Aim 1: Evaluation of Database (File System, DBMS, RDBMS, DDBMS).

Answer:

1) File System

Definition:

A file system is a technique of arranging the files in a storage medium like a hard disk, pen drive, DVD, etc.

Application:

It helps in arranging data having different formats such txt, doc, mp3, etc. which are grouped into directories.

Advantages:

- 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.
- Files data are dependent on each other.
- Fast File System Recovery.

Disadvantages:

- The file system doesn’t have a crash recovery mechanism on the other hand.
- Data inconsistency is higher in the file system.
- The file system provides the details of data representation and storage of data.
- Storing and retrieving of data can’t be done efficiently in a file system.

Application software:

Storage devices, winfs, discs, etc.

2) DBMS (Database Management System)

Definition:

Database management system is a software which is used to manage the database. It serves as an interface between an end-user and a database, allowing users to create, read, update, and delete data in the database.

Application:

The database management system optimizes the organization of data by following a database schema design technique called normalization, which splits a large table into smaller tables when any of its attributes have redundancy in values.

Advantages:

- Redundancy problem can be solved.
- Has a very high security level.
- Presence of Data integrity.
- Avoidance of inconsistency and support multiple users.
- Shared data between authorized users.
- Provide backup of data.

Disadvantages:

- Designers must understand complete functionality of DBMS to utilize its utmost potential, hence it is a complex software.
- Functionality of DBMS use lot of memory.
- The cost of DBMS varies significantly depending on the environment and functionality provided.

Application software:

MySQL, PostgreSQL, Microsoft Access, SQL Server, FileMaker, Oracle, RDBMS, dBASE, Clipper, and FoxPro.

3) RDBMS (Relational Database Management System)

Definition:

A system used to maintain the logical relationship among the different tables and create interaction between them.

Application:

The relational structure makes it possible to run queries across multiple tables at once.

Advantages:

- It ensures that all data stored are in the form of rows and columns
- All data stored in the tables are provided by an RDBMS, which makes it easily understood by the programmer.
- Facilitates primary key, which helps in unique identification of the rows
- Index creation for retrieving data at a higher speed
- Facilitates a common column to be shared amid two or more tables
- Multi-user accessibility is facilitated to be controlled by individual users
- A virtual table creation is enabled to store sensitive data and simplify queries

Disadvantages:

- RDBMS imposes limits on field lengths.
- Extremely difficult to manage high volume of data.
- The expense of maintaining and even setting up a database system is relatively high and one of the drawbacks of relational databases.
- A special software is required for setting up a relational database and this could cost a fortune.

Application software:

MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access.

4) DDBMS (Distributed Database Management System)

Definition:

The distributed database management system contains the data in multiple locations. That can be in different systems in the same place or across different geographical locations.

Application:

DDBMS is widely used in data warehousing, where huge volumes of data are processed and accessed by numerous users or database clients at the same time. This database system is used to manage data in networks, maintain confidentiality and handle data integrity.

Advantages:

- The database can be stored according to the departmental information in an organisation. In that case, it is easier for an organisational hierarchical access.
- In case of natural catastrophe such as fire or an earthquake all the data would not be destroyed it is stored at different locations.
- It is cheaper to create a network of systems containing a part of the database. This database can also be easily increased or decreased.
- Even if some of the data nodes go offline, the rest of the database can continue it's normal functions.
- The database is easier to expand as it is already spread across multiple systems and it is not too complicated to add a system.
- The distributed database can have the data arranged according to different levels of transparency i.e. data with different transparency levels can be stored at different locations.

Disadvantages:

- The DDBMS is more expensive as it is complex and hence, difficult to maintain.
- It is difficult to provide security in a distributed database as the database needs to be secured at all the locations it is stored. Moreover, the infrastructure connecting all the nodes in a distributed database also needs to be secured.
- It is difficult to maintain data integrity in the distributed database because of its nature. There can also be data redundancy in the database as it is stored at multiple locations.
- The distributed database is complicated and it is difficult to find people with the necessary experience who can manage and maintain it.
- The distributed database is quite complex and it is difficult to make sure that a user gets a uniform view of the database because it is spread across multiple locations.

Application software:

MySQL, Oracle, SQL Server, dBASE, FoxPro, PostgreSQL.

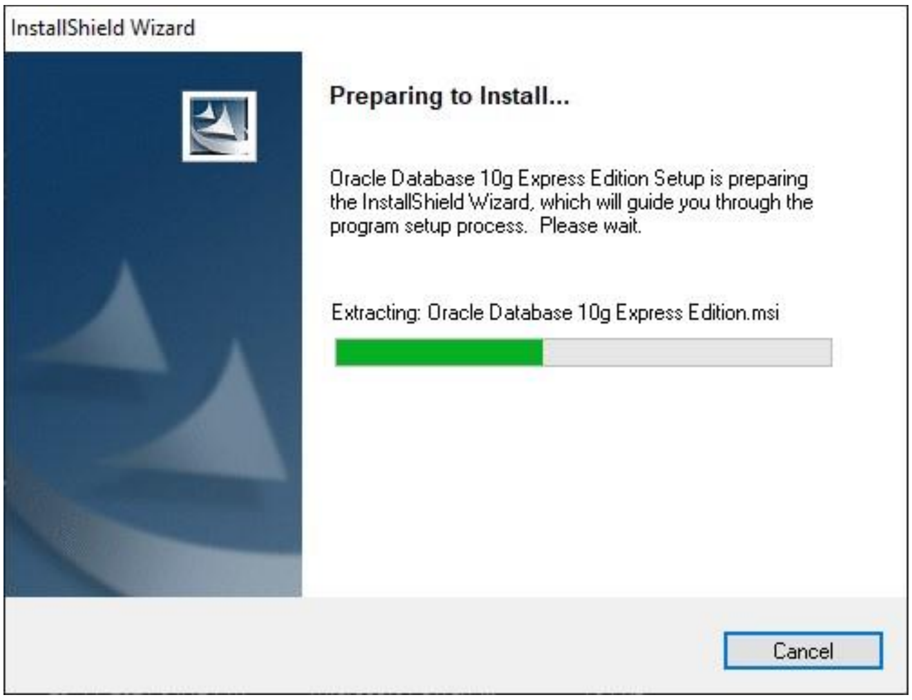
Aim 2: Introduction to Oracle (step by step installation, introduction, introduction to sql, plsql).

Answer:

Step 1: Downloading Oracle 10 g from below link:

https://www.oracle.com/database/technologies/database10gr2-doc.html

Step 2: Installing it by double clicking .exe which I have downloaded.

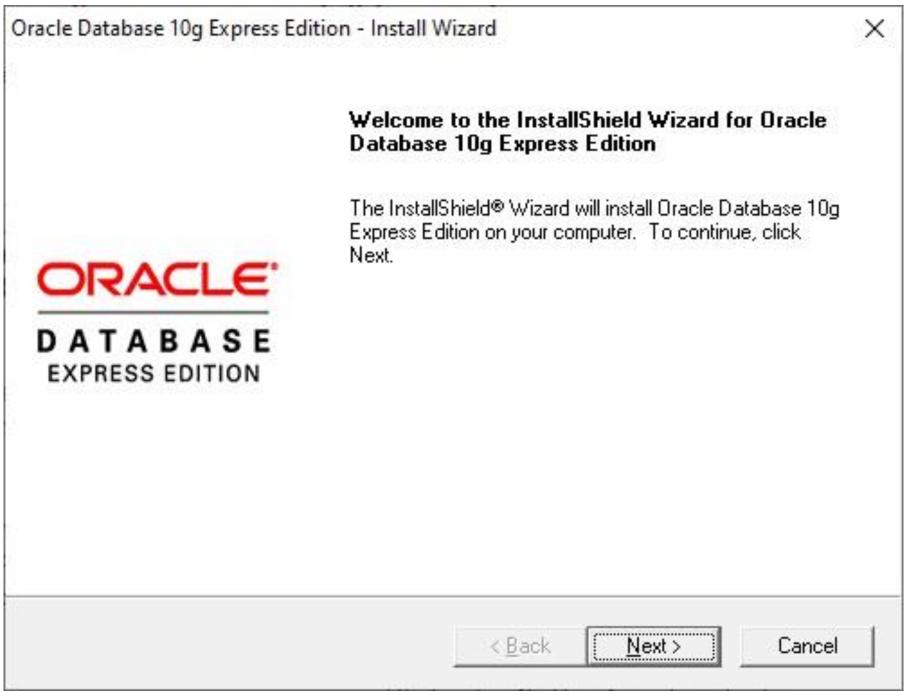

InstallShield Wizard

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

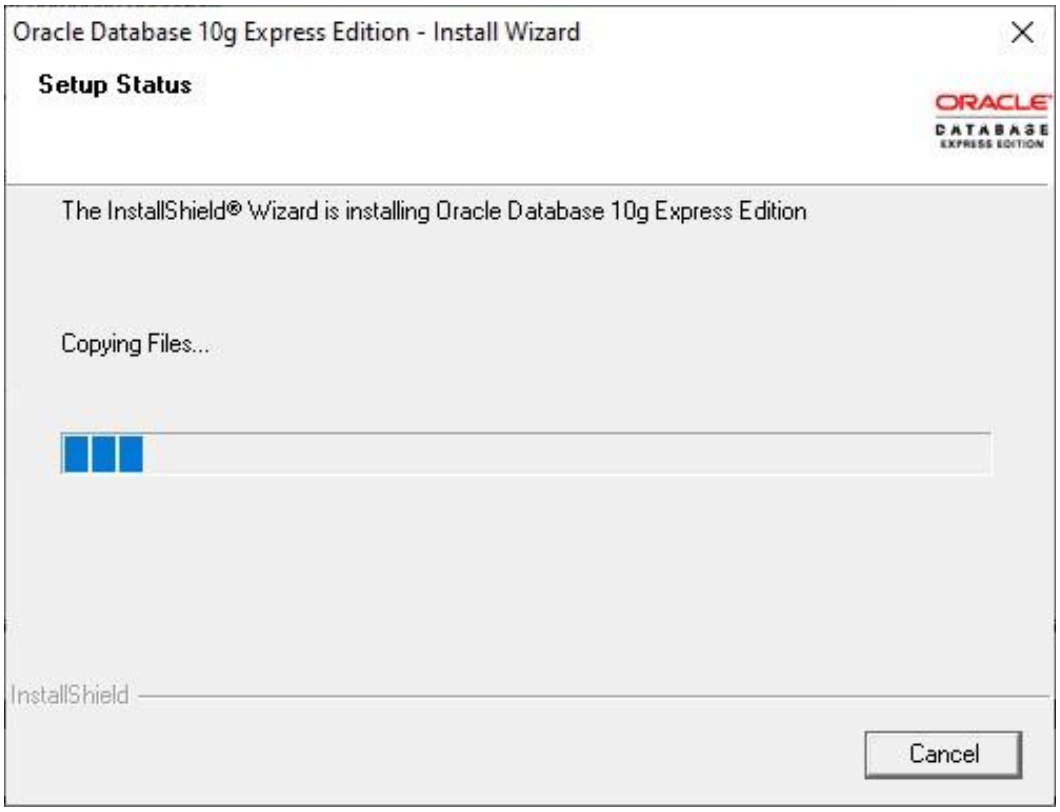
Step 3: Clicking on Next button.


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.

Step 4: Installing Oracle 10 g.



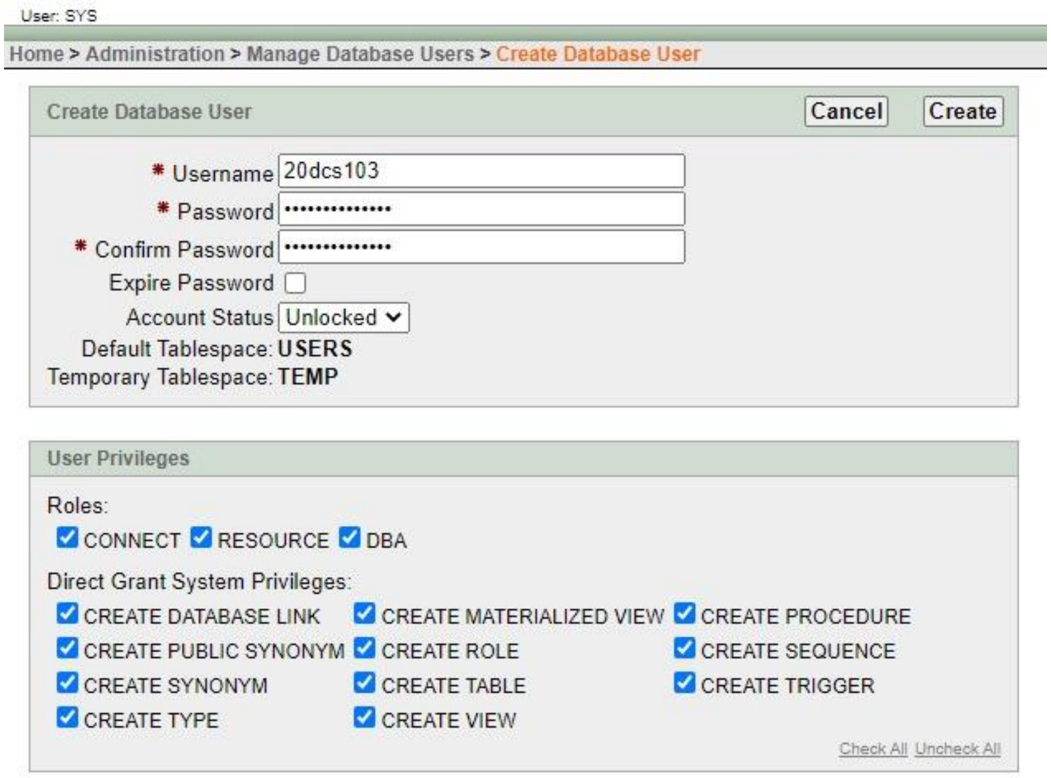
Oracle Database 10g Express Edition - Install Wizard

Setup Status

The InstallShield® Wizard is installing Oracle Database 10g Express Edition

Copying Files...

Step 5: Creating a new user database by entering the username and password.



User: SYS

Home > Administration > Manage Database Users > Create Database User

Create Database User

* Username	20dcs103
* 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	

Aim 3: To study DDL-create and DML-insert commands.

Answer:

DEPOSIT:

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


INSERT ALL

INTO DEPOSIT VALUES(100, 'ANIL', 'VRCE', 1000.00, '1-MAR-95')

INTO DEPOSIT VALUES(101, 'SUNIL', 'AJNI', 5000.00, '4-JAN-96')

INTO DEPOSIT VALUES(102, 'MEHUL', 'KAROLBAGH', 3500.00, '17-NOV-95')

INTO DEPOSIT VALUES(104, 'MADHURI', 'CHANDI', 1200.00, '17-DEC-95')

INTO DEPOSIT VALUES(105, 'PRAMOD', 'M.G.ROAD', 3000.00, '27-MAR-96')

INTO DEPOSIT VALUES(106, 'SANDIP', 'ANDHERI', 2000.00, '31-MAR-96')

INTO DEPOSIT VALUES(107, 'SHIVANI', 'VIRAR', 1000.00, '5-SEP-95')

INTO DEPOSIT VALUES(108, 'KRANTI', 'NEHRU PLACE', 5000.00, '2-JUL-95')

INTO DEPOSIT VALUES(109, 'MINU', 'POWAI', 7000.00, '10-AUG-95')

SELECT * FROM DUAL
```

BRANCH:

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


INSERT ALL

INTO BRANCH VALUES('VRCE', 'NAGPUR')

INTO BRANCH VALUES('AJNI', 'NAGPUR')

INTO BRANCH VALUES('KAROLBAGH', 'DELHI')

INTO BRANCH VALUES('CHANDI', 'DELHI')

INTO BRANCH VALUES('DHARAMPETH', 'NAGPUR')

INTO BRANCH VALUES('M.G.ROAD', 'BANGLORE')

INTO BRANCH VALUES('ANDHERI', 'BOMBAY')

INTO BRANCH VALUES('VIRAR', 'BOMBAY')

INTO BRANCH VALUES('NEHRU PLACE', 'DELHI')

INTO BRANCH VALUES('POWAI', 'BOMBAY')

SELECT * FROM DUAL
```

CUSTOMERS:

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


INSERT ALL
INTO CUSTOMERS VALUES('ANIL', 'CALCUTTA')
INTO CUSTOMERS VALUES('SUNIL', 'DELHI')
INTO CUSTOMERS VALUES('MEHUL', 'BARODA')
INTO CUSTOMERS VALUES('MANDAR', 'PATNA')
INTO CUSTOMERS VALUES('MADHURI', 'NAGPUR')
INTO CUSTOMERS VALUES('PRAMOD', 'NAGPUR')
INTO CUSTOMERS VALUES('SANDIP', 'SURAT')
INTO CUSTOMERS VALUES('SHIVANI', 'BOMBAY')
INTO CUSTOMERS VALUES('KRANTI', 'BOMBAY')
INTO CUSTOMERS VALUES('NAREN', 'BOMBAY')
SELECT * FROM DUAL
```

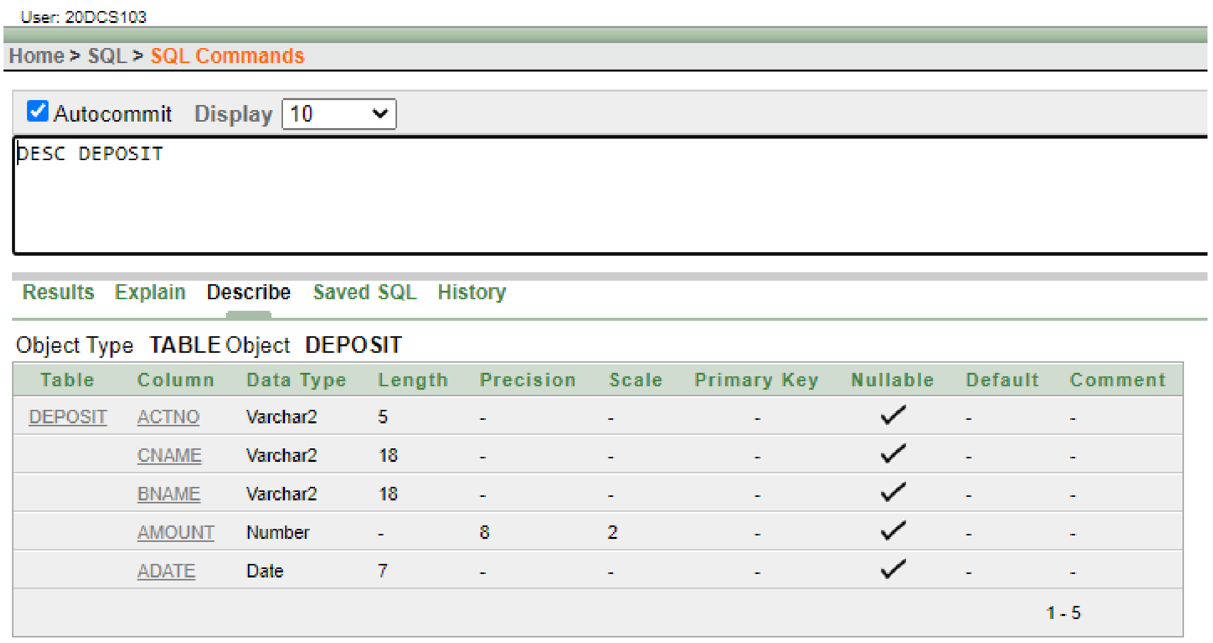
BORROW:

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


INSERT ALL
INTO BORROW VALUES(201 , 'ANIL', 'VRCE', 1000.00)
INTO BORROW VALUES(206 , 'MEHUL', 'AJNI', 5000.00)
INTO BORROW VALUES(311 , 'SUNIL', 'DHARAMPETH', 3000.00)
INTO BORROW VALUES(321 , 'MADHURI', 'ANDHERI', 2000.00)
INTO BORROW VALUES(375 , 'PRMOD', 'VIRAR', 8000.00)
INTO BORROW VALUES(481 , 'KRANTI', 'NEHRU PLACE', 3000.00)
SELECT * FROM DUAL
```

1) Describe deposit, branch.

<u>Answer:</u>

DESC DEPOSIT


User: 20DCS103

Home > SQL > SQL Commands

☑ Autocommit Display 10

```
DESC DEPOSIT
```

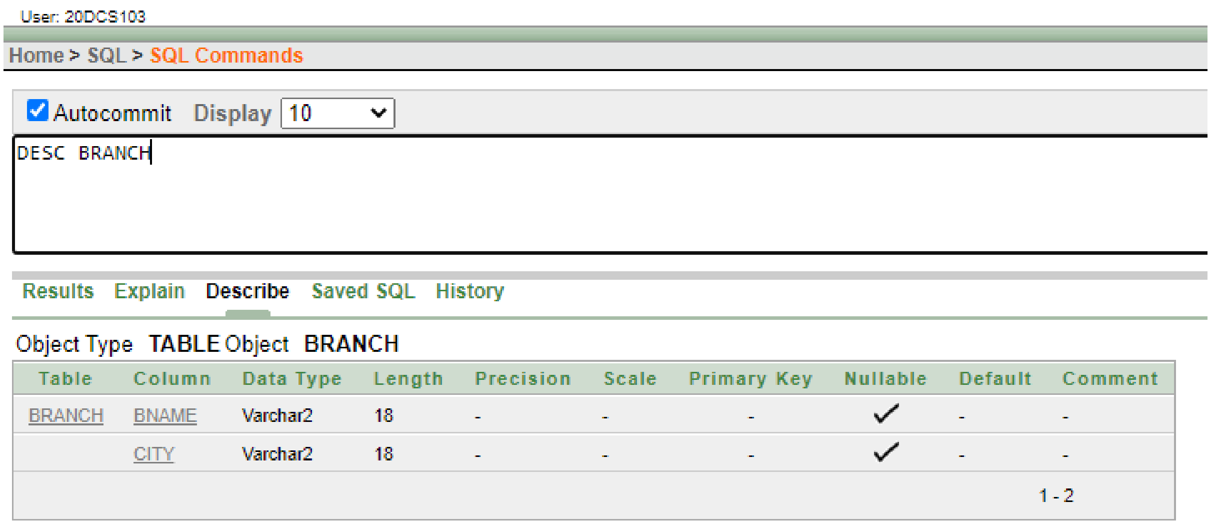
Results Explain Describe Saved SQL History

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

DESC BRANCH


User: 20DCS103

Home > SQL > SQL Commands

☑ Autocommit Display 10

```
DESC BRANCH
```

Results Explain Describe Saved SQL History

Object Type TABLE Object BRANCH

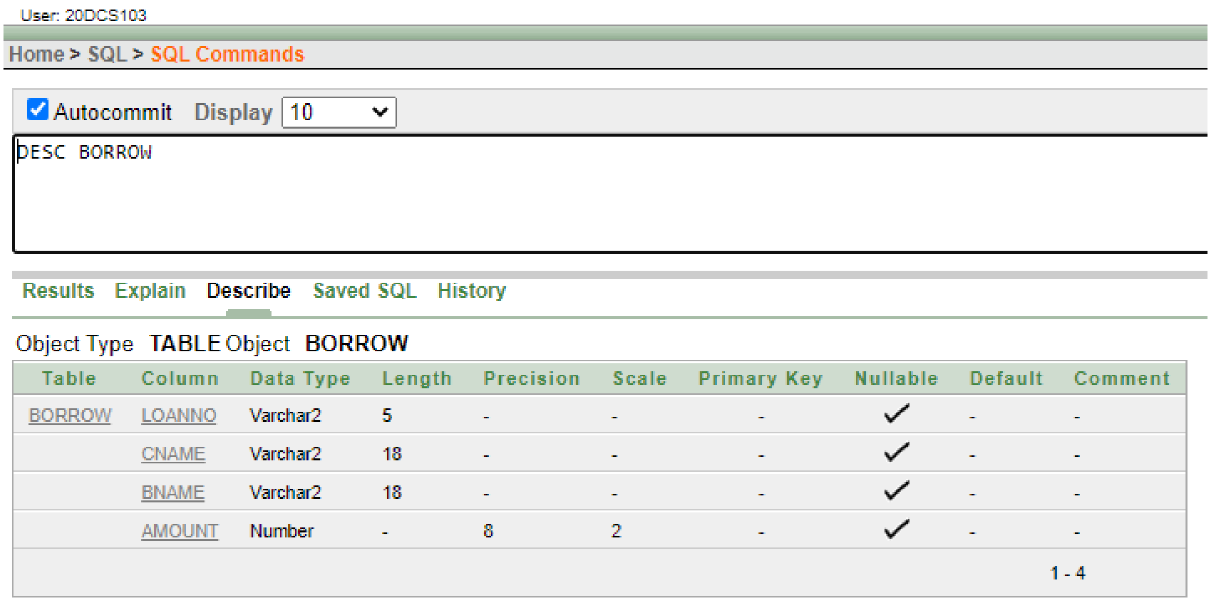
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
BRANCH	BNAME	Varchar2	18	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-

1 - 2

2) Describe borrow, customers.

<u>Answer:</u>

DESC BORROW


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
DESC BORROW
```

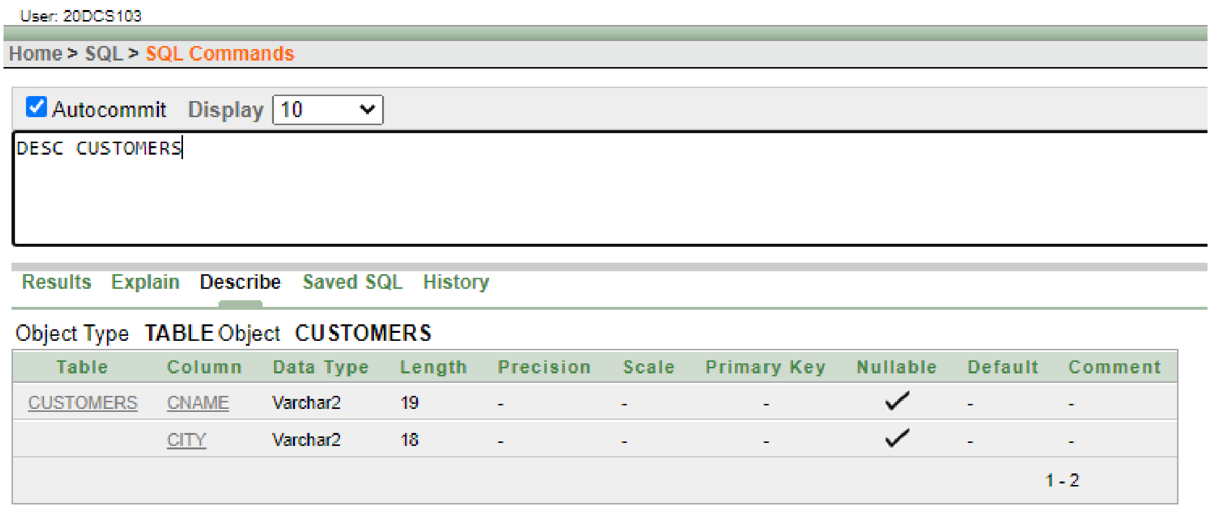
Results Explain Describe Saved SQL History

Object Type TABLE Object BORROW

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

1 - 4

DESC CUSTOMERS


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
DESC CUSTOMERS
```

Results Explain Describe Saved SQL History

Object Type TABLE Object CUSTOMERS

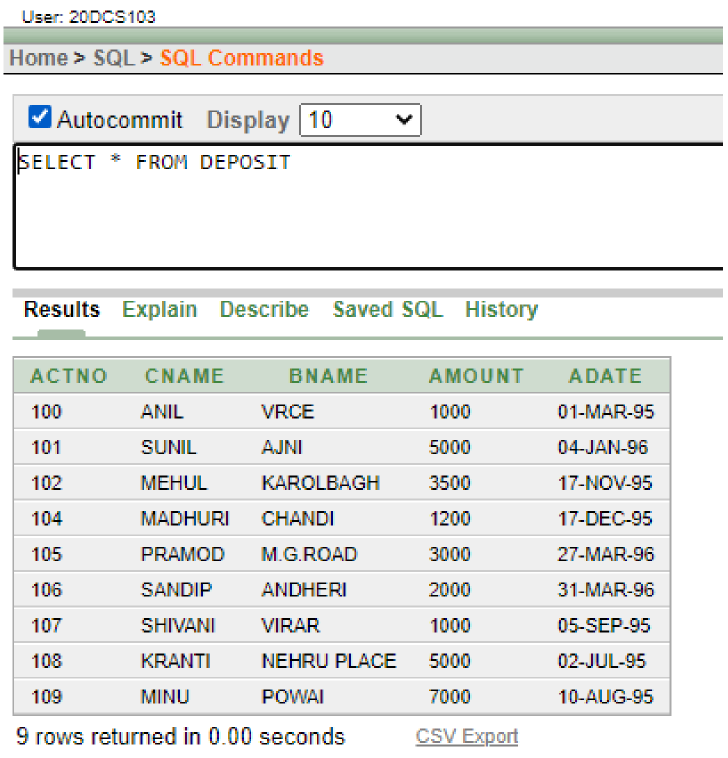
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
CUSTOMERS	CNAME	Varchar2	19	-	-	-	✓	-	-
	CITY	Varchar2	18	-	-	-	✓	-	-

1 - 2

3) List all data from table DEPOSIT.

<u>Answer:</u>

SELECT * FROM DEPOSIT


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM DEPOSIT
```

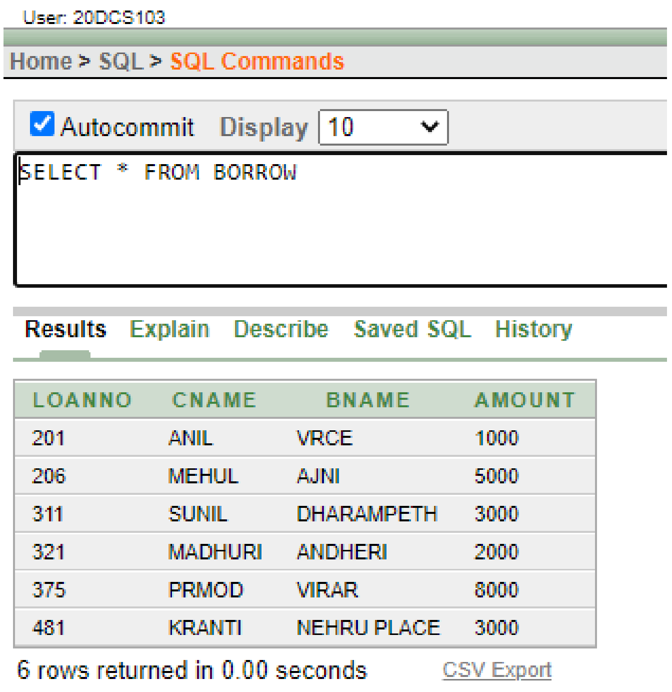
ACTNO	CNAME	BNAME	AMOUNT	ADATE
100	ANIL	VRCE	1000	01-MAR-95
101	SUNIL	AJNI	5000	04-JAN-96
102	MEHUL	KAROLBAGH	3500	17-NOV-95
104	MADHURI	CHANDI	1200	17-DEC-95
105	PRAMOD	M.G.ROAD	3000	27-MAR-96
106	SANDIP	ANDHERI	2000	31-MAR-96
107	SHIVANI	VIRAR	1000	05-SEP-95
108	KRANTI	NEHRU PLACE	5000	02-JUL-95
109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

9 rows returned in 0.00 seconds

4) List all data from table BORROW.

<u>Answer:</u>

SELECT * FROM BORROW


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM BORROW
```

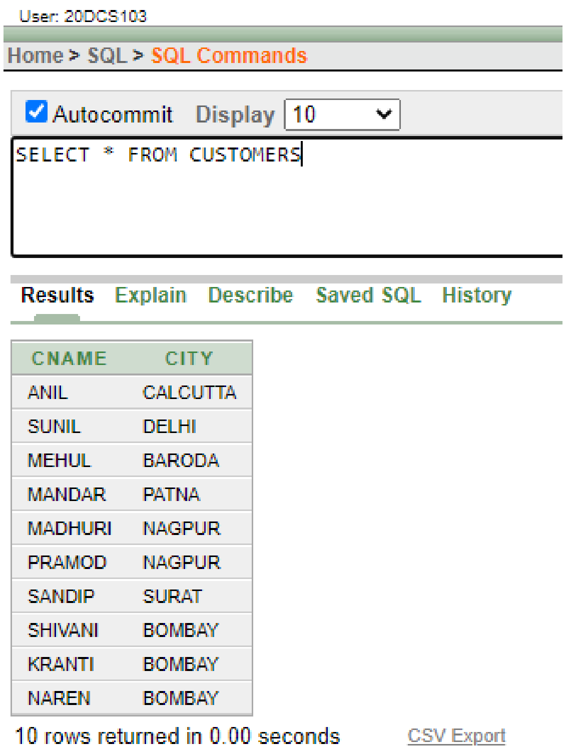
LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000
206	MEHUL	AJNI	5000
311	SUNIL	DHARAMPETH	3000
321	MADHURI	ANDHERI	2000
375	PRMOD	VIRAR	8000
481	KRANTI	NEHRU PLACE	3000

6 rows returned in 0.00 seconds

5) List all data from table CUSTOMERS.

