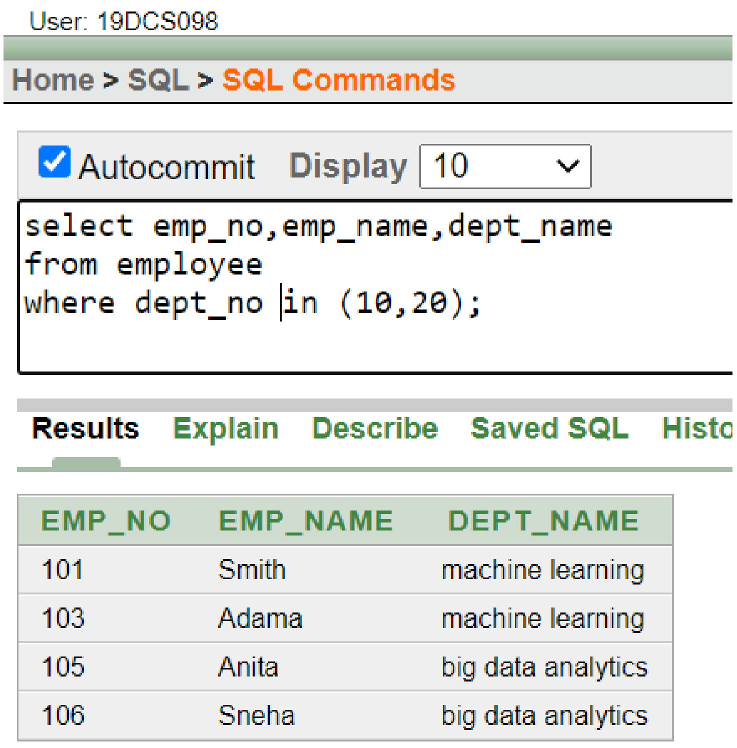
(4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

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

NAME	EMP_SAL
Smith	800
Adama	1100

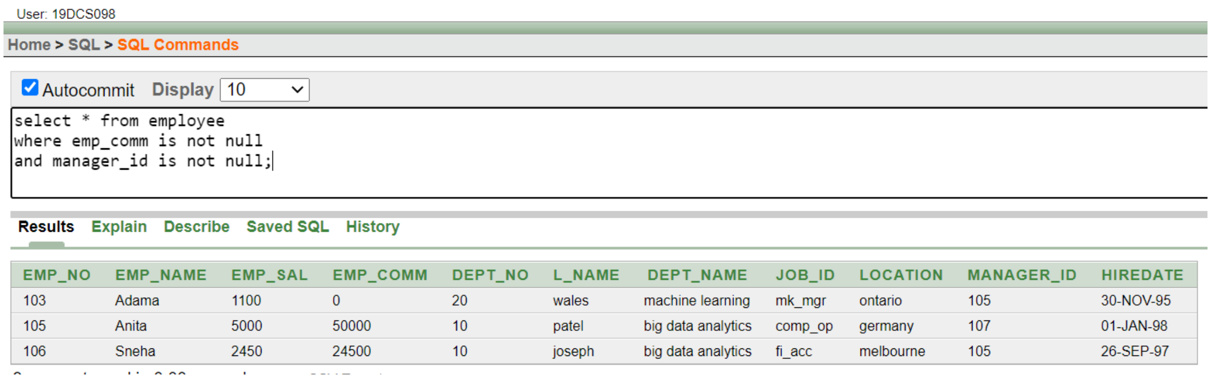
(5) Display employee no, name and department details of those employee whose department lies in (10,20).


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL Histo

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

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

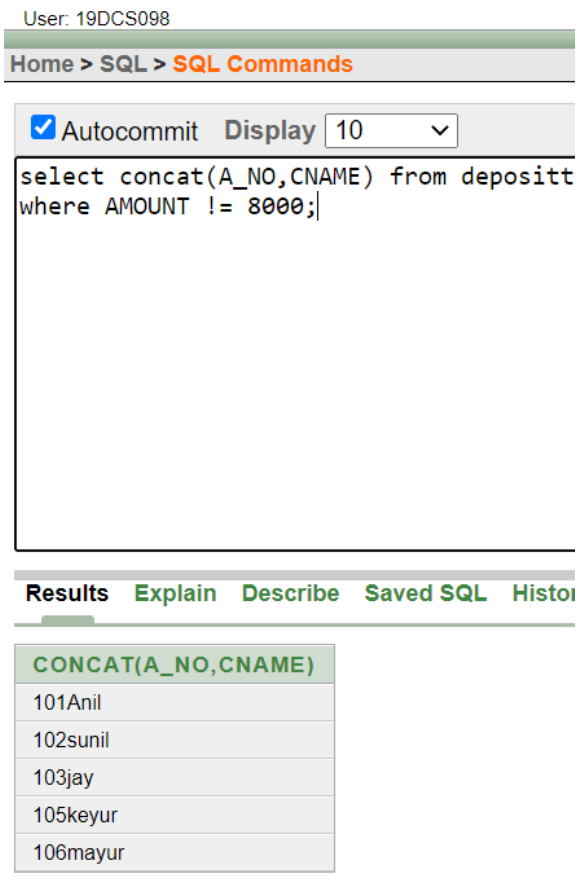
(6) **Display the non-null values of employees.**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
select * from employee
where emp_comm is not null
and manager_id is not null;
```

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	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

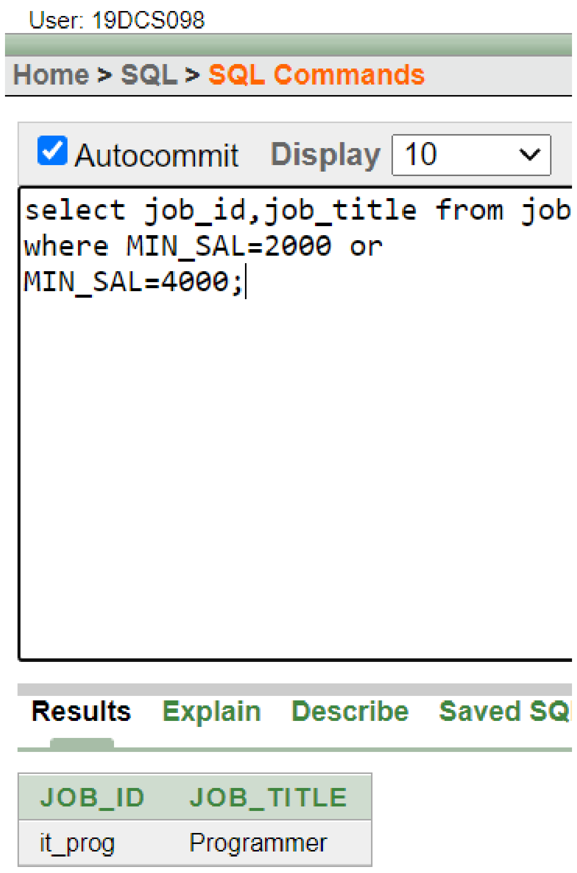
(7) **Display name of customer along with its account no (both column should be displayed as one) whose amount is not equal to 8000 Rs.**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL Histo

```
select concat(A_NO,CNAME) from depositt
where AMOUNT != 8000;
```

CONCAT(A_NO,CNAME)
101Anil
102sunil
103jay
105keyur
106mayur

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

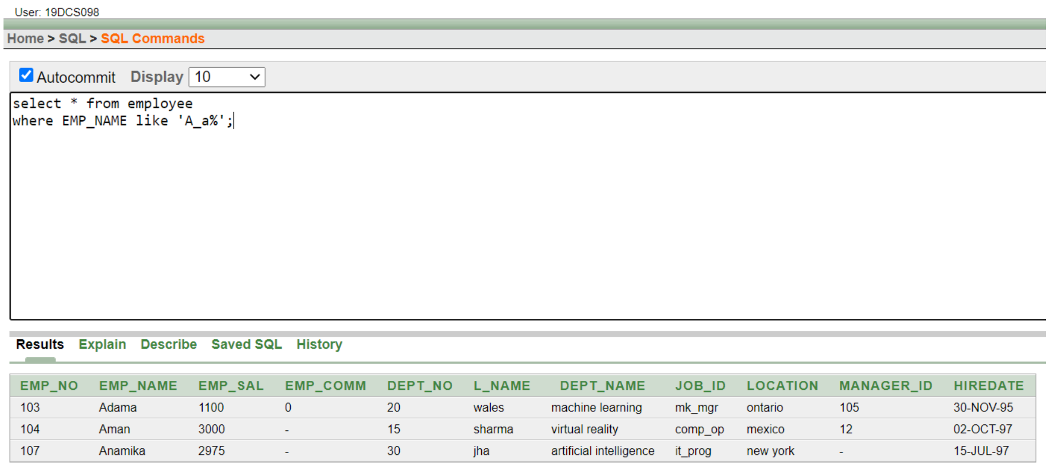


```
select job_id,job_title from job
where MIN_SAL=2000 or
MIN_SAL=4000;
```

JOB_ID	JOB_TITLE
it_prog	Programmer

To study various options of LIKE predicate

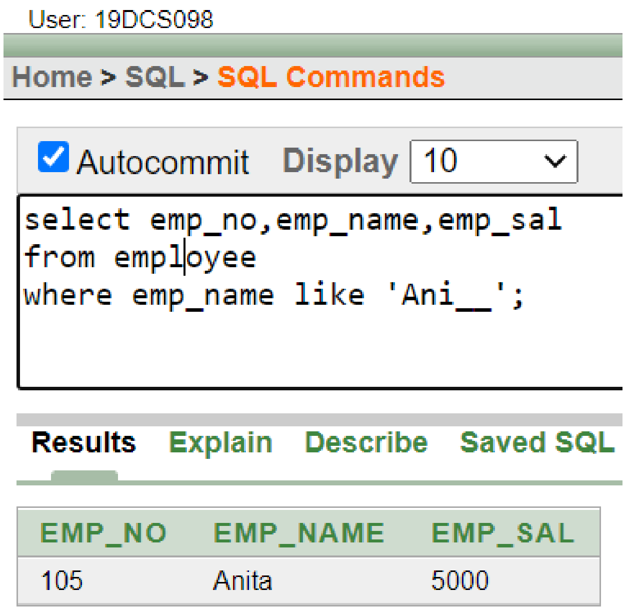
(1) Display all employee whose name start with 'A' and third character is "'a'.



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

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

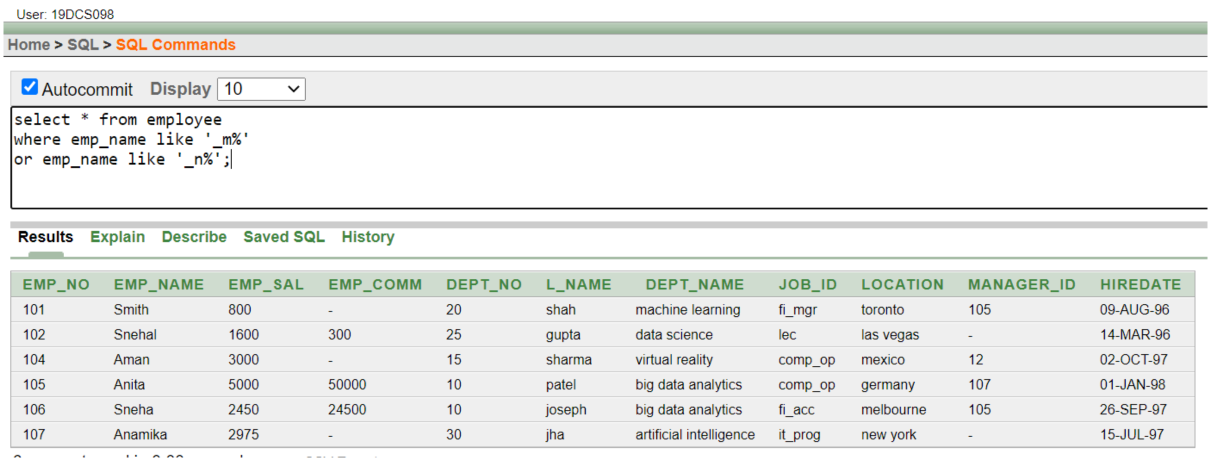
(2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'



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

EMP_NO	EMP_NAME	EMP_SAL
105	Anita	5000

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



User: 19DCS098

Home > SQL > SQL Commands

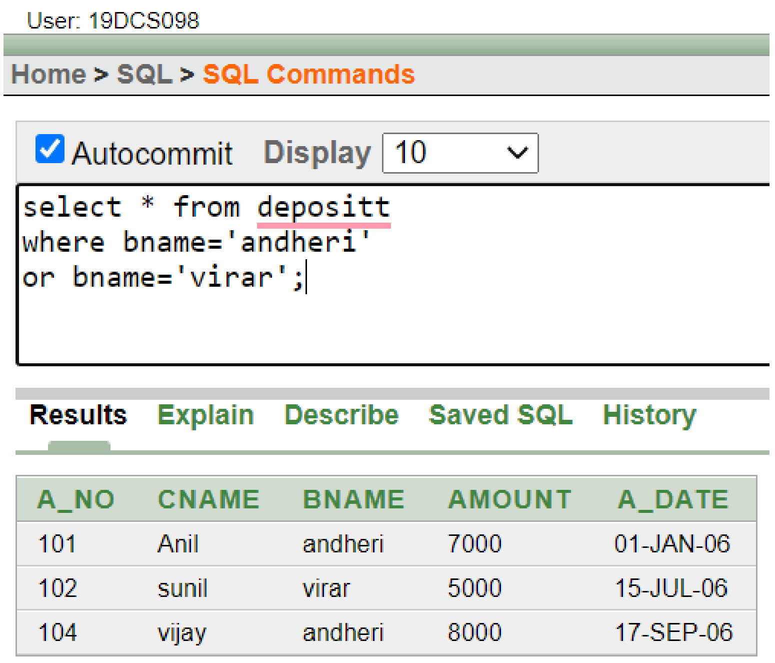
Autocommit Display 10

```
select * from employee
where emp_name like '_m%'
or emp_name like '_n%';
```

Results Explain Describe Saved SQL History

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	2450	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

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



User: 19DCS098

Home > SQL > SQL Commands

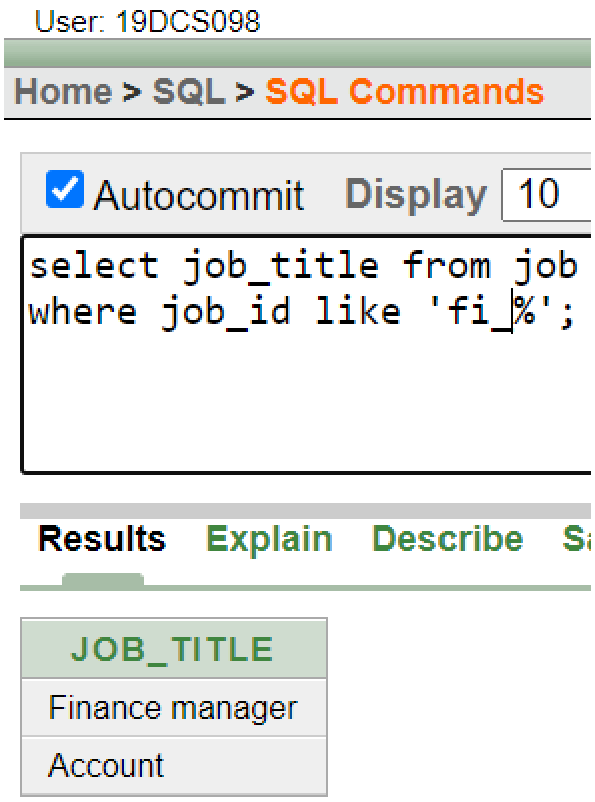
Autocommit Display 10

```
select * from depositt
where bname='andheri'
or bname='virar';
```

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
104	vijay	andheri	8000	17-SEP-06

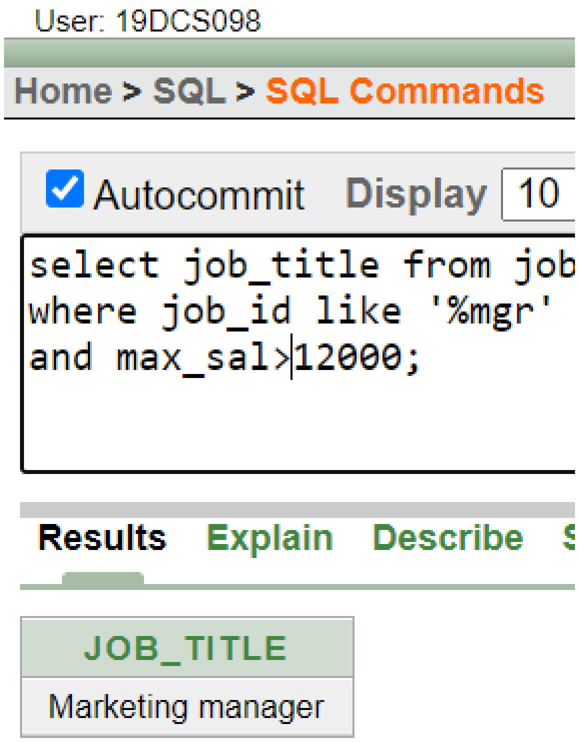
(5) Display the job name whose first three character in job id field is 'FI_'



```
select job_title from job
where job_id like 'fi_%';
```

JOB_TITLE
Finance manager
Account

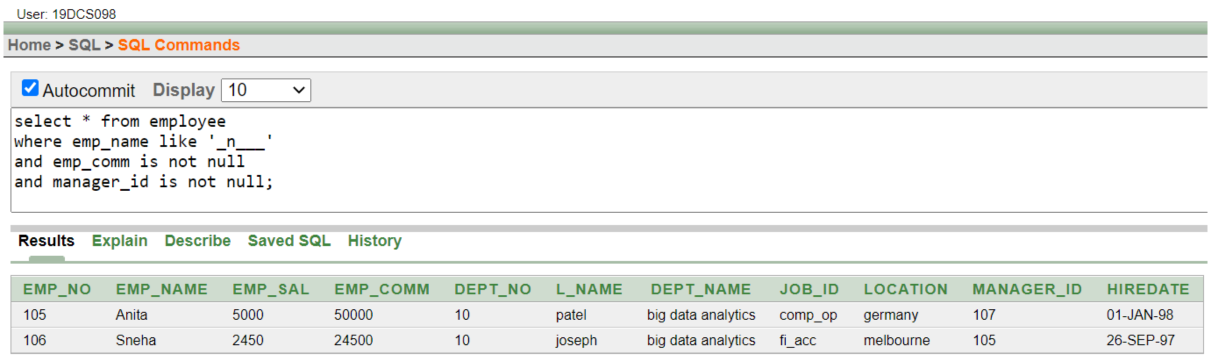
(6) Display the title/name of job who's last three character are '_MGR' and their maximum salary is greater than Rs 12000



```
select job_title from job
where job_id like '%mgr'
and max_sal>12000;
```

JOB_TITLE
Marketing manager

(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: 19DCS098

Home > SQL > SQL Commands

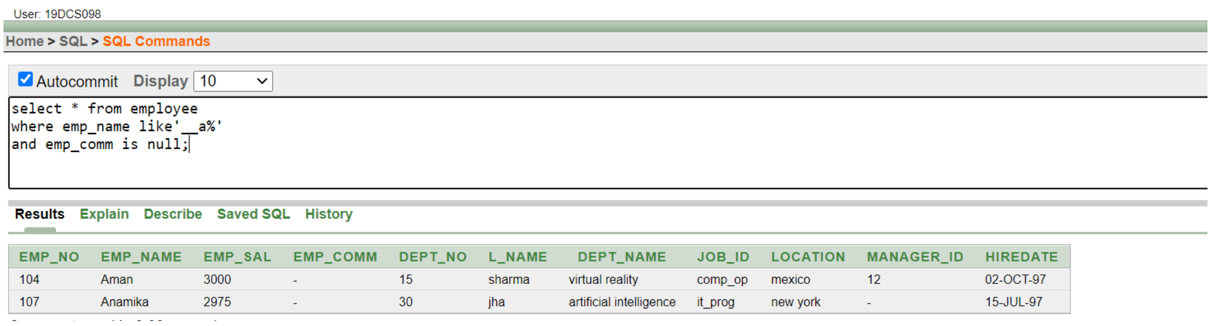
Autocommit Display 10

```
select * 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_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	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

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



User: 19DCS098

Home > SQL > SQL Commands

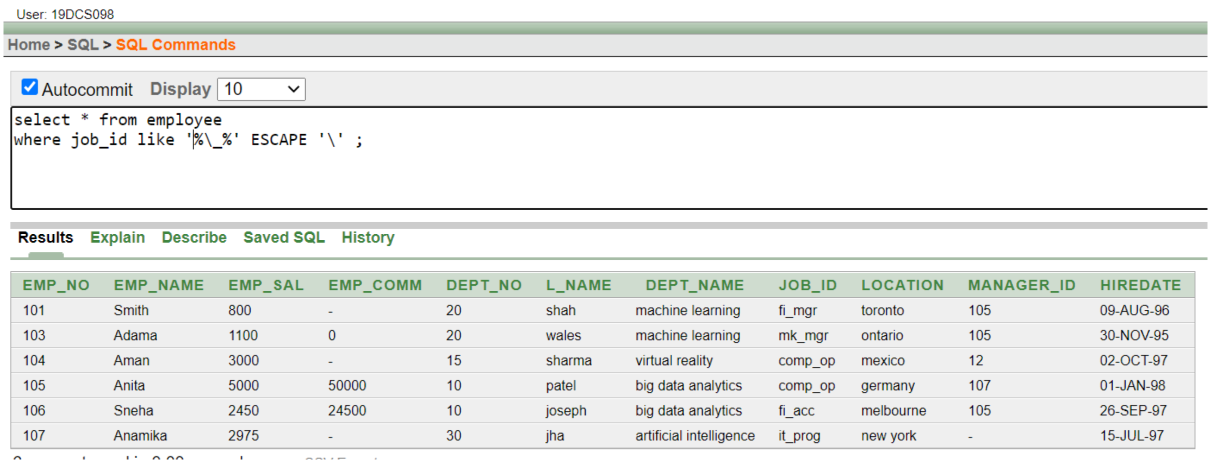
Autocommit Display 10

```
select * from employee
where emp_name like'__a%'
and emp_comm is 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
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

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



User: 19DCS098

Home > SQL > SQL Commands

Autocommit Display 10

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

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
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	2450	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

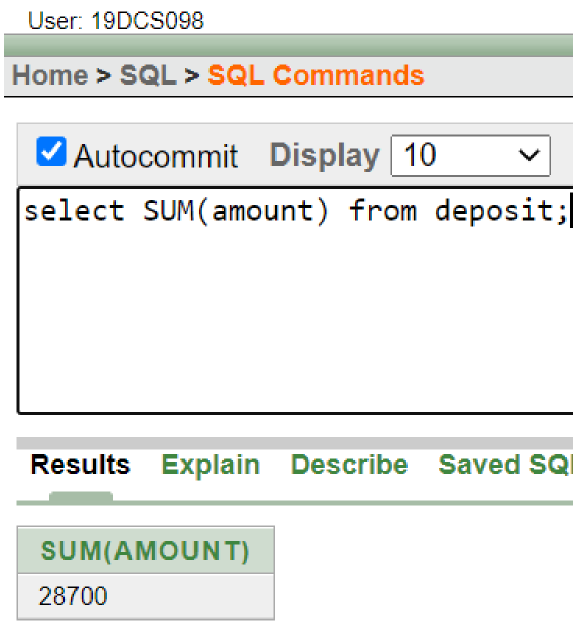
CONCLUSION:

In the above practical, we learned DDL,DML and the concept of ‘LIKE’

PRACTICAL-5

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

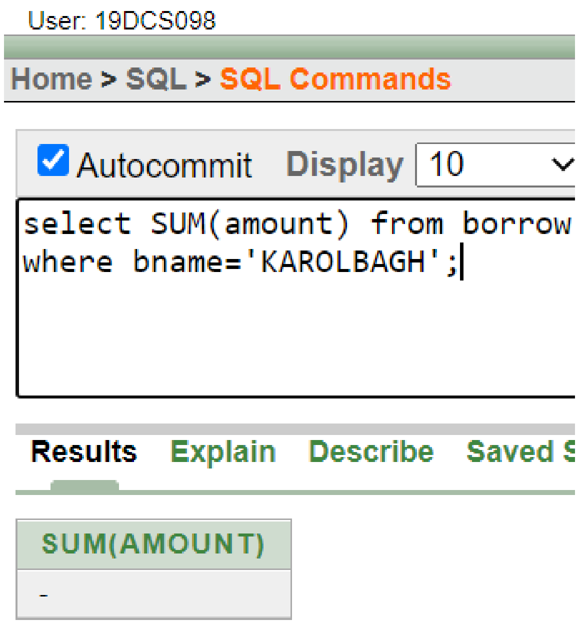
(1) List total deposit from deposit.



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

```
SUM(AMOUNT)
28700
```

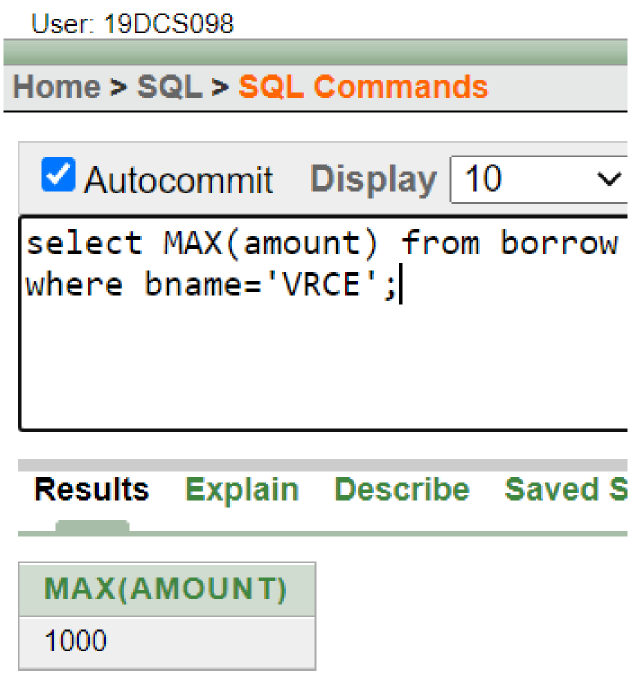
(2) List total loan from karolbagh branch



```
select SUM(amount) from borrow
where bname='KAROLBAGH';
```

```
SUM(AMOUNT)
-
```

(3) Give maximum loan from branch vrce



User: 19DCS098

Home > SQL > SQL Commands

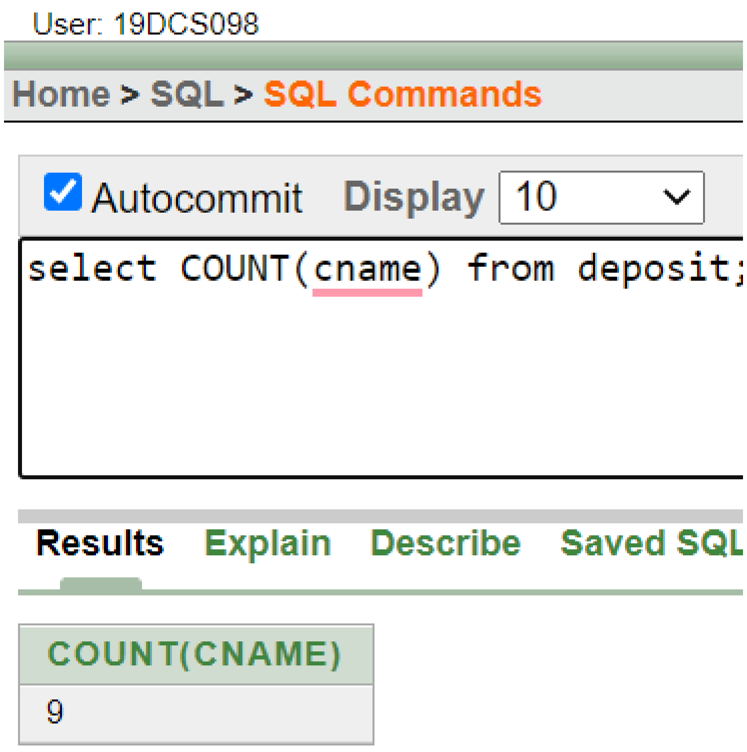
Autocommit Display 10

```
select MAX(amount) from borrow
where bname='VRCE';
```

Results Explain Describe Saved S

MAX(AMOUNT)
1000

(4) Count total number of customers



User: 19DCS098

Home > SQL > SQL Commands

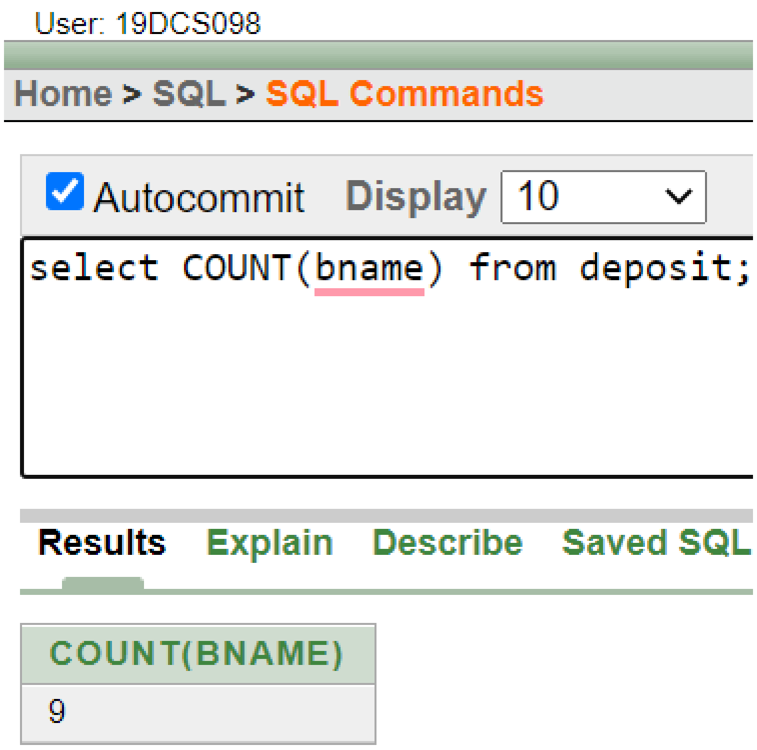
Autocommit Display 10

```
select COUNT(cname) from deposit;
```

Results Explain Describe Saved SQL

COUNT(CNAME)
9

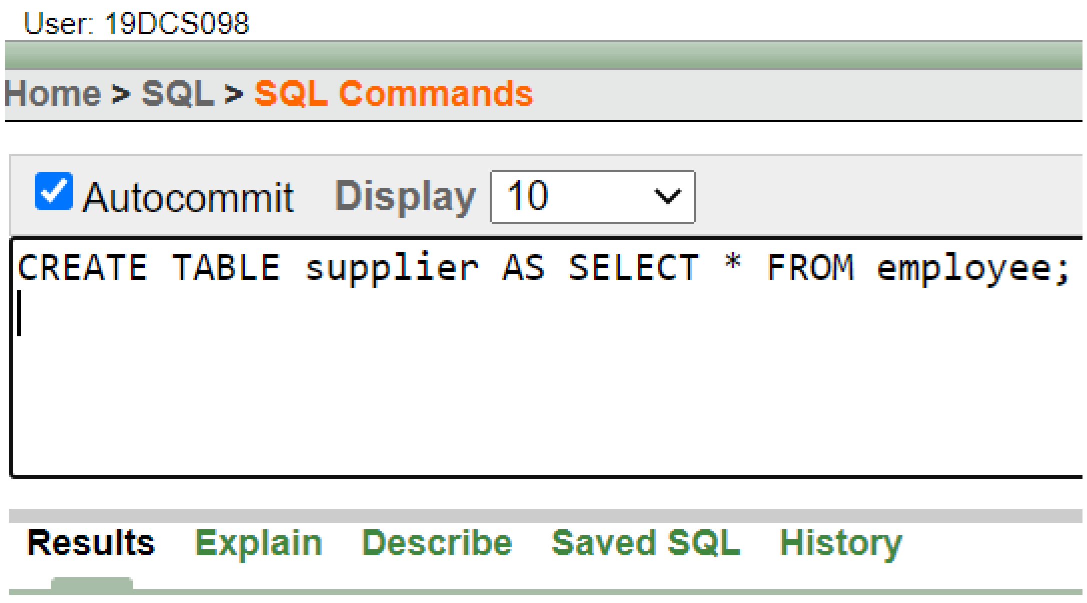
(5) Count total number of customer's cities



```
select COUNT(bname) from deposit;
```

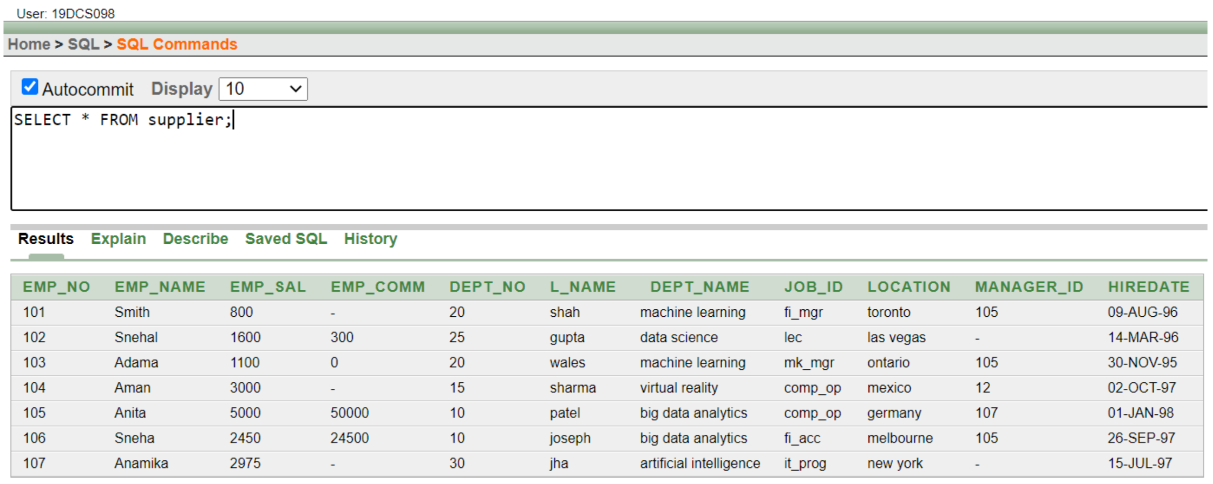
COUNT(BNAME)
9

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



```
CREATE TABLE supplier AS SELECT * FROM employee;
```

```
Table created.
```


User: 19DCS098

Home > SQL > SQL Commands

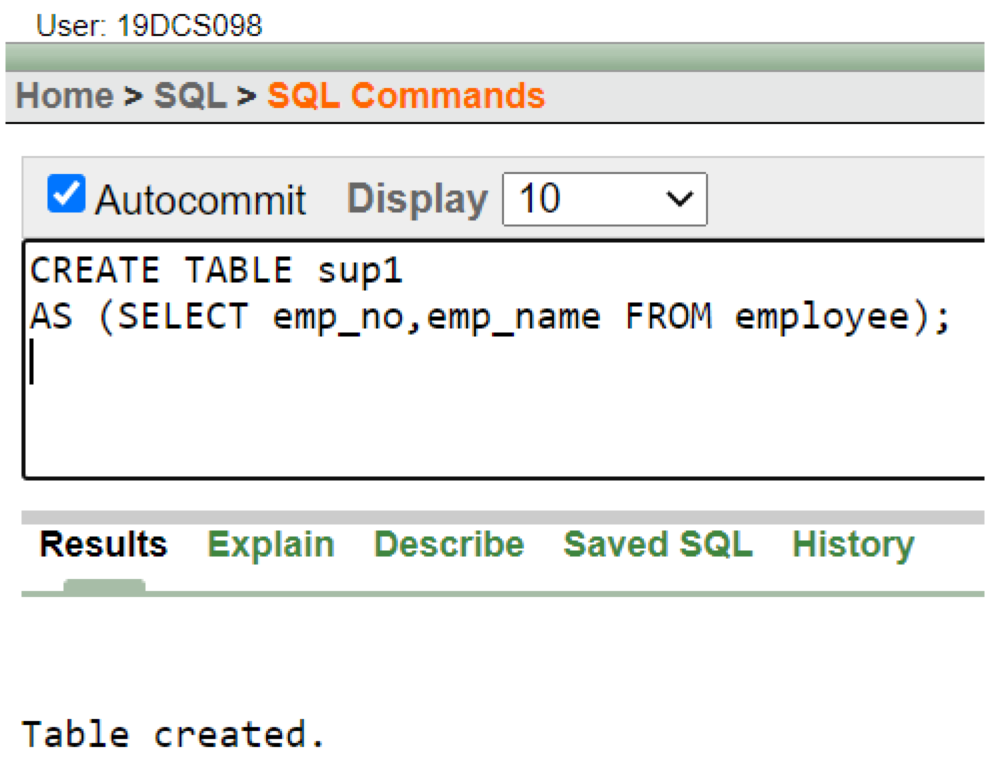
Autocommit Display 10

```
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	2450	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) Create table sup1 from employee with first two columns.


User: 19DCS098

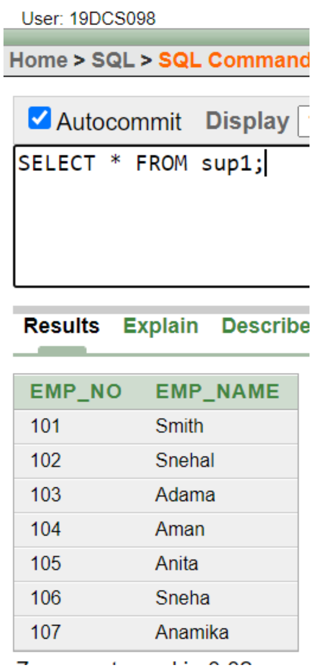
Home > SQL > SQL Commands

Autocommit Display 10

```
CREATE TABLE sup1
AS (SELECT emp_no,emp_name FROM employee);
```

Results Explain Describe Saved SQL History

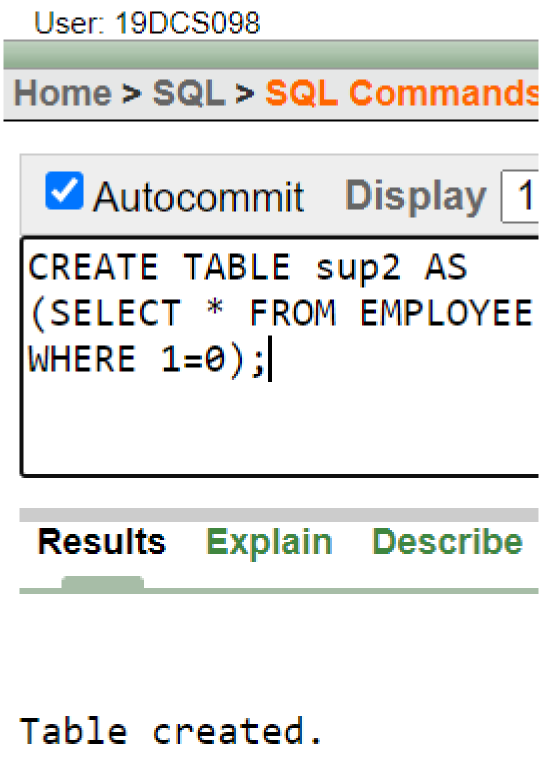
```
Table created.
```



```
SELECT * FROM sup1;
```

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

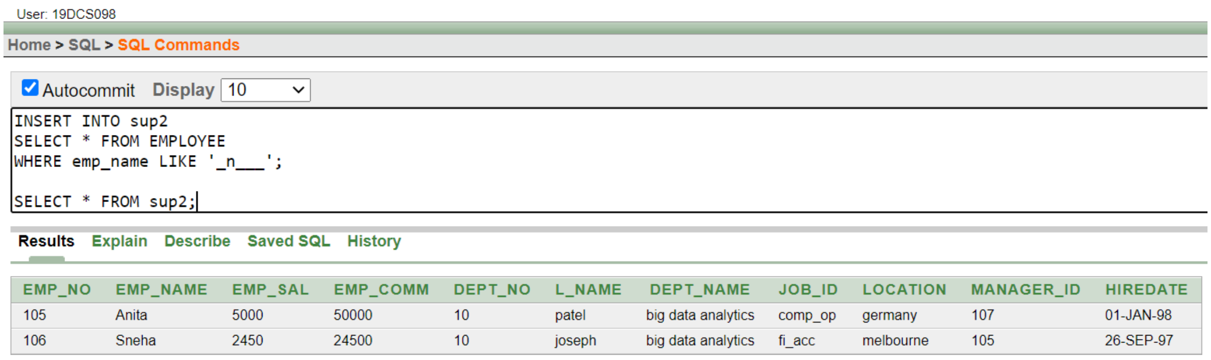
(8) Create table sup2 from employee with no data



```
CREATE TABLE sup2 AS
(SELECT * FROM EMPLOYEE
WHERE 1=0);
```

```
Table created.
```

(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: 19DCS098

Home > SQL > SQL Commands

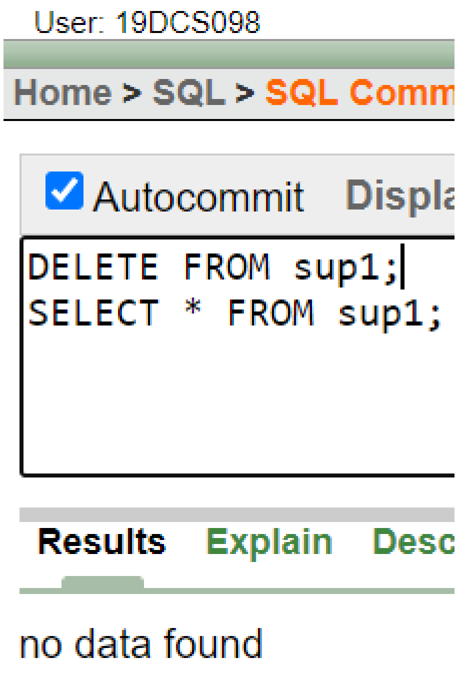
Autocommit Display 10

```
INSERT INTO sup2
SELECT * FROM EMPLOYEE
WHERE emp_name LIKE '_n___';

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	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

(10) Delete all the rows from sup1.


User: 19DCS098

Home > SQL > SQL Comm

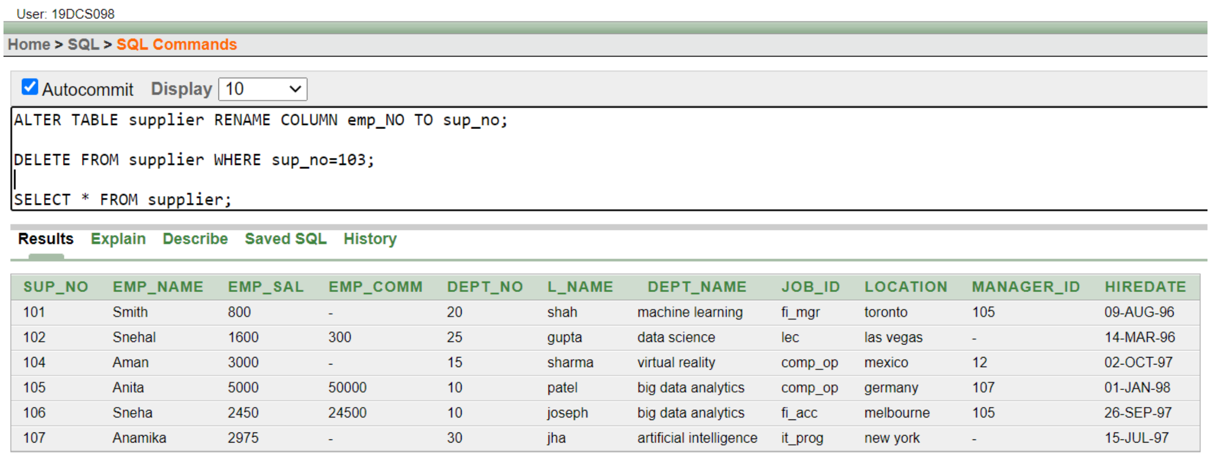
Autocommit Displa

```
DELETE FROM sup1;
SELECT * FROM sup1;
```

Results Explain Desc

no data found

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


User: 19DCS098

Home > SQL > SQL Commands

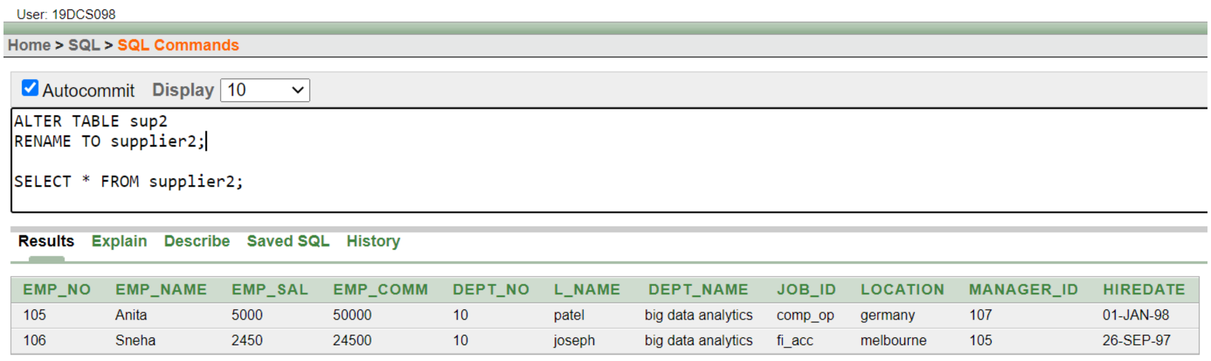
Autocommit Display 10