<u>Answer:</u>

SELECT * FROM CUSTOMERS


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM CUSTOMERS
```

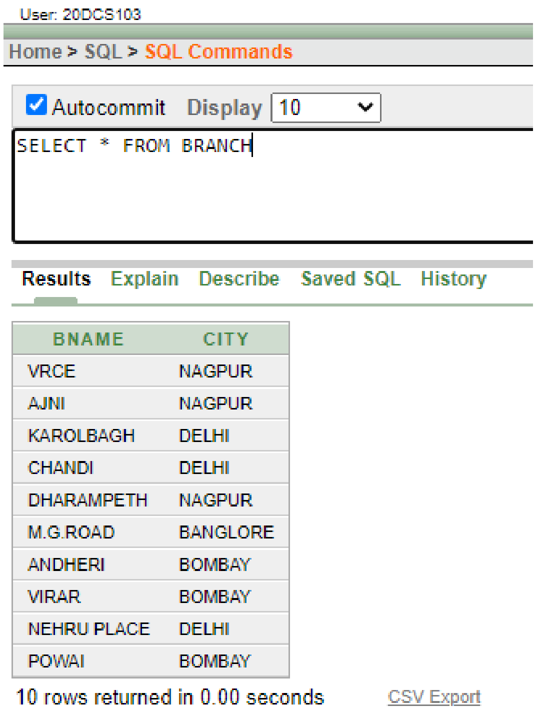
CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

10 rows returned in 0.00 seconds

6) List all data from table BRANCH.

<u>Answer:</u>

SELECT * FROM BRANCH


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM BRANCH
```

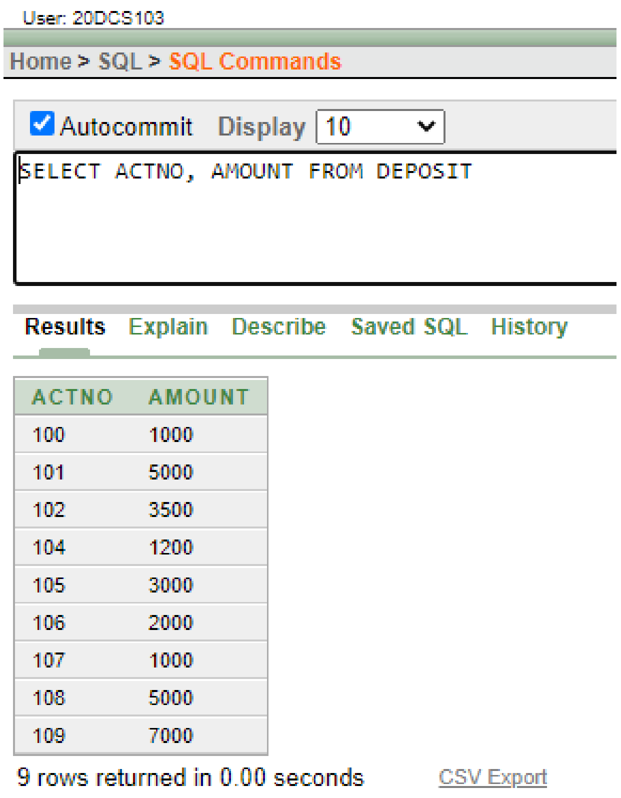
BNAME	CITY
VRCE	NAGPUR
AJNI	NAGPUR
KAROLBAGH	DELHI
CHANDI	DELHI
DHARAMPETH	NAGPUR
M.G.ROAD	BANGLORE
ANDHERI	BOMBAY
VIRAR	BOMBAY
NEHRU PLACE	DELHI
POWAI	BOMBAY

10 rows returned in 0.00 seconds

7) Give account no and amount of depositors.

Answer:

SELECT ACTNO, AMOUNT FROM DEPOSIT


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
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

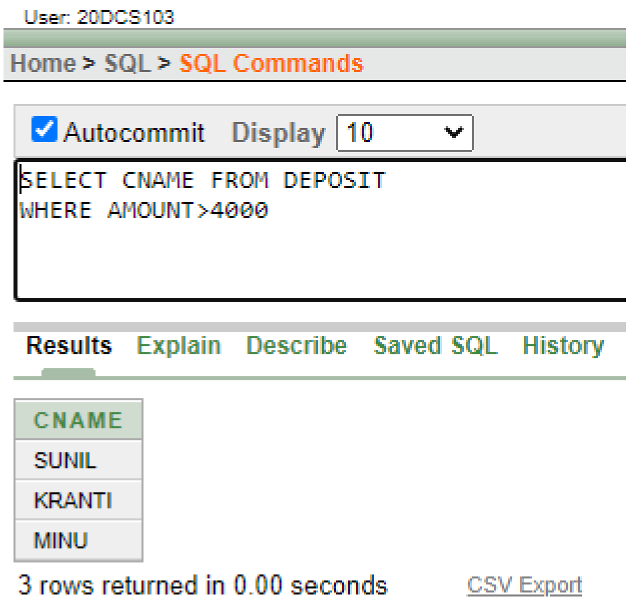
9 rows returned in 0.00 seconds

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

Answer:

SELECT CNAME FROM DEPOSIT

WHERE AMOUNT>4000


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT CNAME FROM DEPOSIT
WHERE AMOUNT>4000
```

CNAME
SUNIL
KRANTI
MINU

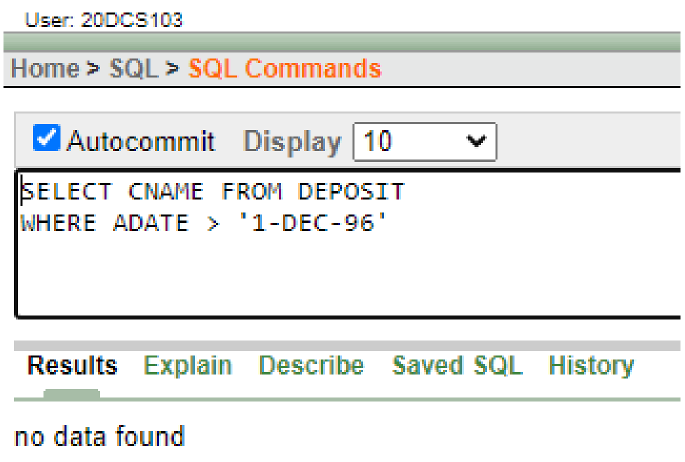
3 rows returned in 0.00 seconds

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

Answer:

SELECT CNAME FROM DEPOSIT

WHERE ADATE > '1-DEC-96'


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT CNAME FROM DEPOSIT
WHERE ADATE > '1-DEC-96'
```

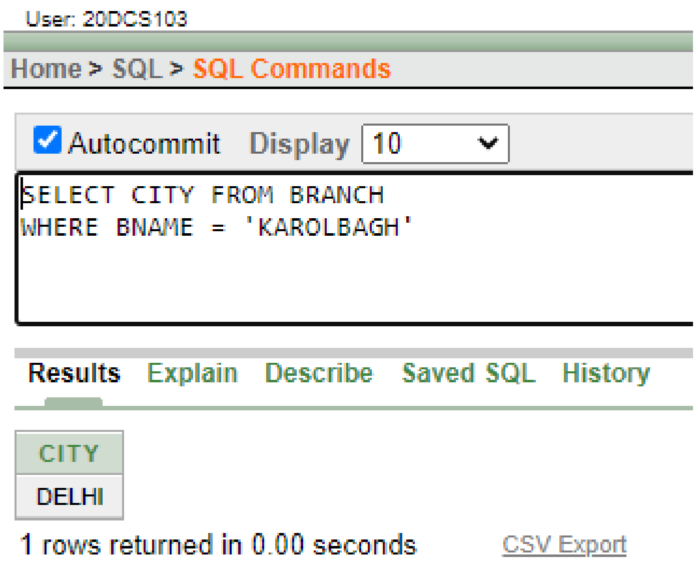
```
no data found
```

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

Answer:

SELECT CITY FROM BRANCH

WHERE BNAME = 'KAROLBAGH'


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT CITY FROM BRANCH
WHERE BNAME = 'KAROLBAGH'
```

CITY
DELHI

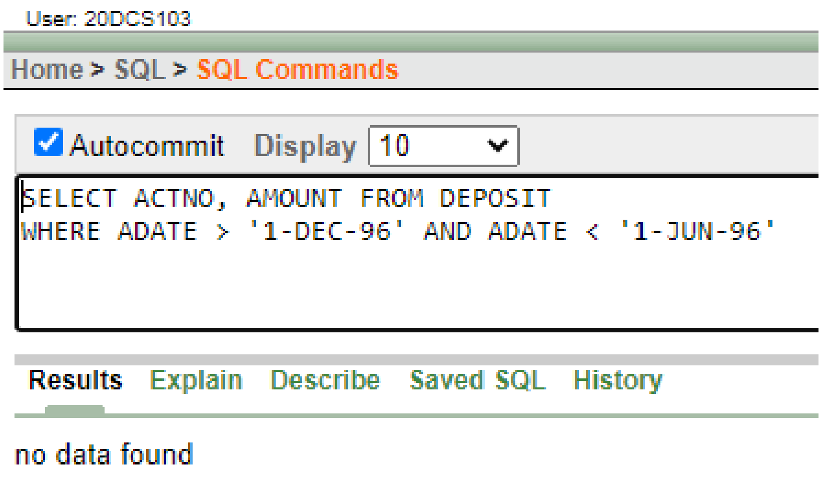
1 rows returned in 0.00 seconds

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

Answer:

SELECT ACTNO, AMOUNT FROM DEPOSIT

WHERE ADATE > '1-DEC-96' AND ADATE < '1-JUN-96'



```
SELECT ACTNO, AMOUNT FROM DEPOSIT
WHERE ADATE > '1-DEC-96' AND ADATE < '1-JUN-96'
```

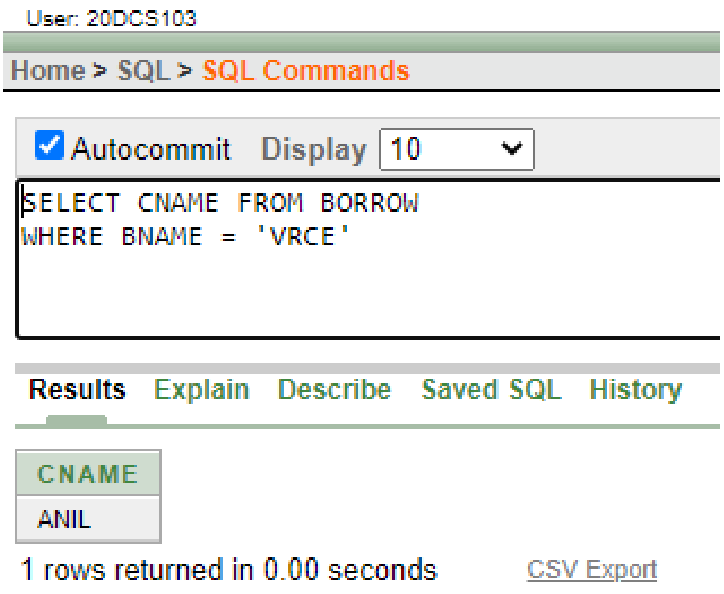
```
no data found
```

12) Give names of depositors having account at VRCE.

Answer:

SELECT CNAME FROM BORROW

WHERE BNAME = 'VRCE'



```
SELECT CNAME FROM BORROW
WHERE BNAME = 'VRCE'
```

CNAME
ANIL

1 rows returned in 0.00 seconds

Aim 4: Create the below given table and insert the data accordingly.

Answer:

<u>EMPLOYEE:</u>

```
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

)


INSERT ALL

INTO Employee VALUES(101, 'Smith',   800,  NULL,  20, 'shah',   'machine
learning',        'fig_mgr',  'toronto' ,    105,   '09-aug-96')

INTO Employee VALUES(102, 'Snehal',  1600, 300,   25, 'gupta',  'data science',
'lec',      'las vegas' ,  NULL,  '09-aug-96')

INTO Employee VALUES(103, 'Adama',   1100, 0,     20, 'wales',  'machine
learning',        'mk_mgr',   'ontario' ,    105,   '30-nov-95')

INTO Employee VALUES(104, 'Aman',    3000, NULL,  15, 'sharma', 'virtual reality',
'comp_op',  'mexico' ,     12,    '02-oct-97')

INTO Employee VALUES(105, 'Anita',   5000, 50000, 10, 'patel',  'big data
analytics',      'comp_op',  'germany' ,    107,   '01-jan-98')

INTO Employee VALUES(106, 'Sneha',   2450, 24500, 10, 'joseph', 'big data
analytics',      'fi_acc',   'melbourne' ,  105,   '26-sep-97')

INTO Employee VALUES(107, 'Anamika', 2975, NULL,  30, 'jha',    'artificial
intelligence', 'it_prog',  'new york' ,   NULL,  '15-jul-97')

SELECT * FROM DUAL
```

JOB:

```
CREATE TABLE Job
(
job_id varchar2(15),
job_title varchar2(30),
min_sal number(7,2),
max_sal number(7,2)
)

INSERT ALL
INTO Job VALUES ( 'it_prog', 'Programmer', 4000, 10000)
INTO Job VALUES ( 'mk_mgr', 'Marketing manager', 9000, 15000)
INTO Job VALUES ( 'fi_mgr', 'Finance manager', 8200, 12000)
INTO Job VALUES ( 'fi_acc', 'Account', 4200, 9000)
INTO Job VALUES ( 'lec', 'Lecturer', 6000, 17000)
INTO Job VALUES ( 'comp_op', 'Computer Operator', 1500, 3000)
SELECT * FROM DUAL
```

DEPOSIT:

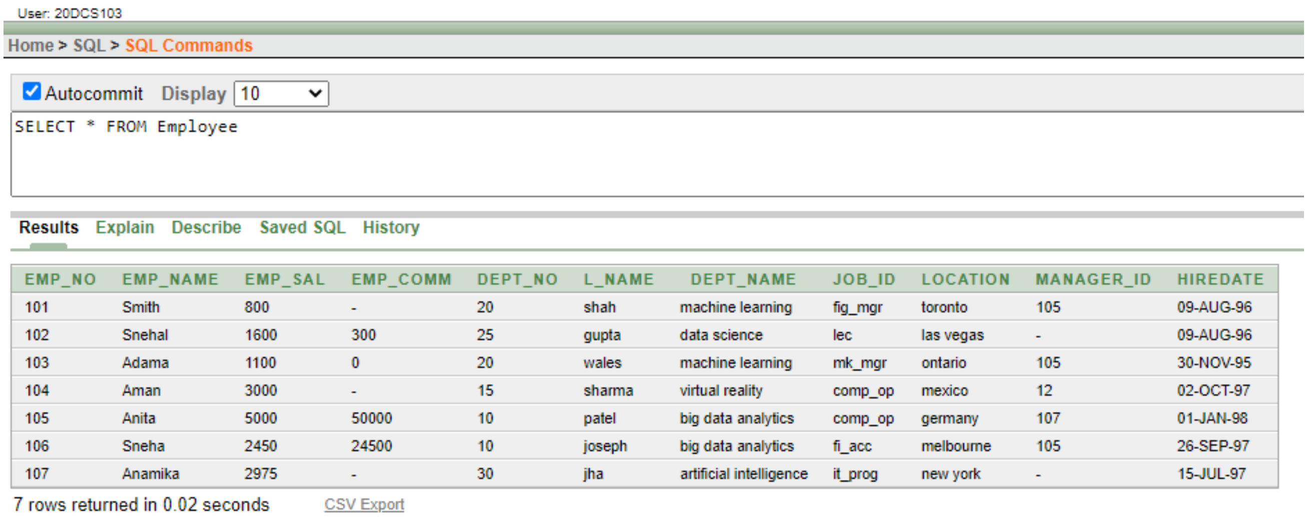
```
CREATE TABLE deposit
(
a_no varchar2(5),
cname varchar2(15),
bname varchar2(10),
amount number(7,2),
a_date date
)

INSERT ALL
INTO deposit VALUES(101, 'Anil', 'andheri', 7000, '01-jan-06')
INTO deposit VALUES(102, 'sunil', 'virar', 5000, '15-jul-06')
INTO deposit VALUES(103, 'jay', 'villeparle', 6500, '12-mar-06')
INTO deposit VALUES(104, 'vijay', 'andheri ', 8000, '01-sep-06')

INTO deposit VALUES(105, 'keyur', 'dadar', 7500, '01-nov-06')

INTO deposit VALUES(106, 'mayur', 'borivali', 5500, '01-dec-06')

SELECT * FROM DUAL
```

PERFORM FOLLOWING QUERIES.

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

Answer:

SELECT * FROM Employee

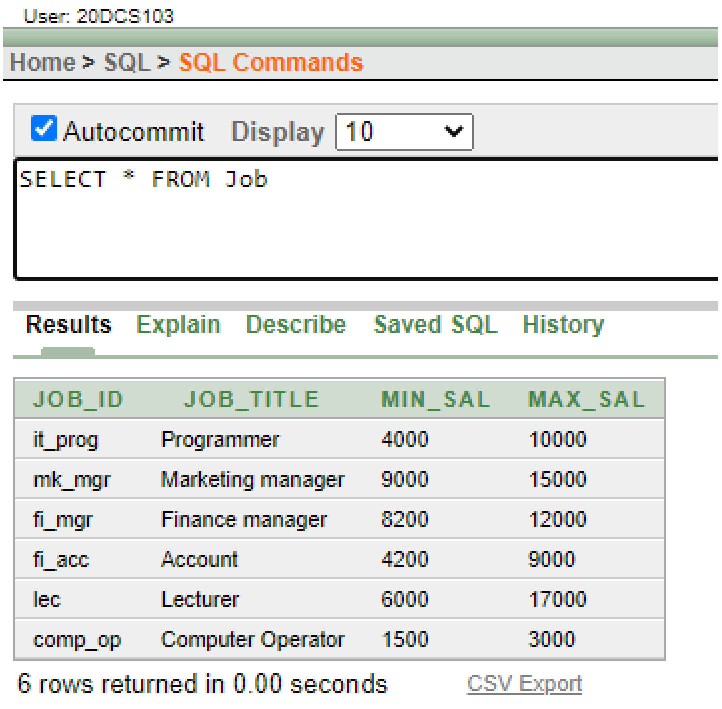

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
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	fig_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	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

7 rows returned in 0.02 seconds CSV Export

SELECT * FROM Job

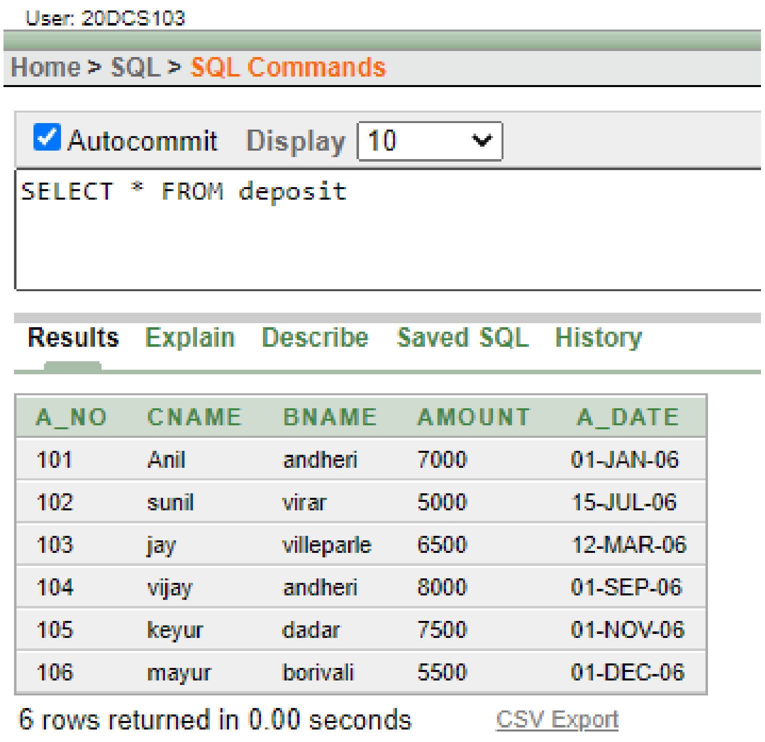

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
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

6 rows returned in 0.00 seconds CSV Export

SELECT * FROM deposit



```
SELECT * FROM deposit
```

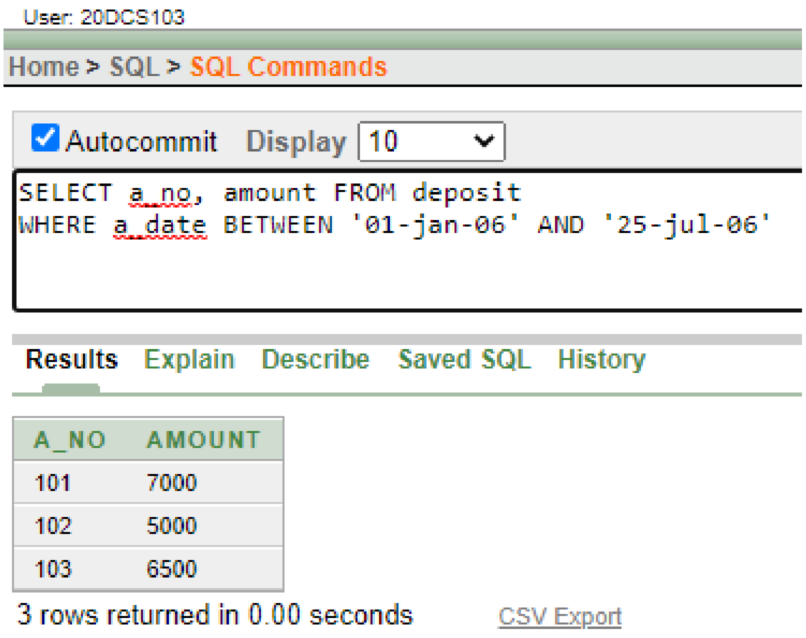
A_NO	CNAME	BNAME	AMOUNT	A_DATE
101	Anil	andheri	7000	01-JAN-06
102	sunil	virar	5000	15-JUL-06
103	jay	villeparle	6500	12-MAR-06
104	vijay	andheri	8000	01-SEP-06
105	keyur	dadar	7500	01-NOV-06
106	mayur	borivali	5500	01-DEC-06

6 rows returned in 0.00 seconds

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

Answer:

SELECT a_no, amount FROM deposit
WHERE a_date BETWEEN '01-jan-06' AND '25-jul-06'



```
SELECT a_no, amount FROM deposit
WHERE a_date BETWEEN '01-jan-06' AND '25-jul-06'
```

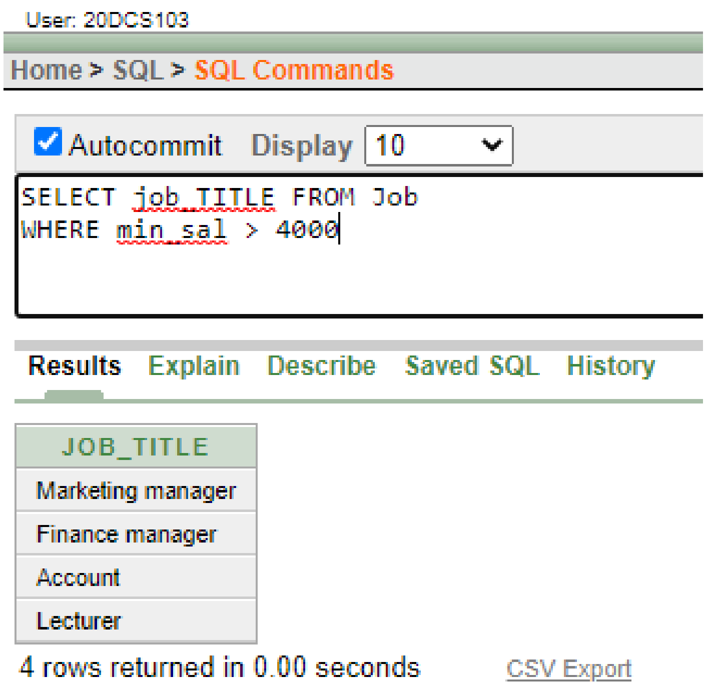
A_NO	AMOUNT
101	7000
102	5000
103	6500

3 rows returned in 0.00 seconds

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

Answer:

SELECT job_TITLE FROM Job
WHERE min_sal > 4000



```
SELECT job_TITLE FROM Job
WHERE min_sal > 4000
```

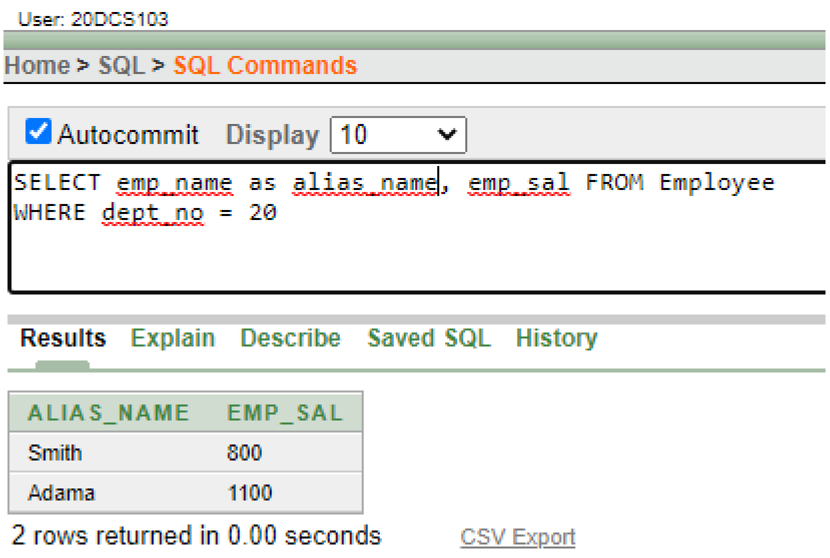
JOB_TITLE
Marketing manager
Finance manager
Account
Lecturer

4 rows returned in 0.00 seconds

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

Answer:

SELECT emp_name as alias_name, emp_sal FROM Employee
WHERE dept_no = 20



```
SELECT emp_name as alias_name, emp_sal FROM Employee
WHERE dept_no = 20
```

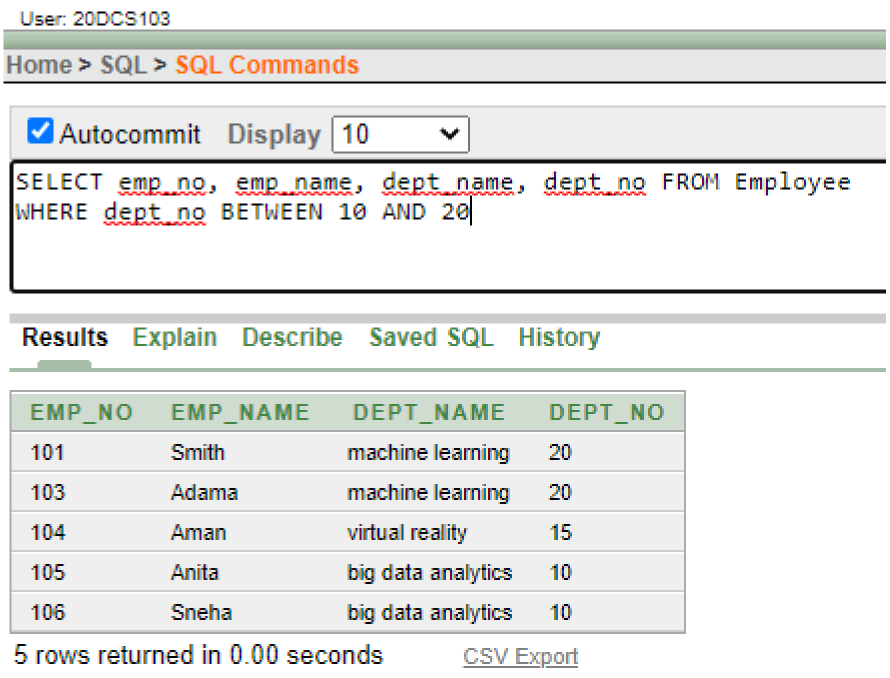
ALIAS_NAME	EMP_SAL
Smith	800
Adama	1100

2 rows returned in 0.00 seconds

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

Answer:

SELECT emp_no, emp_name, dept_name, dept_no FROM Employee
WHERE dept_no BETWEEN 10 AND 20


User: 20DCS103

Home > SQL > SQL Commands

Autocommit: ✓ Display: 10

```
SELECT emp_no, emp_name, dept_name, dept_no FROM Employee
WHERE dept_no BETWEEN 10 AND 20
```

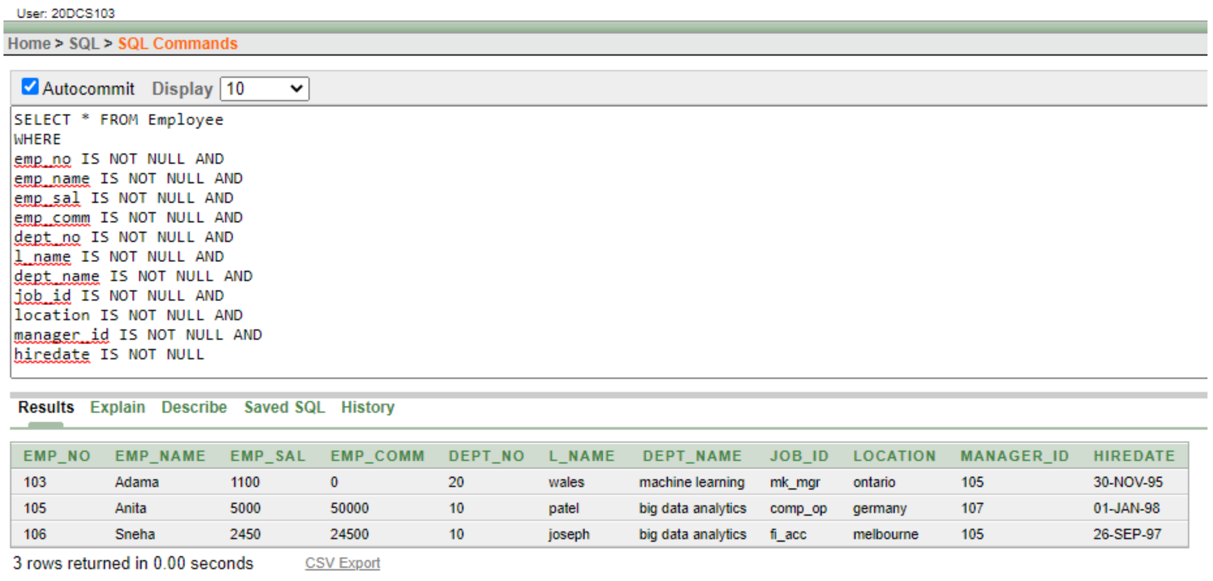
EMP_NO	EMP_NAME	DEPT_NAME	DEPT_NO
101	Smith	machine learning	20
103	Adama	machine learning	20
104	Aman	virtual reality	15
105	Anita	big data analytics	10
106	Sneha	big data analytics	10

5 rows returned in 0.00 seconds

6) Display the non-null values of employees.

Answer:

SELECT * FROM Employee
WHERE
emp_no IS NOT NULL AND
emp_name IS NOT NULL AND
emp_sal IS NOT NULL AND
emp_comm IS NOT NULL AND
dept_no IS NOT NULL AND
l_name IS NOT NULL AND
dept_name IS NOT NULL AND
job_id IS NOT NULL AND
location IS NOT NULL AND
manager_id IS NOT NULL AND
hiredate IS NOT NULL


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM Employee
WHERE
emp_no IS NOT NULL AND
emp_name IS NOT NULL AND
emp_sal IS NOT NULL AND
emp_comm IS NOT NULL AND
dept_no IS NOT NULL AND
l_name IS NOT NULL AND
dept_name IS NOT NULL AND
job_id IS NOT NULL AND
location IS NOT NULL AND
manager_id IS NOT NULL AND
hiredate IS NOT NULL
```

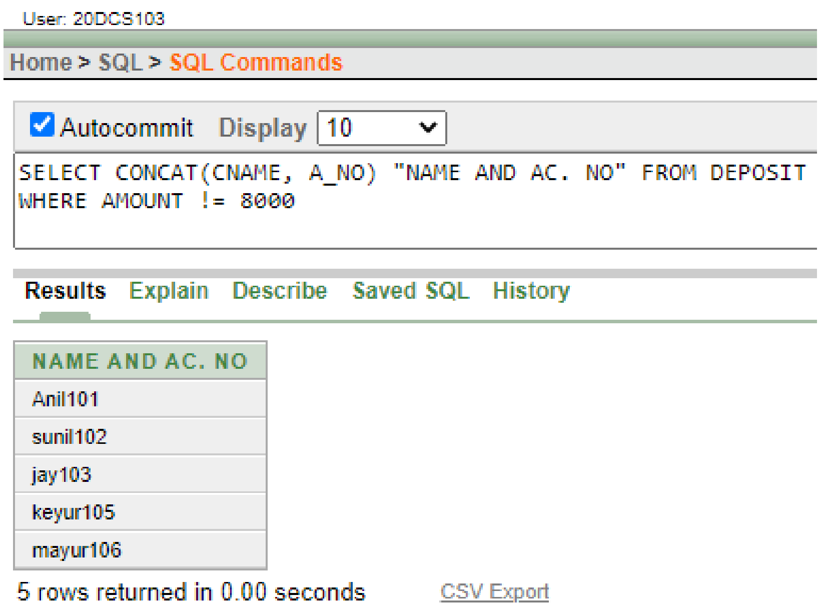
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

3 rows returned in 0.00 seconds

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.