```
ALTER TABLE supplier RENAME COLUMN emp_NO TO sup_no;

DELETE FROM supplier WHERE sup_no=103;

SELECT * FROM supplier;
```

Results Explain Describe Saved SQL History

SUP_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	2450	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

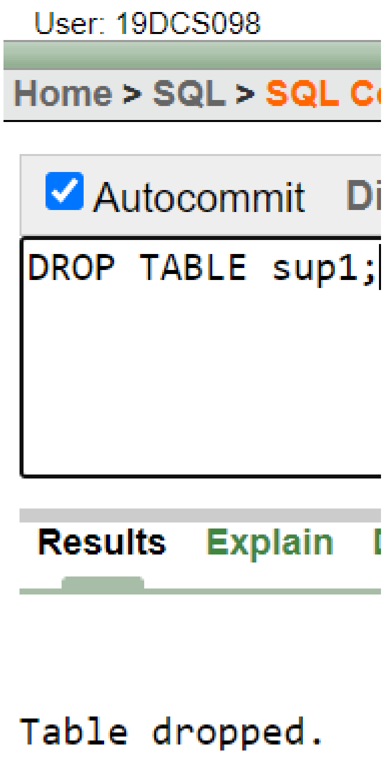
(12) Rename the table sup2.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
ALTER TABLE sup2
RENAME TO supplier2;

SELECT * FROM supplier2;
```

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	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

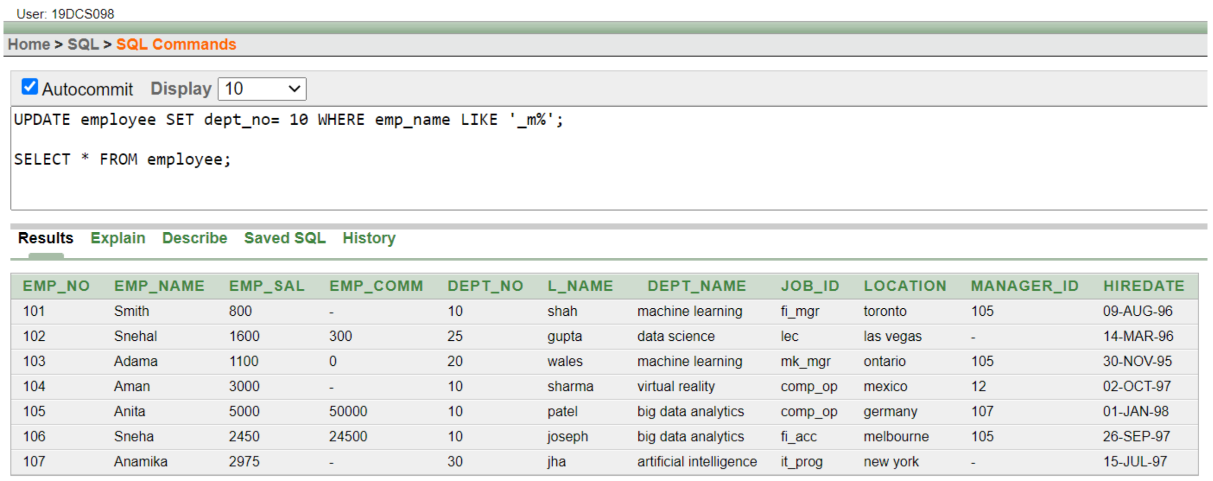
(13) Destroy table sup1 with all the data.



```
DROP TABLE sup1;
```

```
Table dropped.
```

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



User: 19DCS098

Home > SQL > SQL Commands

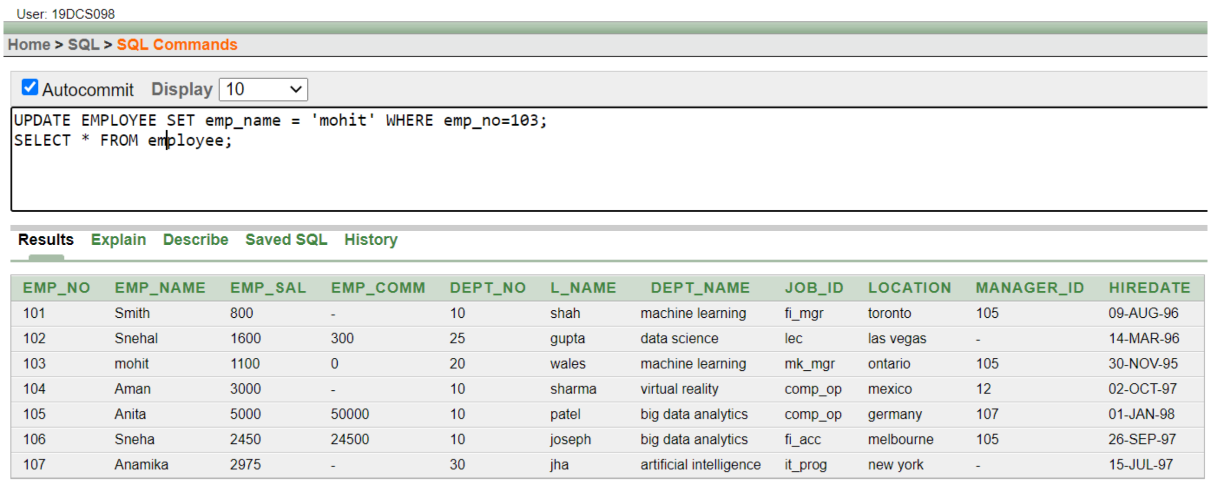
Autocommit Display 10

```
UPDATE employee SET dept_no= 10 WHERE emp_name LIKE '_m%';

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	-	10	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	-	10	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	2450	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

Update the value of employee name whose employee number is 103.



User: 19DCS098

Home > SQL > SQL Commands

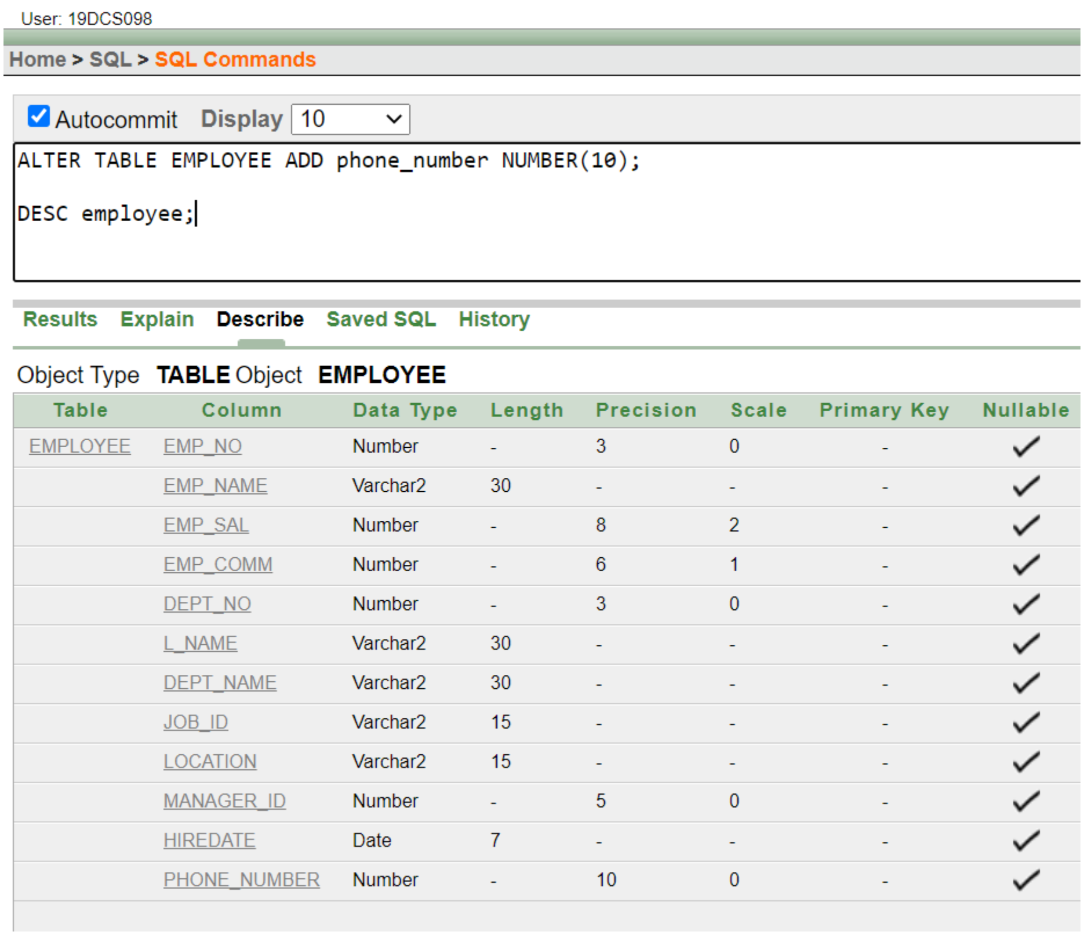
Autocommit Display 10

```
UPDATE EMPLOYEE SET emp_name = 'mohit' 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	-	10	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	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
104	Aman	3000	-	10	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	2450	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

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



User: 19DCS098

Home > SQL > SQL Commands

✓ Autocommit Display 10

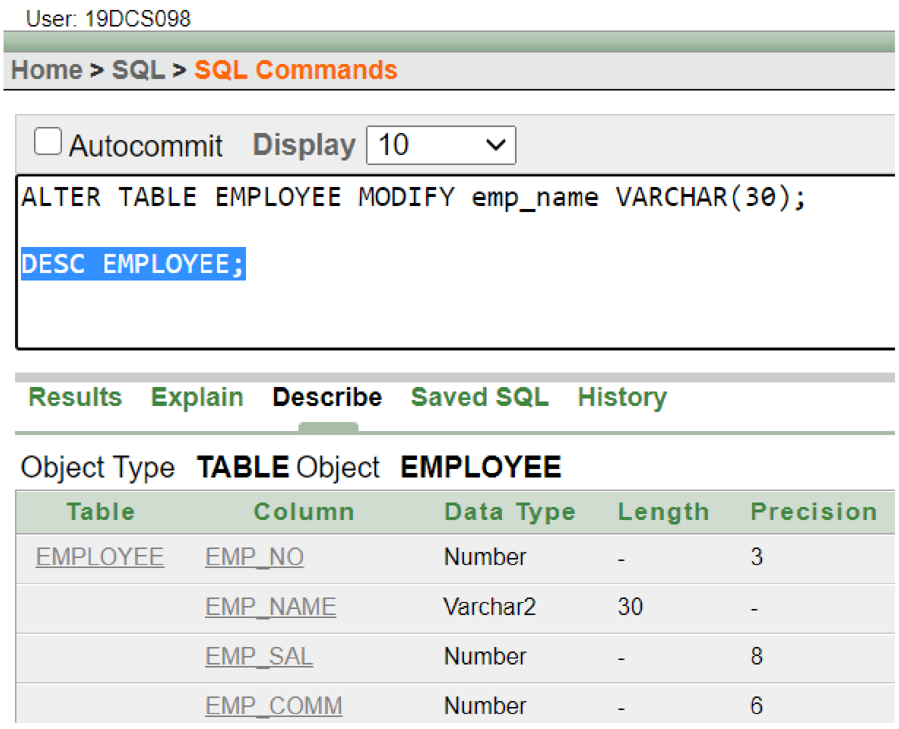
```
ALTER TABLE EMPLOYEE ADD phone_number NUMBER(10);

DESC employee;
```

Results Explain Describe Saved SQL History

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

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable
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	Number	-	10	0	-	✓

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



User: 19DCS098

Home > SQL > SQL Commands

Autocommit Display 10

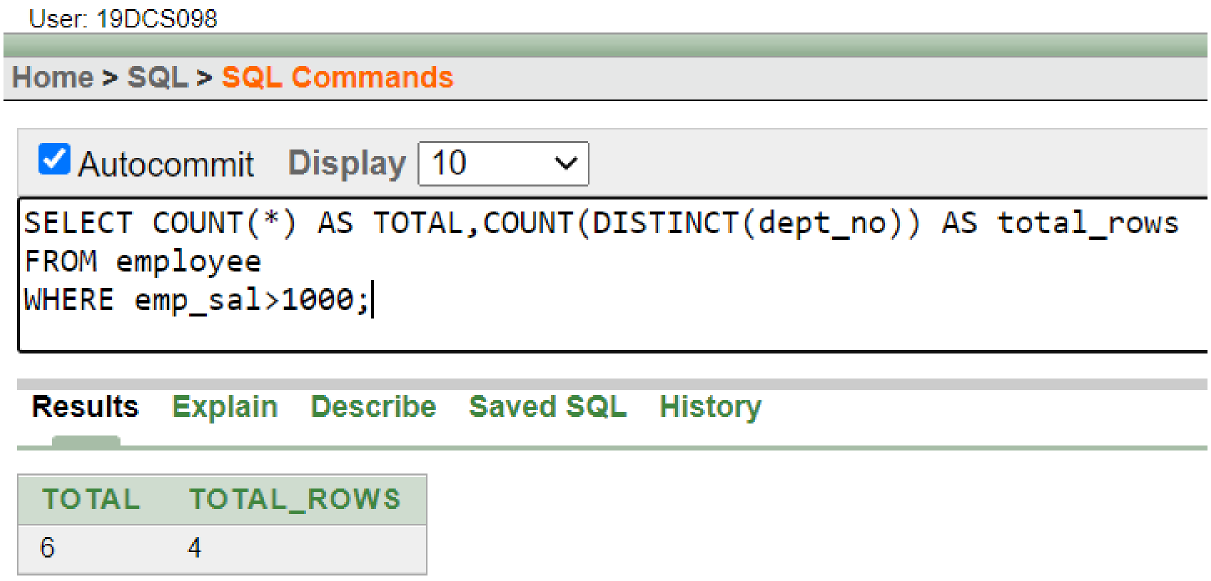
```
ALTER TABLE EMPLOYEE MODIFY emp_name VARCHAR(30);

DESC EMPLOYEE;
```

Results Explain Describe Saved SQL History

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

Table	Column	Data Type	Length	Precision
EMPLOYEE	EMP_NO	Number	-	3
	EMP_NAME	Varchar2	30	-
	EMP_SAL	Number	-	8
	EMP_COMM	Number	-	6

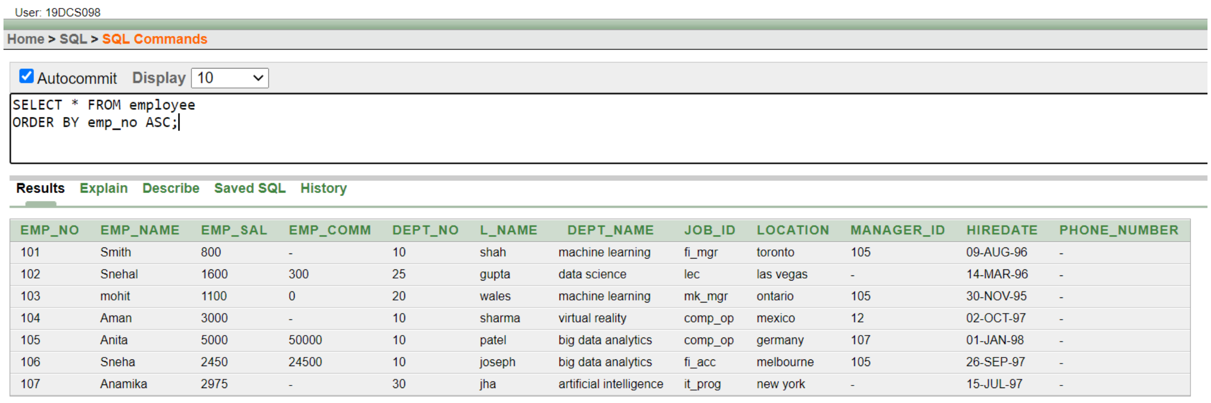
(16) Count the total no as well as distinct rows in dept_no column with a condition of salary greater than 1000 of employee


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT COUNT(*) AS TOTAL,COUNT(DISTINCT(dept_no)) AS total_rows
FROM employee
WHERE emp_sal>1000;
```

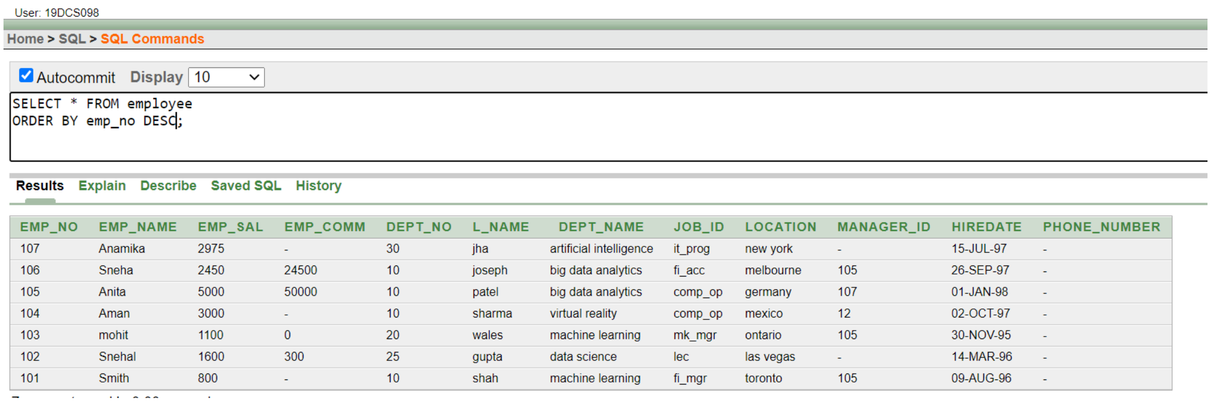
TOTAL	TOTAL_ROWS
6	4

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


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

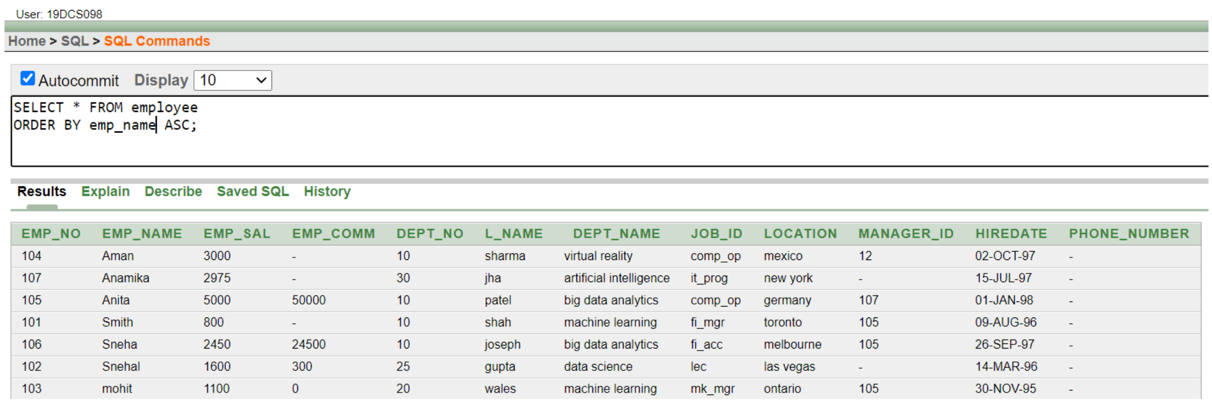
```
SELECT * FROM employee
ORDER BY emp_no ASC;
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
101	Smith	800	-	10	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	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	10	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	2450	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	-


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

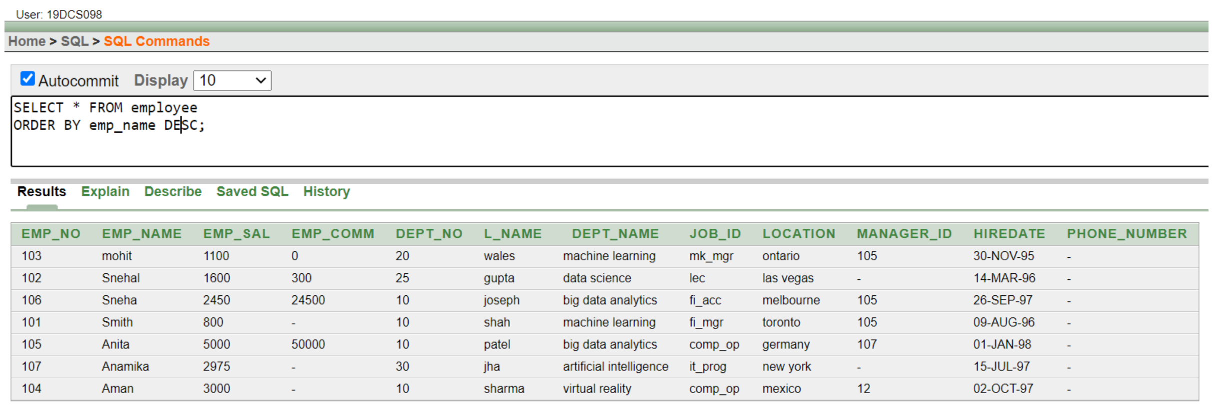
```
SELECT * FROM employee
ORDER BY emp_no DESC;
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
103	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT * FROM employee
ORDER BY emp_name ASC;
```

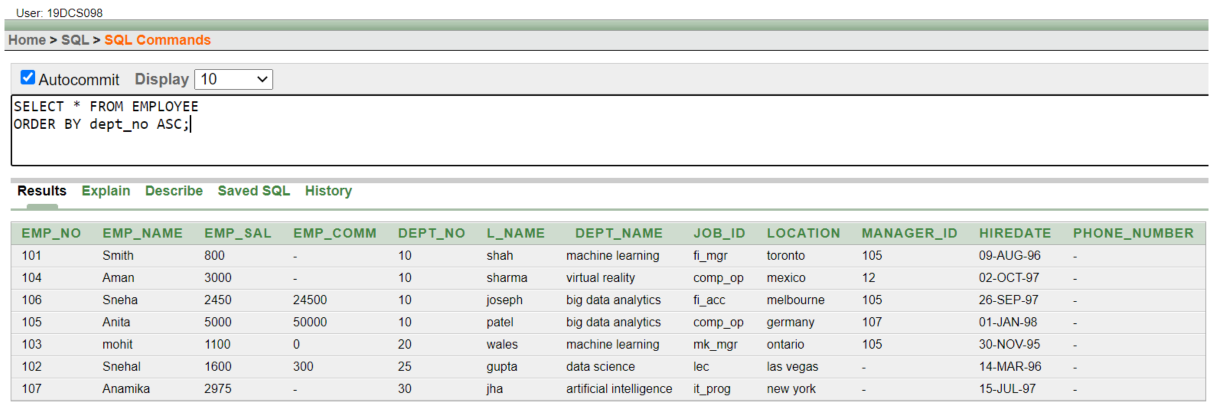
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
106	Sneha	2450	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	-
103	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT * FROM employee
ORDER BY emp_name DESC;
```

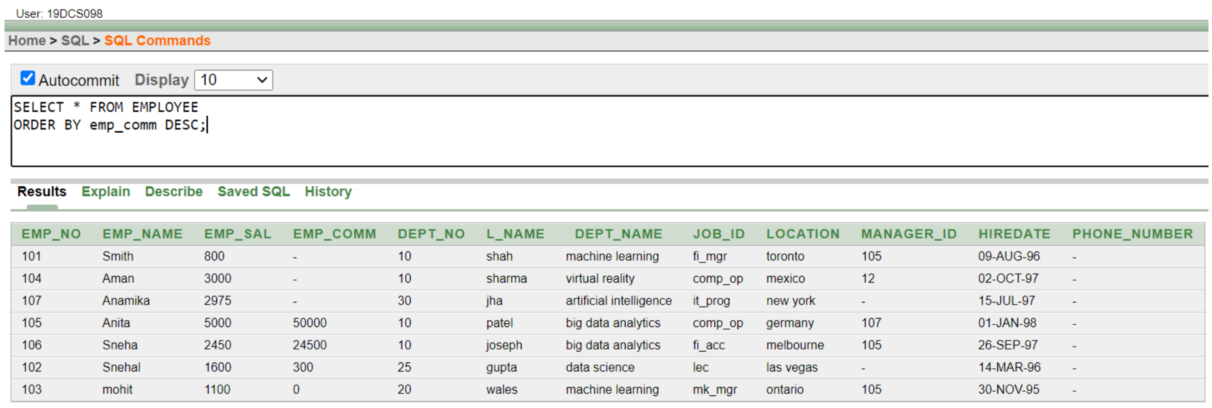
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
103	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-

(18) **Display the dept_no in ascending order and accordingly display emp_comm in descending order**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT * FROM EMPLOYEE
ORDER BY dept_no ASC;
```

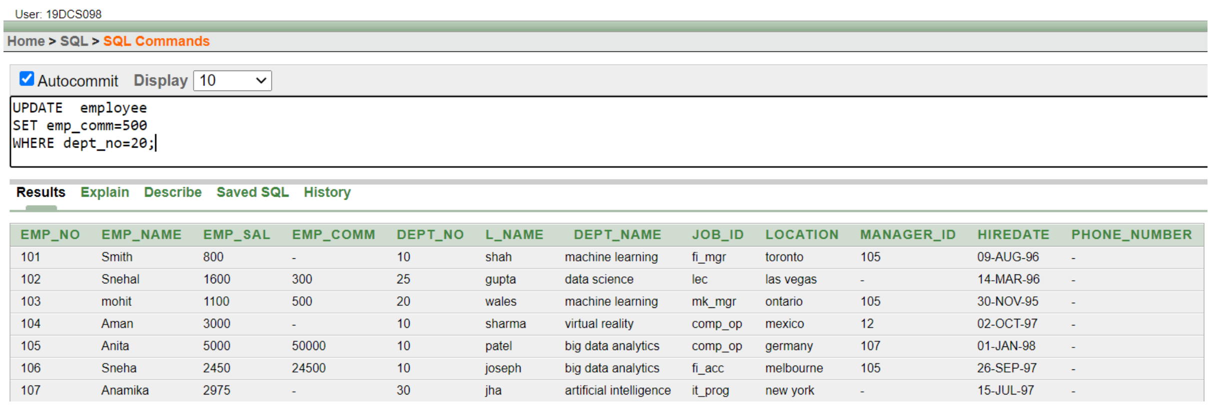
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-
103	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT * FROM EMPLOYEE
ORDER BY emp_comm DESC;
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
104	Aman	3000	-	10	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	-
106	Sneha	2450	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	-
103	mohit	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-

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


User: 19DCS098

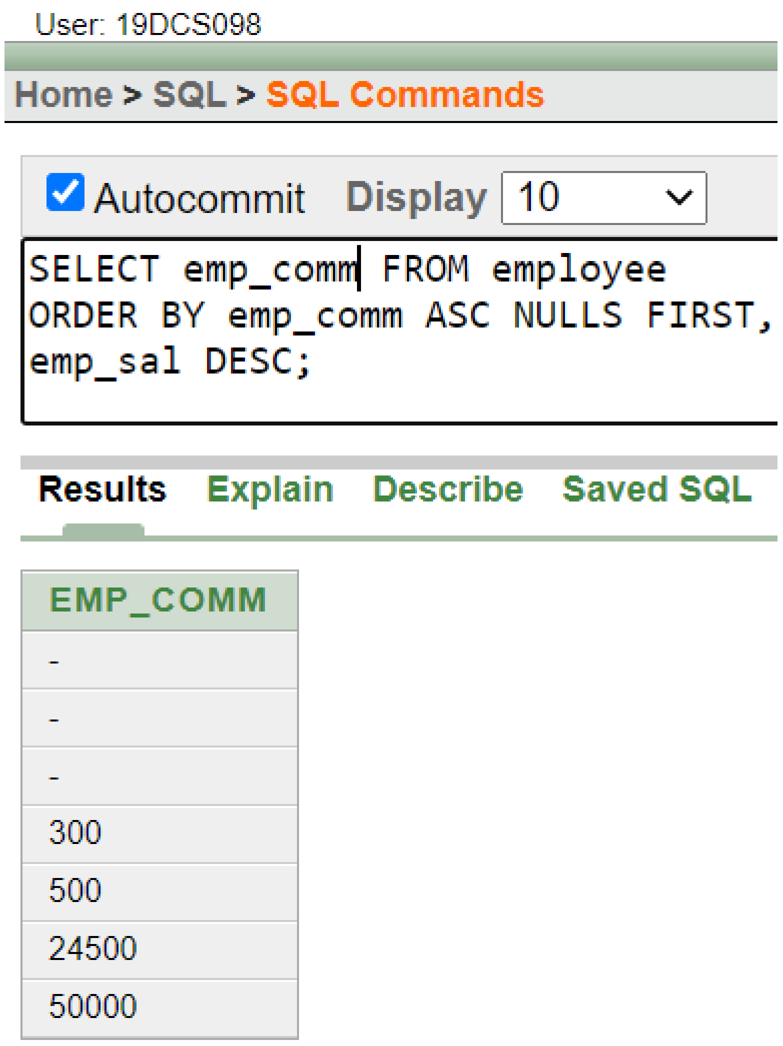
Home > SQL > SQL Commands

Autocommit Display 10

```
UPDATE  employee
SET emp_comm=500
WHERE dept_no=20;
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NUMBER
101	Smith	800	-	10	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	mohit	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	10	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	2450	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	-

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


User: 19DCS098

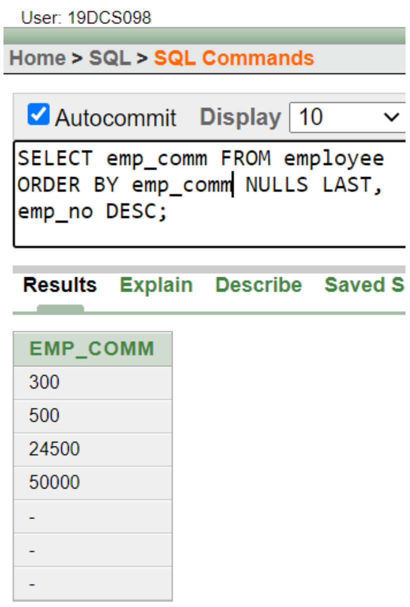
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT emp_comm FROM employee
ORDER BY emp_comm ASC NULLS FIRST,
emp_sal DESC;
```

EMP_COMM
-
-
-
300
500
24500
50000

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



```
SELECT emp_comm FROM employee
ORDER BY emp_comm NULLS LAST,
emp_no DESC;
```

```
EMP_COMM
300
500
24500
50000
-
-
-
```

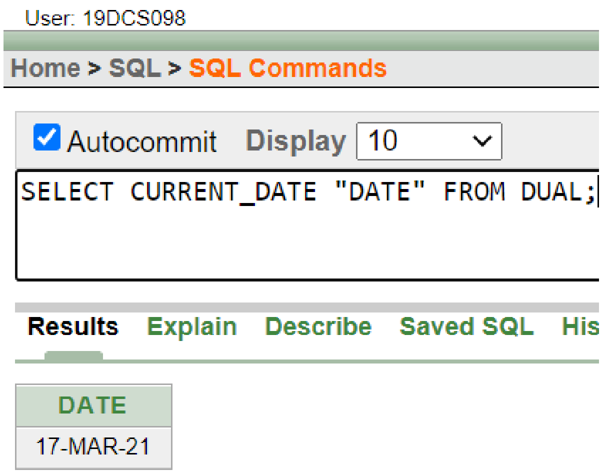
CONCLUSION:

In the above practical, we learned the various data manipulation commands and aggregate functions.

PRACTICAL-6

To study Single-row functions.

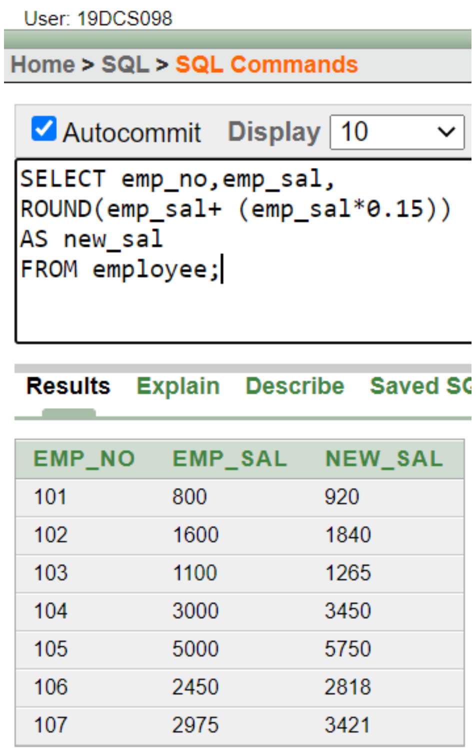
(1) Write a query to display the current date. Label the column Date


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL His

```
SELECT CURRENT_DATE "DATE" FROM DUAL;
```

DATE
17-MAR-21

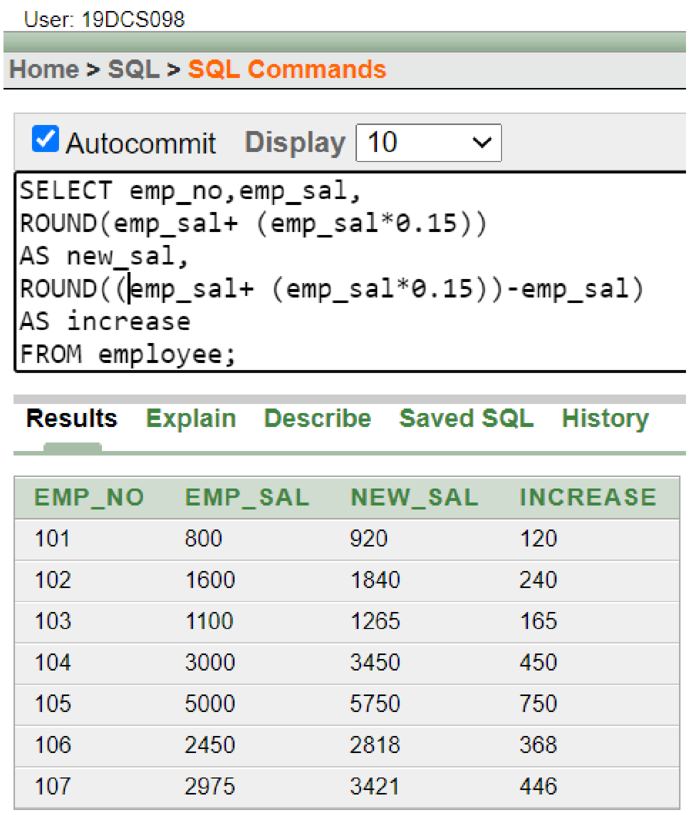
(2) For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number. Label the column New Salary



```
SELECT emp_no,emp_sal,
ROUND(emp_sal+ (emp_sal*0.15))
AS new_sal
FROM employee;
```

EMP_NO	EMP_SAL	NEW_SAL
101	800	920
102	1600	1840
103	1100	1265
104	3000	3450
105	5000	5750
106	2450	2818
107	2975	3421

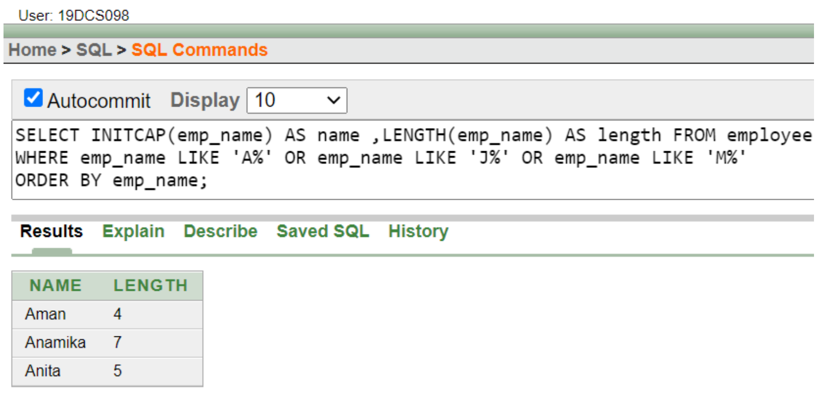
(3) Modify your query no (2) to add a column that subtracts the old salary from the new salary. Label the column Increase



```
SELECT emp_no,emp_sal,
ROUND(emp_sal+ (emp_sal*0.15))
AS new_sal,
ROUND((emp_sal+ (emp_sal*0.15))-emp_sal)
AS increase
FROM employee;
```

EMP_NO	EMP_SAL	NEW_SAL	INCREASE
101	800	920	120
102	1600	1840	240
103	1100	1265	165
104	3000	3450	450
105	5000	5750	750
106	2450	2818	368
107	2975	3421	446

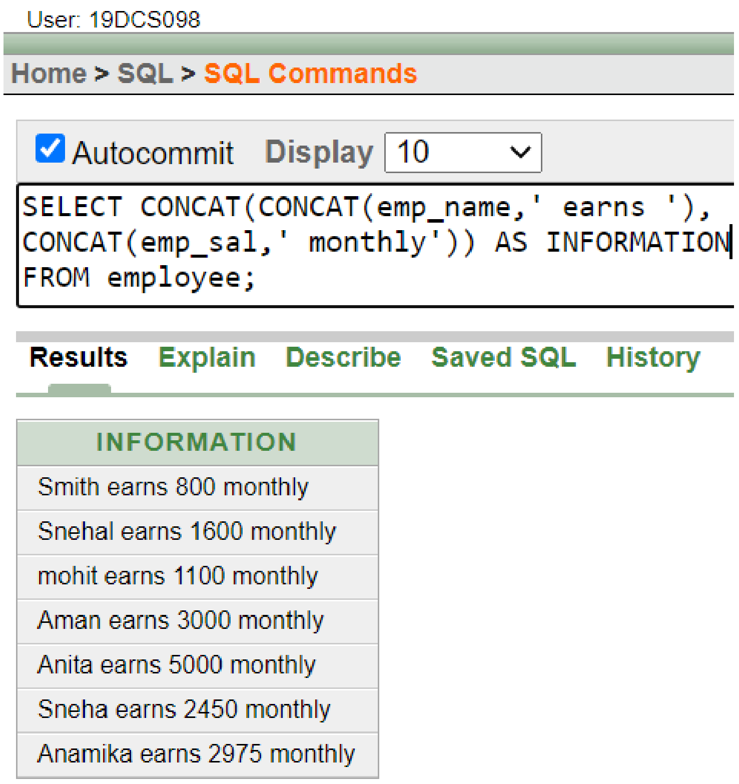
(4) Write a query that displays the employee's names with the first letter capitalized and all other letters lowercase, and the length of the names, for all employees whose name starts with J, A, or M. Give each column an appropriate label. Sort the results by the employees' last names.



```
SELECT INITCAP(emp_name) AS name ,LENGTH(emp_name) AS length FROM employee
WHERE emp_name LIKE 'A%' OR emp_name LIKE 'J%' OR emp_name LIKE 'M%'
ORDER BY emp_name;
```

NAME	LENGTH
Aman	4
Anamika	7
Anita	5

(5) Write a query that produces the following for each employee: earns monthly


User: 19DCS098

Home > SQL > SQL Commands

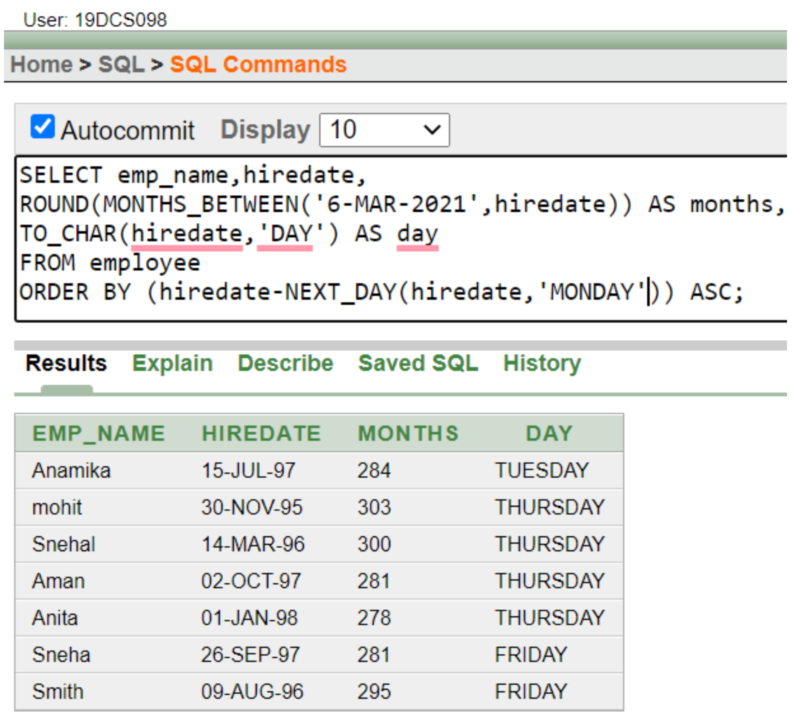
Autocommit Display 10

```
SELECT CONCAT(CONCAT(emp_name,' earns '),
CONCAT(emp_sal,' monthly')) AS INFORMATION
FROM employee;
```

Results Explain Describe Saved SQL History

INFORMATION
Smith earns 800 monthly
Snehal earns 1600 monthly
mohit earns 1100 monthly
Aman earns 3000 monthly
Anita earns 5000 monthly
Sneha earns 2450 monthly
Anamika earns 2975 monthly

(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

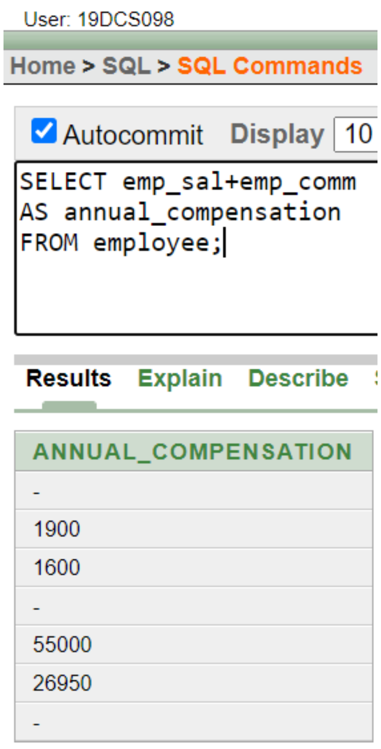


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT emp_name,hiredate,
ROUND(MONTHS_BETWEEN('6-MAR-2021',hiredate)) AS months,
TO_CHAR(hiredate,'DAY') AS day
FROM employee
ORDER BY (hiredate-NEXT_DAY(hiredate,'MONDAY')) ASC;
```

EMP_NAME	HIREDATE	MONTHS	DAY
Anamika	15-JUL-97	284	TUESDAY
mohit	30-NOV-95	303	THURSDAY
Snehal	14-MAR-96	300	THURSDAY
Aman	02-OCT-97	281	THURSDAY
Anita	01-JAN-98	278	THURSDAY
Sneha	26-SEP-97	281	FRIDAY
Smith	09-AUG-96	295	FRIDAY

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



```
SELECT emp_sal+emp_comm
AS annual_compensation
FROM employee;
```

ANNUAL_COMPENSATION
-
1900
1600
-
55000
26950
-

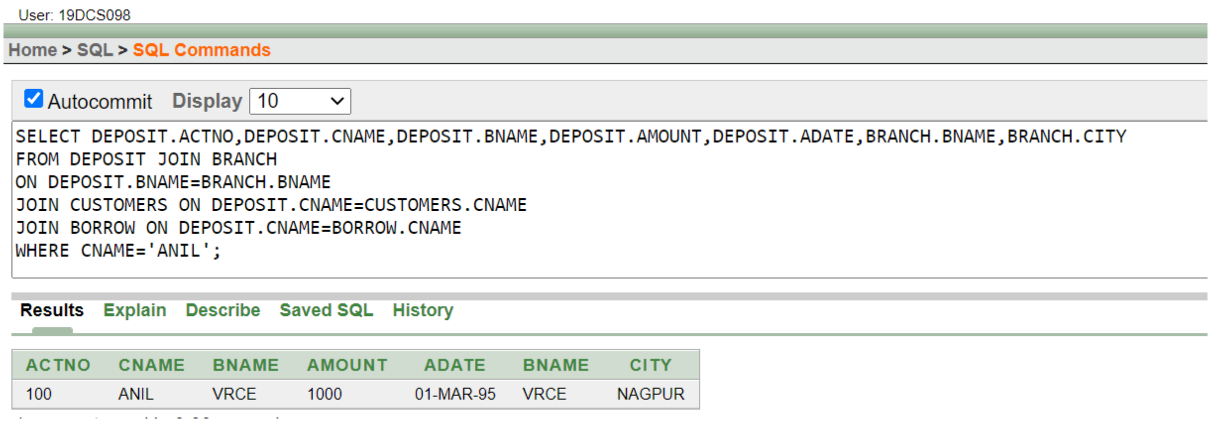
CONCLUSION:

In the above practical, we can learned the concept and application of single row functions.

PRACTICAL-7

Displaying data from Multiple Tables (join)

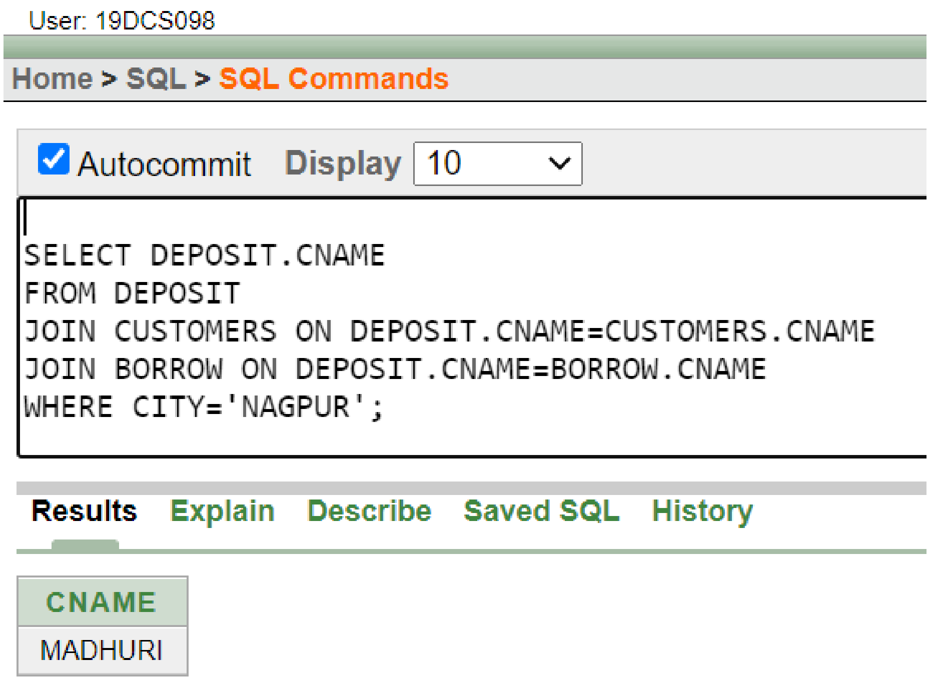
(1)Give details of customers ANIL.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT DEPOSIT.ACTNO,DEPOSIT.CNAME,DEPOSIT.BNAME,DEPOSIT.AMOUNT,DEPOSIT.ADATE,BRANCH.BNAME,BRANCH.CITY
FROM DEPOSIT JOIN BRANCH
ON DEPOSIT.BNAME=BRANCH.BNAME
JOIN CUSTOMERS ON DEPOSIT.CNAME=CUSTOMERS.CNAME
JOIN BORROW ON DEPOSIT.CNAME=BORROW.CNAME
WHERE CNAME='ANIL';
```