Answer:

SELECT CONCAT(CNAME, A_NO) "NAME AND AC. NO" FROM DEPOSIT
WHERE AMOUNT != 8000


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT CONCAT(CNAME, A_NO) "NAME AND AC. NO" FROM DEPOSIT
WHERE AMOUNT != 8000
```

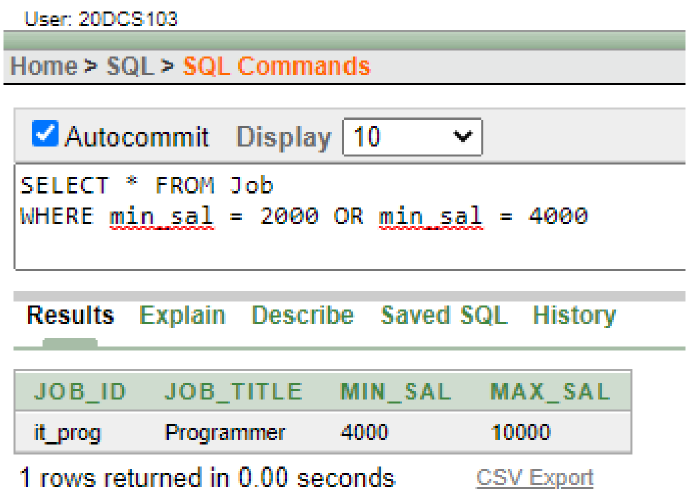
NAME AND AC. NO
Anil101
sunil102
jay103
keyur105
mayur106

5 rows returned in 0.00 seconds

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

Answer:

SELECT * FROM Job
WHERE min_sal = 2000 OR min_sal = 4000


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM Job
WHERE min_sal = 2000 OR min_sal = 4000
```

JOB_ID	JOB_TITLE	MIN_SAL	MAX_SAL
it_prog	Programmer	4000	10000

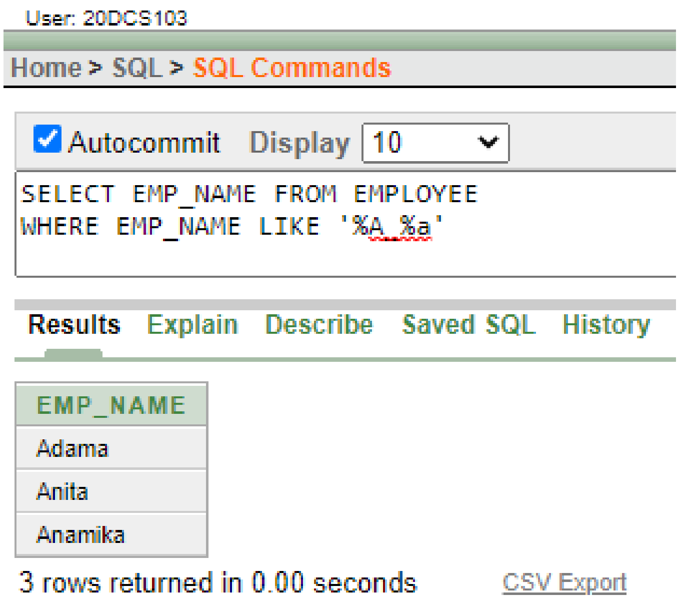
1 rows returned in 0.00 seconds

TO STUDY VARIOUS OPTIONS OF LIKE PREDICATE.

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

Answer:

SELECT EMP_NAME FROM EMPLOYEE
WHERE EMP_NAME LIKE 'A%'



User: 20DCS103

Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT EMP_NAME FROM EMPLOYEE
WHERE EMP_NAME LIKE '%A_%a'
```

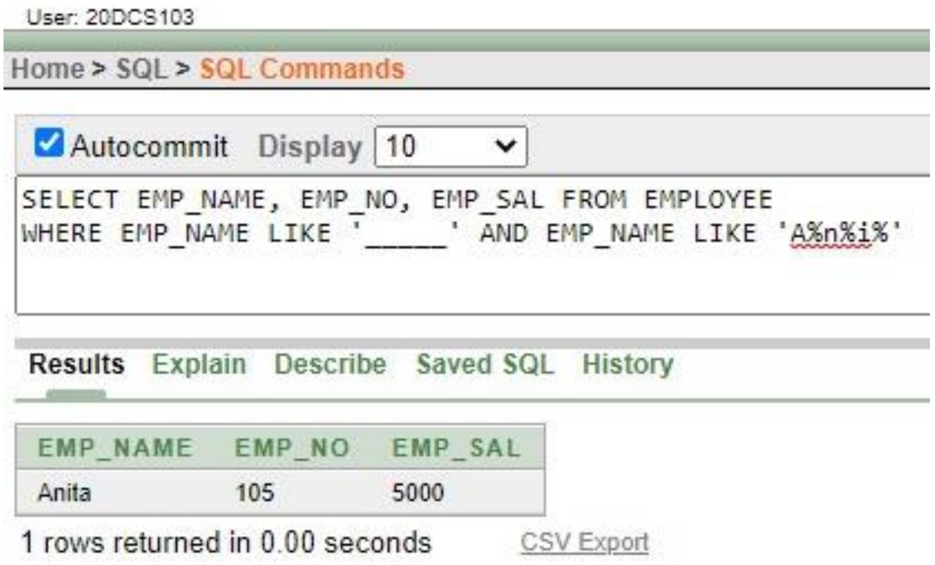
EMP_NAME
Adama
Anita
Anamika

3 rows returned in 0.00 seconds

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

Answer:

SELECT EMP_NAME, EMP_NO, EMP_SAL FROM EMPLOYEE
WHERE EMP_NAME LIKE '_____' AND EMP_NAME LIKE 'A%n%i%'



User: 20DCS103

Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT EMP_NAME, EMP_NO, EMP_SAL FROM EMPLOYEE
WHERE EMP_NAME LIKE '_____' AND EMP_NAME LIKE 'A%n%i%'
```

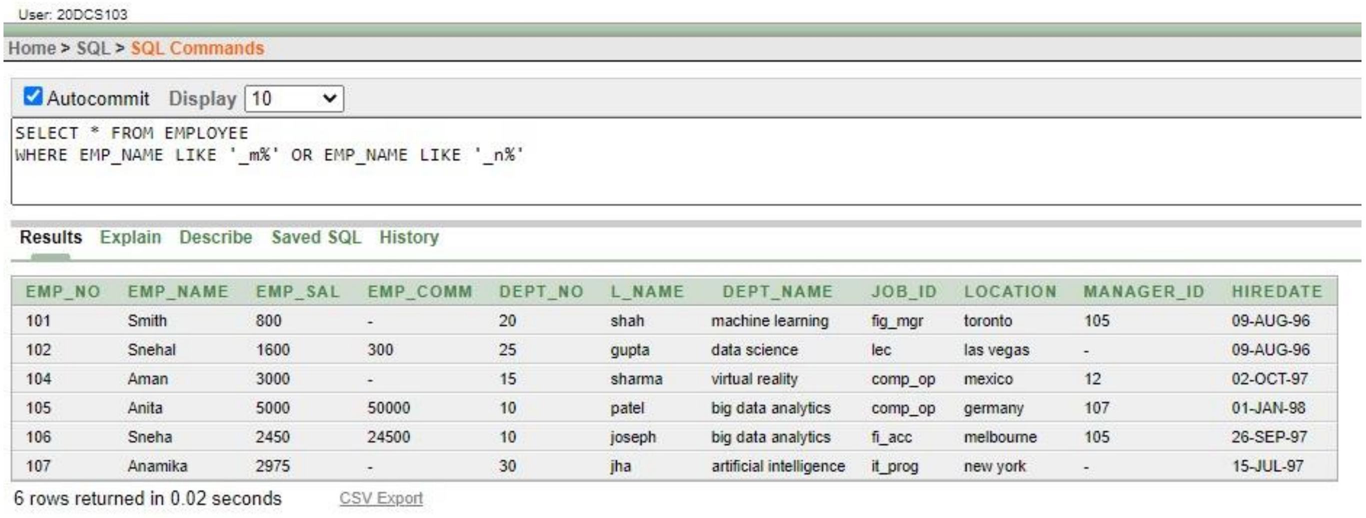
EMP_NAME	EMP_NO	EMP_SAL
Anita	105	5000

1 rows returned in 0.00 seconds

3) Display all information of employee whose second character of name is either ‘m’ or ‘n’.

Answer:

SELECT * FROM EMPLOYEE
WHERE EMP_NAME LIKE '_m%' OR EMP_NAME LIKE '_n%'


User: 20DCS103

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

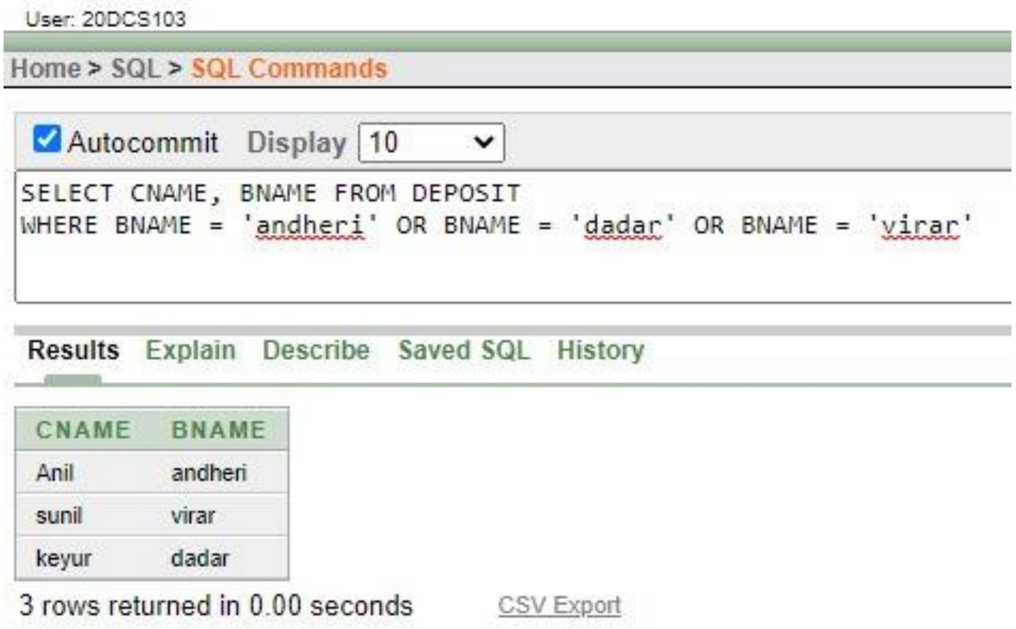
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fig_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
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

6 rows returned in 0.02 seconds CSV Export

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

Answer:

SELECT CNAME, BNAME FROM DEPOSIT
WHERE BNAME = 'andheri' OR BNAME = 'dadar' OR BNAME = 'virar '


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT CNAME, BNAME FROM DEPOSIT
WHERE BNAME = 'andheri' OR BNAME = 'dadar' OR BNAME = 'virar'
```

Results Explain Describe Saved SQL History

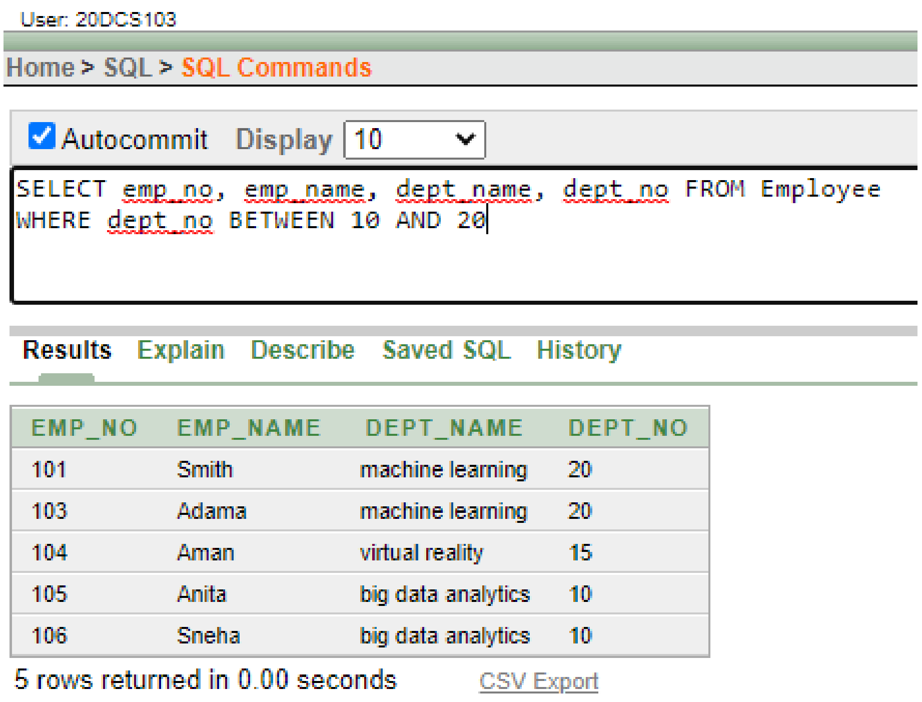
CNAME	BNAME
Anil	andheri
sunil	virar
keyur	dadar

3 rows returned in 0.00 seconds CSV Export

5) Display the job name whose first three character in job id field is ‘fi_’.

Answer:

SELECT JOB_TITLE FROM JOB
WHERE JOB_TITLE LIKE 'f%i%_'


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT emp_no, emp_name, dept_name, dept_no FROM Employee
WHERE dept_no BETWEEN 10 AND 20
```

Results Explain Describe Saved SQL History

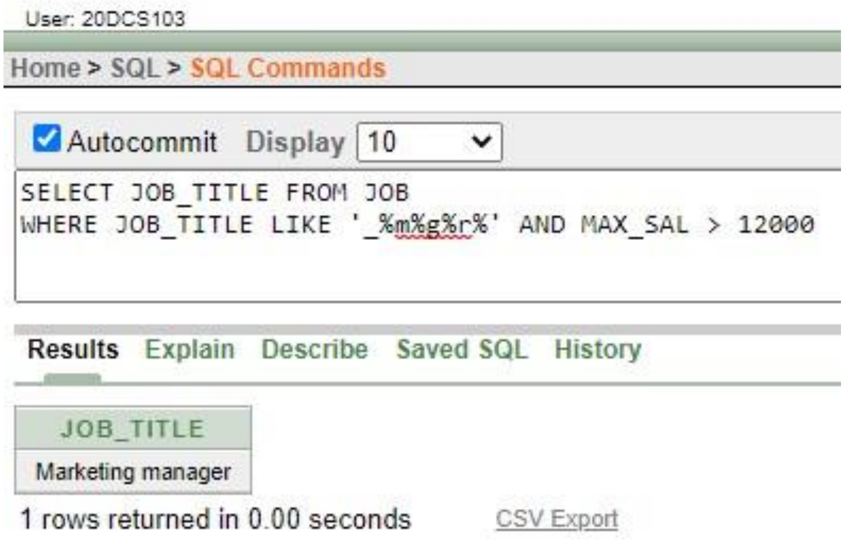
EMP_NO	EMP_NAME	DEPT_NAME	DEPT_NO
101	Smith	machine learning	20
103	Adama	machine learning	20
104	Aman	virtual reality	15
105	Anita	big data analytics	10
106	Sneha	big data analytics	10

5 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT JOB_TITLE FROM JOB
WHERE JOB_TITLE LIKE '_%m%g%r%' AND MAX_SAL > 12000


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT JOB_TITLE FROM JOB
WHERE JOB_TITLE LIKE '_%m%g%r%' AND MAX_SAL > 12000
```

Results Explain Describe Saved SQL History

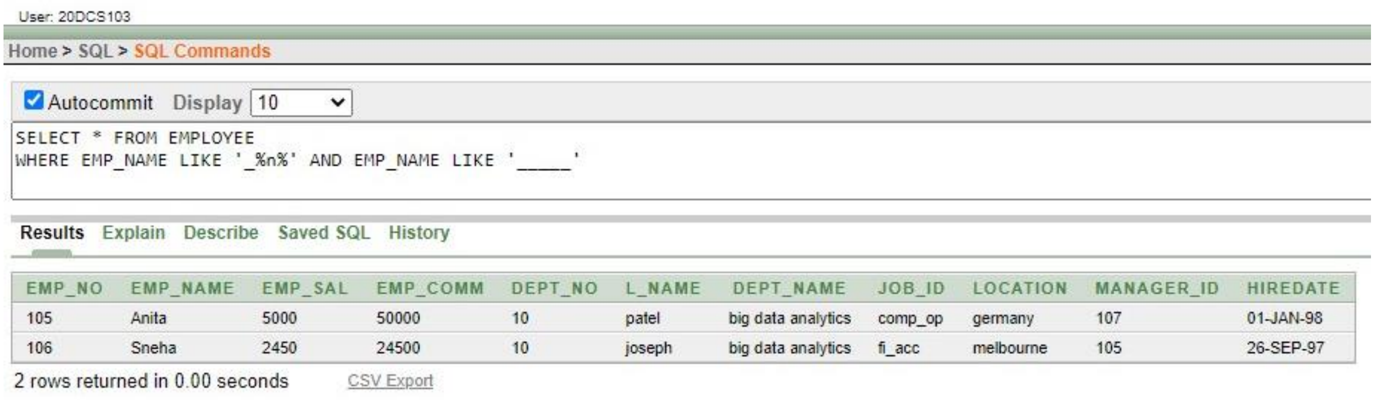
JOB_TITLE
Marketing manager

1 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT * FROM EMPLOYEE
WHERE EMP_NAME LIKE ‘_%n%’ AND EMP_NAME LIKE ‘_____’


User: 20DCS103
Home > SQL > SQL Commands

```
SELECT * FROM EMPLOYEE
WHERE EMP_NAME LIKE '_%n%' AND EMP_NAME LIKE '_____'
```

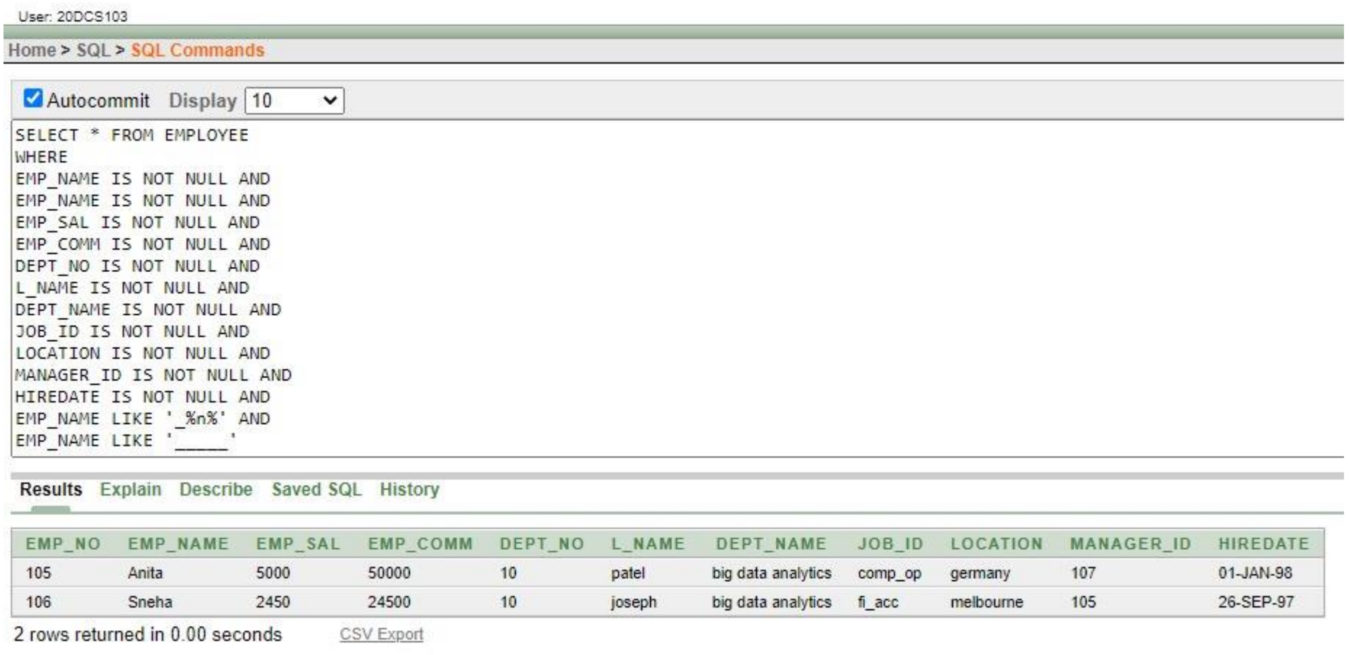
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds

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

Answer:

SELECT * FROM EMPLOYEE
WHERE
EMP_NAME IS NOT NULL AND
EMP_NAME IS NOT NULL AND
EMP_SAL IS NOT NULL AND
EMP_COMM IS NOT NULL AND
DEPT_NO IS NOT NULL AND
L_NAME IS NOT NULL AND
DEPT_NAME IS NOT NULL AND
JOB_ID IS NOT NULL AND
LOCATION IS NOT NULL AND
MANAGER_ID IS NOT NULL AND
HIREDATE IS NOT NULL AND
EMP_NAME LIKE '_%n%' AND
EMP_NAME LIKE '_____'


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM EMPLOYEE
WHERE
EMP_NAME IS NOT NULL AND
EMP_NAME IS NOT NULL AND
EMP_SAL IS NOT NULL AND
EMP_COMM IS NOT NULL AND
DEPT_NO IS NOT NULL AND
L_NAME IS NOT NULL AND
DEPT_NAME IS NOT NULL AND
JOB_ID IS NOT NULL AND
LOCATION IS NOT NULL AND
MANAGER_ID IS NOT NULL AND
HIREDATE IS NOT NULL AND
EMP_NAME LIKE '_%n%' AND
EMP_NAME LIKE '_____'
```

Results Explain Describe Saved SQL History

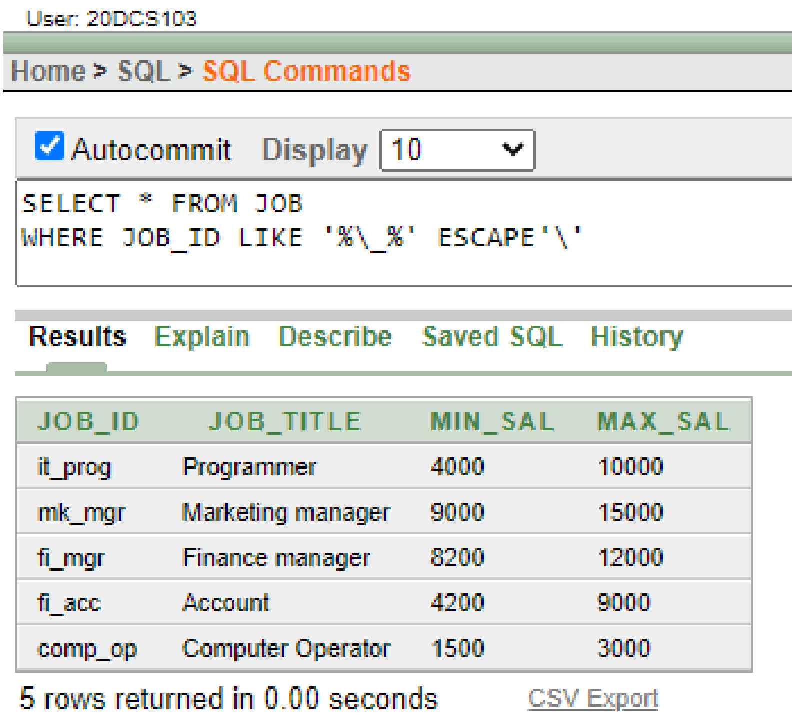
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT * FROM JOB
WHERE JOB_ID LIKE '%_%' ESCAPE'\'


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM JOB
WHERE JOB_ID LIKE '%\_%' ESCAPE'\'
```

Results Explain Describe Saved SQL History

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

5 rows returned in 0.00 seconds CSV Export

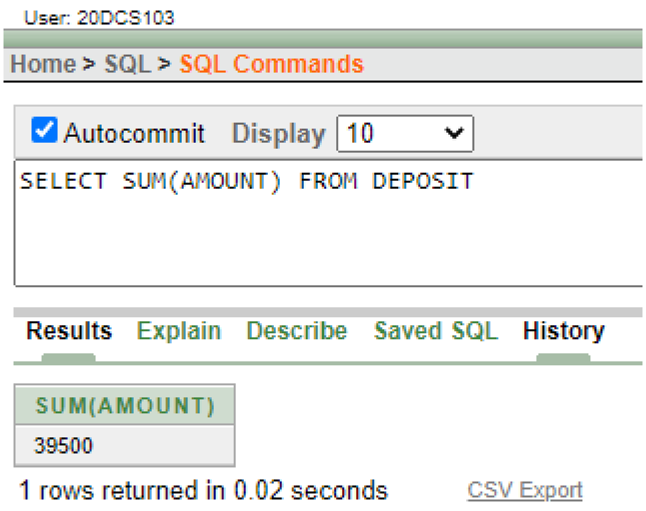
Aim 5: To Perform various data manipulation commands, aggregate functions and sorting concept on all created tables.

Answer:

1) List total deposit from deposit.

Answer:

SELECT SUM(AMOUNT) FROM DEPOSIT


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT SUM(AMOUNT) FROM DEPOSIT
```

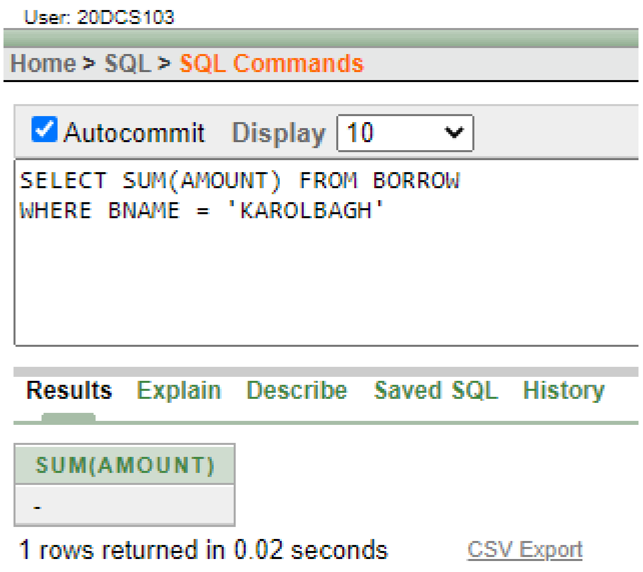
SUM(AMOUNT)
39500

1 rows returned in 0.02 seconds

2) List total loan from karolbagh branch

Answer:

SELECT SUM(LOAN) FROM BORROW
WHERE B_NAME = 'KAROLBAGH'


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT SUM(AMOUNT) FROM BORROW
WHERE BNAME = 'KAROLBAGH'
```

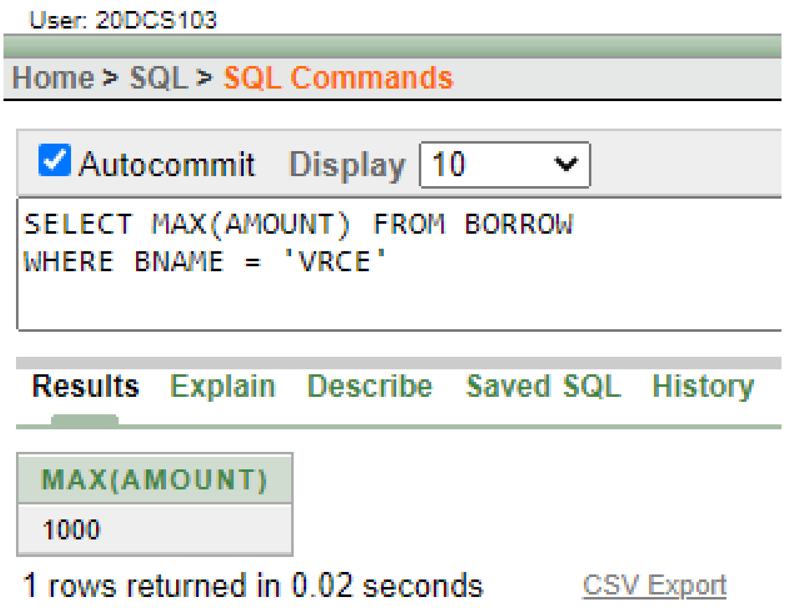
SUM(AMOUNT)
-

1 rows returned in 0.02 seconds

3) Give maximum loan from branch vrce.

Answer:

SELECT MAX(AMOUNT) FROM BORROW
WHERE BNAME = ‘VRCE’


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT MAX(AMOUNT) FROM BORROW
WHERE BNAME = 'VRCE'
```

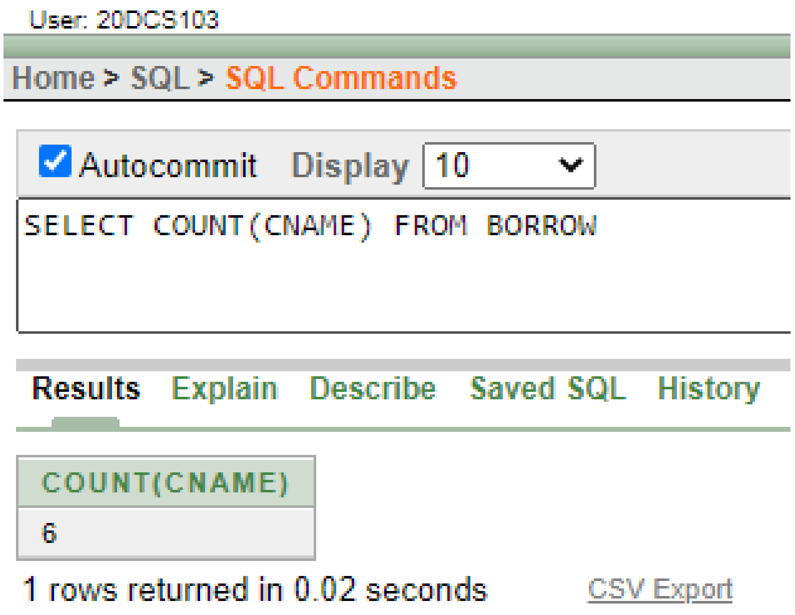
MAX(AMOUNT)
1000

1 rows returned in 0.02 seconds

4) Count total number of customers

Answer:

SELECT COUNT(CNAME) FROM BORROW


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT COUNT(CNAME) FROM BORROW
```

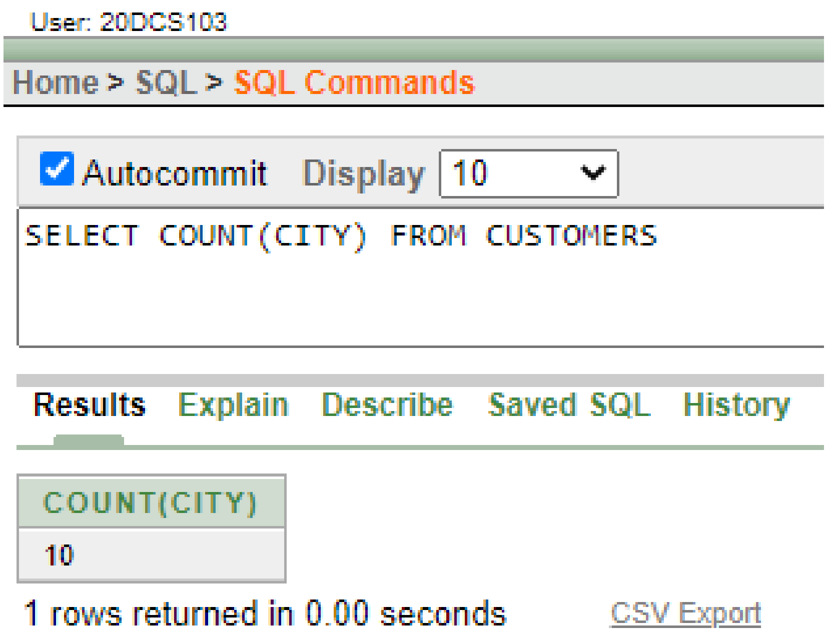
COUNT(CNAME)
6

1 rows returned in 0.02 seconds

5) Count total number of customer's cities.

Answer:

SELECT COUNT(CITY) FROM CUSTOMERS


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT COUNT(CITY) FROM CUSTOMERS
```

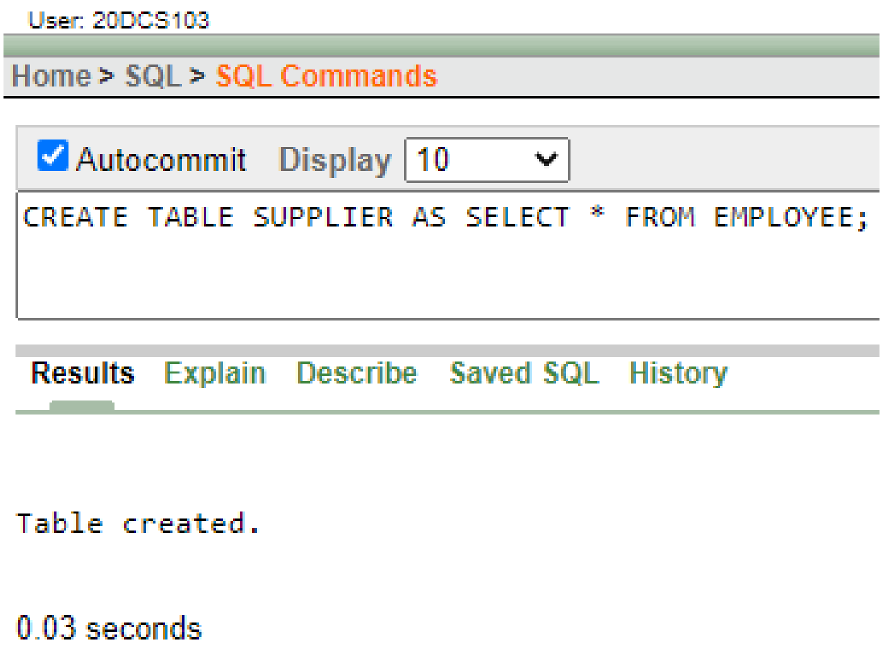
COUNT(CITY)
10

1 rows returned in 0.00 seconds

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

Answer:

CREATE TABLE SUPPLIER AS SELECT * FROM EMPLOYEE


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE TABLE SUPPLIER AS SELECT * FROM EMPLOYEE;
```

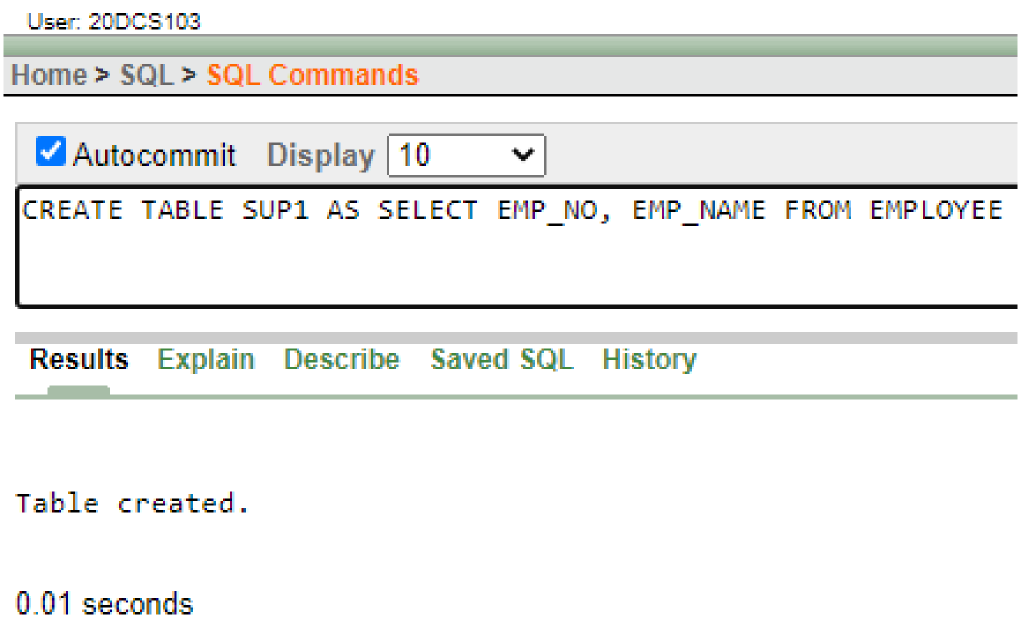
```
Table created.
```

0.03 seconds

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

Answer:

CREATE TABLE SUP1 AS SELECT EMP_NO, EMP_NAME FROM EMPLOYEE


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

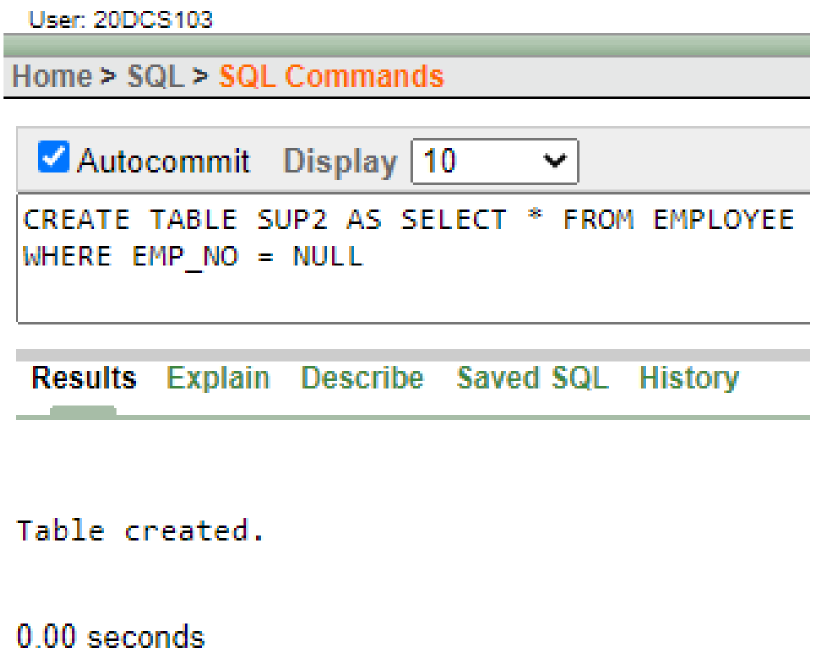
```
CREATE TABLE SUP1 AS SELECT EMP_NO, EMP_NAME FROM EMPLOYEE
```

```
Table created.

0.01 seconds
```

8) Create table sup2 from employee with no data

Answer:

CREATE TABLE SUP2 AS SELECT * FROM EMPLOYEE
WHERE EMP_NO = NULL


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

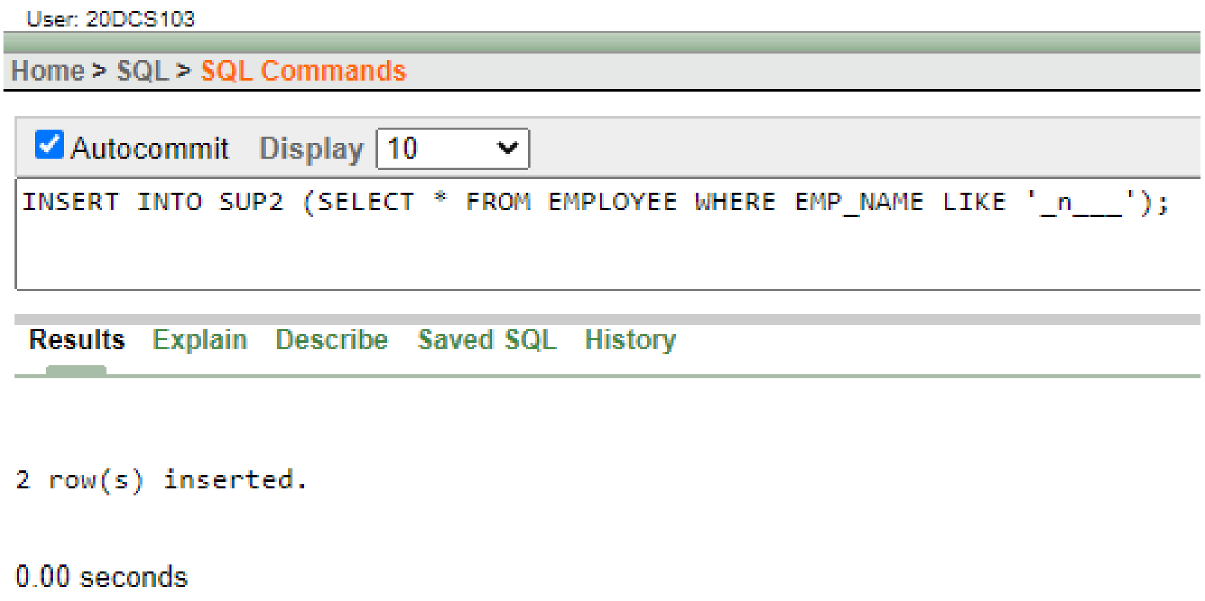
```
CREATE TABLE SUP2 AS SELECT * FROM EMPLOYEE
WHERE EMP_NO = NULL
```

```
Table created.

0.00 seconds
```

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

Answer:

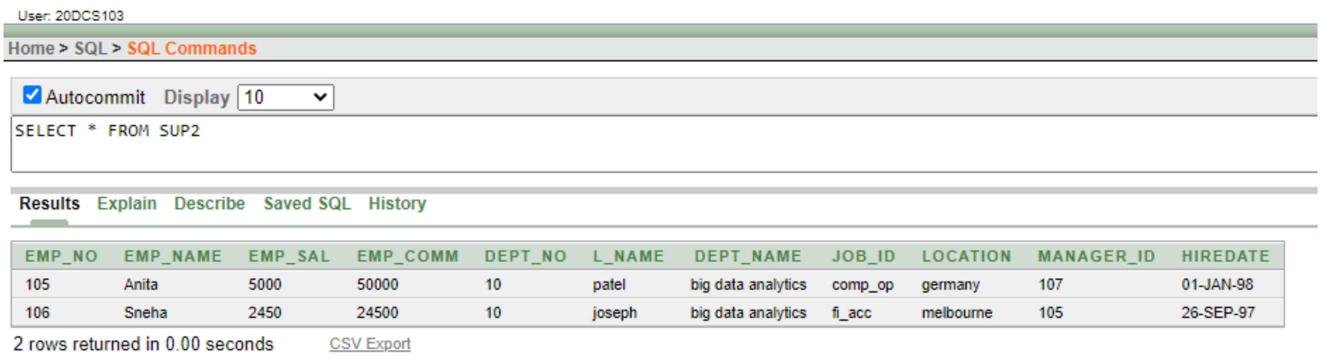
INSERT INTO SUP2 (SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE '_n___');


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
INSERT INTO SUP2 (SELECT * FROM EMPLOYEE WHERE EMP_NAME LIKE '_n___');
```

```
2 row(s) inserted.

0.00 seconds
```

SELECT * FROM SUP2


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM SUP2
```

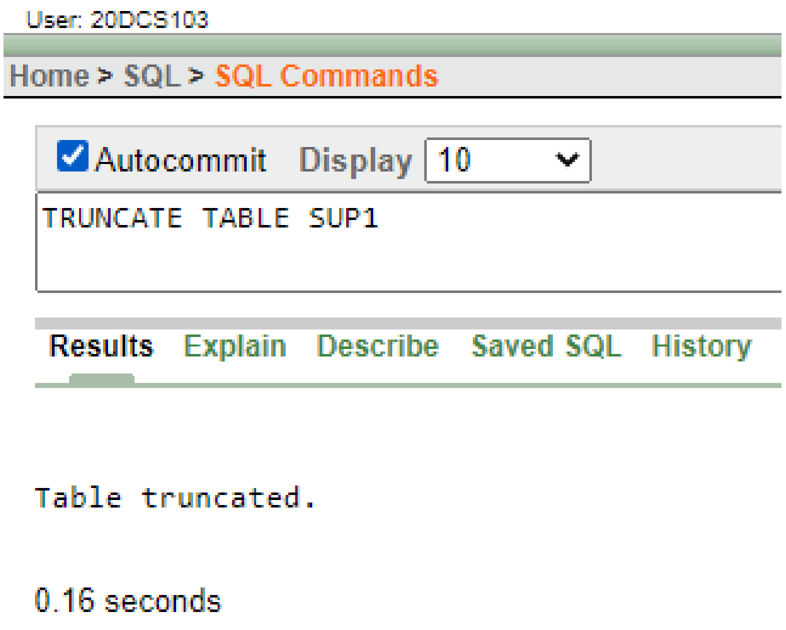
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds

10) Delete all the rows from sup1.

Answer:

TRUNCATE TABLE SUP1



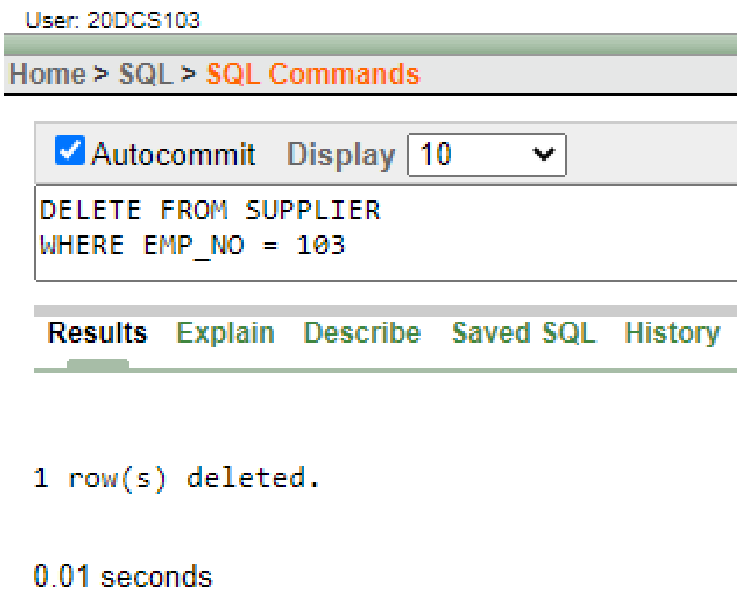
```
TRUNCATE TABLE SUP1
```

```
Table truncated.

0.16 seconds
```

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

Answer:

DELETE FROM SUPPLIER
WHERE SUP_NO = 103



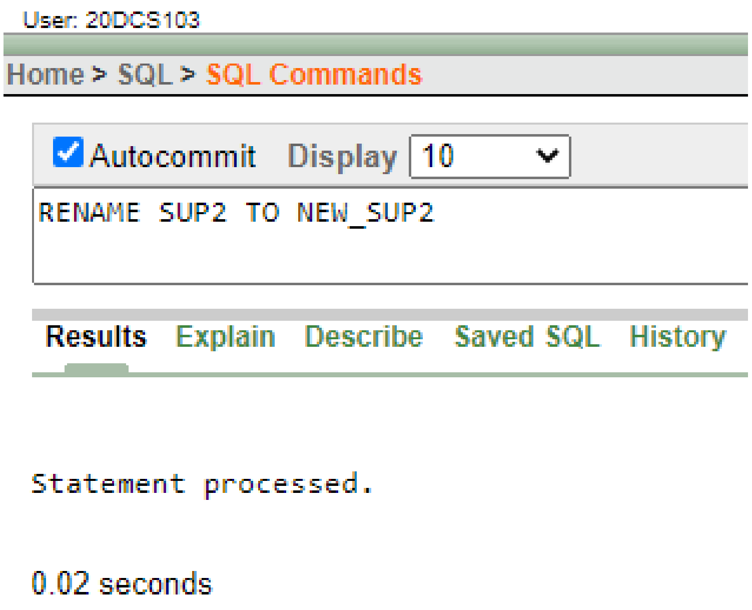
```
DELETE FROM SUPPLIER
WHERE EMP_NO = 103
```

```
1 row(s) deleted.

0.01 seconds
```

12) Rename the table sup2.

<u>Answer:</u>

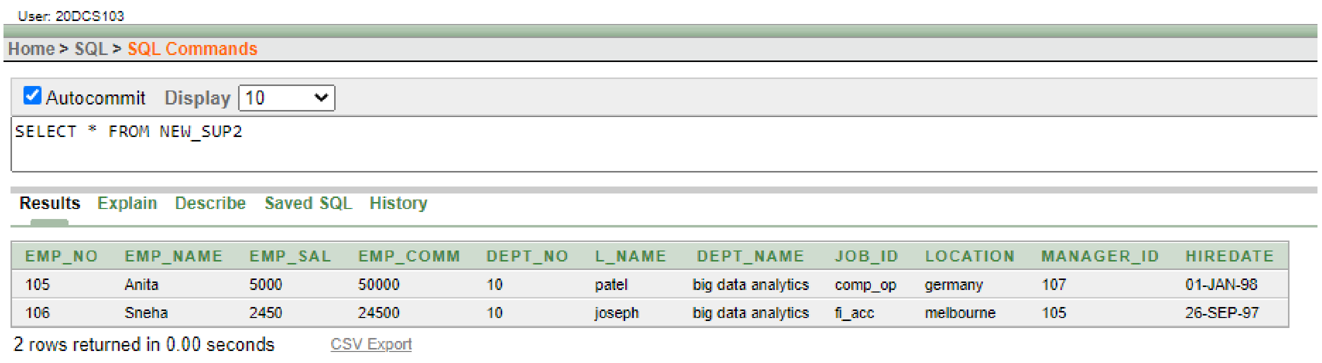
RENAME SUP2 TO NEW_SUP2



```
RENAME SUP2 TO NEW_SUP2
```

```
Statement processed.

0.02 seconds
```

SELECT * FROM NEW_SUP2



```
SELECT * FROM NEW_SUP2
```

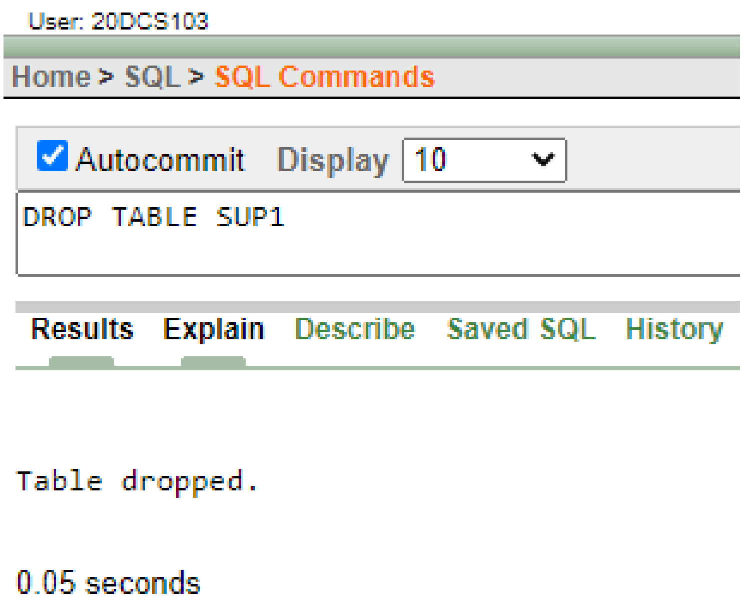
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97

2 rows returned in 0.00 seconds

13) Destroy table sup1 with all the data.

Answer:

DROP TABLE SUP1


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

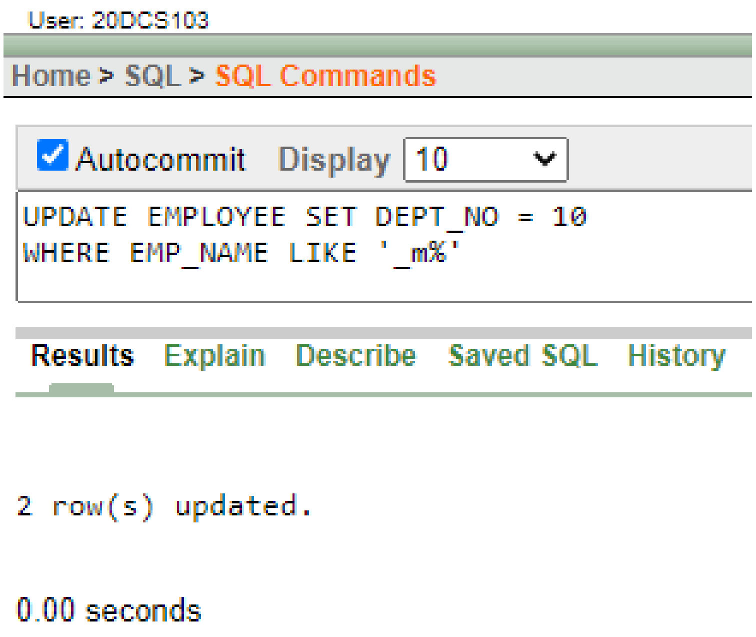
```
DROP TABLE SUP1
```

```
Table dropped.

0.05 seconds
```

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

Answer:

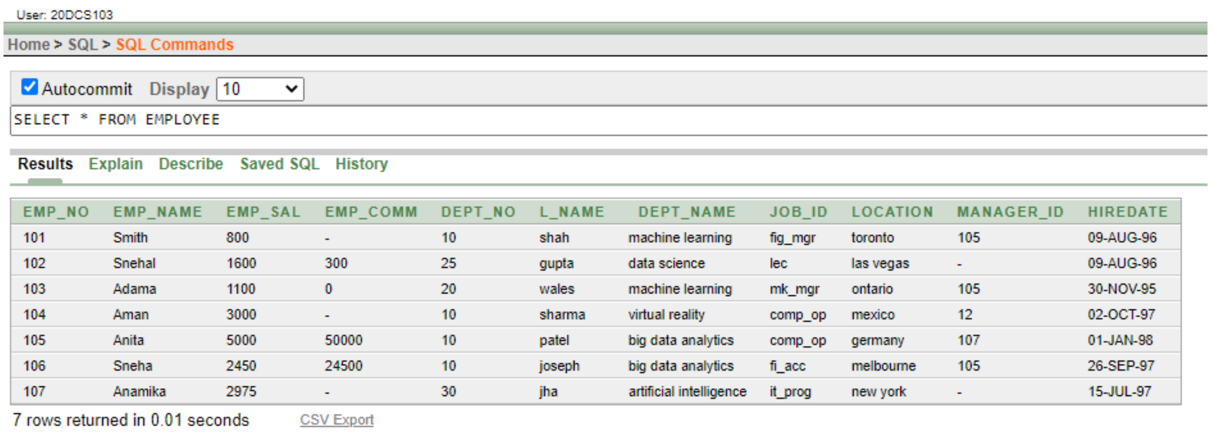
UPDATE EMPLOYEE SET DEPT_NO = 10
WHERE EMP_NAME LIKE '_m%'


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
UPDATE EMPLOYEE SET DEPT_NO = 10
WHERE EMP_NAME LIKE '_m%'
```

```
2 row(s) updated.

0.00 seconds
```

SELECT * FROM EMPLOYEE


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
SELECT * FROM EMPLOYEE
```

Results Explain Describe Saved SQL History

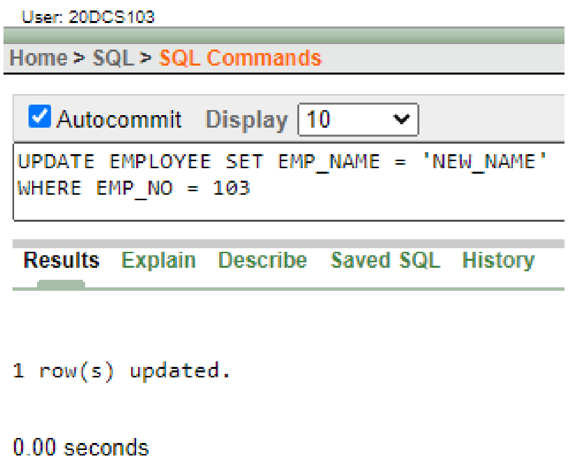
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
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

7 rows returned in 0.01 seconds CSV Export

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

<u>Answer:</u>

UPDATE EMPLOYEE SET EMP_NAME = 'NEW NAME'
WHERE EMP_NO = 103

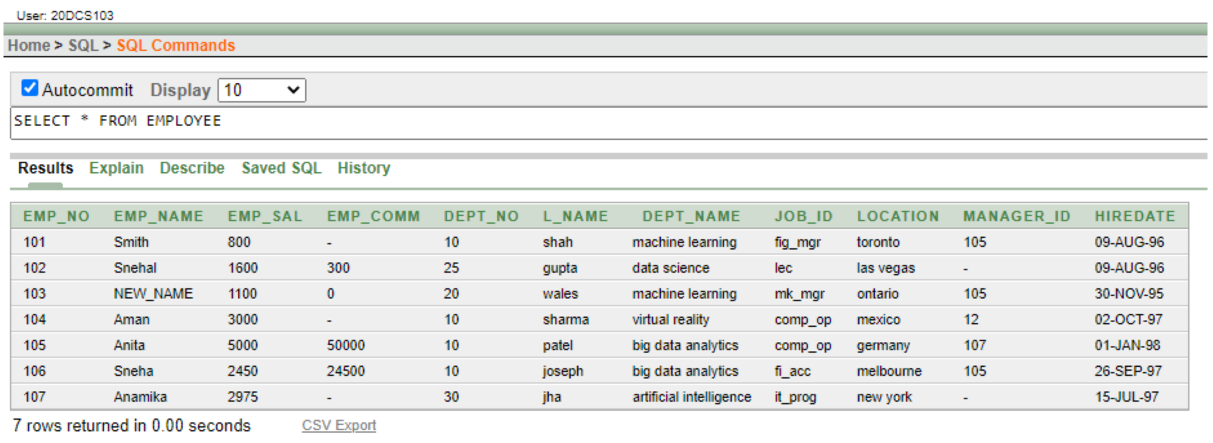

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
UPDATE EMPLOYEE SET EMP_NAME = 'NEW_NAME'
WHERE EMP_NO = 103
```

Results Explain Describe Saved SQL History

```
1 row(s) updated.

0.00 seconds
```

SELECT * FROM EMPLOYEE


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
SELECT * FROM EMPLOYEE
```

Results Explain Describe Saved SQL History

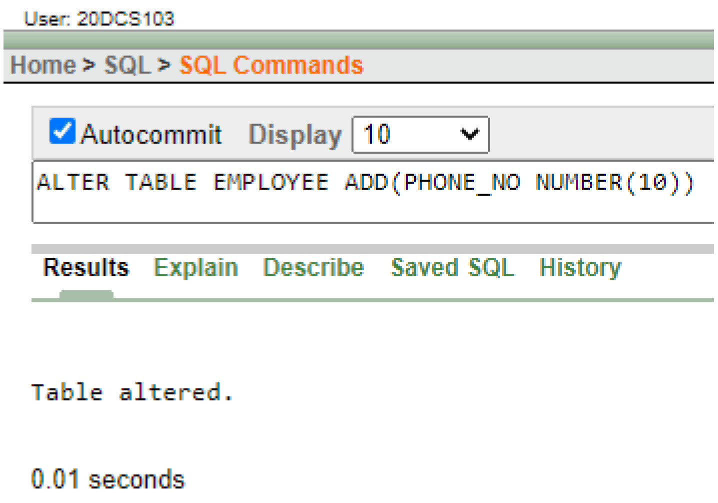
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96
103	NEW_NAME	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95
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

7 rows returned in 0.00 seconds CSV Export

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

Answer:

ALTER TABLE EMPLOYEE ADD(PHONE NUMBER NUMBER(10));


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

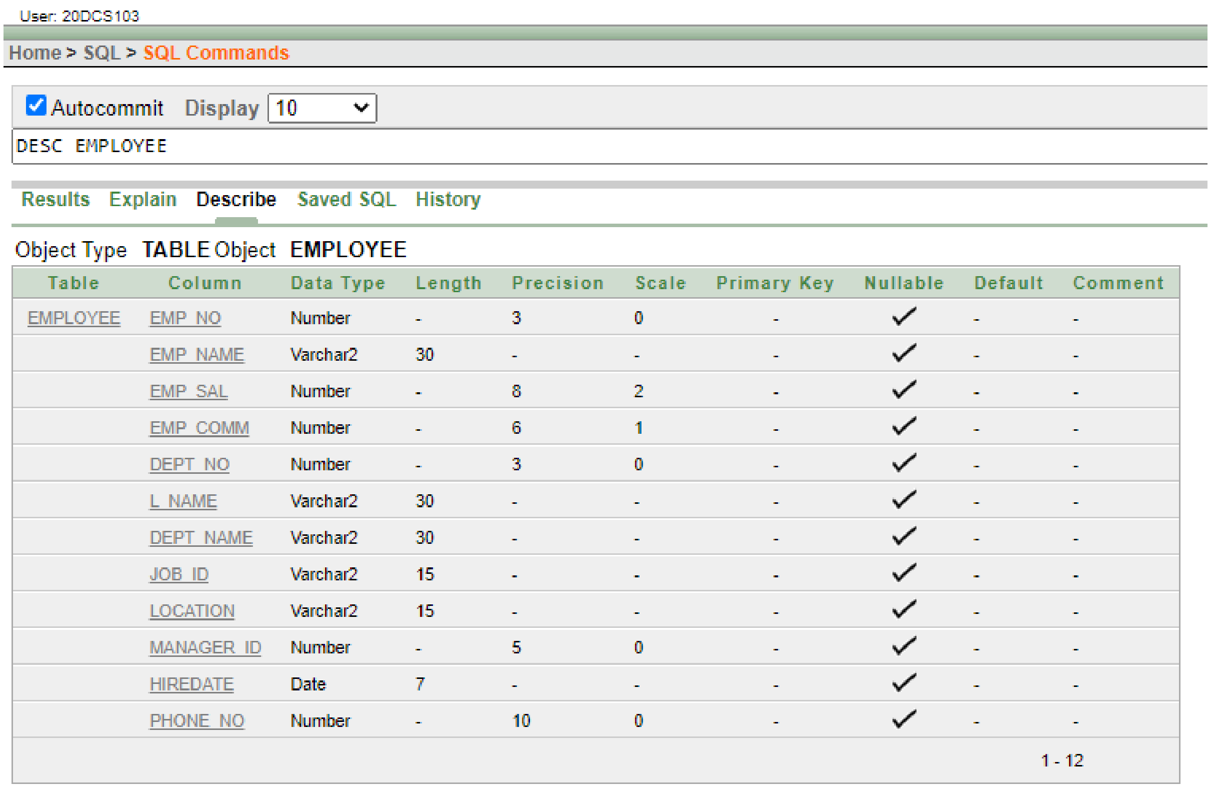
```
ALTER TABLE EMPLOYEE ADD(PHONE_NO NUMBER(10))
```

Results Explain Describe Saved SQL History

```
Table altered.
```

0.01 seconds

DESC EMPLOYEE


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
DESC EMPLOYEE
```

Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE

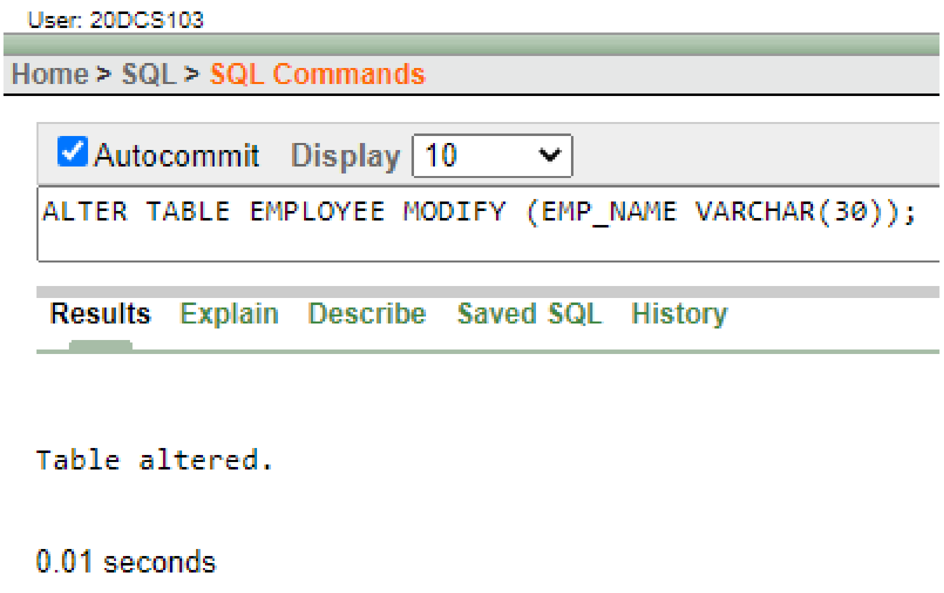
Table	Column	Data Type	Length	Precision	Scale	Primary Key	Nullable	Default	Comment
EMPLOYEE	EMP_NO	Number	-	3	0	-	✓	-	-
	EMP_NAME	Varchar2	30	-	-	-	✓	-	-
	EMP_SAL	Number	-	8	2	-	✓	-	-
	EMP_COMM	Number	-	6	1	-	✓	-	-
	DEPT_NO	Number	-	3	0	-	✓	-	-
	L_NAME	Varchar2	30	-	-	-	✓	-	-
	DEPT_NAME	Varchar2	30	-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

1 - 12

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

Answer:

ALTER TABLE EMPLOYEE MODIFY (EMP_NAME VARCHAR(30));


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

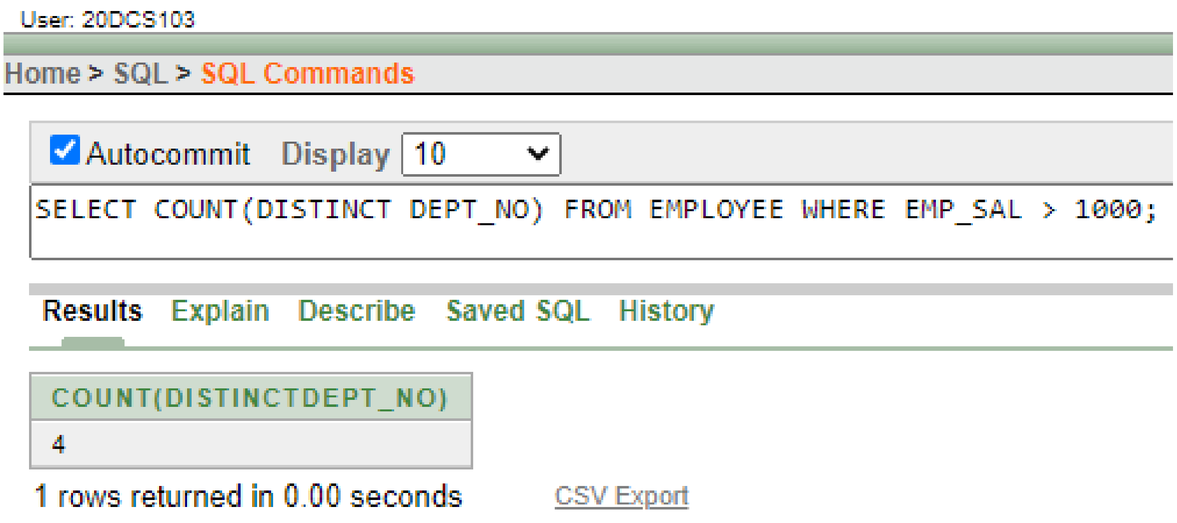
```
ALTER TABLE EMPLOYEE MODIFY (EMP_NAME VARCHAR(30));
```

```
Table altered.

0.01 seconds
```

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

Answer:

SELECT COUNT(DISTINCT DEPT_NO) FROM EMPLOYEE WHERE EMP_SAL > 1000;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT COUNT(DISTINCT DEPT_NO) FROM EMPLOYEE WHERE EMP_SAL > 1000;
```

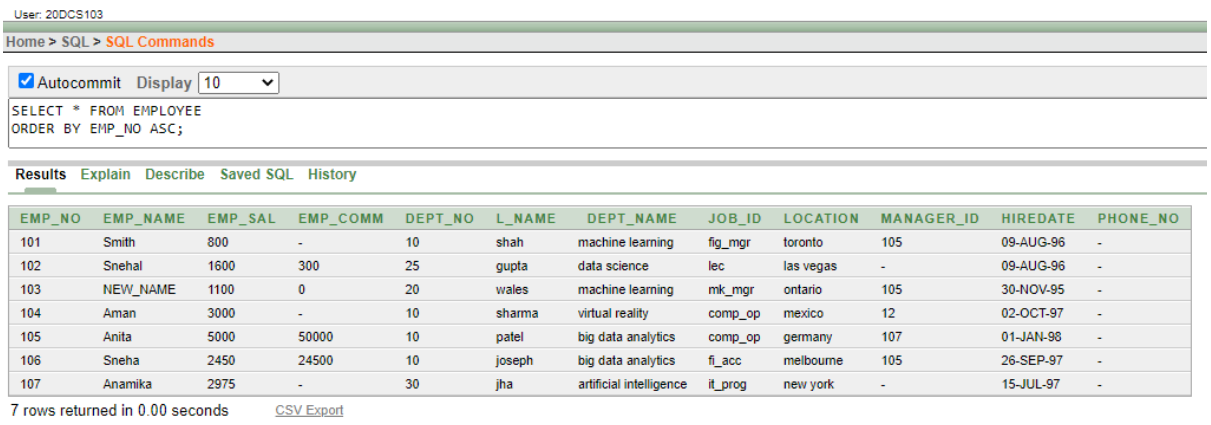
COUNT(DISTINCTDEPT_NO)
4

1 rows returned in 0.00 seconds

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

Answer:

SELECT * FROM EMPLOYEE
ORDER BY EMP_NO ASC;

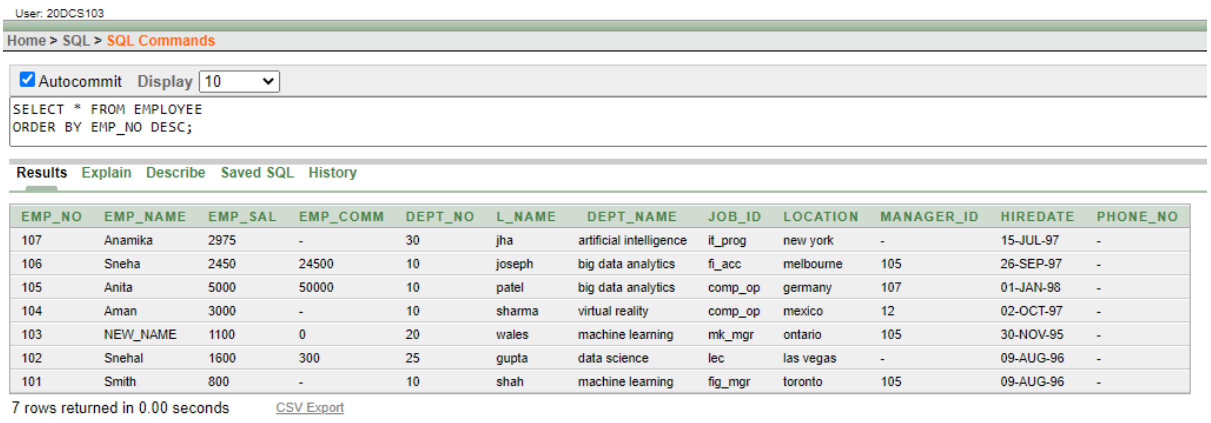

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
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_NO
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
103	NEW_NAME	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	-

7 rows returned in 0.00 seconds

SELECT * FROM EMPLOYEE
ORDER BY EMP_NO DESC;

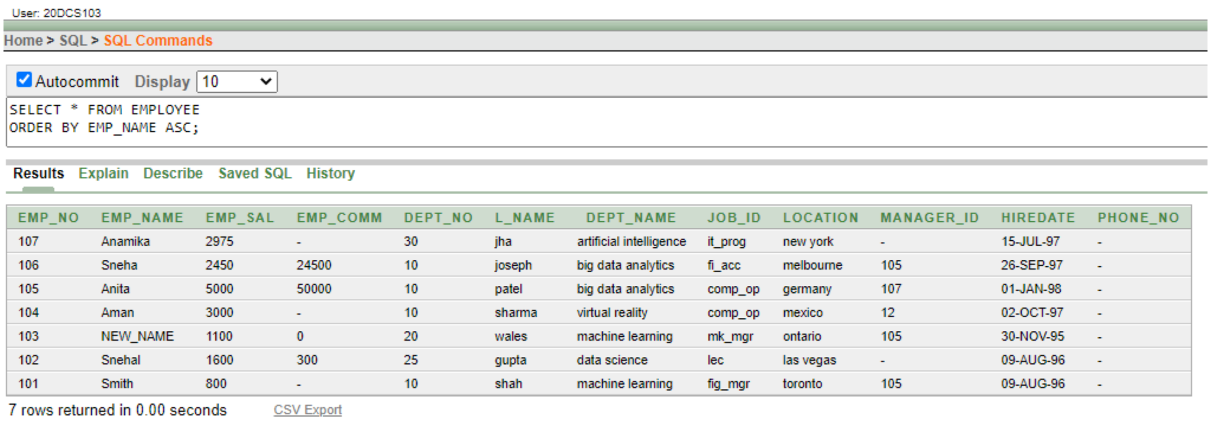

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
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_NO
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	NEW_NAME	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-

7 rows returned in 0.00 seconds

SELECT * FROM EMPLOYEE
ORDER BY EMP_NAME ASC;

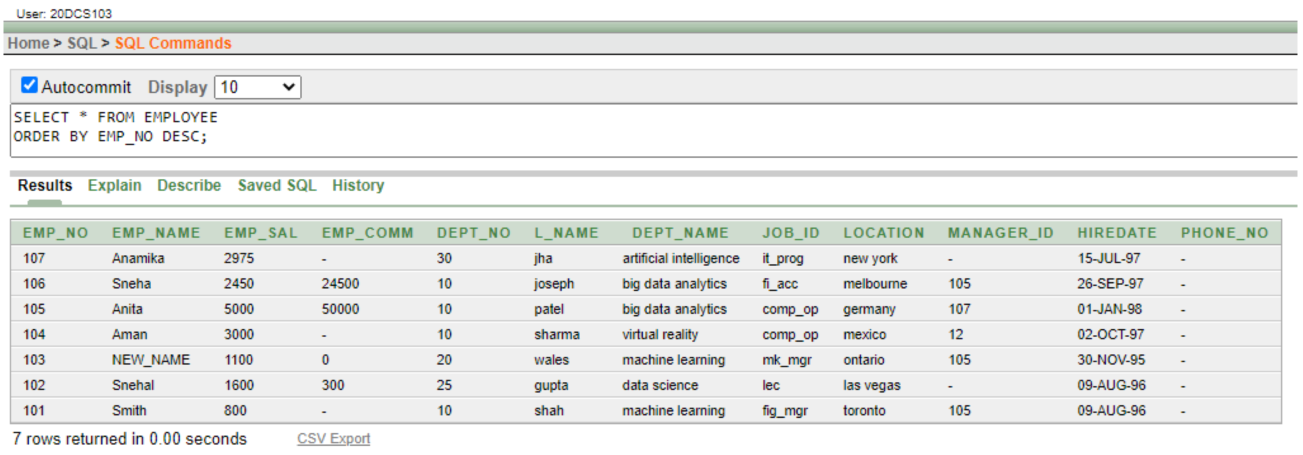

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
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_NO
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	NEW_NAME	1100	0	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-

7 rows returned in 0.00 seconds

SELECT * FROM EMPLOYEE
ORDER BY EMP_NAME ASC;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

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

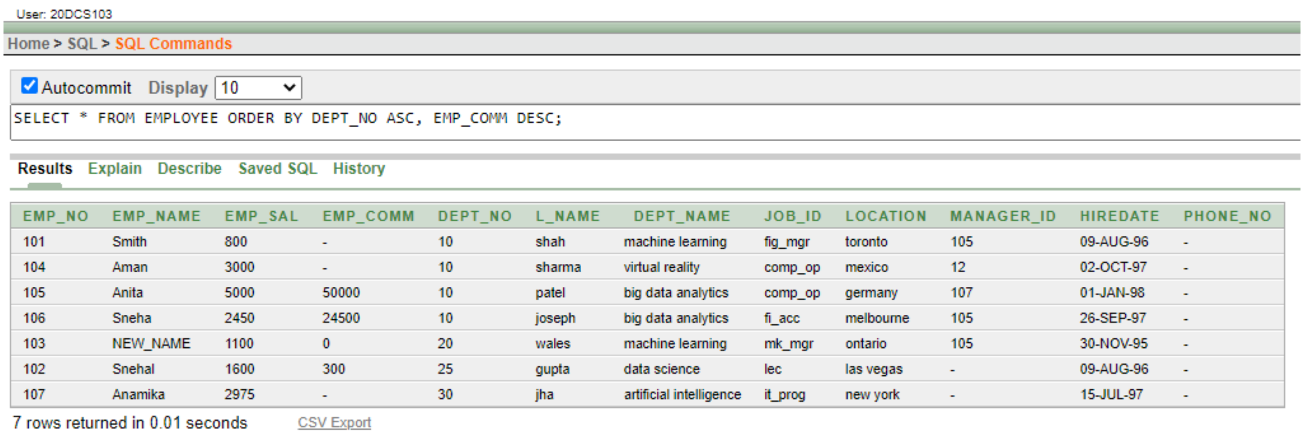
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NO
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
106	Sneha	2450	24500	10	joseph	big data analytics	fi_acc	melbourne	105	26-SEP-97	-
105	Anita	5000	50000	10	patel	big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

7 rows returned in 0.00 seconds

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

<u>Answer:</u>

SELECT * FROM EMPLOYEE ORDER BY DEPT_NO ASC, EMP_COMM DESC;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM EMPLOYEE ORDER BY DEPT_NO ASC, EMP_COMM DESC;
```

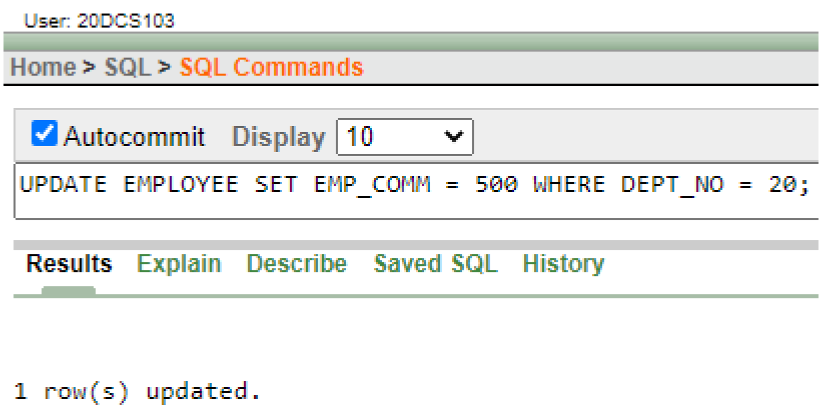
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NO
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
105	Anita	5000	50000	10	patel	big data analytics	comp_op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

7 rows returned in 0.01 seconds

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

Answer:

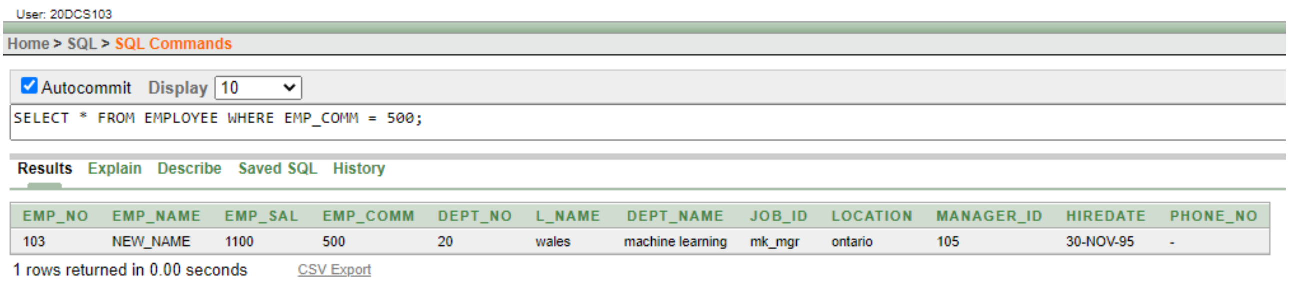
UPDATE EMPLOYEE SET EMP_COMM = 500 WHERE DEPT_NO = 20;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
UPDATE EMPLOYEE SET EMP_COMM = 500 WHERE DEPT_NO = 20;
```

```
1 row(s) updated.
```

SELECT * FROM EMPLOYEE WHERE EMP_COMM = 500;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM EMPLOYEE WHERE EMP_COMM = 500;
```

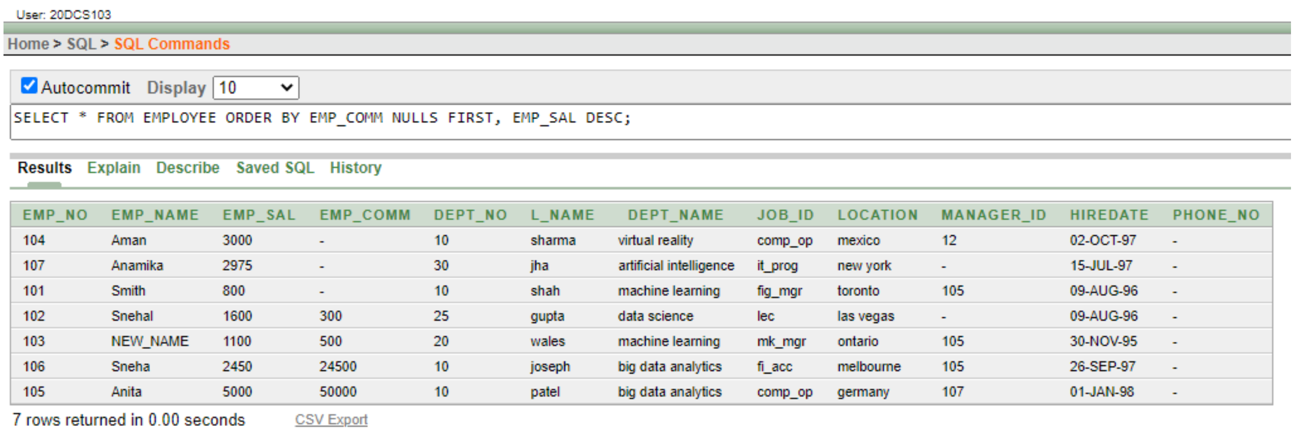
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NO
103	NEW_NAME	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-

1 rows returned in 0.00 seconds

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

Answer:

SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS FIRST, EMP_SAL DESC;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS FIRST, EMP_SAL DESC;
```

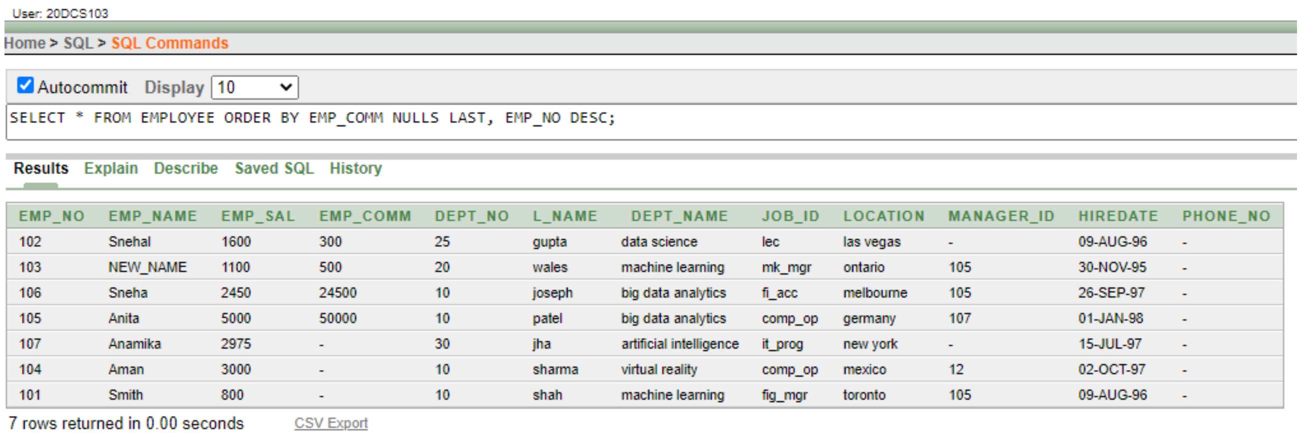
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NO
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-
107	Anamika	2975	-	30	jha	artificial intelligence	it_prog	new york	-	15-JUL-97	-
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
103	NEW_NAME	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
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	-

7 rows returned in 0.00 seconds

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

<u>Answer:</u>

SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS LAST, EMP_NO



User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT * FROM EMPLOYEE ORDER BY EMP_COMM NULLS LAST, EMP_NO DESC;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE_NO
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
103	NEW_NAME	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
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	-
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	-
101	Smith	800	-	10	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-

7 rows returned in 0.00 seconds CSV Export

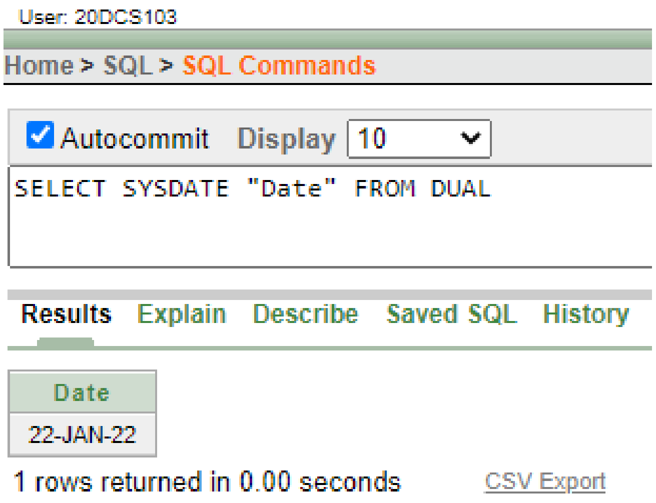
Aim 6: To study Single-row functions.

Answer:

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

Answer:

SELECT SYSDATE "Date" FROM DUAL



```
SELECT SYSDATE "Date" FROM DUAL
```

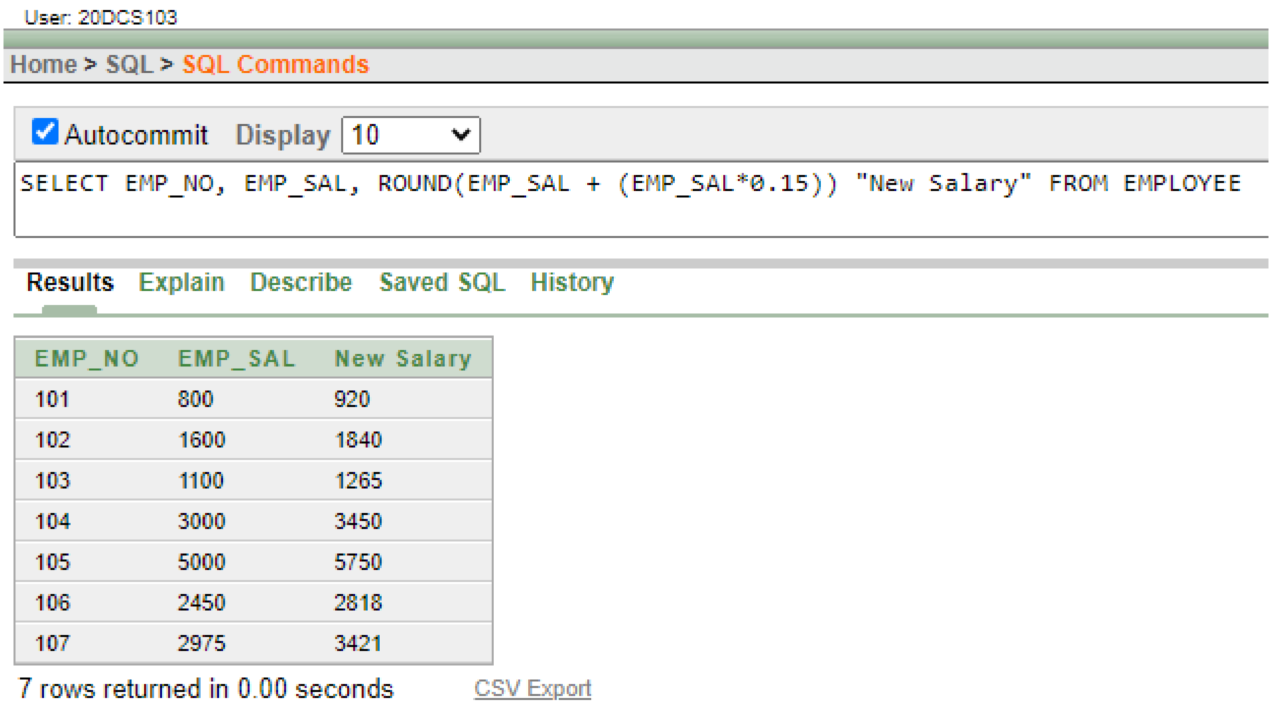
Date
22-JAN-22

1 rows returned in 0.00 seconds

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

Answer:

SELECT EMP_NO, EMP_SAL, ROUND(EMP_SAL + (EMP_SAL*0.15)) "New Salary" FROM EMPLOYEE



```
SELECT EMP_NO, EMP_SAL, ROUND(EMP_SAL + (EMP_SAL*0.15)) "New Salary" FROM EMPLOYEE
```

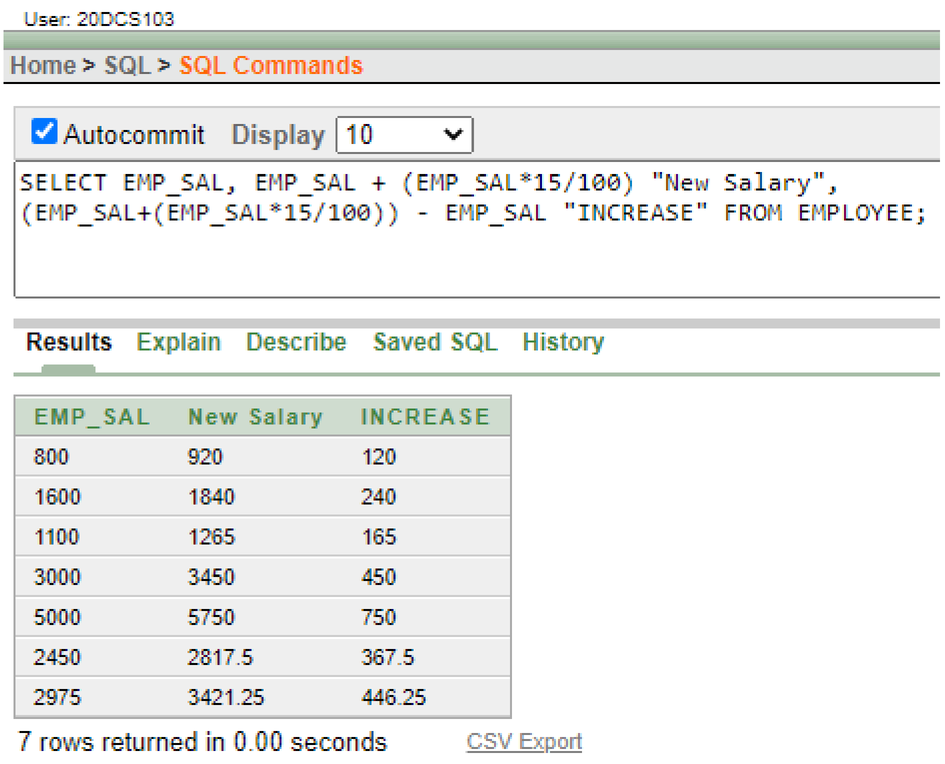
EMP_NO	EMP_SAL	New Salary
101	800	920
102	1600	1840
103	1100	1265
104	3000	3450
105	5000	5750
106	2450	2818
107	2975	3421

7 rows returned in 0.00 seconds

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

Answer:

SELECT EMP_NO, EMP_NAME, EMP_SAL,EMP_SAL+(EMP_SAL*15/100) "New Salary",
(EMP_SAL+(EMP_SAL*15/100)) - EMP_SAL "INCREASE" FROM EMPLOYEE



```
SELECT EMP_SAL, EMP_SAL + (EMP_SAL*15/100) "New Salary",
(EMP_SAL+(EMP_SAL*15/100)) - EMP_SAL "INCREASE" FROM EMPLOYEE;
```

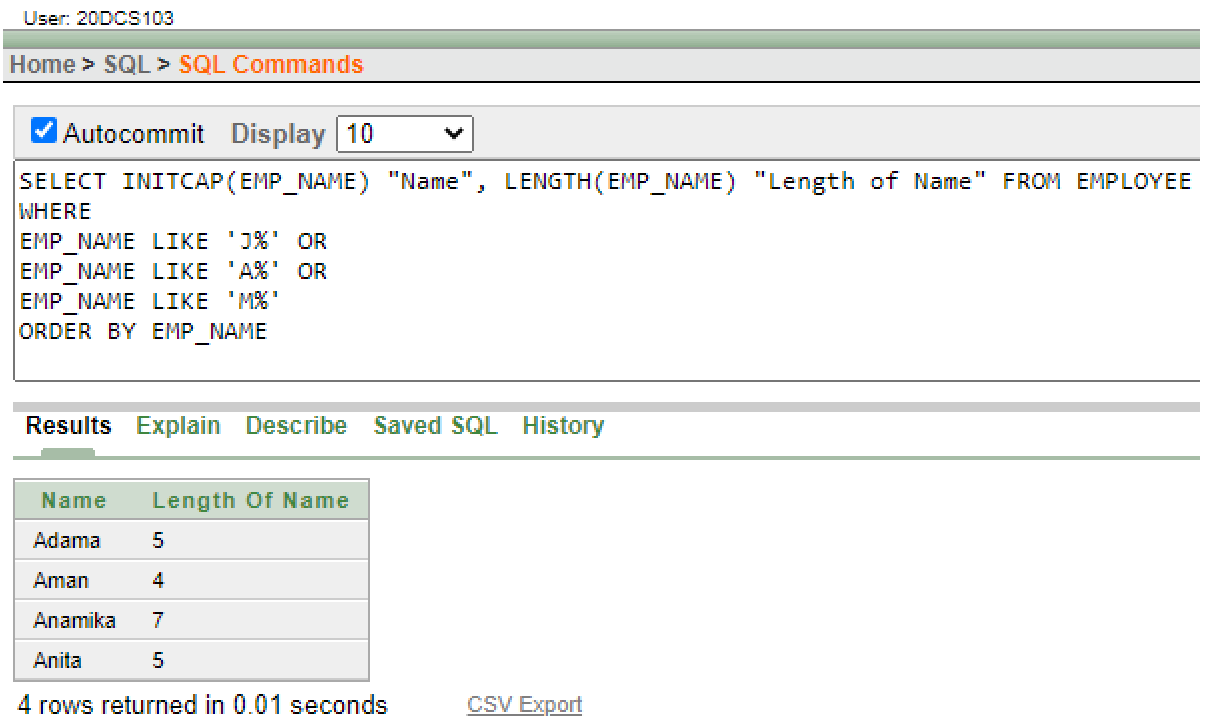
EMP_SAL	New Salary	INCREASE
800	920	120
1600	1840	240
1100	1265	165
3000	3450	450
5000	5750	750
2450	2817.5	367.5
2975	3421.25	446.25

7 rows returned in 0.00 seconds

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.

Answer:

SELECT INITCAP(EMP_NAME) "Name", LENGTH(EMP_NAME) "Length of Name"
FROM EMPLOYEE
WHERE
EMP_NAME LIKE 'J%' OR
EMP_NAME LIKE 'A%' OR
EMP_NAME LIKE 'M%'
ORDER BY EMP_NAME


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT INITCAP(EMP_NAME) "Name", LENGTH(EMP_NAME) "Length of Name" FROM EMPLOYEE
WHERE
EMP_NAME LIKE 'J%' OR
EMP_NAME LIKE 'A%' OR
EMP_NAME LIKE 'M%'
ORDER BY EMP_NAME
```