ACTNO	CNAME	BNAME	AMOUNT	ADATE	BNAME	CITY
100	ANIL	VRCE	1000	01-MAR-95	VRCE	NAGPUR

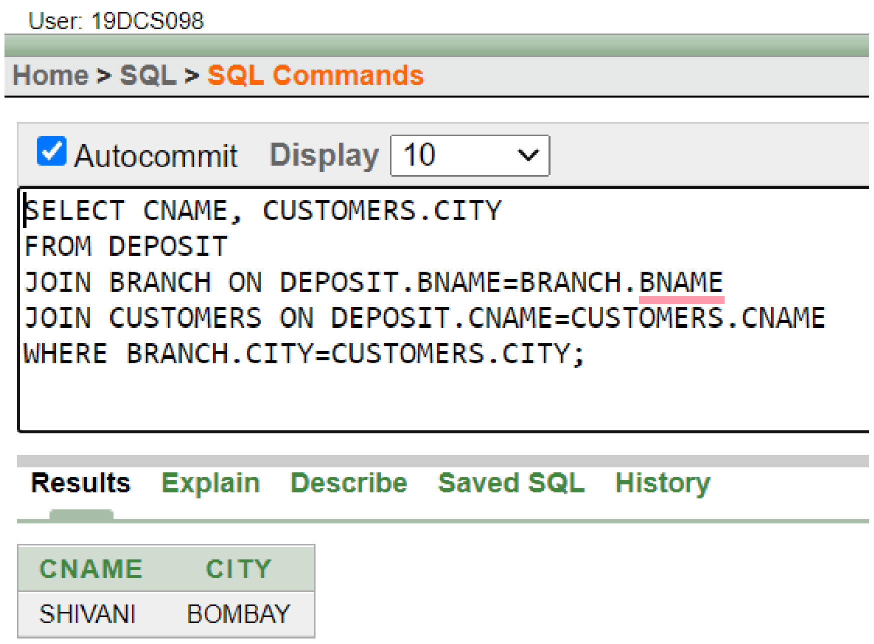
(2)Give name of customer who are borrowers and depositors and having living city Nagpur


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT DEPOSIT.CNAME
FROM DEPOSIT
JOIN CUSTOMERS ON DEPOSIT.CNAME=CUSTOMERS.CNAME
JOIN BORROW ON DEPOSIT.CNAME=BORROW.CNAME
WHERE CITY='NAGPUR';
```

CNAME
MADHURI

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


User: 19DCS098

Home > SQL > SQL Commands

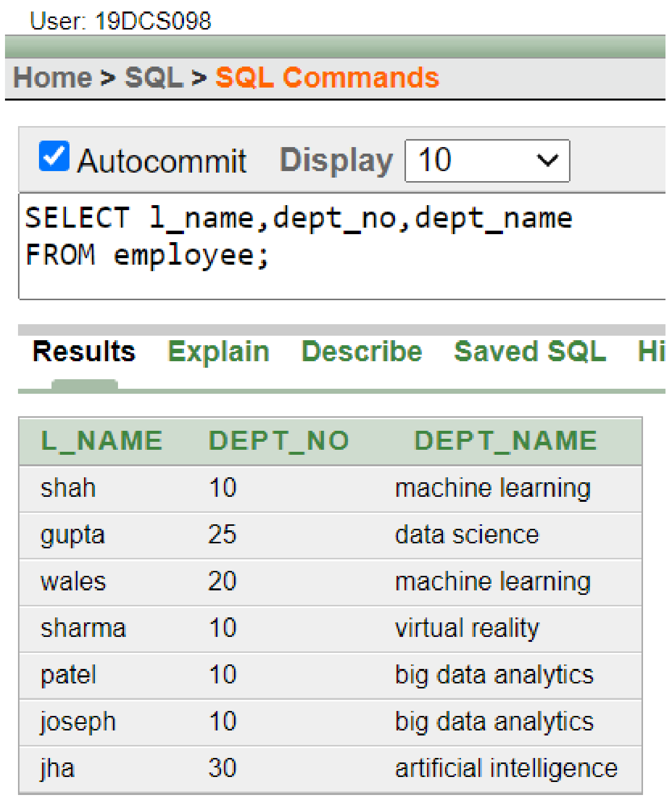
Autocommit Display 10

```
SELECT CNAME, CUSTOMERS.CITY
FROM DEPOSIT
JOIN BRANCH ON DEPOSIT.BNAME=BRANCH.BNAME
JOIN CUSTOMERS ON DEPOSIT.CNAME=CUSTOMERS.CNAME
WHERE BRANCH.CITY=CUSTOMERS.CITY;
```

Results Explain Describe Saved SQL History

CNAME	CITY
SHIVANI	BOMBAY

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


User: 19DCS098

Home > SQL > SQL Commands

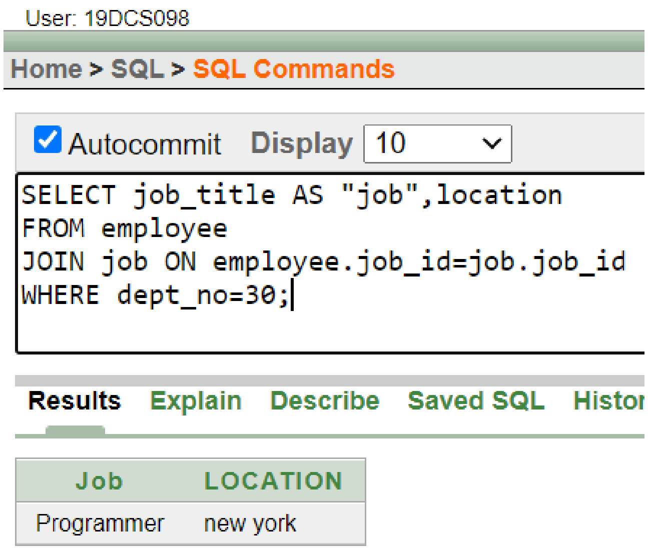
Autocommit Display 10

```
SELECT l_name,dept_no,dept_name
FROM employee;
```

Results Explain Describe Saved SQL Hi

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

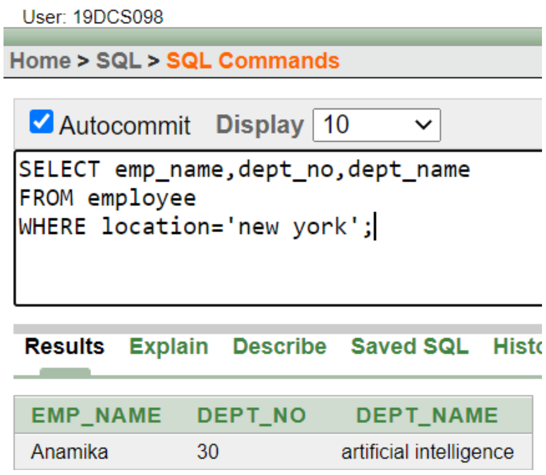
5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output



```
SELECT job_title AS "job",location
FROM employee
JOIN job ON employee.job_id=job.job_id
WHERE dept_no=30;
```

Job	LOCATION
Programmer	new york

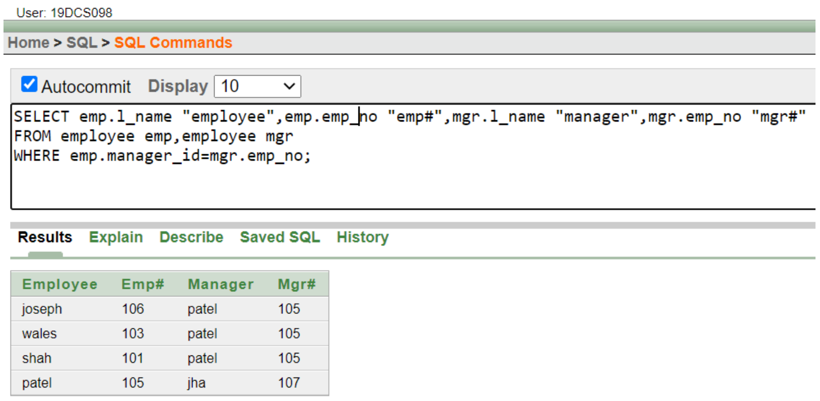
6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.



```
SELECT emp_name,dept_no,dept_name
FROM employee
WHERE location='new york';
```

EMP_NAME	DEPT_NO	DEPT_NAME
Anamika	30	artificial intelligence

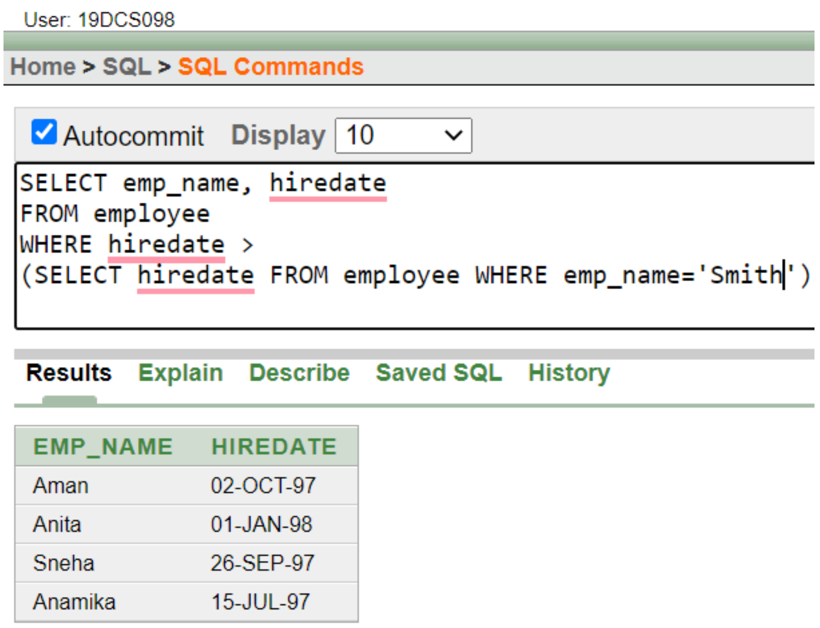
7) Display the employee last name and employee number along with their manager's last name and manager number. Label the columns Employee, Emp#, Manager, and Mgr#, respectively.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT emp.l_name "employee",emp.emp_no "emp#",mgr.l_name "manager",mgr.emp_no "mgr#"
FROM employee emp,employee mgr
WHERE emp.manager_id=mgr.emp_no;
```

Employee	Emp#	Manager	Mgr#
joseph	106	patel	105
wales	103	patel	105
shah	101	patel	105
patel	105	jha	107

8) Create a query to display the name and hire date of any employee hired after employee "smith"


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT emp_name, hiredate
FROM employee
WHERE hiredate >
(SELECT hiredate FROM employee WHERE emp_name='Smith')
```

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

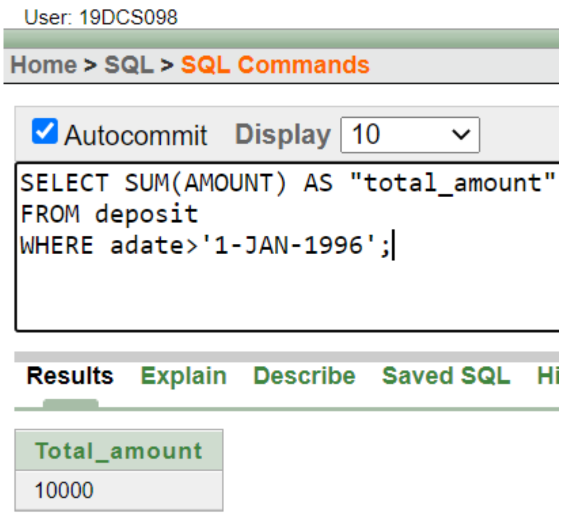
CONCLUSION:

In the above Practical, we learned the concept of JOINS.

Practical-8

To apply the concept of Aggregating Data using Group functions.

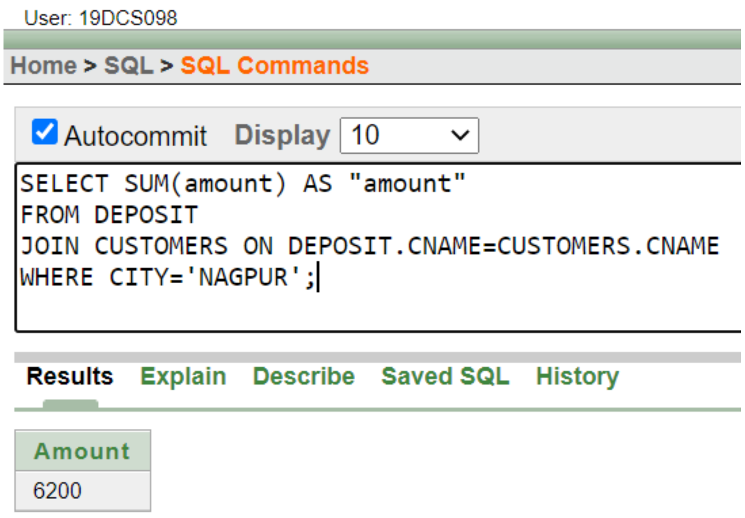
1) **List total deposit of customer having account date after 1-jan-96.**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL Hi

```
SELECT SUM(AMOUNT) AS "total_amount"
FROM deposit
WHERE adate>'1-JAN-1996';
```

```
Total_amount
10000
```

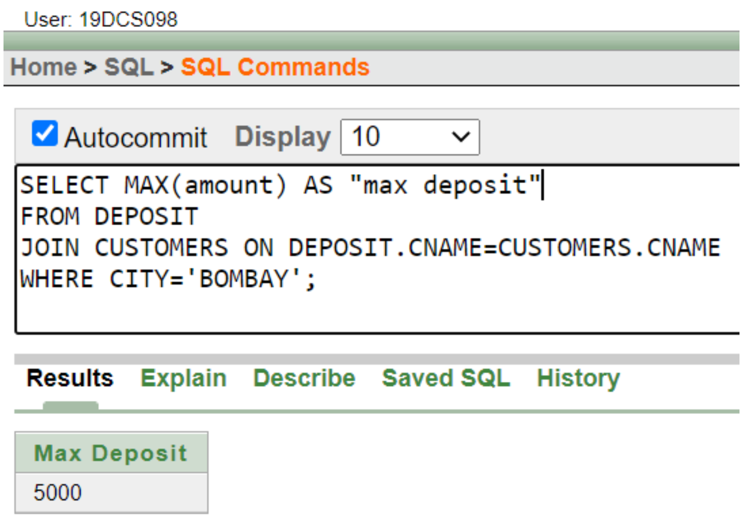
2) **List total deposit of customers living in city Nagpur.**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT SUM(amount) AS "amount"
FROM DEPOSIT
JOIN CUSTOMERS ON DEPOSIT.CNAME=CUSTOMERS.CNAME
WHERE CITY='NAGPUR';
```

```
Amount
6200
```

3) List maximum deposit of customers living in bombay.



User: 19DCS098

Home > SQL > SQL Commands

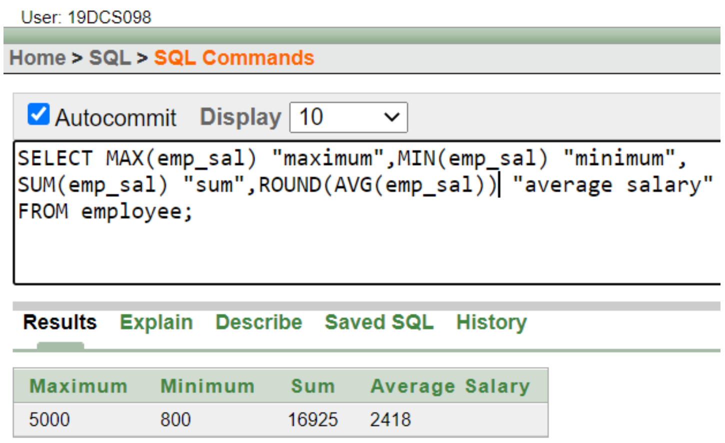
Autocommit Display 10

```
SELECT MAX(amount) AS "max deposit"
FROM DEPOSIT
JOIN CUSTOMERS ON DEPOSIT.CNAME=CUSTOMERS.CNAME
WHERE CITY='BOMBAY';
```

Results Explain Describe Saved SQL History

Max Deposit
5000

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: 19DCS098

Home > SQL > SQL Commands

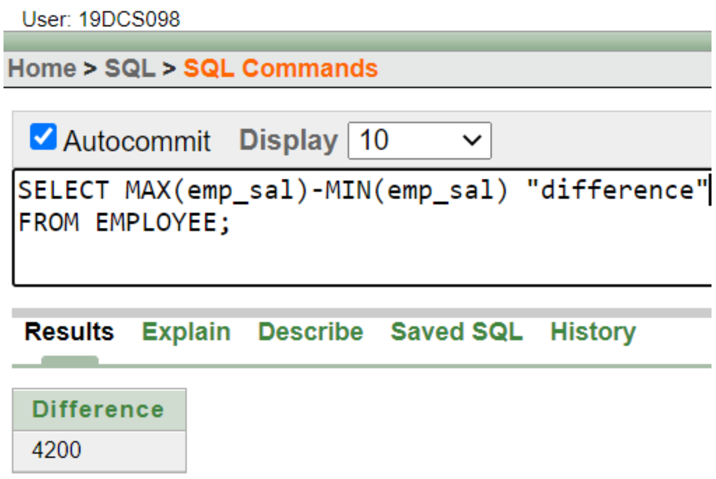
Autocommit Display 10

```
SELECT MAX(emp_sal) "maximum",MIN(emp_sal) "minimum",
SUM(emp_sal) "sum",ROUND(AVG(emp_sal)) "average salary"
FROM employee;
```

Results Explain Describe Saved SQL History

Maximum	Minimum	Sum	Average Salary
5000	800	16925	2418

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



User: 19DCS098

Home > SQL > SQL Commands

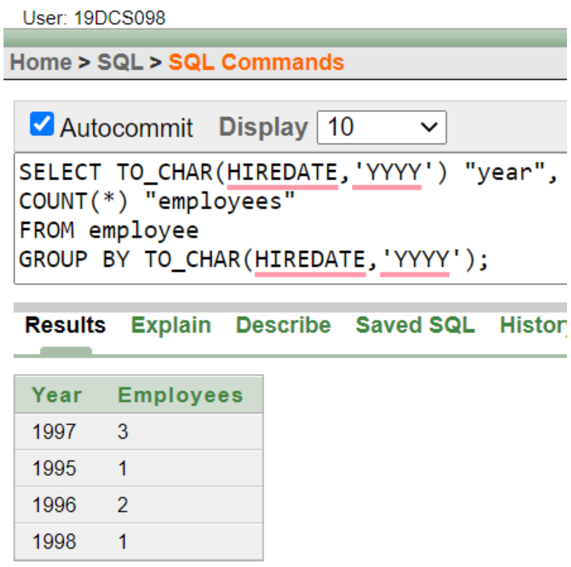
Autocommit Display 10

```
SELECT MAX(emp_sal)-MIN(emp_sal) "difference"
FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

Difference
4200

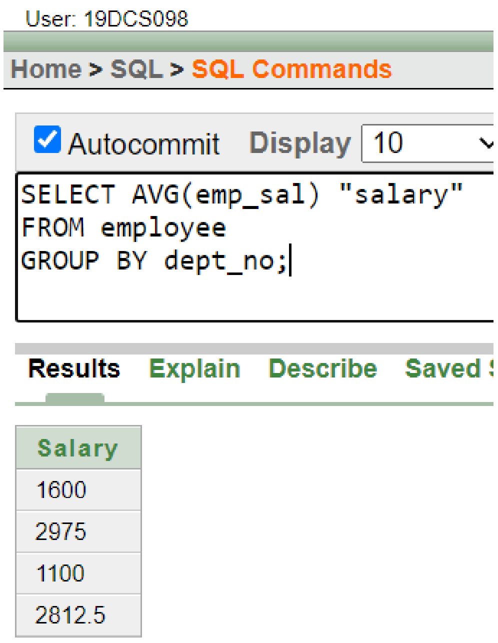
6) **Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998**


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL Histor

```
SELECT TO_CHAR(HIREDATE,'YYYY') "year",
COUNT(*) "employees"
FROM employee
GROUP BY TO_CHAR(HIREDATE,'YYYY');
```

Year	Employees
1997	3
1995	1
1996	2
1998	1

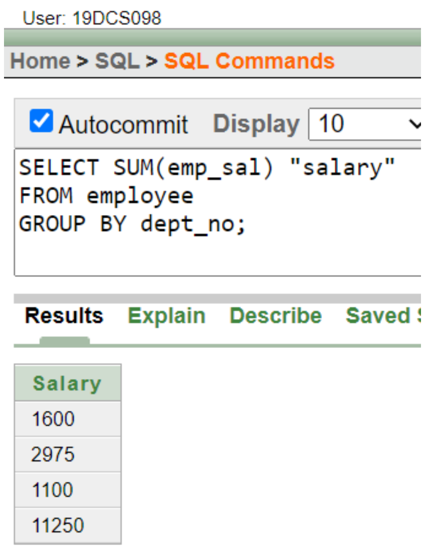
7) **Find the average salaries for each department without displaying the respective department numbers.**



```
SELECT AVG(emp_sal) "salary"
FROM employee
GROUP BY dept_no;
```

Salary
1600
2975
1100
2812.5

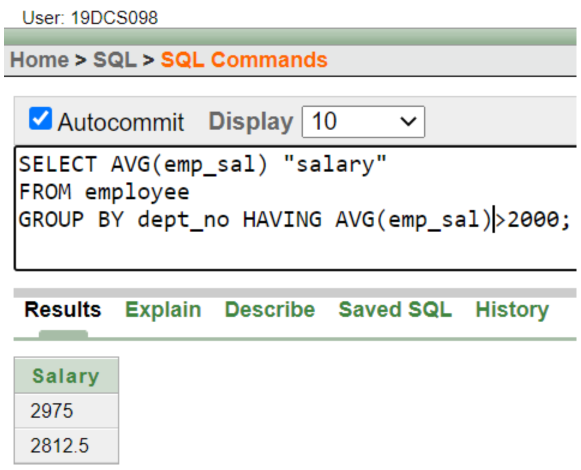
8) Write a query to display the total salary being paid to each job title, within each department.



```
SELECT SUM(emp_sal) "salary"
FROM employee
GROUP BY dept_no;
```

Salary
1600
2975
1100
11250

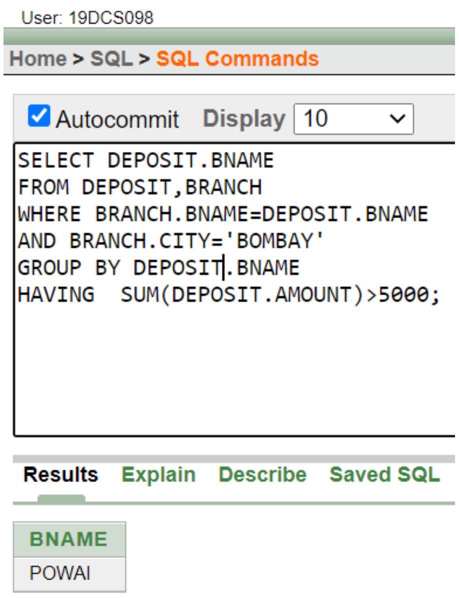
9) Find the average salaries > 2000 for each department without displaying the respective department numbers.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT AVG(emp_sal) "salary"
FROM employee
GROUP BY dept_no HAVING AVG(emp_sal)>2000;
```

Salary
2975
2812.5

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



```
SELECT DEPOSIT.BNAME
FROM DEPOSIT,BRANCH
WHERE BRANCH.BNAME=DEPOSIT.BNAME
AND BRANCH.CITY='BOMBAY'
GROUP BY DEPOSIT.BNAME
HAVING  SUM(DEPOSIT.AMOUNT)>5000;
```

BNAME
POWAI

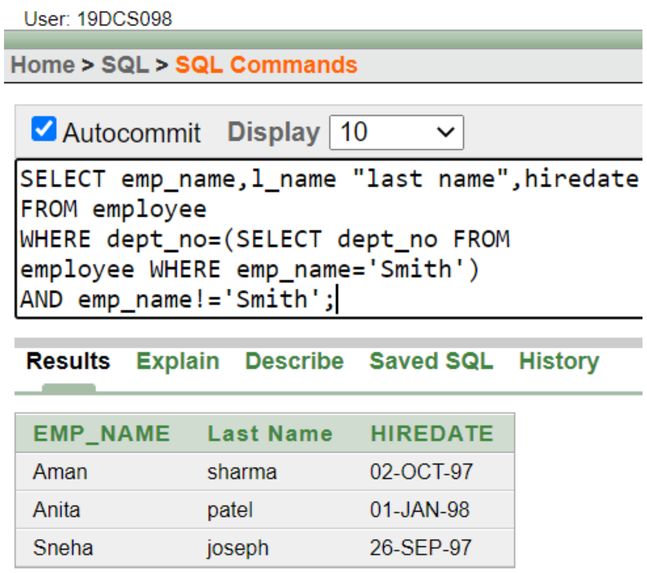
CONCLUSION:

In the above practical, we learned to aggregate data by using GROUP BY.

PRACTICAL-9

To solve queries using the concept of sub query

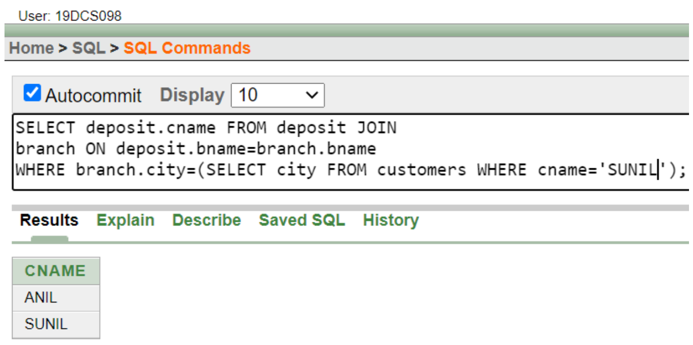
(1) Write a query to display the last name and hire date of any employee in the same department as smith. Exclude smith


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT emp_name,l_name "last name",hiredate
FROM employee
WHERE dept_no=(SELECT dept_no FROM
employee WHERE emp_name='Smith')
AND emp_name!='Smith';
```

EMP_NAME	Last Name	HIREDATE
Aman	sharma	02-OCT-97
Anita	patel	01-JAN-98
Sneha	joseph	26-SEP-97

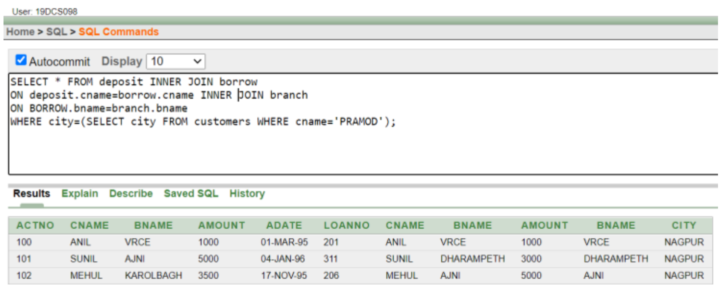
(2) Give name of customers who are depositors having same branch city of mr. sunil.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT deposit.cname FROM deposit JOIN
branch ON deposit.bname=branch.bname
WHERE branch.city=(SELECT city FROM customers WHERE cname='SUNIL');
```

CNAME
ANIL
SUNIL

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



User: 19DCS098

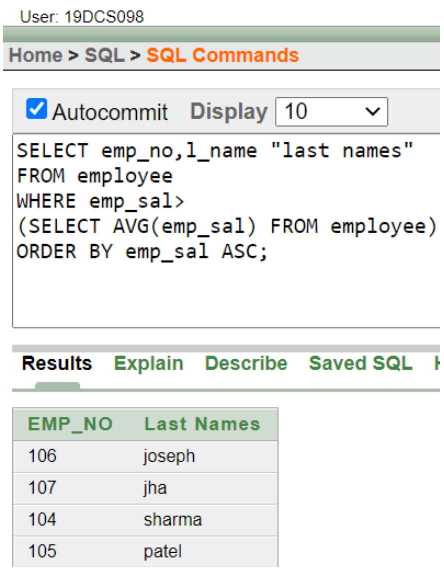
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM deposit INNER JOIN borrow
ON deposit.cname=borrow.cname INNER JOIN branch
ON BORROW.bname=branch.bname
WHERE city=(SELECT city FROM customers WHERE cname='PRAMOD');
```

ACTNO	CNAME	BNAME	AMOUNT	ADATE	LOANNO	CNAME	BNAME	AMOUNT	BNAME	CITY
100	ANIL	VRCE	1000	01-MAR-95	201	ANIL	VRCE	1000	VRCE	NAGPUR
101	SUNIL	AJNI	5000	04-JAN-96	311	SUNIL	DHARAMPETH	3000	DHARAMPETH	NAGPUR
102	MEHUL	KAROLBAGH	3500	17-NOV-95	206	MEHUL	AJNI	5000	AJNI	NAGPUR

(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: 19DCS098

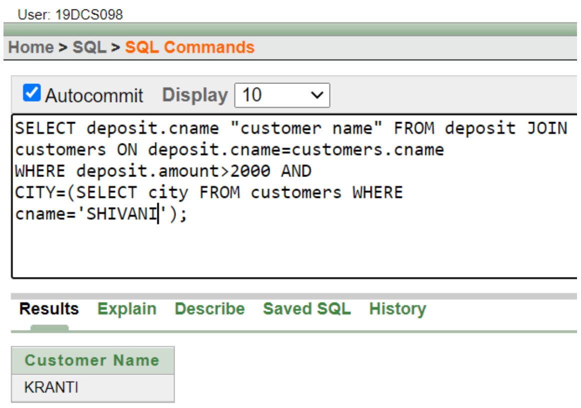
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT emp_no,l_name "last names"
FROM employee
WHERE emp_sal>
(SELECT AVG(emp_sal) FROM employee)
ORDER BY emp_sal ASC;
```

EMP_NO	Last Names
106	joseph
107	jha
104	sharma
105	patel

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



User: 19DCS098

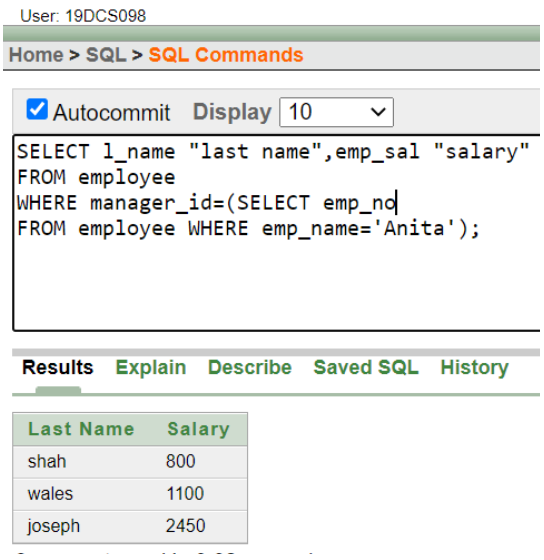
Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT deposit.cname "customer name" FROM deposit JOIN
customers ON deposit.cname=customers.cname
WHERE deposit.amount>2000 AND
CITY=(SELECT city FROM customers WHERE
cname='SHIVANI');
```

Customer Name
KRANTI

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



User: 19DCS098

Home > SQL > SQL Commands

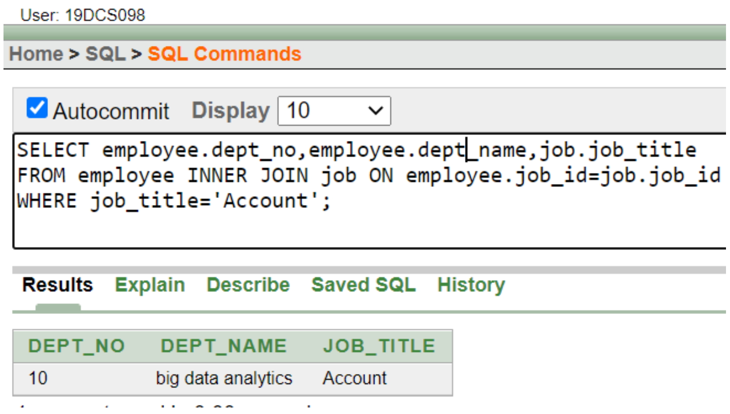
Autocommit Display 10

```
SELECT l_name "last name",emp_sal "salary"
FROM employee
WHERE manager_id=(SELECT emp_no
FROM employee WHERE emp_name='Anita');
```

Results Explain Describe Saved SQL History

Last Name	Salary
shah	800
wales	1100
joseph	2450

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



User: 19DCS098

Home > SQL > SQL Commands

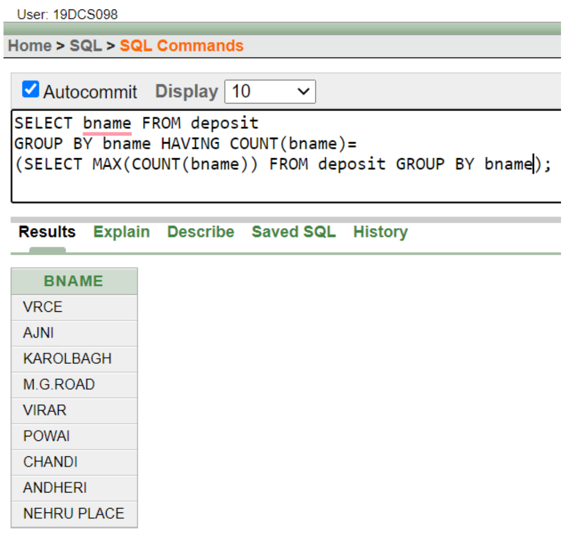
Autocommit Display 10

```
SELECT employee.dept_no,employee.dept_name,job.job_title
FROM employee INNER JOIN job ON employee.job_id=job.job_id
WHERE job_title='Account';
```

Results Explain Describe Saved SQL History

DEPT_NO	DEPT_NAME	JOB_TITLE
10	big data analytics	Account

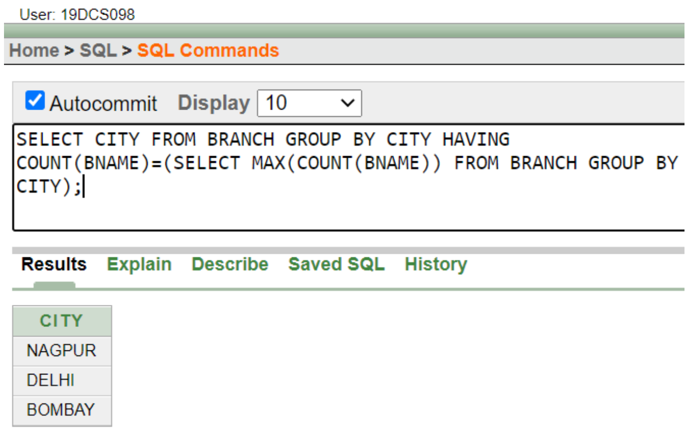
(8) **List the name of branch having highest number of depositors**.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT bname FROM deposit
GROUP BY bname HAVING COUNT(bname)=
(SELECT MAX(COUNT(bname)) FROM deposit GROUP BY bname);
```

BNAME
VRCE
AJNI
KAROLBAGH
M.G.ROAD
VIRAR
POWAI
CHANDI
ANDHERI
NEHRU PLACE

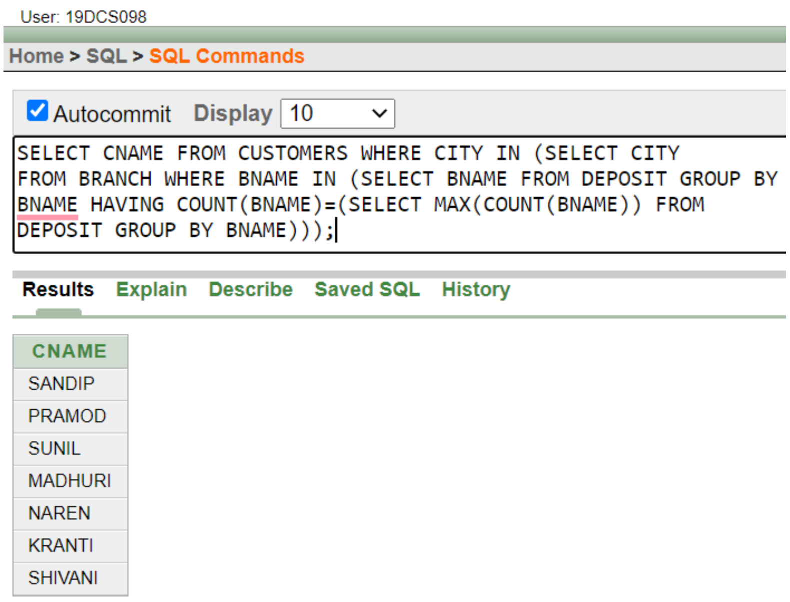
(9) Give the name of cities where in which the maximum numbers of branches are located.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT CITY FROM BRANCH GROUP BY CITY HAVING
COUNT(BNAME)=(SELECT MAX(COUNT(BNAME)) FROM BRANCH GROUP BY
CITY);
```

CITY
NAGPUR
DELHI
BOMBAY

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


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT CNAME FROM CUSTOMERS WHERE CITY IN (SELECT CITY
FROM BRANCH WHERE BNAME IN (SELECT BNAME FROM DEPOSIT GROUP BY
BNAME HAVING COUNT(BNAME)=(SELECT MAX(COUNT(BNAME)) FROM
DEPOSIT GROUP BY BNAME)));
```

CNAME
SANDIP
PRAMOD
SUNIL
MADHURI
NAREN
KRANTI
SHIVANI

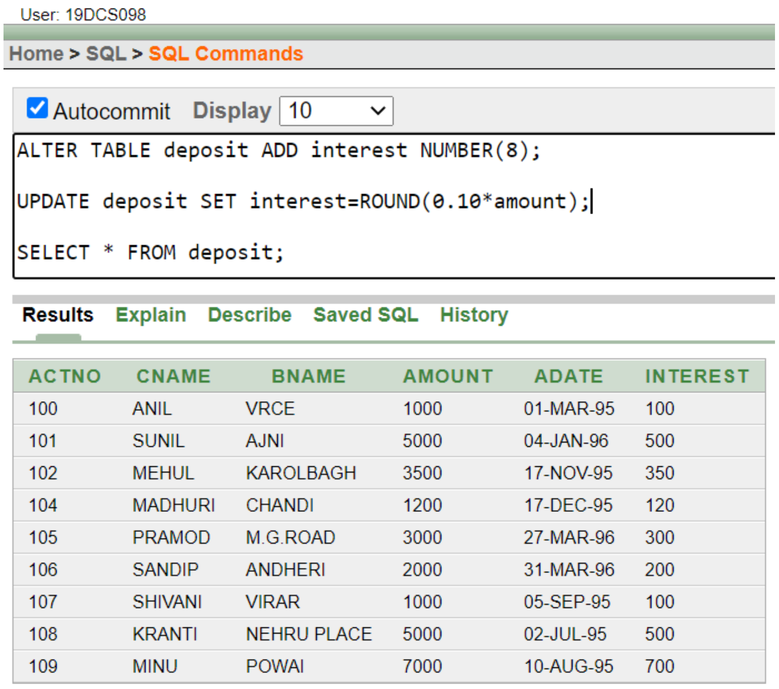
CONCLUSION:

We learned the concept of sub query.

PRACTICAL-10

Manipulating Data

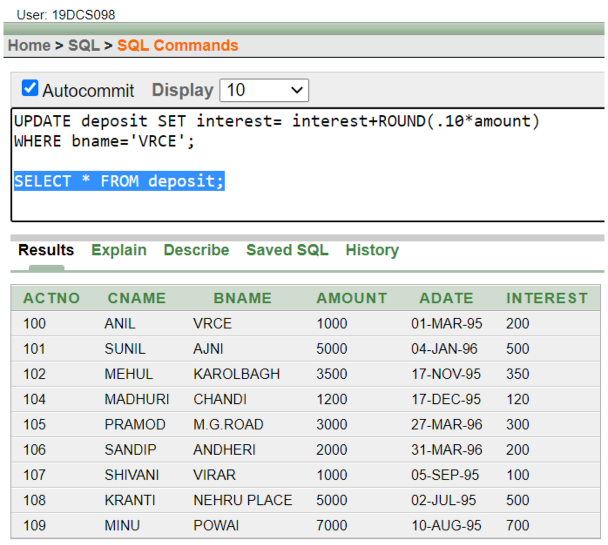
(1) Give 10% interest to all depositors


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
ALTER TABLE deposit ADD interest NUMBER(8);

UPDATE deposit SET interest=ROUND(0.10*amount);

SELECT * FROM deposit;
```

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	1000	01-MAR-95	100
101	SUNIL	AJNI	5000	04-JAN-96	500
102	MEHUL	KAROLBAGH	3500	17-NOV-95	350
104	MADHURI	CHANDI	1200	17-DEC-95	120
105	PRAMOD	M.G.ROAD	3000	27-MAR-96	300
106	SANDIP	ANDHERI	2000	31-MAR-96	200
107	SHIVANI	VIRAR	1000	05-SEP-95	100
108	KRANTI	NEHRU PLACE	5000	02-JUL-95	500
109	MINU	POWAI	7000	10-AUG-95	700

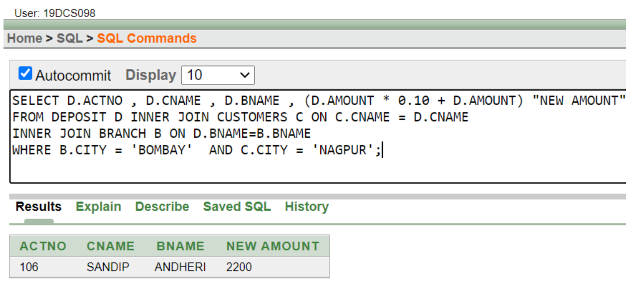
(2) Give 10% interest to all depositors having branch vrce


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
UPDATE deposit SET interest= interest+ROUND(.10*amount)
WHERE bname='VRCE';

SELECT * FROM deposit;
```

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	1000	01-MAR-95	200
101	SUNIL	AJNI	5000	04-JAN-96	500
102	MEHUL	KAROLBAGH	3500	17-NOV-95	350
104	MADHURI	CHANDI	1200	17-DEC-95	120
105	PRAMOD	M.G.ROAD	3000	27-MAR-96	300
106	SANDIP	ANDHERI	2000	31-MAR-96	200
107	SHIVANI	VIRAR	1000	05-SEP-95	100
108	KRANTI	NEHRU PLACE	5000	02-JUL-95	500
109	MINU	POWAI	7000	10-AUG-95	700