Results Explain Describe Saved SQL History

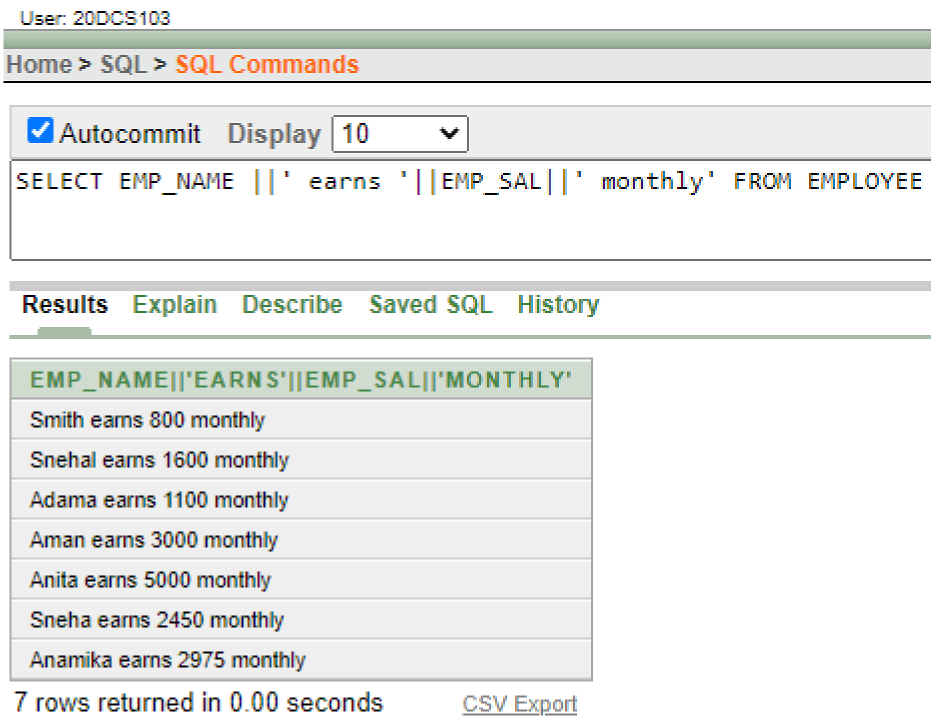
Name	Length Of Name
Adama	5
Aman	4
Anamika	7
Anita	5

4 rows returned in 0.01 seconds CSV Export

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

Answer:

SELECT EMP_NAME ||' earns '||EMP_SAL||' monthly' FROM EMPLOYEE


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT EMP_NAME ||' earns '||EMP_SAL||' monthly' FROM EMPLOYEE
```

Results Explain Describe Saved SQL History

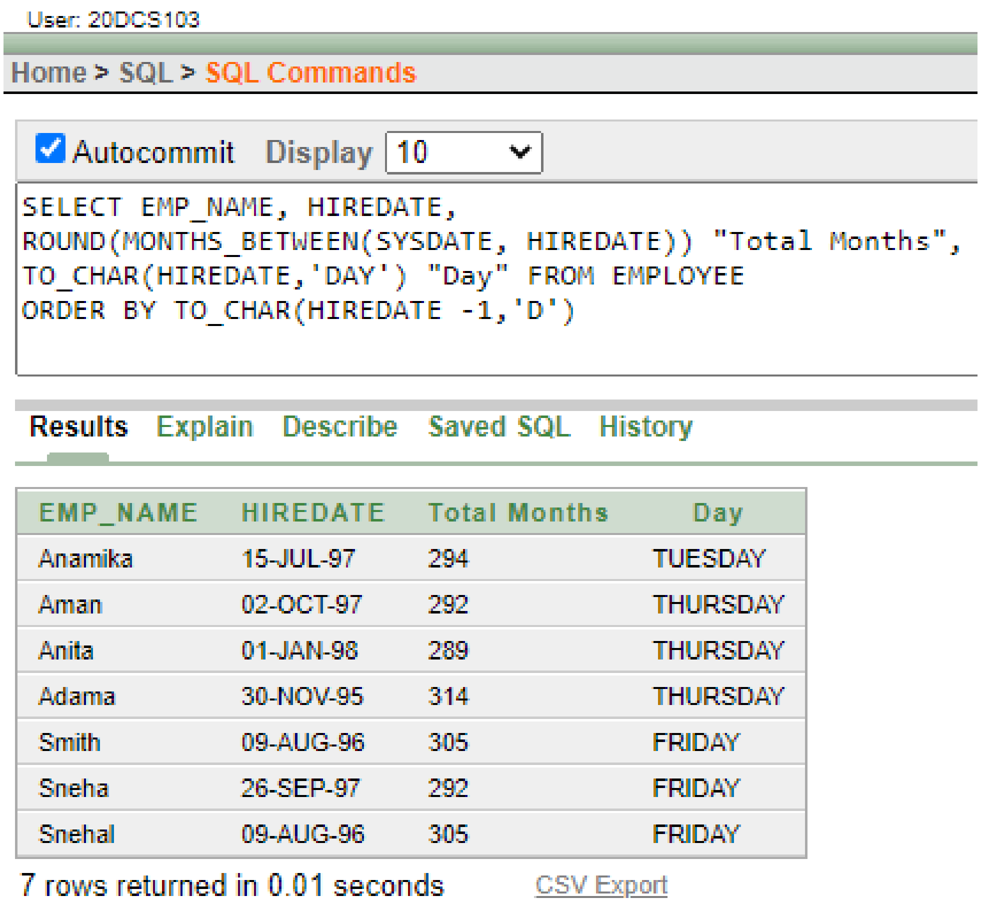
EMP_NAME||'EARNS'||EMP_SAL||'MONTHLY'
Smith earns 800 monthly
Snehal earns 1600 monthly
Adama earns 1100 monthly
Aman earns 3000 monthly
Anita earns 5000 monthly
Sneha earns 2450 monthly
Anamika earns 2975 monthly

7 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT EMP_NAME, HIREDATE,
ROUND(MONTHS_BETWEEN(SYSDATE, HIREDATE)) "Total Months",
TO_CHAR(HIREDATE,'DAY') "Day" FROM EMPLOYEE
ORDER BY TO_CHAR(HIREDATE -1,'D')


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT EMP_NAME, HIREDATE,
ROUND(MONTHS_BETWEEN(SYSDATE, HIREDATE)) "Total Months",
TO_CHAR(HIREDATE,'DAY') "Day" FROM EMPLOYEE
ORDER BY TO_CHAR(HIREDATE -1,'D')
```

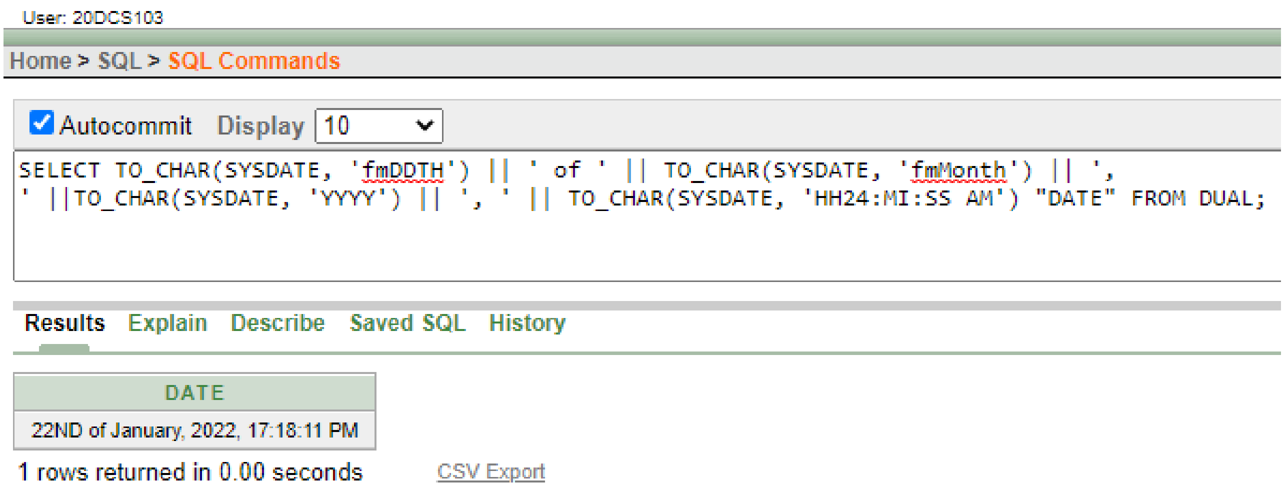
EMP_NAME	HIREDATE	Total Months	Day
Anamika	15-JUL-97	294	TUESDAY
Aman	02-OCT-97	292	THURSDAY
Anita	01-JAN-98	289	THURSDAY
Adama	30-NOV-95	314	THURSDAY
Smith	09-AUG-96	305	FRIDAY
Sneha	26-SEP-97	292	FRIDAY
Snehal	09-AUG-96	305	FRIDAY

7 rows returned in 0.01 seconds CSV Export

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

Answer:

SELECT TO_CHAR(SYSDATE, 'fmDDTH') || ' of ' || TO_CHAR(SYSDATE, 'fmMonth') || ', ' ||TO_CHAR(SYSDATE, 'YYYY') || ', ' || TO_CHAR(SYSDATE, 'HH24:MI:SS AM') "DATE" FROM DUAL;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT TO_CHAR(SYSDATE, 'fmDDTH') || ' of ' || TO_CHAR(SYSDATE, 'fmMonth') || ',
' ||TO_CHAR(SYSDATE, 'YYYY') || ', ' || TO_CHAR(SYSDATE, 'HH24:MI:SS AM') "DATE" FROM DUAL;
```

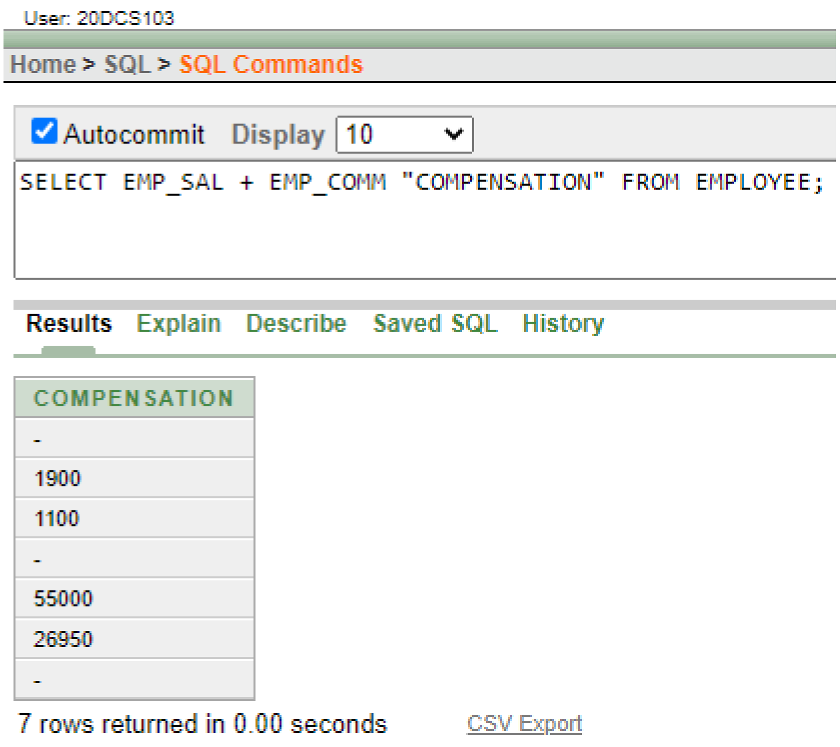
DATE
22ND of January, 2022, 17:18:11 PM

1 rows returned in 0.00 seconds

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

Answer:

SELECT EMP_SAL+EMP_COMM "COMPENSATION" FROM EMPLOYEE;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT EMP_SAL + EMP_COMM "COMPENSATION" FROM EMPLOYEE;
```

COMPENSATION
-
1900
1100
-
55000
26950
-

7 rows returned in 0.00 seconds

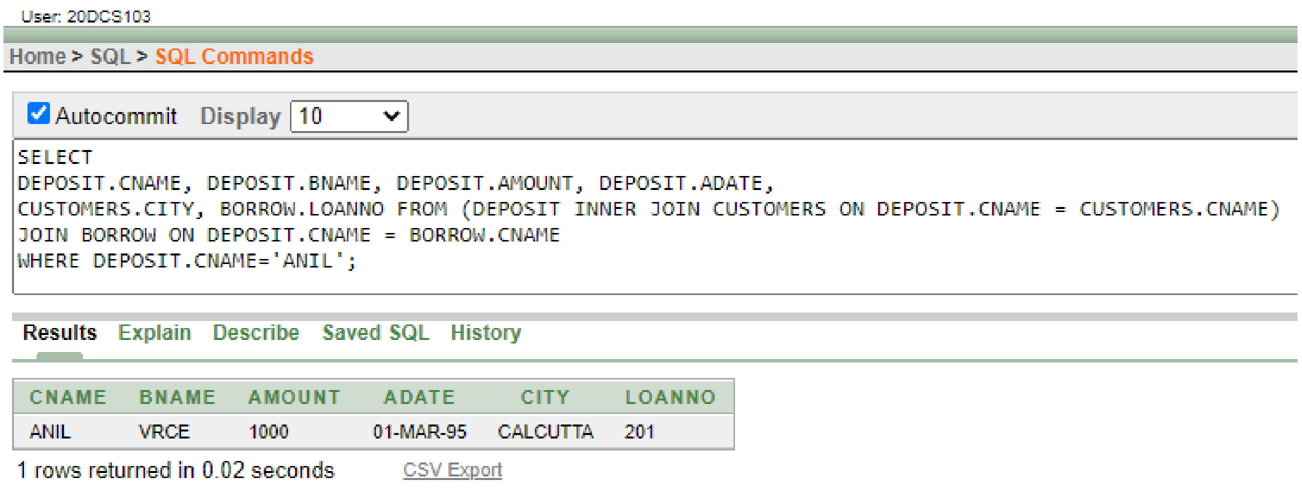
Aim 7: Displaying data from Multiple Tables (join)

Answer:

1) Give details of customers ANIL.

Answer:

SELECT
DEPOSIT.CNAME, DEPOSIT.BNAME, DEPOSIT.AMOUNT,
DEPOSIT.ADATE,
CUSTOMERS.CITY, BORROW.LOANNO FROM (DEPOSIT INNER JOIN
CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME)
JOIN BORROW ON DEPOSIT.CNAME = BORROW.CNAME
WHERE DEPOSIT.CNAME='ANIL';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT
DEPOSIT.CNAME, DEPOSIT.BNAME, DEPOSIT.AMOUNT, DEPOSIT.ADATE,
CUSTOMERS.CITY, BORROW.LOANNO FROM (DEPOSIT INNER JOIN CUSTOMERS ON DEPOSIT.CNAME = CUSTOMERS.CNAME)
JOIN BORROW ON DEPOSIT.CNAME = BORROW.CNAME
WHERE DEPOSIT.CNAME='ANIL';
```

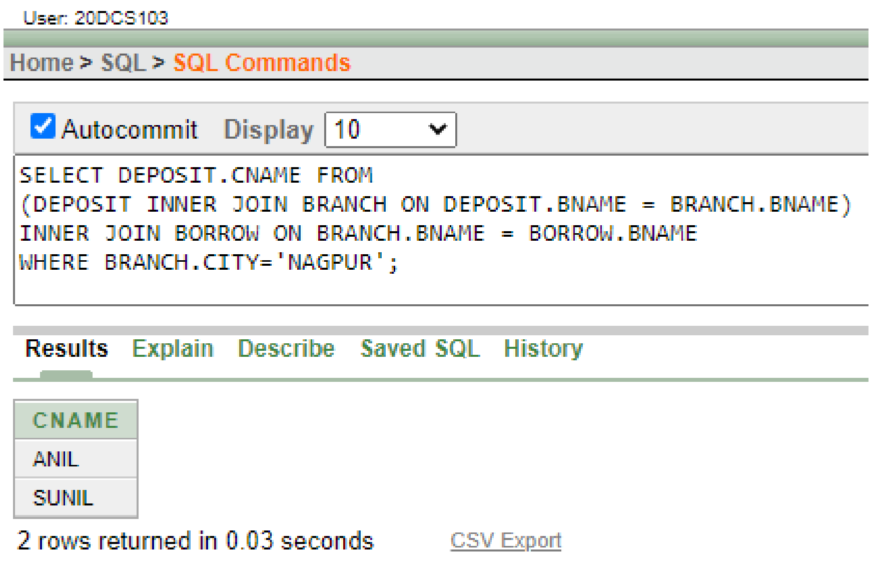
CNAME	BNAME	AMOUNT	ADATE	CITY	LOANNO
ANIL	VRCE	1000	01-MAR-95	CALCUTTA	201

1 rows returned in 0.02 seconds

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

Answer:

SELECT DEPOSIT.CNAME FROM
(DEPOSIT INNER JOIN BRANCH ON DEPOSIT.BNAME =
BRANCH.BNAME)
INNER JOIN BORROW ON BRANCH.BNAME = BORROW.BNAME
WHERE BRANCH.CITY = 'NAGPUR';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

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

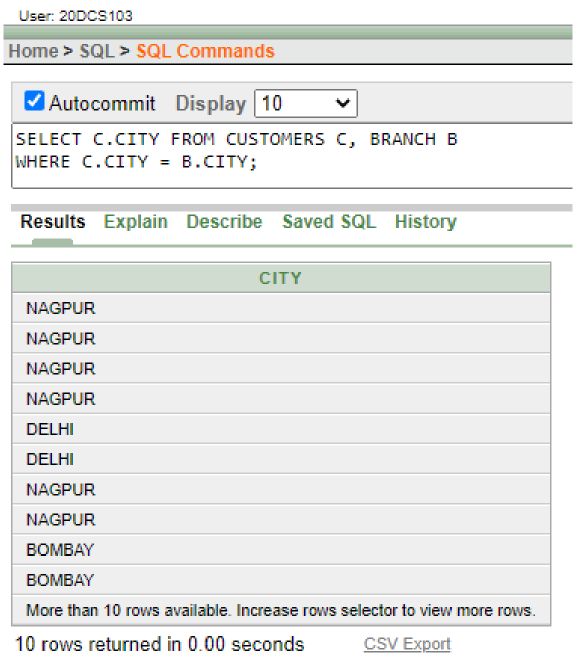
CNAME
ANIL
SUNIL

2 rows returned in 0.03 seconds

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

Answer:

SELECT C.CITY FROM CUSTOMERS C, BRANCH B
WHERE C.CITY = B.CITY;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT C.CITY FROM CUSTOMERS C, BRANCH B
WHERE C.CITY = B.CITY;
```

CITY
NAGPUR
NAGPUR
NAGPUR
NAGPUR
DELHI
DELHI
NAGPUR
NAGPUR
BOMBAY
BOMBAY

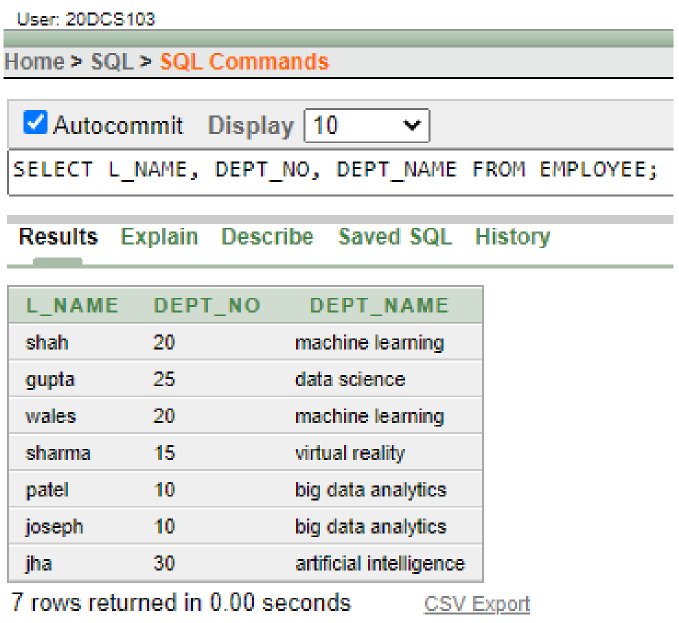
More than 10 rows available. Increase rows selector to view more rows.

10 rows returned in 0.00 seconds

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

Answer:

SELECT L_NAME, DEPT_NO, DEPT_NAME FROM EMPLOYEE;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT L_NAME, DEPT_NO, DEPT_NAME FROM EMPLOYEE;
```

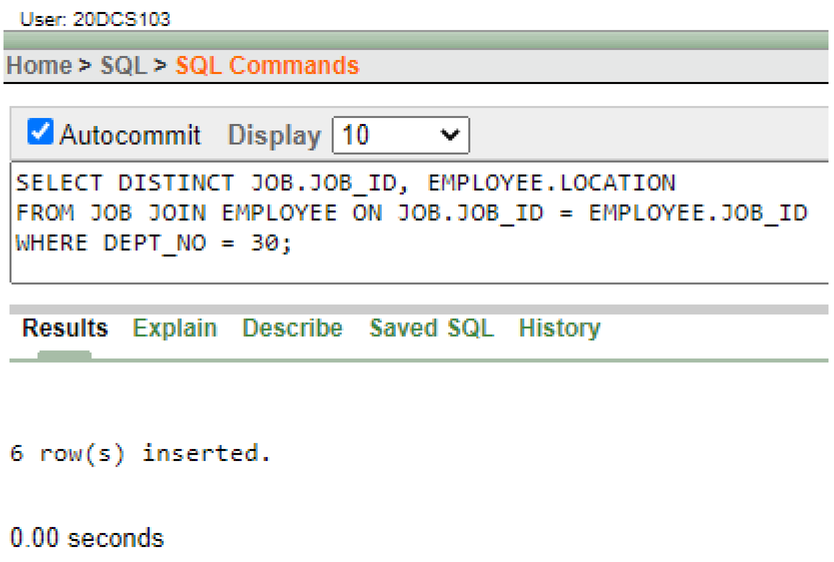
L_NAME	DEPT_NO	DEPT_NAME
shah	20	machine learning
gupta	25	data science
wales	20	machine learning
sharma	15	virtual reality
patel	10	big data analytics
joseph	10	big data analytics
jha	30	artificial intelligence

7 rows returned in 0.00 seconds

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

Answer:

SELECT DISTINCT JOB.JOB_ID, EMPLOYEE.LOCATION
FROM JOB JOIN EMPLOYEE ON JOB.JOB_ID = EMPLOYEE.JOB_ID
WHERE DEPT_NO = 30;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

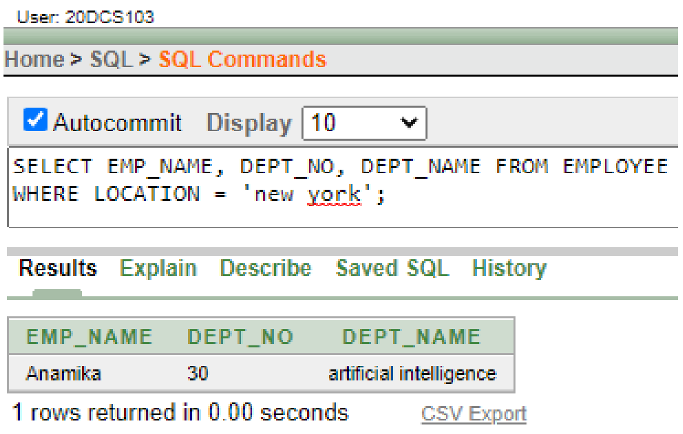
```
SELECT DISTINCT JOB.JOB_ID, EMPLOYEE.LOCATION
FROM JOB JOIN EMPLOYEE ON JOB.JOB_ID = EMPLOYEE.JOB_ID
WHERE DEPT_NO = 30;
```

```
6 row(s) inserted.

0.00 seconds
```

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

Answer:

SELECT EMP_NAME, DEPT_NO, DEPT_NAME FROM EMPLOYEE
WHERE LOCATION = 'new york';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
SELECT EMP_NAME, DEPT_NO, DEPT_NAME FROM EMPLOYEE
WHERE LOCATION = 'new york';
```

Results Explain Describe Saved SQL History

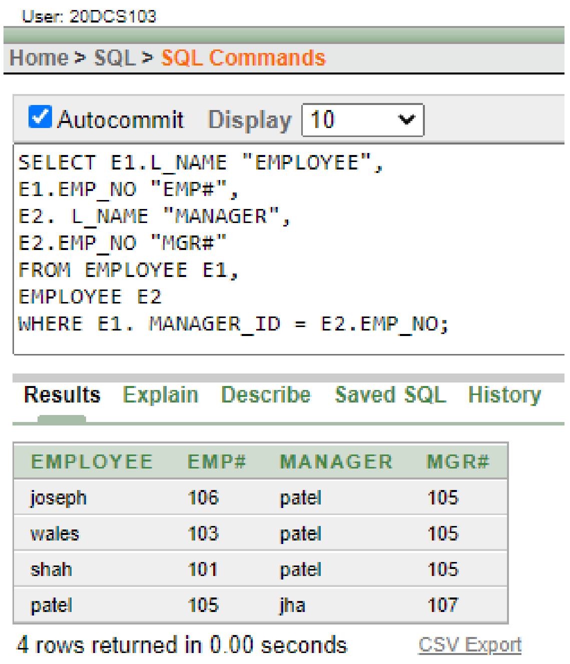
EMP_NAME	DEPT_NO	DEPT_NAME
Anamika	30	artificial intelligence

1 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT E1.L_NAME "EMPLOYEE",
E1.EMP_NO "EMP#", E2. L_NAME "MANAGER", E2.EMP_NO "MGR#"
FROM EMPLOYEE E1,
EMPLOYEE E2
WHERE E1. MANAGER_ID = E2.EMP_NO;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
SELECT E1.L_NAME "EMPLOYEE",
E1.EMP_NO "EMP#",
E2. L_NAME "MANAGER",
E2.EMP_NO "MGR#"
FROM EMPLOYEE E1,
EMPLOYEE E2
WHERE E1. MANAGER_ID = E2.EMP_NO;
```

Results Explain Describe Saved SQL History

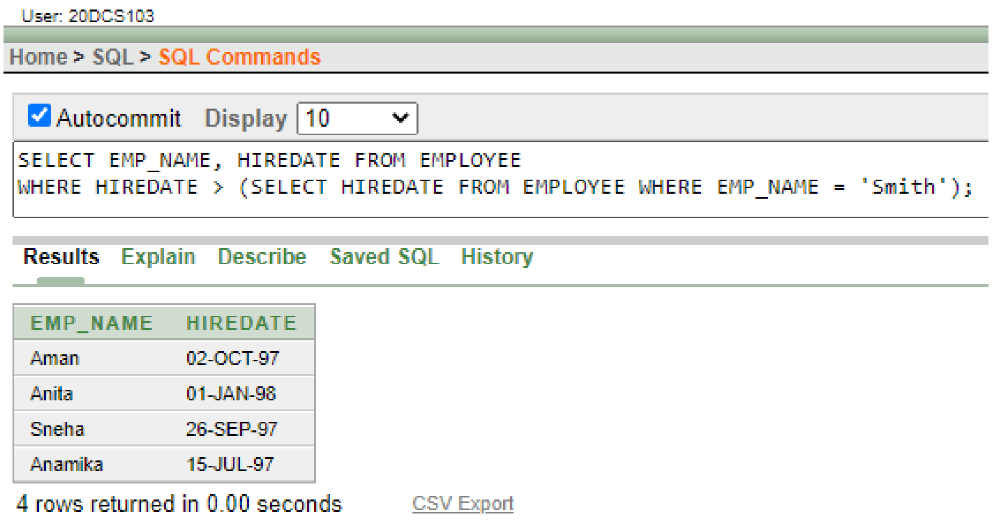
EMPLOYEE	EMP#	MANAGER	MGR#
joseph	106	patel	105
wales	103	patel	105
shah	101	patel	105
patel	105	jha	107

4 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT EMP_NAME, HIREDATE FROM EMPLOYEE
WHERE HIREDATE > (SELECT HIREDATE FROM EMPLOYEE WHERE EMP_NAME='Smith');


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10

```
SELECT EMP_NAME, HIREDATE FROM EMPLOYEE
WHERE HIREDATE > (SELECT HIREDATE FROM EMPLOYEE WHERE EMP_NAME = 'Smith');
```

Results Explain Describe Saved SQL History

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

4 rows returned in 0.00 seconds

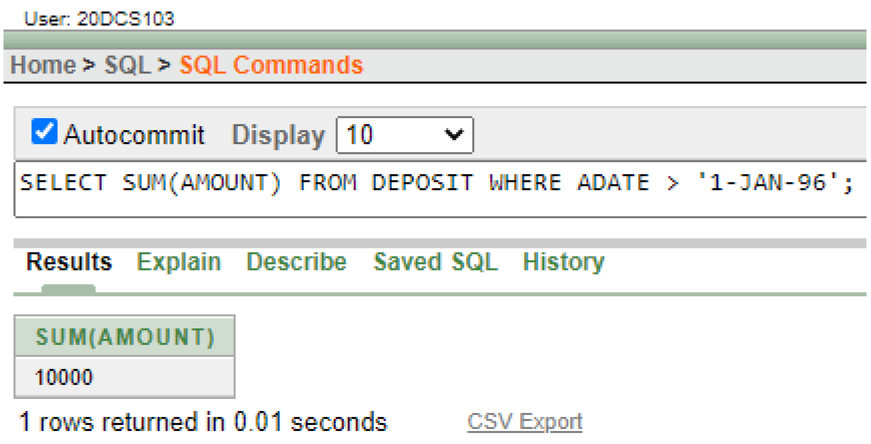
Aim 8: To apply the concept of Aggregating Data using Group functions.

Answer:

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

Answer:

SELECT SUM(AMOUNT) FROM DEPOSIT WHERE ADATE > '1-JAN-96';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT SUM(AMOUNT) FROM DEPOSIT WHERE ADATE > '1-JAN-96';
```

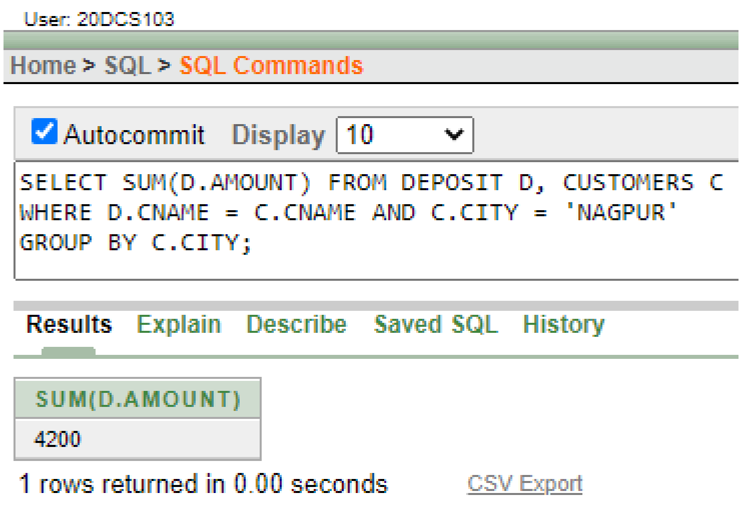
SUM(AMOUNT)
10000

1 rows returned in 0.01 seconds

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

Answer:

SELECT SUM(D.AMOUNT) FROM DEPOSIT D, CUSTOMERS C
WHERE D.CNAME = C.CNAME AND C.CITY = 'NAGPUR'
GROUP BY C.CITY;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT SUM(D.AMOUNT) FROM DEPOSIT D, CUSTOMERS C
WHERE D.CNAME = C.CNAME AND C.CITY = 'NAGPUR'
GROUP BY C.CITY;
```

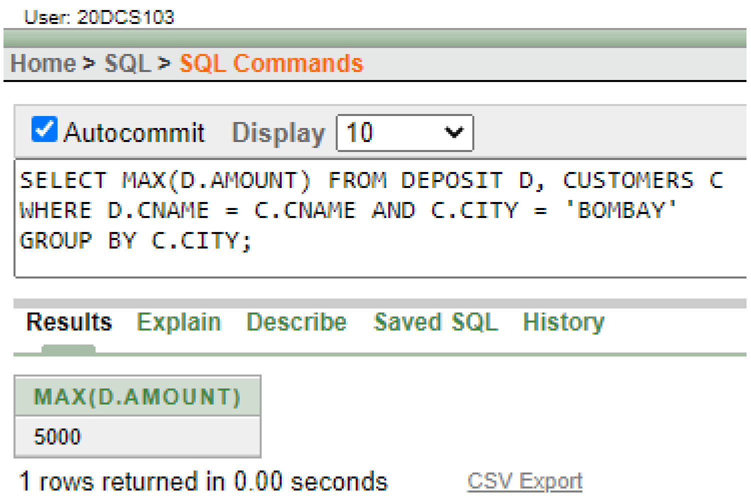
SUM(D.AMOUNT)
4200

1 rows returned in 0.00 seconds

3) List maximum deposit of customers living in Bombay.

Answer:

SELECT MAX(D.AMOUNT) FROM DEPOSIT D, CUSTOMERS C
WHERE D.CNAME = C.CNAME AND C.CITY = 'BOMBAY'
GROUP BY C.CITY;


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT MAX(D.AMOUNT) FROM DEPOSIT D, CUSTOMERS C
WHERE D.CNAME = C.CNAME AND C.CITY = 'BOMBAY'
GROUP BY C.CITY;
```

Results Explain Describe Saved SQL History

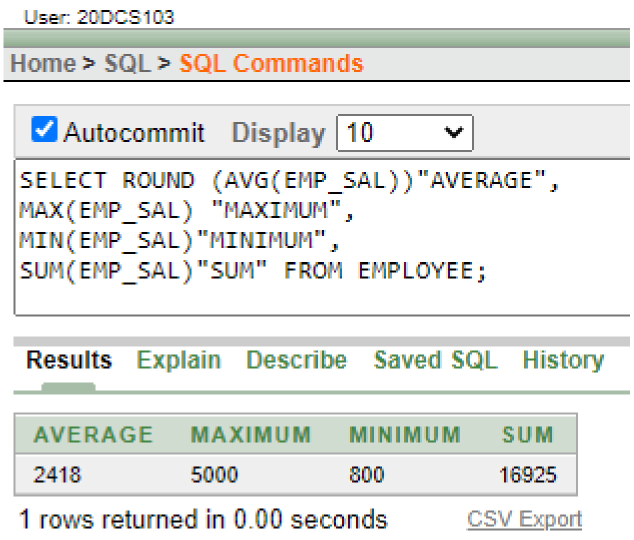
```
MAX(D.AMOUNT)
5000
```

1 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT ROUND (AVG(EMP_SAL))"AVERAGE",
MAX(EMP_SAL) "MAXIMUM",
MIN(EMP_SAL)"MINIMUM",
SUM(EMP_SAL)"SUM" FROM EMPLOYEE;


User: 20DCS103

Home > SQL > SQL Commands

Autocommit Display 10

```
SELECT ROUND (AVG(EMP_SAL))"AVERAGE",
MAX(EMP_SAL) "MAXIMUM",
MIN(EMP_SAL)"MINIMUM",
SUM(EMP_SAL)"SUM" FROM EMPLOYEE;
```

Results Explain Describe Saved SQL History

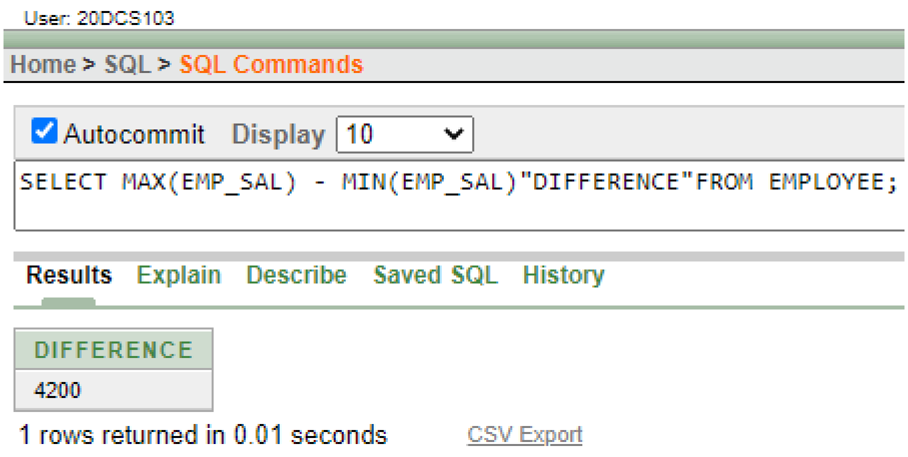
```
AVERAGE   MAXIMUM   MINIMUM   SUM
2418      5000      800       16925
```

1 rows returned in 0.00 seconds CSV Export

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

Answer:

SELECT MAX(EMP_SAL) - MIN(EMP_SAL)"DIFFERENCE"FROM EMPLOYEE;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT MAX(EMP_SAL) - MIN(EMP_SAL)"DIFFERENCE"FROM EMPLOYEE;
```

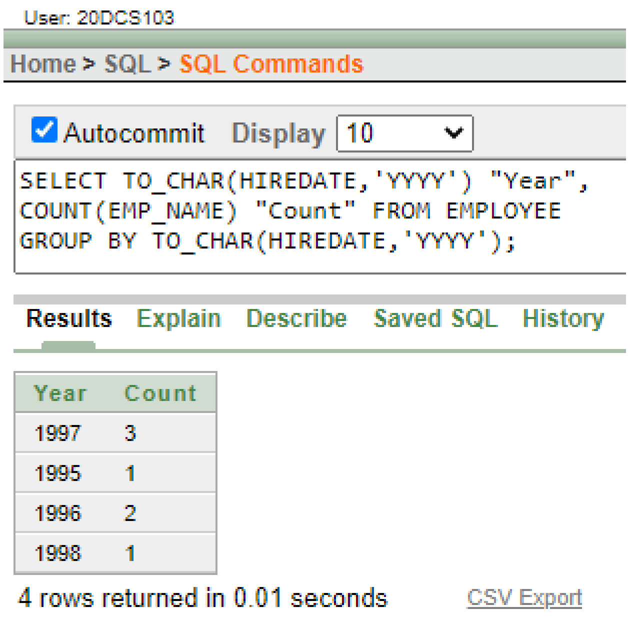
DIFFERENCE
4200

1 rows returned in 0.01 seconds

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.

Answer:

SELECT TO_CHAR(HIREDATE,'YYYY') "Year",
COUNT(EMP_NAME) "Count" FROM EMPLOYEE
GROUP BY TO_CHAR(HIREDATE,'YYYY');


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT TO_CHAR(HIREDATE,'YYYY') "Year",
COUNT(EMP_NAME) "Count" FROM EMPLOYEE
GROUP BY TO_CHAR(HIREDATE,'YYYY');
```

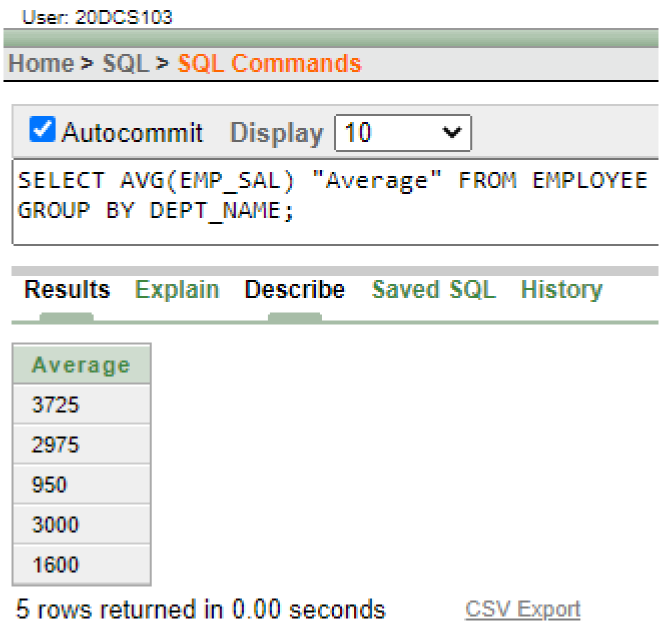
Year	Count
1997	3
1995	1
1996	2
1998	1

4 rows returned in 0.01 seconds

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

Answer:

SELECT AVG(EMP_SAL) "Average" FROM EMPLOYEE
GROUP BY DEPT_NAME;



```
SELECT AVG(EMP_SAL) "Average" FROM EMPLOYEE
GROUP BY DEPT_NAME;
```

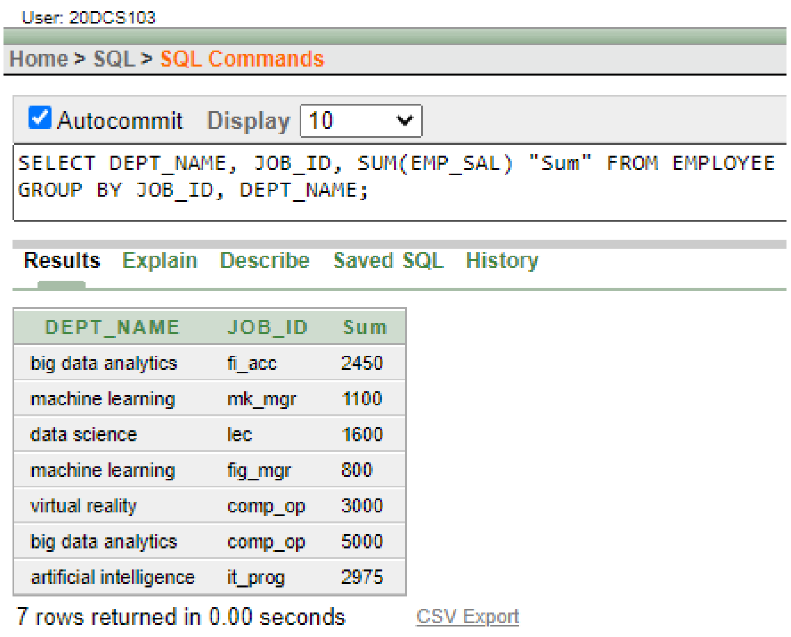
Average
3725
2975
950
3000
1600

5 rows returned in 0.00 seconds

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

Answer:

SELECT DEPT_NAME, JOB_ID, SUM(EMP_SAL) "Sum" FROM EMPLOYEE
GROUP BY JOB_ID, DEPT_NAME;



```
SELECT DEPT_NAME, JOB_ID, SUM(EMP_SAL) "Sum" FROM EMPLOYEE
GROUP BY JOB_ID, DEPT_NAME;
```

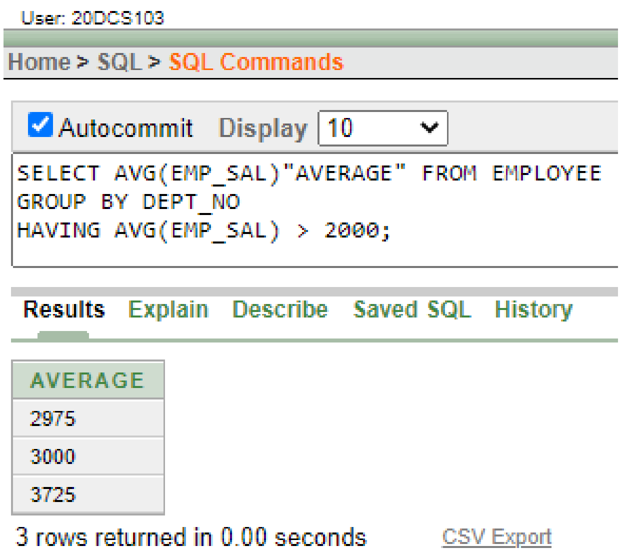
DEPT_NAME	JOB_ID	Sum
big data analytics	fi_acc	2450
machine learning	mk_mgr	1100
data science	lec	1600
machine learning	fig_mgr	800
virtual reality	comp_op	3000
big data analytics	comp_op	5000
artificial intelligence	it_prog	2975

7 rows returned in 0.00 seconds

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

Answer:

SELECT AVG(EMP_SAL)"AVERAGE" FROM EMPLOYEE
GROUP BY DEPT_NO
HAVING AVG(EMP_SAL) > 2000;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT AVG(EMP_SAL)"AVERAGE" FROM EMPLOYEE
GROUP BY DEPT_NO
HAVING AVG(EMP_SAL) > 2000;
```

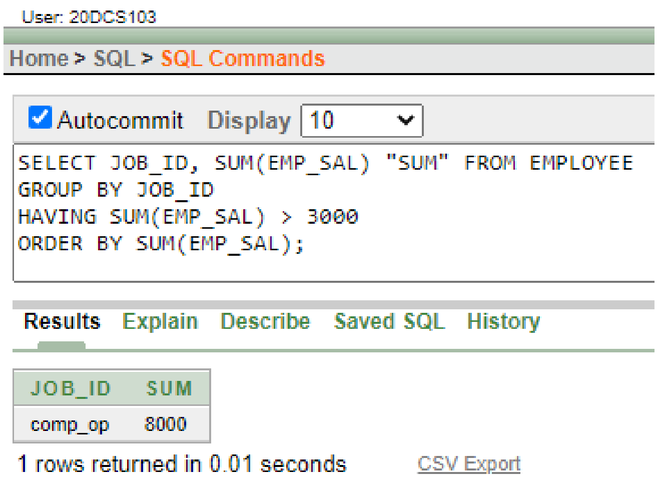
AVERAGE
2975
3000
3725

3 rows returned in 0.00 seconds

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

Answer:

SELECT JOB_ID, SUM(EMP_SAL) "SUM" FROM EMPLOYEE
GROUP BY JOB_ID
HAVING SUM(EMP_SAL) > 3000
ORDER BY SUM(EMP_SAL);


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT JOB_ID, SUM(EMP_SAL) "SUM" FROM EMPLOYEE
GROUP BY JOB_ID
HAVING SUM(EMP_SAL) > 3000
ORDER BY SUM(EMP_SAL);
```

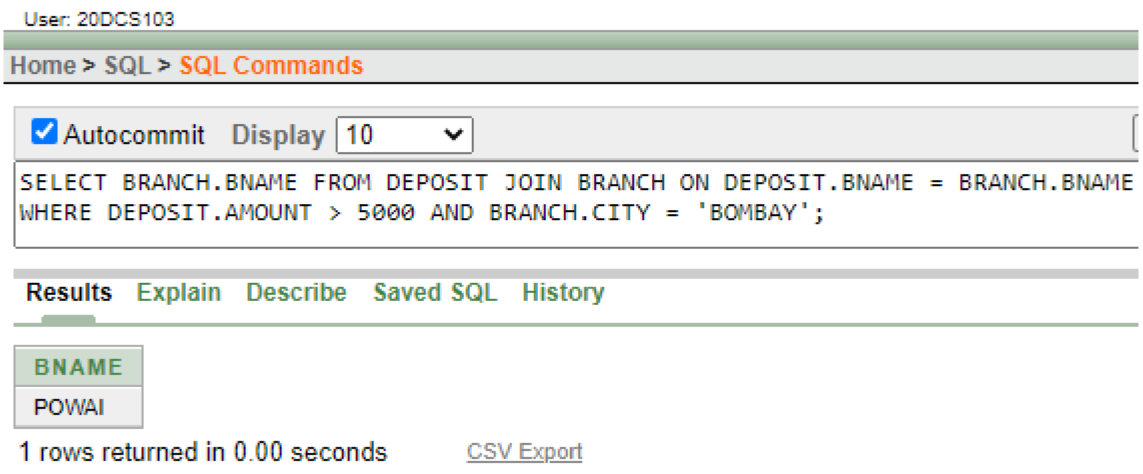
JOB_ID	SUM
comp_op	8000

1 rows returned in 0.01 seconds

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

Answer:

SELECT BRANCH.BNAME FROM DEPOSIT JOIN BRANCH
ON DEPOSIT.BNAME = BRANCH.BNAME
WHERE DEPOSIT.AMOUNT > 5000 AND BRANCH.CITY =
'BOMBAY';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT BRANCH.BNAME FROM DEPOSIT JOIN BRANCH ON DEPOSIT.BNAME = BRANCH.BNAME
WHERE DEPOSIT.AMOUNT > 5000 AND BRANCH.CITY = 'BOMBAY';
```

BNAME
POWAI

1 rows returned in 0.00 seconds

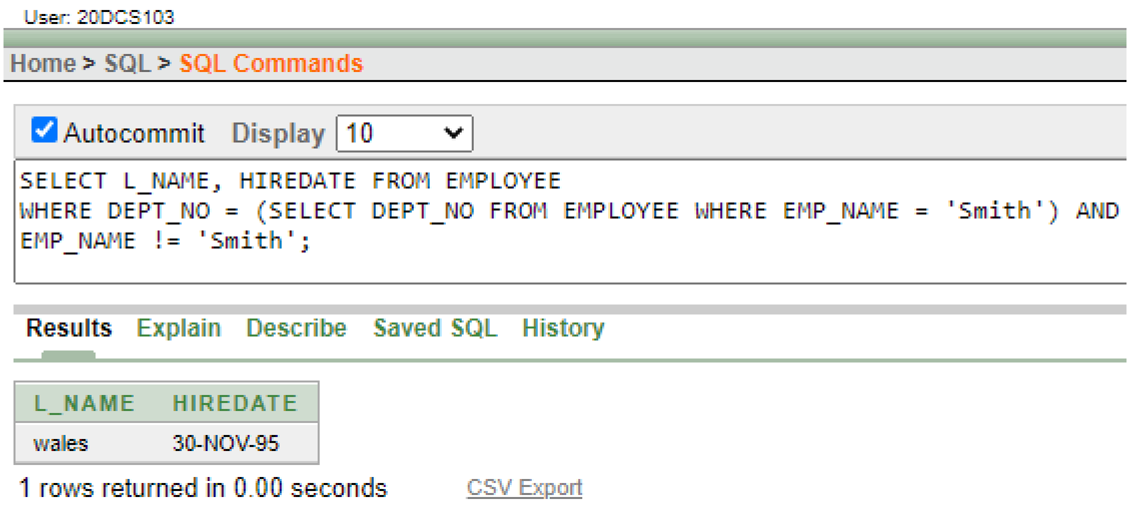
Aim 9: To solve queries using the concept of sub query.

Answer:

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

Answer:

SELECT L_NAME, HIREDATE FROM EMPLOYEE
WHERE DEPT_NO = (SELECT DEPT_NO FROM EMPLOYEE WHERE
EMP_NAME = 'Smith') AND
EMP_NAME != 'Smith';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT L_NAME, HIREDATE FROM EMPLOYEE
WHERE DEPT_NO = (SELECT DEPT_NO FROM EMPLOYEE WHERE EMP_NAME = 'Smith') AND
EMP_NAME != 'Smith';
```

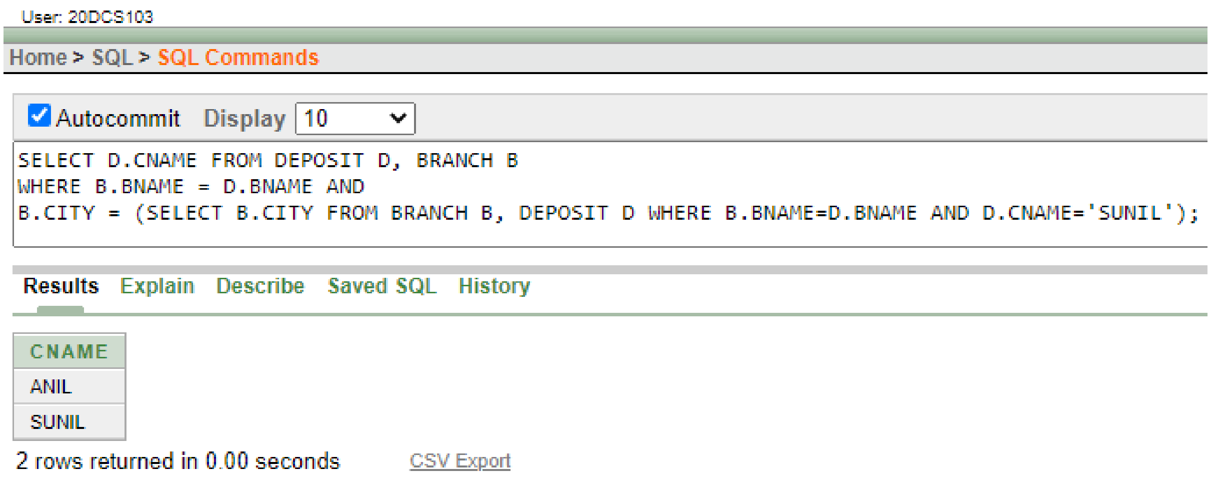
L_NAME	HIREDATE
wales	30-NOV-95

1 rows returned in 0.00 seconds

2) Give name of customers who are depositors having same branch city of Mr. Sunil.

Answer:

SELECT D.CNAME FROM DEPOSIT D, BRANCH B
WHERE B.BNAME = D.BNAME AND
B.CITY = (SELECT B.CITY FROM BRANCH B, DEPOSIT D WHERE B.BNAME=D.BNAME AND D.CNAME='SUNIL');


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT D.CNAME FROM DEPOSIT D, BRANCH B
WHERE B.BNAME = D.BNAME AND
B.CITY = (SELECT B.CITY FROM BRANCH B, DEPOSIT D WHERE B.BNAME=D.BNAME AND D.CNAME='SUNIL');
```

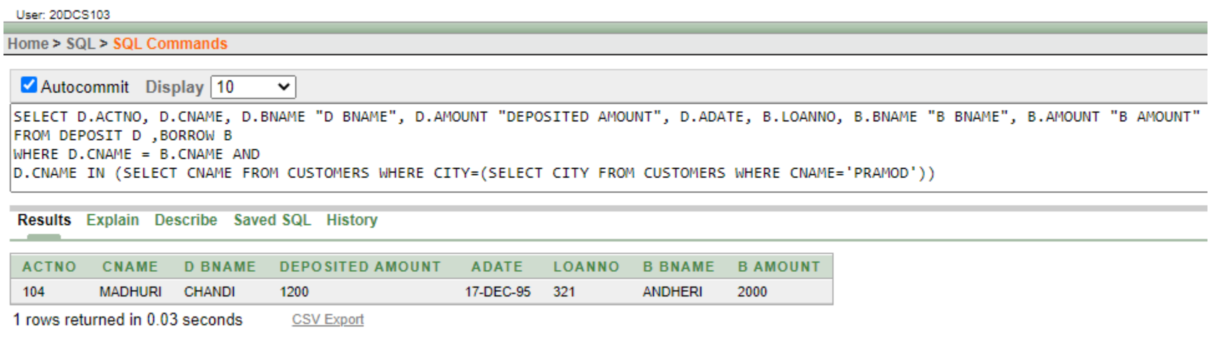
CNAME
ANIL
SUNIL

2 rows returned in 0.00 seconds

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

Answer:

SELECT D.ACTNO, D.CNAME, D.BNAME "D BNAME", D.AMOUNT "DEPOSITED AMOUNT", D.ADATE, B.LOANNO, B.BNAME "B BNAME", B.AMOUNT "B AMOUNT"
FROM DEPOSIT D ,BORROW B
WHERE D.CNAME = B.CNAME AND
D.CNAME IN (SELECT CNAME FROM CUSTOMERS WHERE CITY=(SELECT CITY FROM CUSTOMERS WHERE CNAME='PRAMOD'))


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT D.ACTNO, D.CNAME, D.BNAME "D BNAME", D.AMOUNT "DEPOSITED AMOUNT", D.ADATE, B.LOANNO, B.BNAME "B BNAME", B.AMOUNT "B AMOUNT"
FROM DEPOSIT D ,BORROW B
WHERE D.CNAME = B.CNAME AND
D.CNAME IN (SELECT CNAME FROM CUSTOMERS WHERE CITY=(SELECT CITY FROM CUSTOMERS WHERE CNAME='PRAMOD'))
```

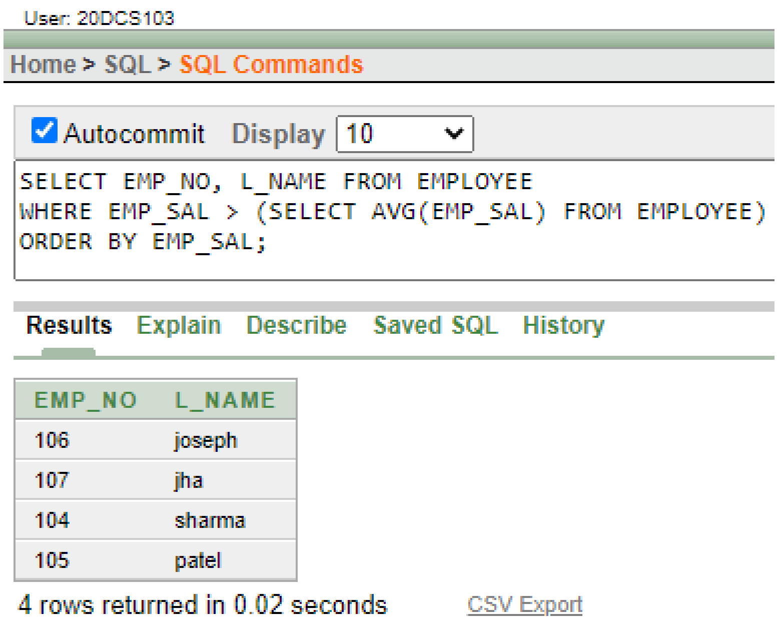
ACTNO	CNAME	D BNAME	DEPOSITED AMOUNT	ADATE	LOANNO	B BNAME	B AMOUNT
104	MADHURI	CHANDI	1200	17-DEC-95	321	ANDHERI	2000

1 rows returned in 0.03 seconds

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.

Answer:

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


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

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

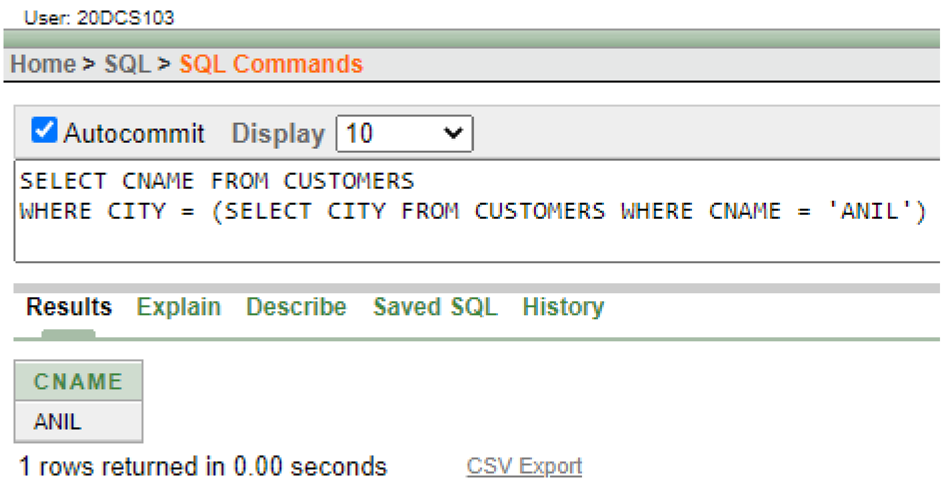
EMP_NO	L_NAME
106	joseph
107	jha
104	sharma
105	patel

4 rows returned in 0.02 seconds

5) Give names of depositors having same living city as Mr. Anil and having deposit amount greater than 2000.