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



User: 19DCS098

Home > SQL > SQL Commands

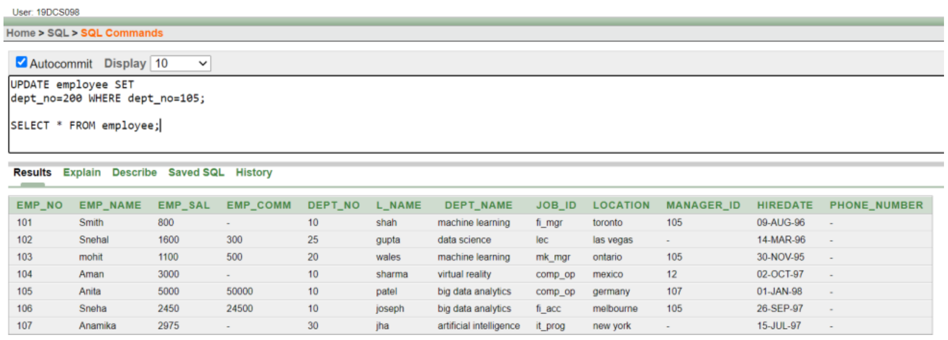
Autocommit Display 10

```
SELECT D.ACTNO , D.CNAME , D.BNAME , (D.AMOUNT * 0.10 + D.AMOUNT) "NEW AMOUNT"
FROM DEPOSIT D INNER JOIN CUSTOMERS C ON C.CNAME = D.CNAME
INNER JOIN BRANCH B ON D.BNAME=B.BNAME
WHERE B.CITY = 'BOMBAY'  AND C.CITY = 'NAGPUR';
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	NEW AMOUNT
106	SANDIP	ANDHERI	2200

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



User: 19DCS098

Home > SQL > SQL Commands

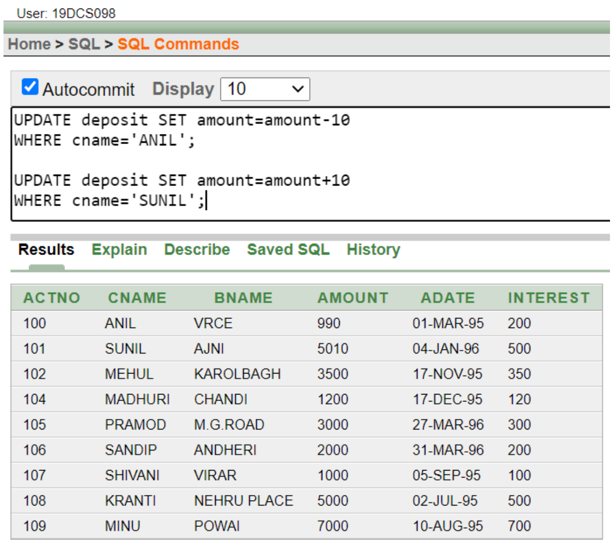
Autocommit Display 10

```
UPDATE employee SET
dept_no=200 WHERE dept_no=105;

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_NUMBER
101	Smith	800	-	10	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	mohit	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
104	Aman	3000	-	10	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	2450	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	-

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



User: 19DCS098

Home > SQL > SQL Commands

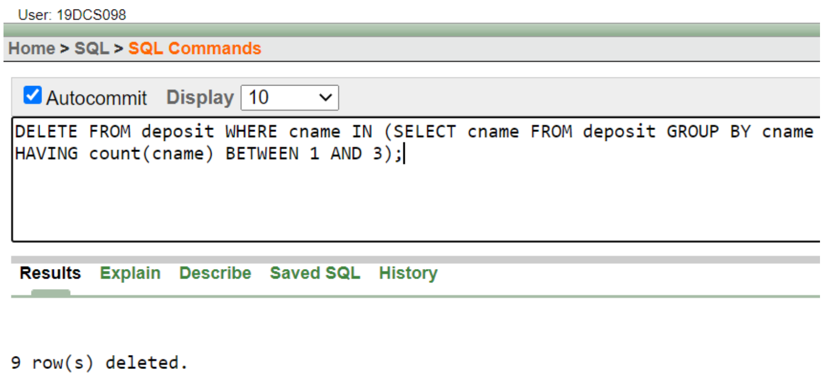
Autocommit Display 10

```
UPDATE deposit SET amount=amount-10
WHERE cname='ANIL';

UPDATE deposit SET amount=amount+10
WHERE cname='SUNIL';
```

Results Explain Describe Saved SQL History

ACTNO	CNAME	BNAME	AMOUNT	ADATE	INTEREST
100	ANIL	VRCE	990	01-MAR-95	200
101	SUNIL	AJNI	5010	04-JAN-96	500
102	MEHUL	KAROLBAGH	3500	17-NOV-95	350
104	MADHURI	CHANDI	1200	17-DEC-95	120
105	PRAMOD	M.G.ROAD	3000	27-MAR-96	300
106	SANDIP	ANDHERI	2000	31-MAR-96	200
107	SHIVANI	VIRAR	1000	05-SEP-95	100
108	KRANTI	NEHRU PLACE	5000	02-JUL-95	500
109	MINU	POWAI	7000	10-AUG-95	700

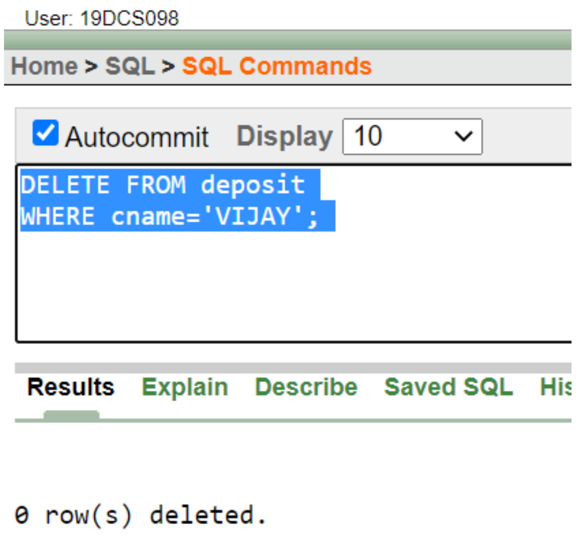
(6) Delete depositors of branches having number of customers between 1 to 3.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DELETE FROM deposit WHERE cname IN (SELECT cname FROM deposit GROUP BY cname
HAVING count(cname) BETWEEN 1 AND 3);
```

```
9 row(s) deleted.
```

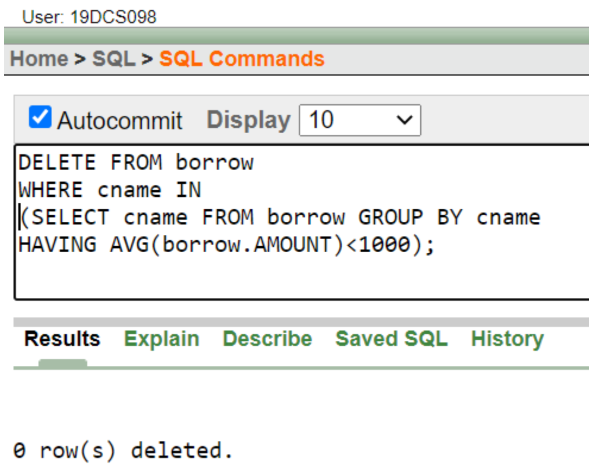
(7) Delete deposit of vijay.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL His

```
DELETE FROM deposit
WHERE cname='VIJAY';
```

```
0 row(s) deleted.
```

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


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DELETE FROM borrow
WHERE cname IN
(SELECT cname FROM borrow GROUP BY cname
HAVING AVG(borrow.AMOUNT)<1000);
```

```
0 row(s) deleted.
```

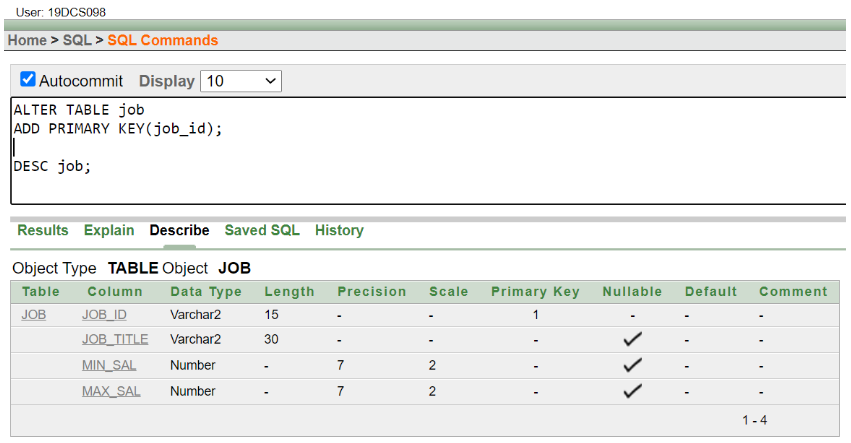
CONCLUSION:

In the above practical, we learned the concept of data manipulation.

PRACTICAL-11

Add and Remove constraint

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



User: 19DCS098

Home > SQL > SQL Commands

✓ Autocommit Display 10

```
ALTER TABLE job
ADD PRIMARY KEY(job_id);

DESC job;
```

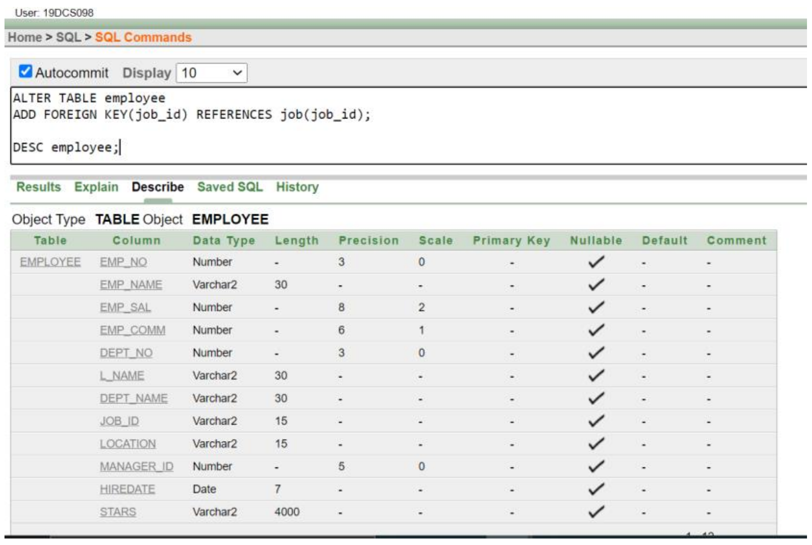
Results Explain Describe Saved SQL History

Object Type **TABLE** Object **JOB**

Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
JOB	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: 19DCS098

Home > SQL > SQL Commands

✓ Autocommit Display 10

```
ALTER TABLE employee
ADD FOREIGN KEY(job_id) REFERENCES job(job_id);

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	-	-	-	✓	-	-
	STARS	Varchar2	4000	-	-	-	✓	-	-

(4)Remove primary key constraint on job_id

ALTER TABLE job DROP CONSTRAINT job_id ;

(5)Remove foreign key constraint on employee table

ALTER TABLE EMPLOYEE DROP CONSTRAINT job_id ;

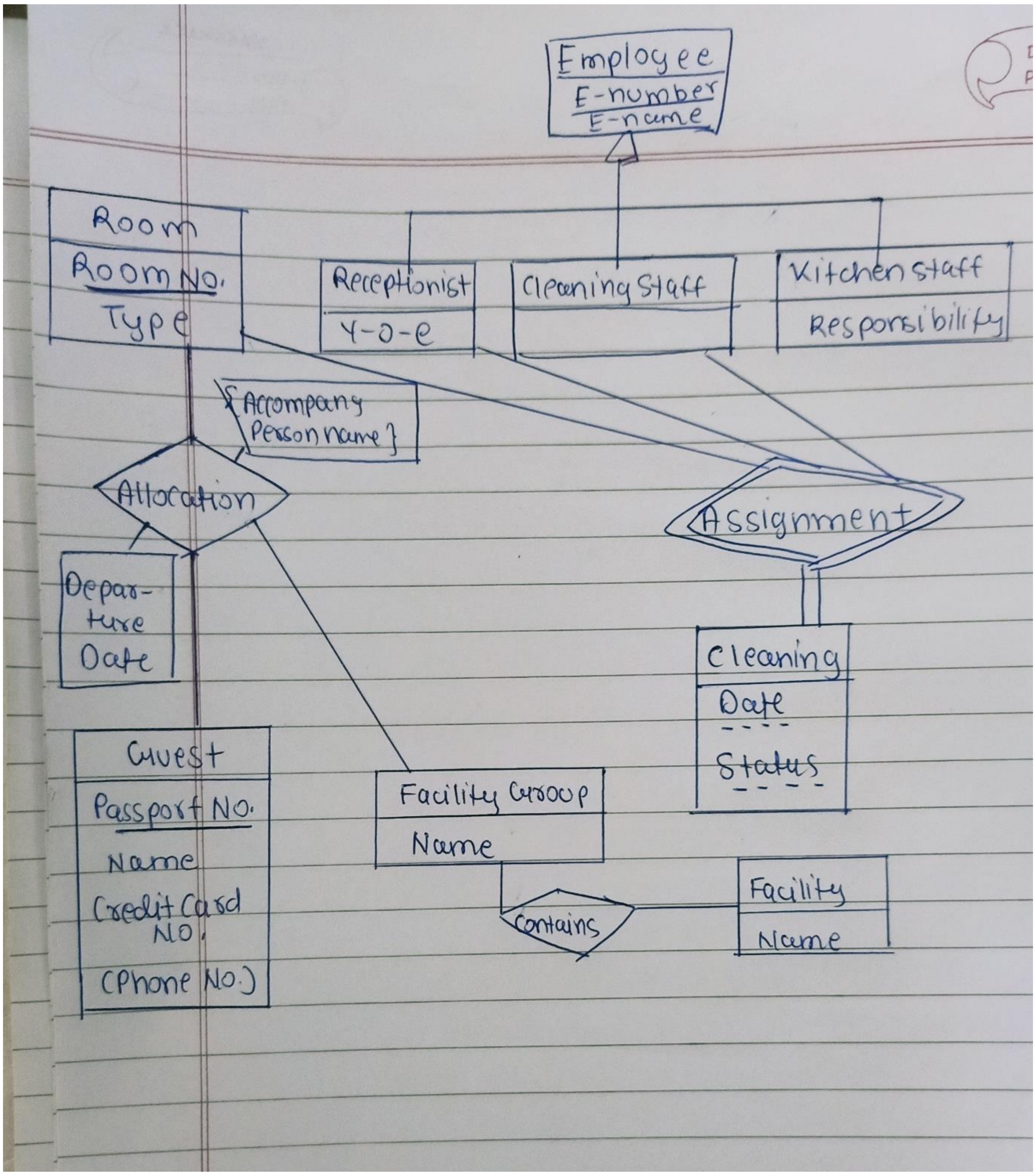
CONCLUSION:

In the above practical, we learned how to add and remove constraints.

PRACTICAL-12

Data Dictionary and E-R Diagram

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

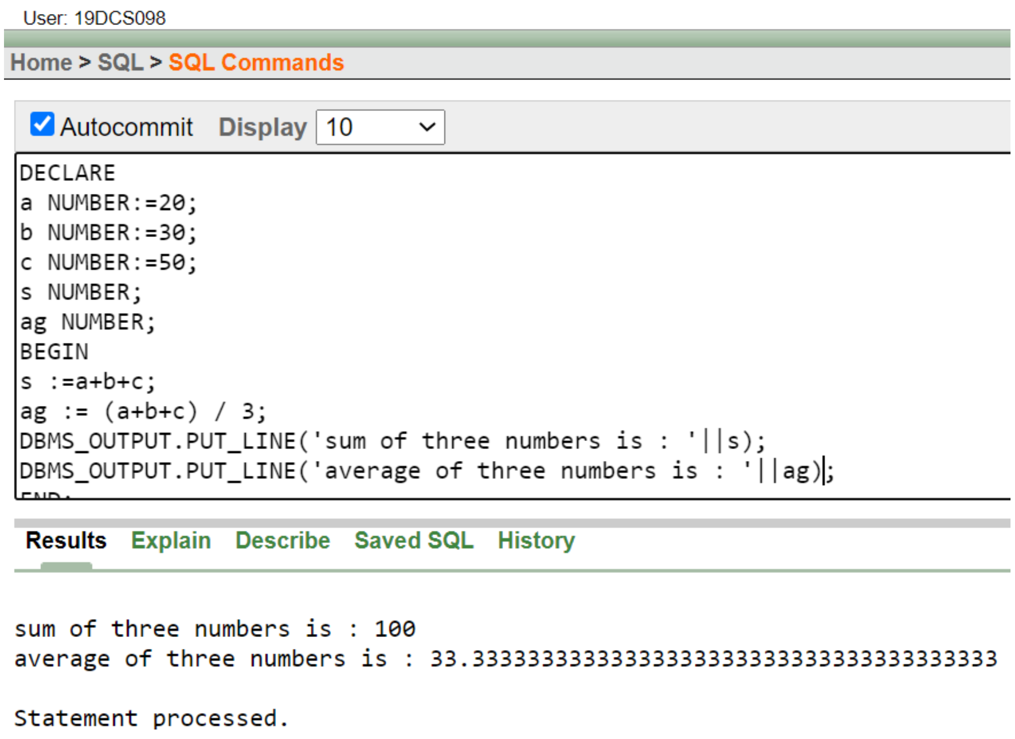


CONCLUSION:

In the above practical, we learned the concept of E-R diagram.

PRACTICAL-13

Write a PL-SQL block to find Sum and average of three numbers.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
a NUMBER:=20;
b NUMBER:=30;
c NUMBER:=50;
s NUMBER;
ag NUMBER;
BEGIN
s :=a+b+c;
ag := (a+b+c) / 3;
DBMS_OUTPUT.PUT_LINE('sum of three numbers is : '||s);
DBMS_OUTPUT.PUT_LINE('average of three numbers is : '||ag);
```

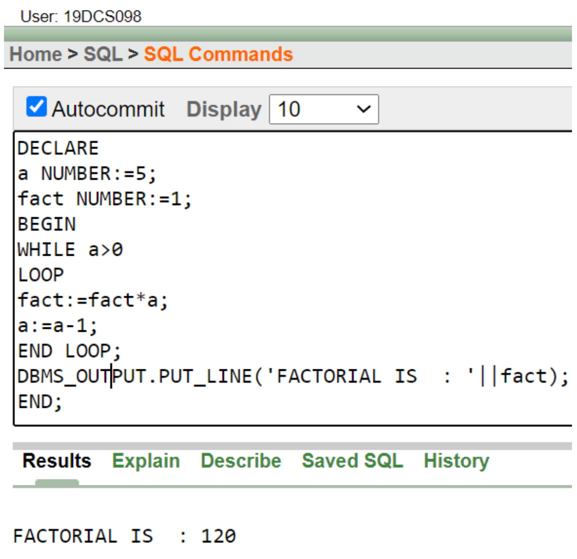
```
sum of three numbers is : 100
average of three numbers is : 33.33333333333333333333333333333333333333

Statement processed.
```

CONCLUSION:

In the above practical, we learned the concept of PL/SQL

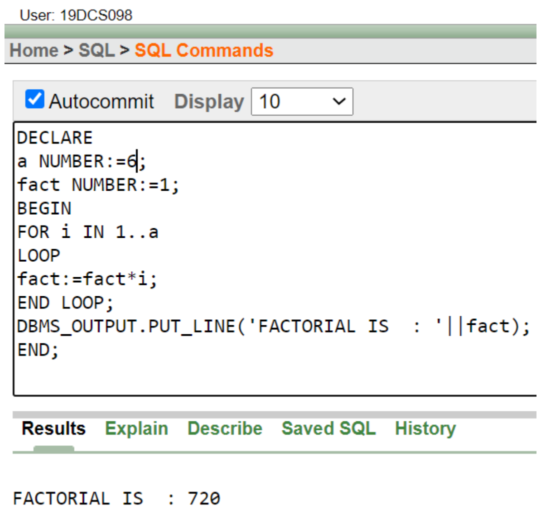
PRACTICAL-14

Find the factorial of a number in pl/sql using for, While and Simple Loop


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
a NUMBER:=5;
fact NUMBER:=1;
BEGIN
WHILE a>0
LOOP
fact:=fact*a;
a:=a-1;
END LOOP;
DBMS_OUTPUT.PUT_LINE('FACTORIAL IS  : '||fact);
END;
```

```
FACTORIAL IS  : 120
```


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
a NUMBER:=6;
fact NUMBER:=1;
BEGIN
FOR i IN 1..a
LOOP
fact:=fact*i;
END LOOP;
DBMS_OUTPUT.PUT_LINE('FACTORIAL IS  : '||fact);
END;
```

```
FACTORIAL IS  : 720
```

CONCLUSION:

In the above practical, we learned the concept of loops using PL/SQL

PRACTICAL-15

To understand the concept of “select into” and “% type” attribute.

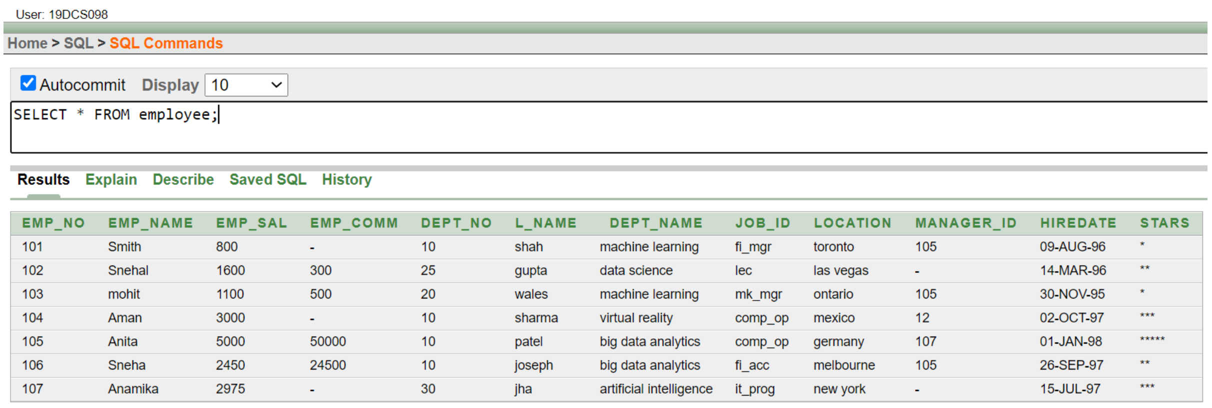
Create an EMPLOYEES table that is a replica of the EMP table. Add a new column, STARS, of VARCHAR2 data type and length of 50 to the EMPLOYEES table for storing asterisk (*). Create a PL/SQL block that rewards an employee by appending an asterisk in the STARS column for every Rs1000/- of the employee’s salary. For example, if the employee has a salary amount of Rs8000/-, the string of asterisks should contain eight asterisks. If the employee has a salary amount of Rs12500/-, the string of asterisks should contain 13 asterisks. Update the STARS column for the employee with the string of asterisks.

PROGRAM CODE:

```
DECLARE
EMPS_NO EMPLOYEE.EMP_NO%TYPE;
EMPS_NAME EMPLOYEE.EMP_NAME%TYPE;
EMPS_SAL EMPLOYEE.EMP_SAL%TYPE;
EMPS_COMM EMPLOYEE.EMP_COMM%TYPE;
DEPT_NO1 EMPLOYEE.DEPT_NO%TYPE;
EMPS_STAR VARCHAR2(50);
i number;
BEGIN
for i in 101..107
loop
select NVL( ROUND (EMP_SAL/1000),0) INTO EMPS_SAL FROM EMPLOYEE WHERE EMP_NO=i;
EMPS_STAR:=NULL;
FOR J IN 1..EMPS_SAL
```

```
LOOP
EMPS_STAR:=EMPS_STAR || '*' ;
END LOOP;
UPDATE employee SET stars=emps_star WHERE emp_no=i;
END LOOP;
END;
```

OUTPUT:


User: 19DCS098

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	STARS
101	Smith	800	-	10	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	mohit	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	*
104	Aman	3000	-	10	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	2450	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	***

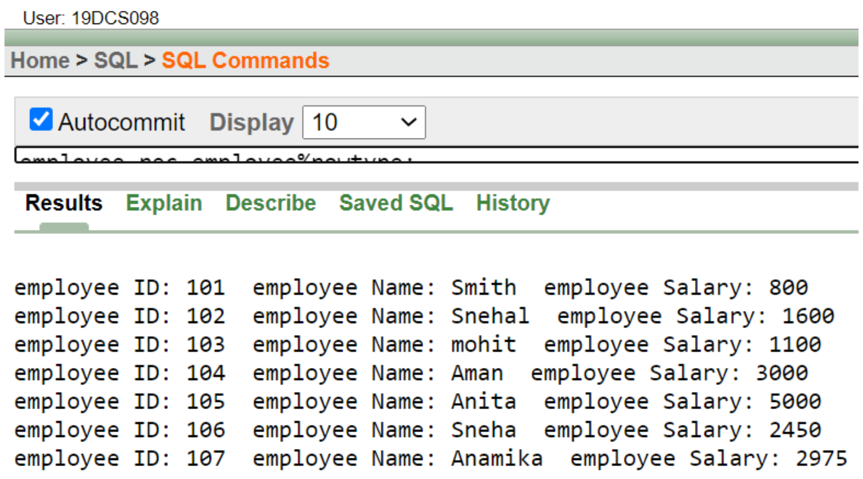
CONCLUSION:

In the above practical, we learned the concept of “select into” and “% type” attribute.

PRACTICAL-16

To perform the concept of cursor (a) Display all the information of EMP table using %ROWTYPE.

```
DECLARE

employee_rec employee%rowtype;


CURSOR c_employee is

SELECT * FROM employee;

BEGIN

OPEN c_employee;

LOOP

FETCH c_employee into employee_rec;

EXIT WHEN c_employee%notfound;

dbms_output.put_line('employee ID: ' || employee_rec.emp_no || ' employee Name: ' || employee_rec.emp_name || ' employee Salary: ' || employee_rec.emp_sal);

END LOOP;

CLOSE c_employee;

END;
```



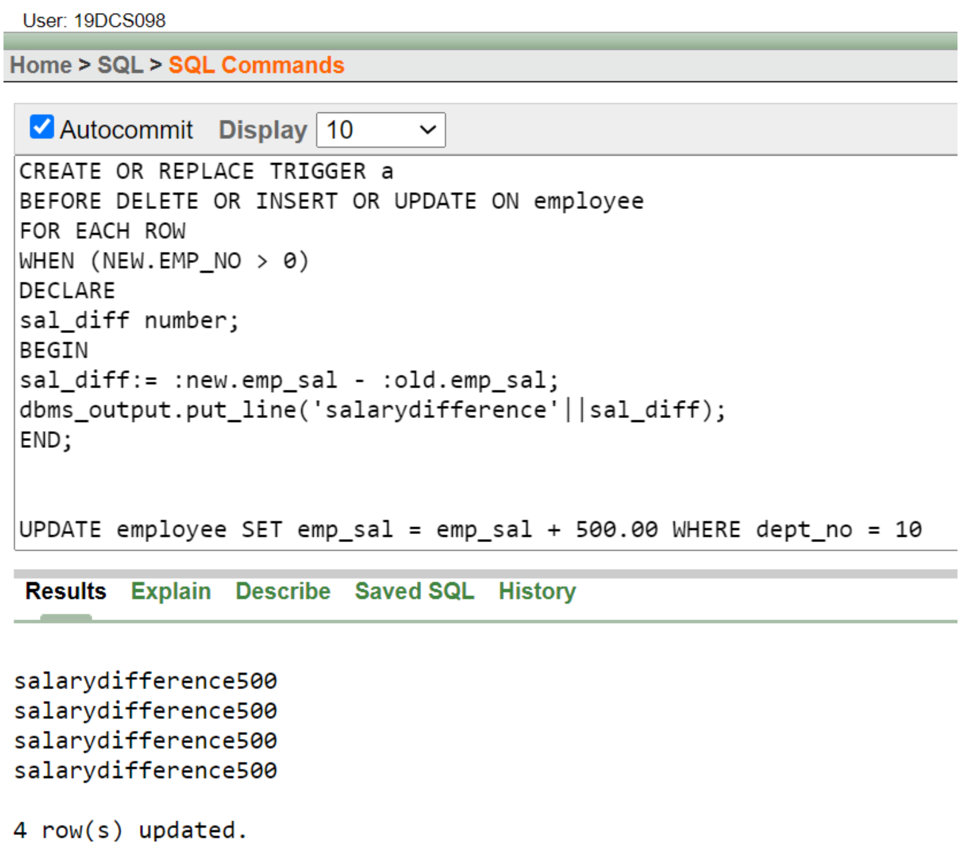
```
employee ID: 101  employee Name: Smith  employee Salary: 800
employee ID: 102  employee Name: Snehal  employee Salary: 1600
employee ID: 103  employee Name: mohit  employee Salary: 1100
employee ID: 104  employee Name: Aman  employee Salary: 3000
employee ID: 105  employee Name: Anita  employee Salary: 5000
employee ID: 106  employee Name: Sneha  employee Salary: 2450
employee ID: 107  employee Name: Anamika  employee Salary: 2975
```

CONCLUSION:

In the above practical, we learnt the concept of cursor

PRACTICAL-17

Write a PL/SQL block to update the salary where deptno is 10. Generate trigger that will store the original record in other table before updation take place


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE OR REPLACE TRIGGER a
BEFORE DELETE OR INSERT OR UPDATE ON employee
FOR EACH ROW
WHEN (NEW.EMP_NO > 0)
DECLARE
sal_diff number;
BEGIN
sal_diff:= :new.emp_sal - :old.emp_sal;
dbms_output.put_line('salarydifference'||sal_diff);
END;


UPDATE employee SET emp_sal = emp_sal + 500.00 WHERE dept_no = 10
```

```
salarydifference500
salarydifference500
salarydifference500
salarydifference500

4 row(s) updated.
```

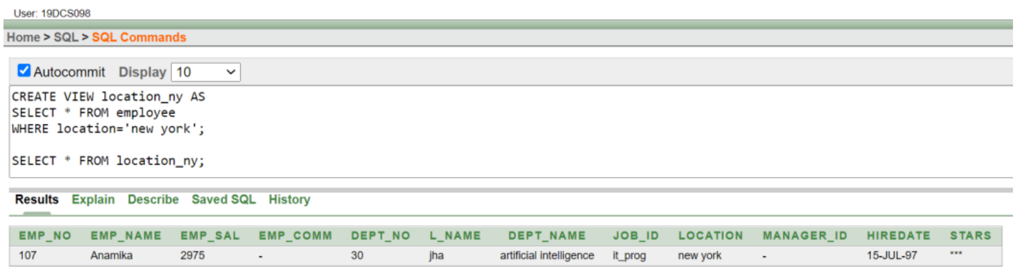
CONCLUSION:

In the above practical, we learned the concept of cursor.

PRACTICAL-18

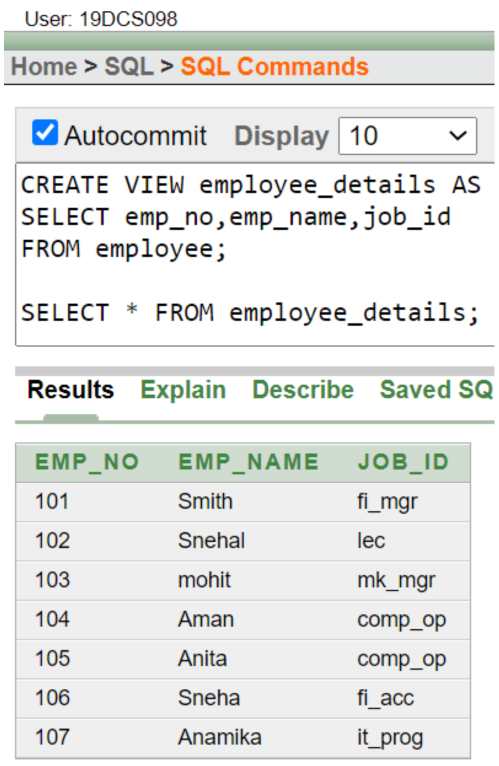
To solve queries using the concept of View.

(1) Write a query to create a view for those employee belongs to the location New York.


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```sql
CREATE VIEW location_ny AS
SELECT * FROM employee
WHERE location='new york';

SELECT * FROM location_ny;
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	STARS
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	***



```sql
CREATE VIEW employee_details AS
SELECT emp_no,emp_name,job_id
FROM employee;

SELECT * FROM employee_details;
```

EMP_NO	EMP_NAME	JOB_ID
101	Smith	fi_mgr
102	Snehal	lec
103	mohit	mk_mgr
104	Aman	comp_op
105	Anita	comp_op
106	Sneha	fi_acc
107	Anamika	it_prog

CONCLUSION:

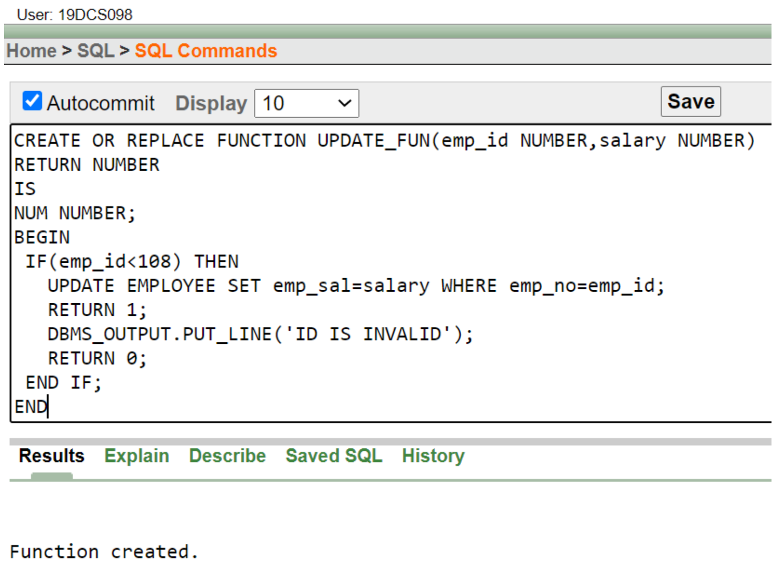
In the above practical, we learnt the concept of views.

PRACTICAL-19

To perform the concept of function and procedure

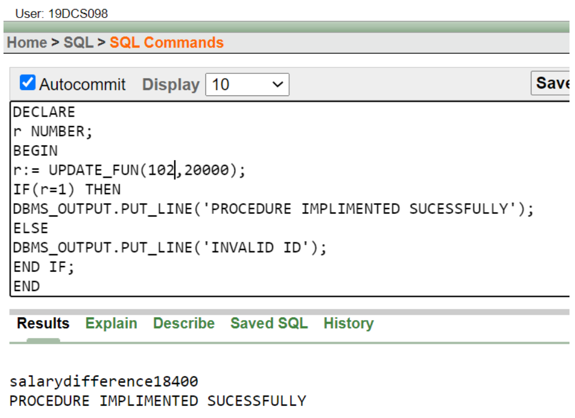
Write a PL/SQL block to update the salary of employee specified by empid. If record exist, then update the salary otherwise display appropriate message. Write a function as well as procedure for updating salary.

FUNCTION:


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

```
CREATE OR REPLACE FUNCTION UPDATE_FUN(emp_id NUMBER,salary NUMBER)
RETURN NUMBER
IS
NUM NUMBER;
BEGIN
 IF(emp_id<108) THEN
   UPDATE EMPLOYEE SET emp_sal=salary WHERE emp_no=emp_id;
   RETURN 1;
   DBMS_OUTPUT.PUT_LINE('ID IS INVALID');
   RETURN 0;
 END IF;
END
```

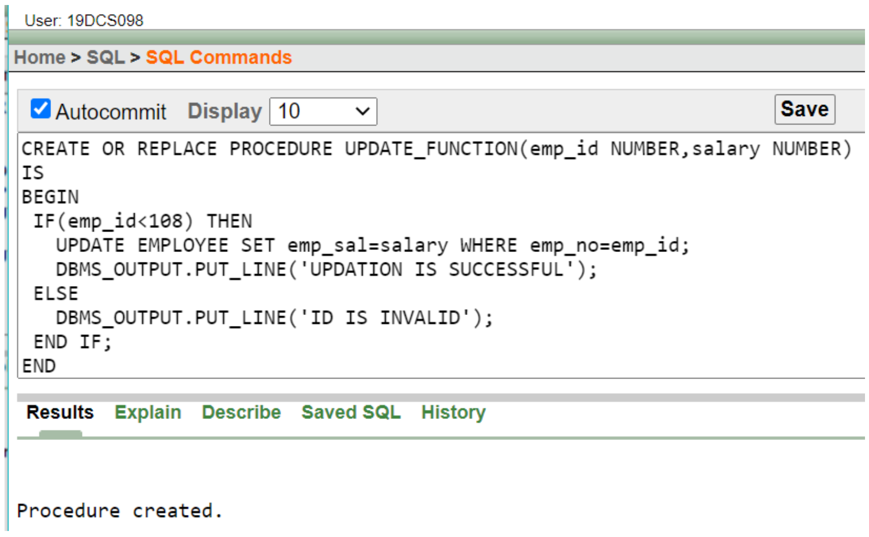
```
Function created.
```


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

```
DECLARE
r NUMBER;
BEGIN
r:= UPDATE_FUN(102,20000);
IF(r=1) THEN
DBMS_OUTPUT.PUT_LINE('PROCEDURE IMPLIMENTED SUCESSFULLY');
ELSE
DBMS_OUTPUT.PUT_LINE('INVALID ID');
END IF;
END
```

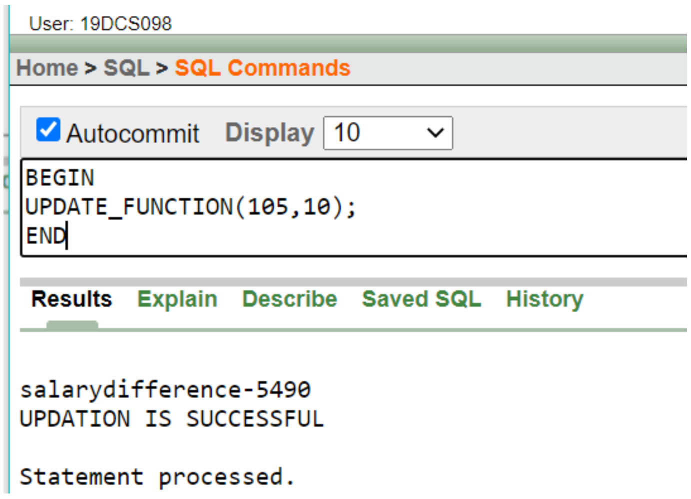
```
salarydifference18400
PROCEDURE IMPLIMENTED SUCESSFULLY
```

PROCEDURE:


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

```
CREATE OR REPLACE PROCEDURE UPDATE_FUNCTION(emp_id NUMBER,salary NUMBER)
IS
BEGIN
 IF(emp_id<108) THEN
   UPDATE EMPLOYEE SET emp_sal=salary WHERE emp_no=emp_id;
   DBMS_OUTPUT.PUT_LINE('UPDATION IS SUCCESSFUL');
 ELSE
   DBMS_OUTPUT.PUT_LINE('ID IS INVALID');
 END IF;
END
```

```
Procedure created.
```


User: 19DCS098
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
BEGIN
UPDATE_FUNCTION(105,10);
END
```

```
salarydifference-5490
UPDATION IS SUCCESSFUL

Statement processed.
```

CONCLUSION:

In the above practical, we learned the concept of functions and procedure.

PRACTICAL-20

To perform the concept of exception handler

Write a PL/SQL block that will accept the employee code, amount and operation. Based on specified operation amount is added or deducted from salary of said employee. Use user defined exception handler for handling the exception.

```
SELECT * FROM employee;
DECLARE
CURSOR c IS SELECT * FROM employee;
greater exception Exception;
emp_code employee.emp_no%type :=101;
V c%rowtype;
amount NUMBER(5):=900;
operation NUMBER(2):=1;
newsl NUMBER(5);
BEGIN
OPEN c;
LOOP
FETCH c INTO v;
EXIT WHEN c%notfound;
IF(v.emp_no=emp_code) THEN
CASE OPERATION
  WHEN 0 THEN
  IF amount>v.emp_sal THEN
  RAISE greater exception;
ELSE
  new_sal=v.emp_sal-amount;
  DBMS_OUTPUT.PUT_LINE('AMOUNT : '||amount);
  DBMS_OUTPUT.PUT_LINE('NEW SALARY : '||new_sal);
END IF;
  WHEN 1 THEN
   new_sal=v.emp_sal+amount;
   DBMS_OUTPUT.PUT_LINE('AMOUNT : '||amount);
   DBMS_OUTPUT.PUT_LINE('NEW SALARY : '||new_sal);
ELSE
   DBMS_OUTPUT.PUT_LINE('INVALID EXPRESSION');
END CASE;
END IF;
END LOOP;
CLOSE c;

exception
WHEN greater exception THEN
DBMS_OUTPUT.PUT_LINE('AMOUNT : '||amount||' BALANCE : '||v.emp_sal);
DBMS_OUTPUT.PUT_LINE('WITHDRAW!!');
WHEN OTHERS THEN DBMS_OUTPUT.PUT_LINE('ERROR');
END;
```

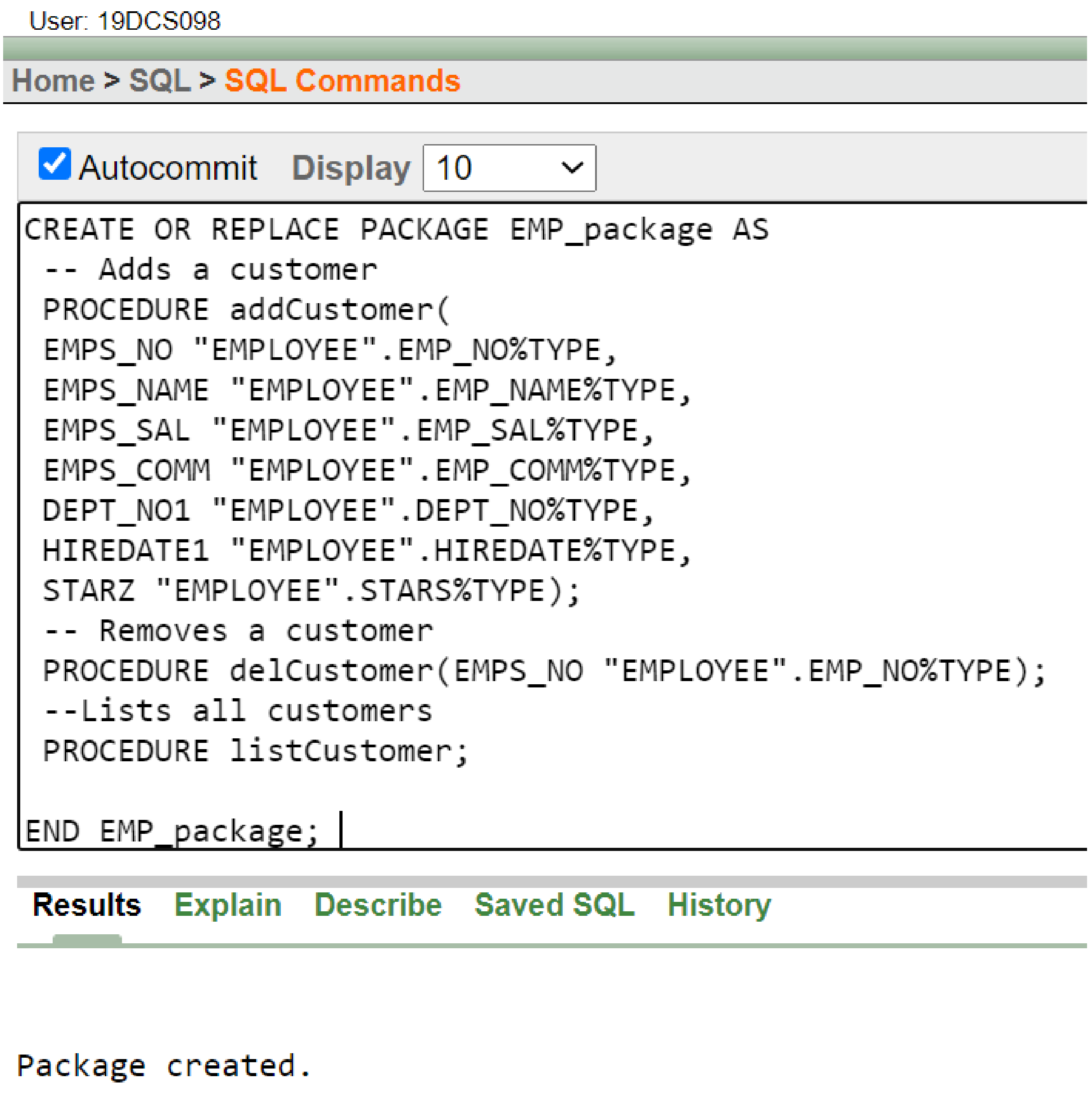
CONCLUSION:

In the above practical, we learned the concept of exceptional handling.

PRACTICAL-21

To perform the concept of package

Create and invoke a package that contains private and public constructs.



```
CREATE OR REPLACE PACKAGE EMP_package AS
 -- Adds a customer
 PROCEDURE addCustomer(
 EMPS_NO "EMPLOYEE".EMP_NO%TYPE,
 EMPS_NAME "EMPLOYEE".EMP_NAME%TYPE,
 EMPS_SAL "EMPLOYEE".EMP_SAL%TYPE,
 EMPS_COMM "EMPLOYEE".EMP_COMM%TYPE,
 DEPT_NO1 "EMPLOYEE".DEPT_NO%TYPE,
 HIREDATE1 "EMPLOYEE".HIREDATE%TYPE,
 STARZ "EMPLOYEE".STARS%TYPE);
 -- Removes a customer
 PROCEDURE delCustomer(EMPS_NO "EMPLOYEE".EMP_NO%TYPE);
 --Lists all customers
 PROCEDURE listCustomer;

END EMP_package;
```

```
Package created.
```

CONCLUSION:

In the above practical, we learnt the concept of packages.