Answer:

SELECT CNAME FROM CUSTOMERS
WHERE CITY = (SELECT CITY FROM CUSTOMERS WHERE
CNAME = 'ANIL')



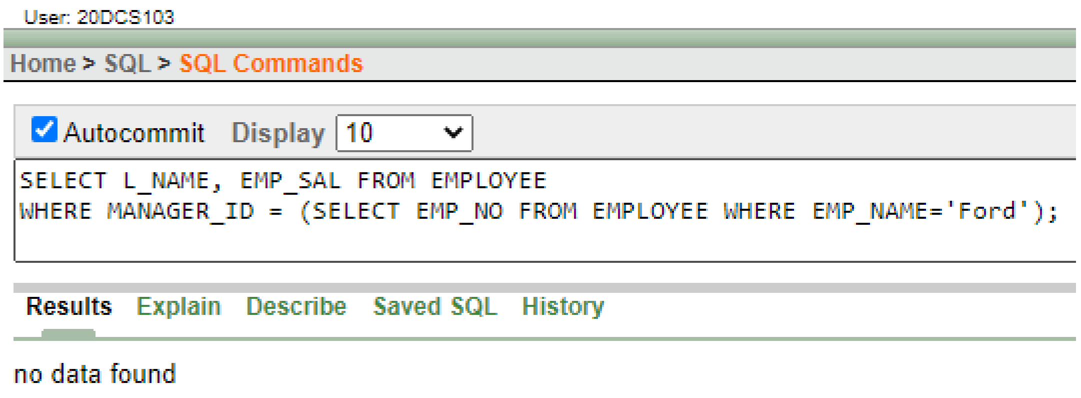
```
SELECT CNAME FROM CUSTOMERS
WHERE CITY = (SELECT CITY FROM CUSTOMERS WHERE CNAME = 'ANIL')
```

```
CNAME
ANIL
1 rows returned in 0.00 seconds
```

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

Answer:

SELECT L_NAME, EMP_SAL FROM EMPLOYEE
WHERE MANAGER_ID = (SELECT EMP_NO FROM EMPLOYEE
WHERE EMP_NAME='Ford');



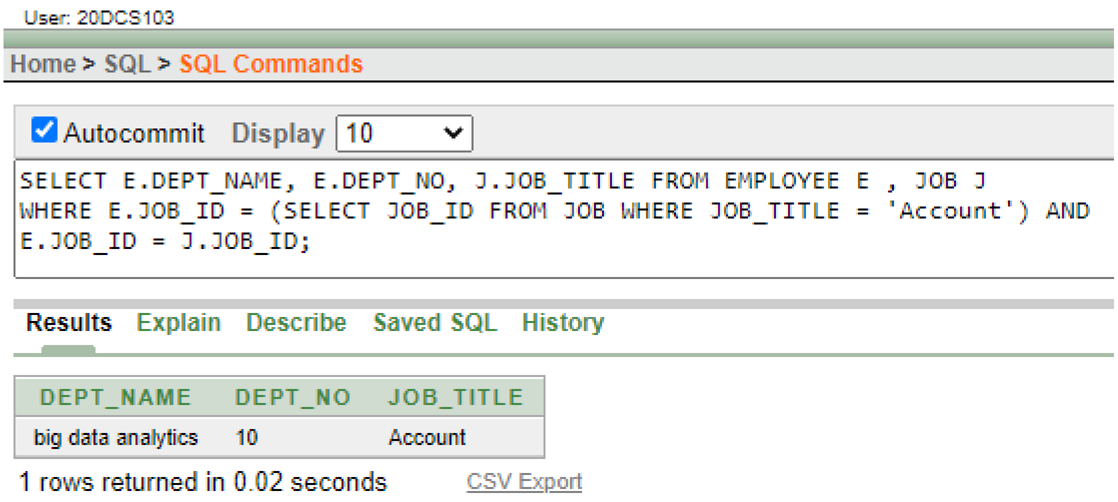
```
SELECT L_NAME, EMP_SAL FROM EMPLOYEE
WHERE MANAGER_ID = (SELECT EMP_NO FROM EMPLOYEE WHERE EMP_NAME='Ford');
```

```
no data found
```

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

Answer:

SELECT E.DEPT_NAME, E.DEPT_NO, J.JOB_TITLE FROM
EMPLOYEE E , JOB J
WHERE E.JOB_ID = (SELECT JOB_ID FROM JOB WHERE JOB_TITLE
= 'Account') AND
E.JOB_ID = J.JOB_ID;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT E.DEPT_NAME, E.DEPT_NO, J.JOB_TITLE FROM EMPLOYEE E , JOB J
WHERE E.JOB_ID = (SELECT JOB_ID FROM JOB WHERE JOB_TITLE = 'Account') AND
E.JOB_ID = J.JOB_ID;
```

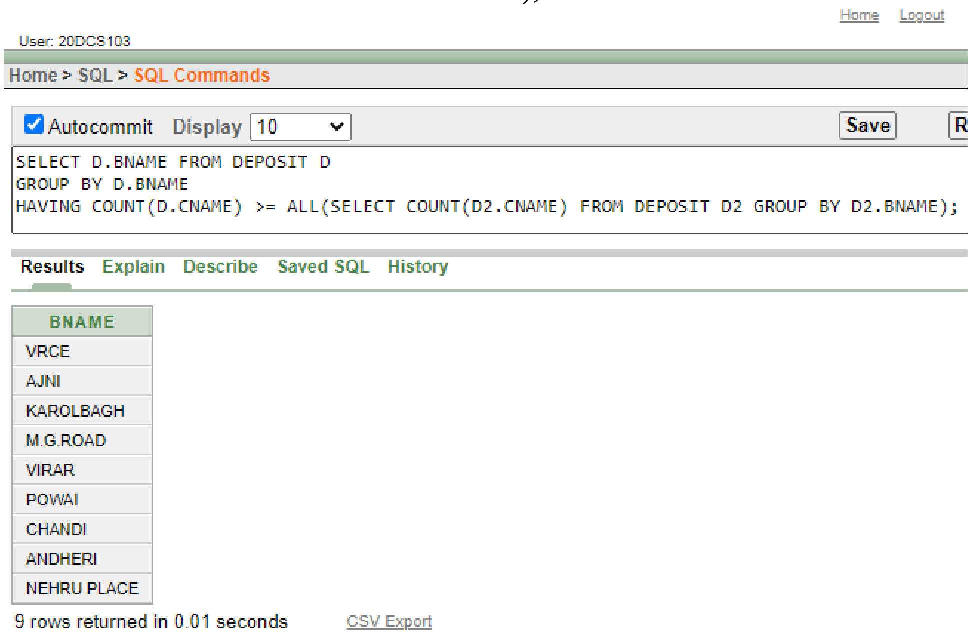
DEPT_NAME	DEPT_NO	JOB_TITLE
big data analytics	10	Account

1 rows returned in 0.02 seconds

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

Answer:

SELECT D.BNAME FROM DEPOSIT D
GROUP BY D.BNAME
HAVING COUNT(D.CNAME) >= ALL(SELECT COUNT(D2.CNAME) FROM
DEPOSIT D2 GROUP BY D2.BNAME);


Home Logout
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Save
R
Results Explain Describe Saved SQL History
CSV Export

```
SELECT D.BNAME FROM DEPOSIT D
GROUP BY D.BNAME
HAVING COUNT(D.CNAME) >= ALL(SELECT COUNT(D2.CNAME) FROM DEPOSIT D2 GROUP BY D2.BNAME);
```

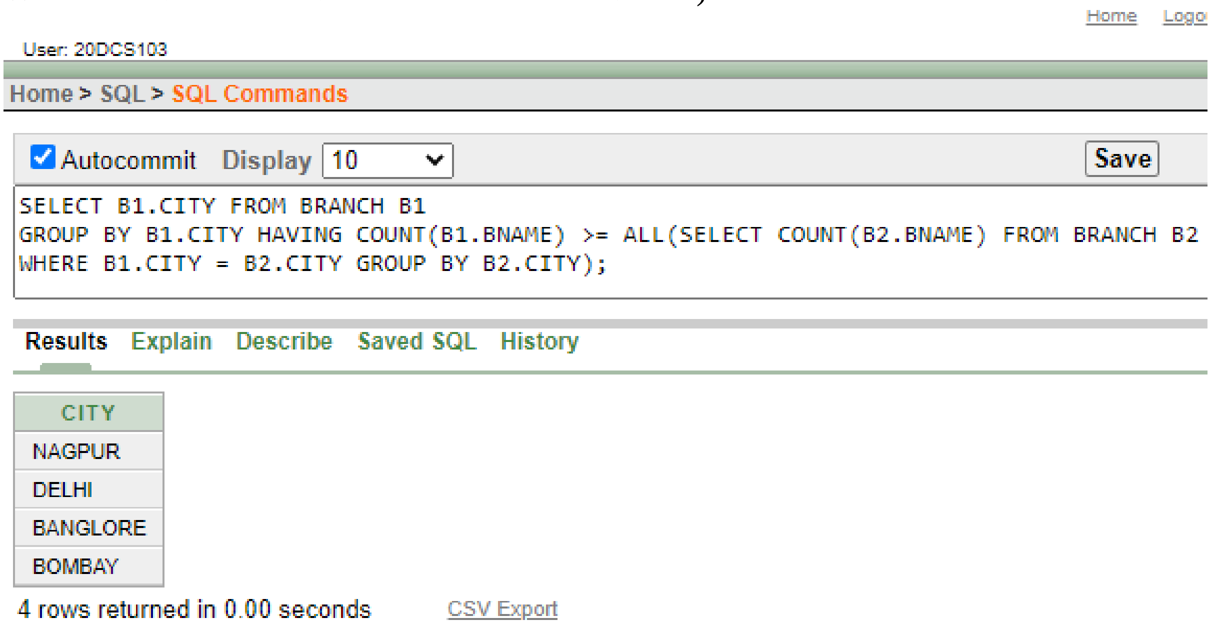
BNAME
VRCE
AJNI
KAROLBAGH
M.G.ROAD
VIRAR
POWAI
CHANDI
ANDHERI
NEHRU PLACE

9 rows returned in 0.01 seconds

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

Answer:

SELECT B1.CITY FROM BRANCH B1
GROUP BY B1.CITY HAVING COUNT(B1.BNAME) >= ALL(SELECT COUNT(B2.BNAME) FROM BRANCH B2
WHERE B1.CITY = B2.CITY GROUP BY B2.CITY)



```
SELECT B1.CITY FROM BRANCH B1
GROUP BY B1.CITY HAVING COUNT(B1.BNAME) >= ALL(SELECT COUNT(B2.BNAME) FROM BRANCH B2
WHERE B1.CITY = B2.CITY GROUP BY B2.CITY);
```

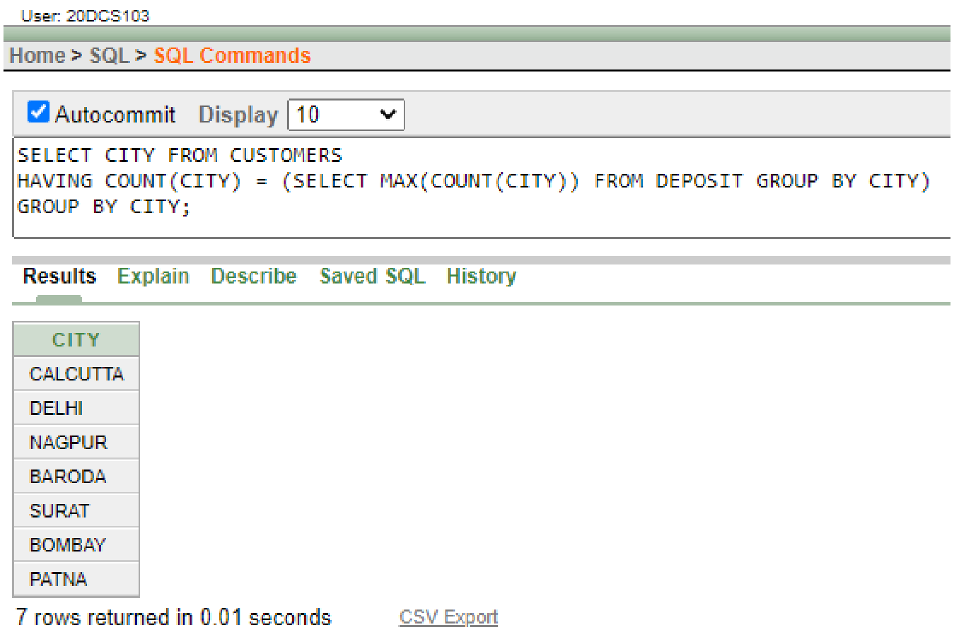
CITY
NAGPUR
DELHI
BANGLORE
BOMBAY

4 rows returned in 0.00 seconds

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

Answer:

SELECT CITY FROM CUSTOMERS HAVING COUNT(CITY) = (SELECT MAX(COUNT(CITY)) FROM DEPOSIT GROUP BY CITY) GROUP BY CITY;



```
SELECT CITY FROM CUSTOMERS
HAVING COUNT(CITY) = (SELECT MAX(COUNT(CITY)) FROM DEPOSIT GROUP BY CITY)
GROUP BY CITY;
```

CITY
CALCUTTA
DELHI
NAGPUR
BARODA
SURAT
BOMBAY
PATNA

7 rows returned in 0.01 seconds

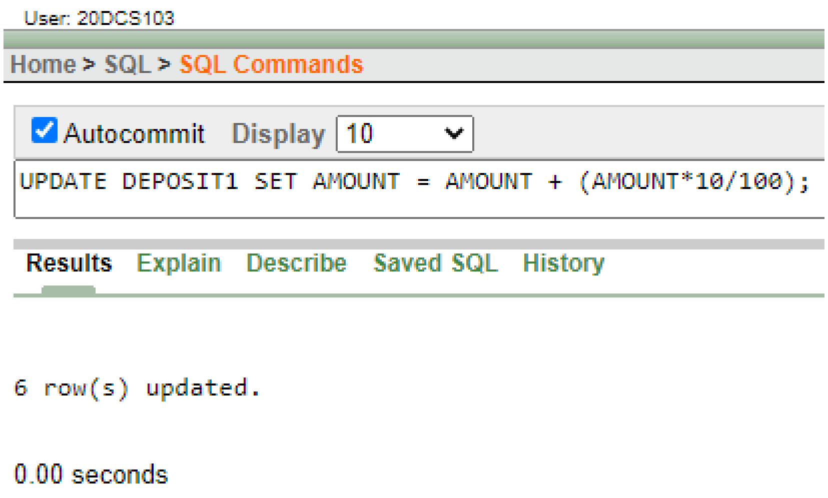
Aim 10: Manipulating Data

Answer:

1) Give 10% interest to all depositors.

Answer:

UPDATE DEPOSIT1 SET AMOUNT = AMOUNT + (AMOUNT*10/100);


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

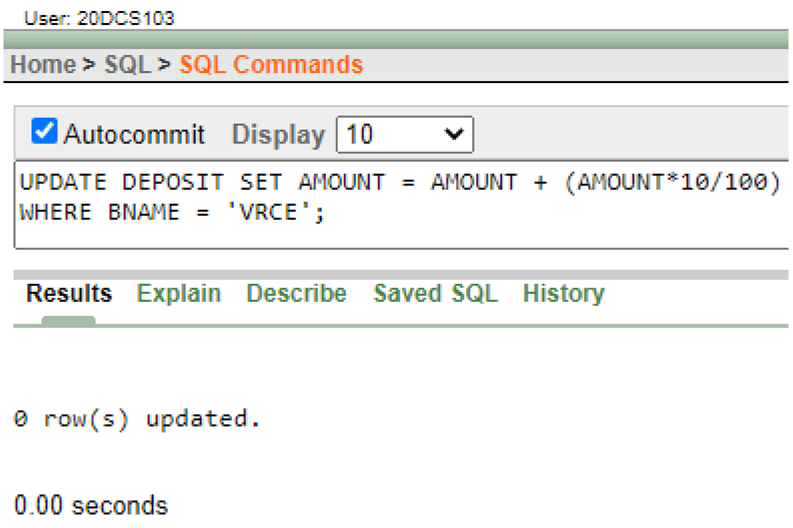
```
UPDATE DEPOSIT1 SET AMOUNT = AMOUNT + (AMOUNT*10/100);
```

```
6 row(s) updated.

0.00 seconds
```

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

Answer:

UPDATE DEPOSIT1 SET AMOUNT = AMOUNT +
(AMOUNT*10/100)
WHERE BNAME = 'VRCE';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

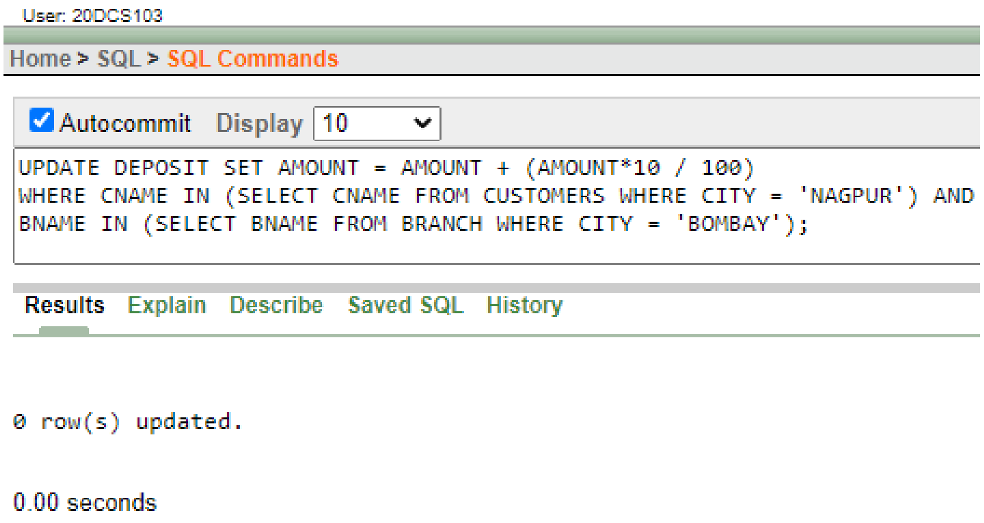
```
UPDATE DEPOSIT SET AMOUNT = AMOUNT + (AMOUNT*10/100)
WHERE BNAME = 'VRCE';
```

```
0 row(s) updated.

0.00 seconds
```

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

Answer:

UPDATE DEPOSIT SET AMOUNT = AMOUNT +
(AMOUNT*10 / 100)
WHERE CNAME IN (SELECT CNAME FROM
CUSTOMERS WHERE CITY = 'NAGPUR') AND
BNAME IN (SELECT BNAME FROM BRANCH WHERE
CITY = 'BOMBAY');


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

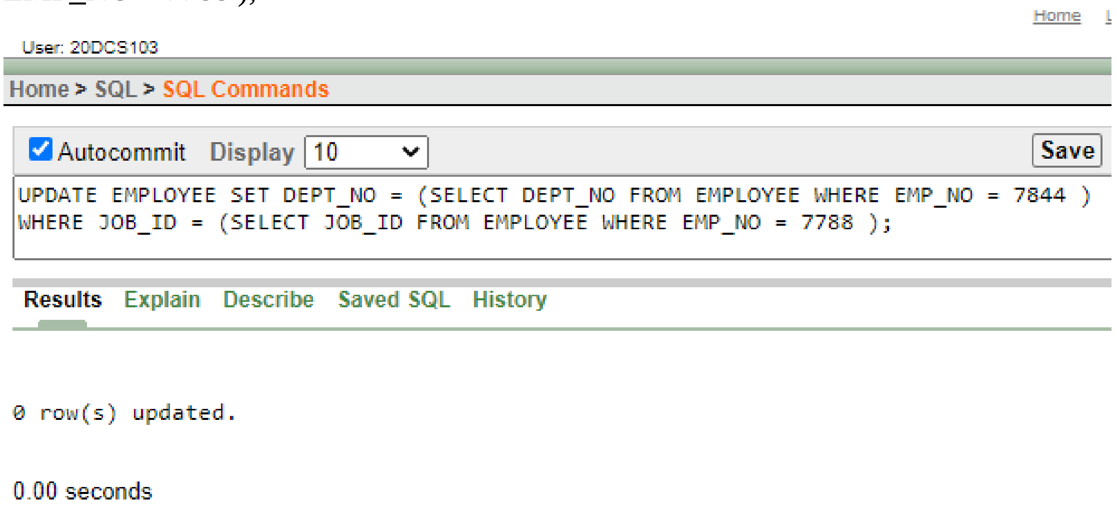
```
UPDATE DEPOSIT SET AMOUNT = AMOUNT + (AMOUNT*10 / 100)
WHERE CNAME IN (SELECT CNAME FROM CUSTOMERS WHERE CITY = 'NAGPUR') AND
BNAME IN (SELECT BNAME FROM BRANCH WHERE CITY = 'BOMBAY');
```

```
0 row(s) updated.

0.00 seconds
```

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

Answer:

UPDATE EMPLOYEE SET DEPT_NO = (SELECT DEPT_NO FROM
EMPLOYEE WHERE EMP_NO = 7844)
WHERE JOB_ID = (SELECT JOB_ID FROM EMPLOYEE WHERE
EMP_NO = 7788);


Home
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

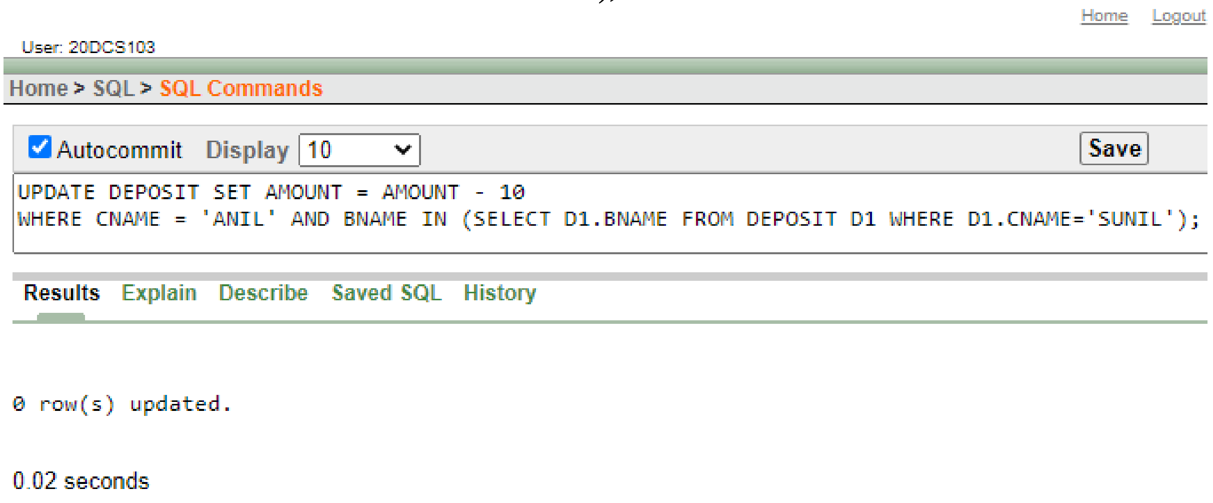
```
UPDATE EMPLOYEE SET DEPT_NO = (SELECT DEPT_NO FROM EMPLOYEE WHERE EMP_NO = 7844 )
WHERE JOB_ID = (SELECT JOB_ID FROM EMPLOYEE WHERE EMP_NO = 7788 );
```

```
0 row(s) updated.

0.00 seconds
```

5) Transfer 10 Rs from account of Anil to Sunil if both are having same branch.

Answer:

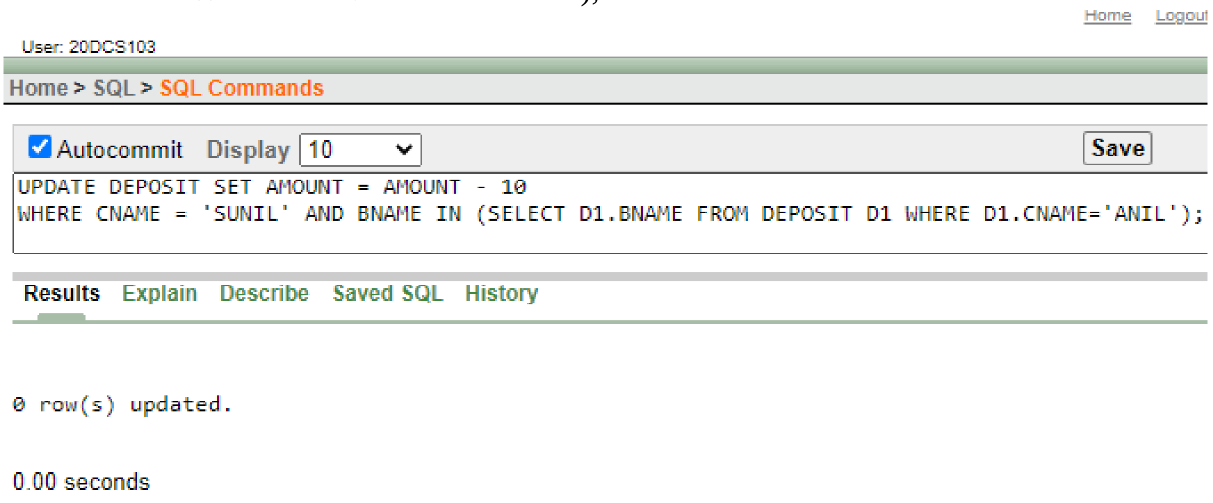
UPDATE DEPOSIT SET AMOUNT = AMOUNT -10
WHERE CNAME ='ANIL' AND BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME='SUNIL');


Home Logout
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10 Save
Results Explain Describe Saved SQL History

```
UPDATE DEPOSIT SET AMOUNT = AMOUNT - 10
WHERE CNAME = 'ANIL' AND BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME='SUNIL');
```

```
0 row(s) updated.

0.02 seconds
```

UPDATE DEPOSIT SET AMOUNT = AMOUNT - 10
WHERE CNAME ='SUNIL' AND BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME='ANIL');


Home Logout
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10 Save
Results Explain Describe Saved SQL History

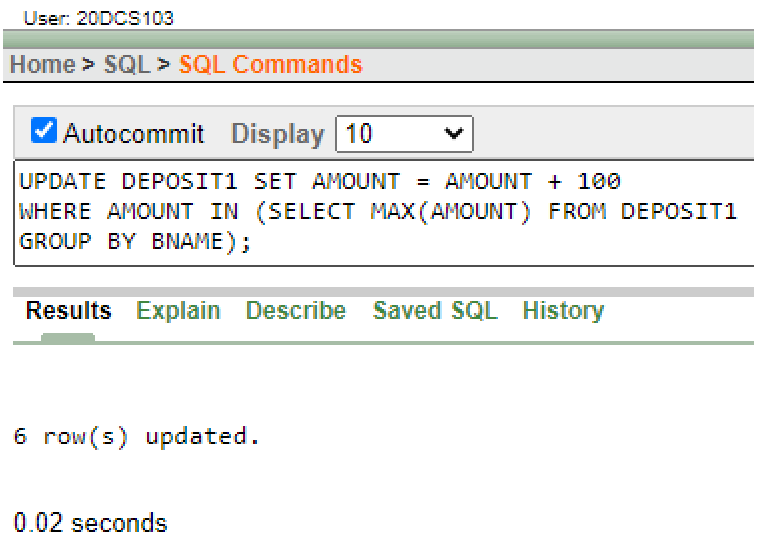
```
UPDATE DEPOSIT SET AMOUNT = AMOUNT - 10
WHERE CNAME = 'SUNIL' AND BNAME IN (SELECT D1.BNAME FROM DEPOSIT D1 WHERE D1.CNAME='ANIL');
```

```
0 row(s) updated.

0.00 seconds
```

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

Answer:

UPDATE DEPOSIT SET AMOUNT = AMOUNT + 100
WHERE AMOUNT IN (SELECT MAX(AMOUNT) FROM DEPOSIT
GROUP BY BNAME);



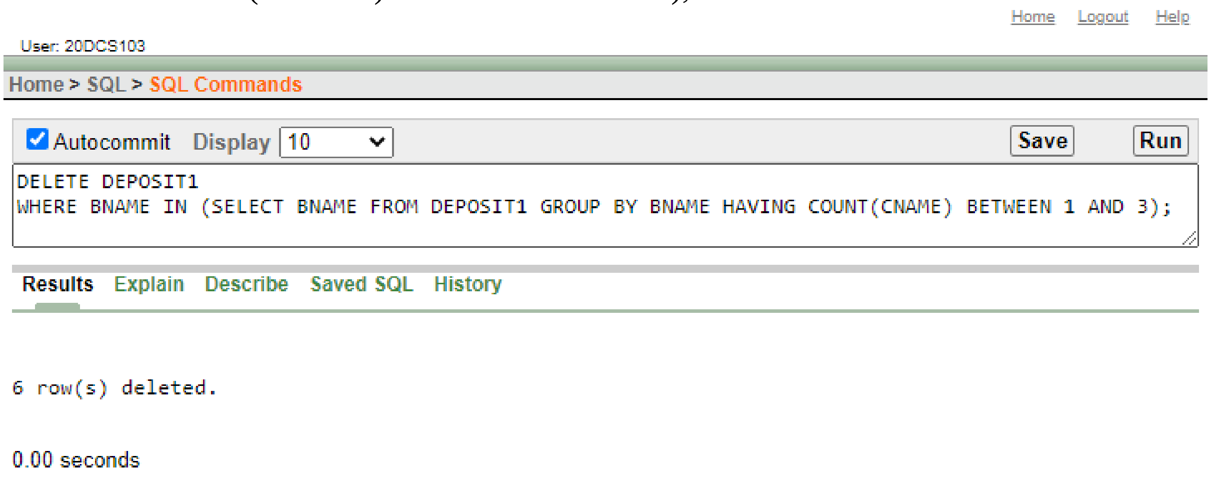
```
UPDATE DEPOSIT1 SET AMOUNT = AMOUNT + 100
WHERE AMOUNT IN (SELECT MAX(AMOUNT) FROM DEPOSIT1
GROUP BY BNAME);
```

```
6 row(s) updated.

0.02 seconds
```

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

Answer:

DELETE DEPOSIT1
WHERE BNAME IN (SELECT BNAME FROM DEPOSIT1 GROUP BY BNAME
HAVING COUNT(CNAME) BETWEEN 1 AND 3);



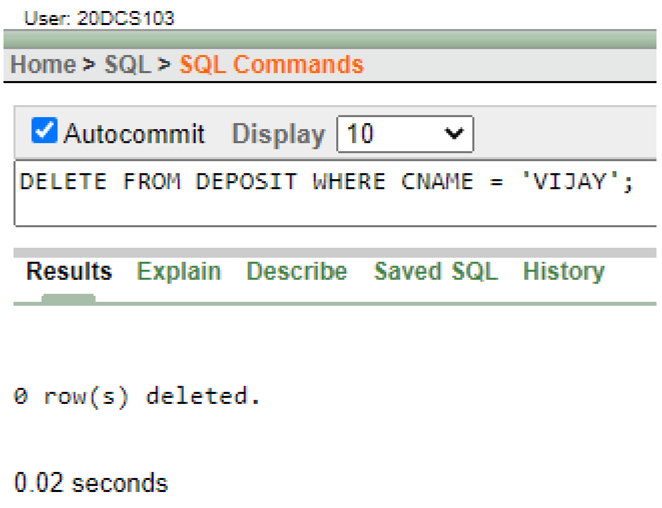
```
DELETE DEPOSIT1
WHERE BNAME IN (SELECT BNAME FROM DEPOSIT1 GROUP BY BNAME HAVING COUNT(CNAME) BETWEEN 1 AND 3);
```

```
6 row(s) deleted.

0.00 seconds
```

8) Delete deposit of Vijay.

Answer:

DELETE FROM DEPOSIT1 WHERE CNAME = 'VIJAY';


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

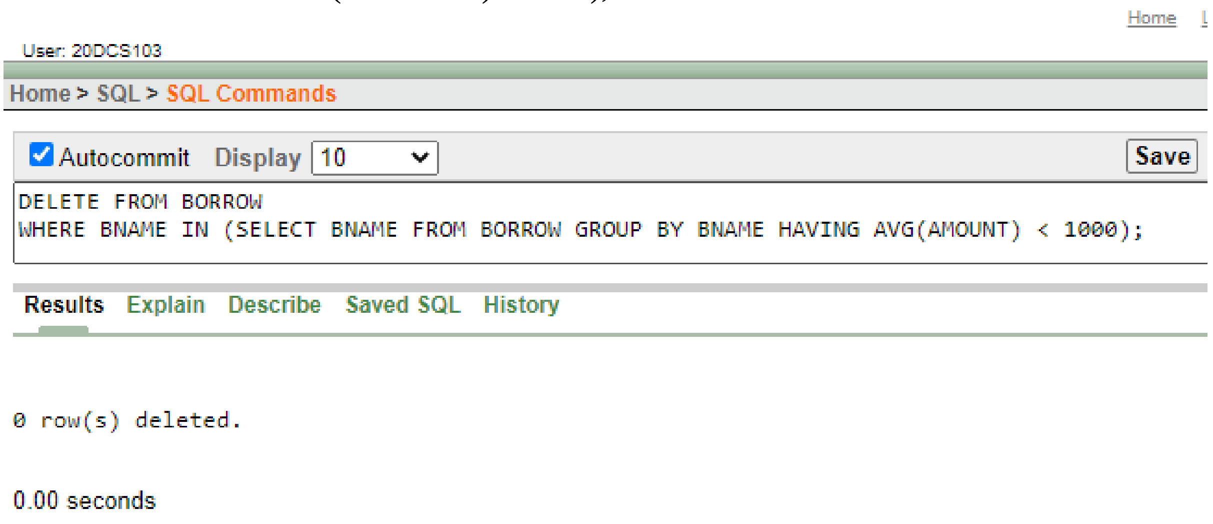
```
DELETE FROM DEPOSIT WHERE CNAME = 'VIJAY';
```

```
0 row(s) deleted.

0.02 seconds
```

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

Answer:

DELETE FROM BORROW
WHERE BNAME IN (SELECT BNAME FROM BORROW GROUP BY
BNAME HAVING AVG(AMOUNT) < 1000);


Home
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

```
DELETE FROM BORROW
WHERE BNAME IN (SELECT BNAME FROM BORROW GROUP BY BNAME HAVING AVG(AMOUNT) < 1000);
```

```
0 row(s) deleted.

0.00 seconds
```

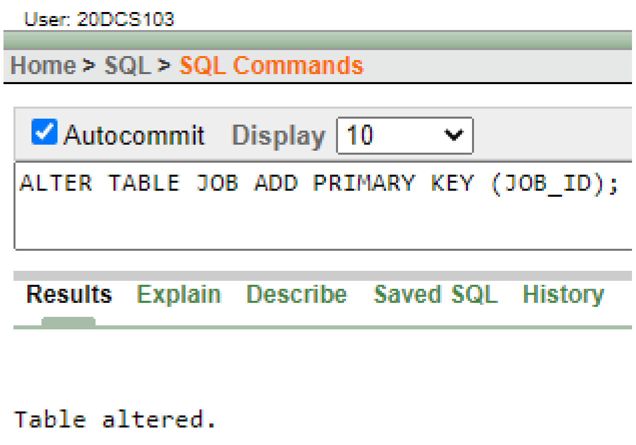
Aim 11: Add and Remove constraint.

Answer:

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

<u>Answer:</u>

ALTER TABLE JOB ADD PRIMARY KEY (JOB_ID);


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

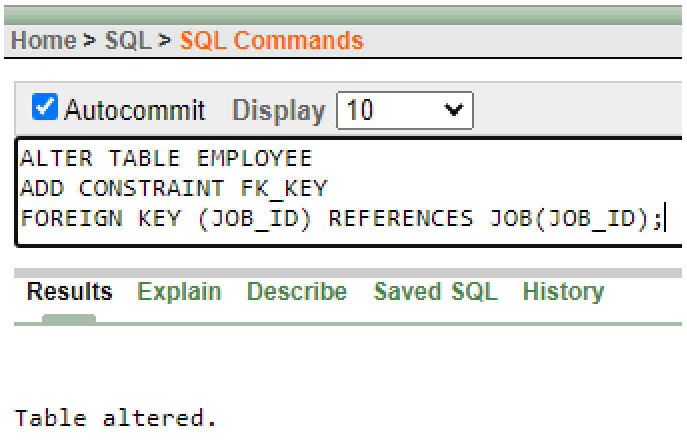
```
ALTER TABLE JOB ADD PRIMARY KEY (JOB_ID);
```

```
Table altered.
```

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

<u>Answer:</u>

ALTER TABLE EMPLOYEE ADD CONSTRAINT FK_KEY FOREIGN KEY (JOB_ID) REFERENCES JOB (JOB_ID);


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

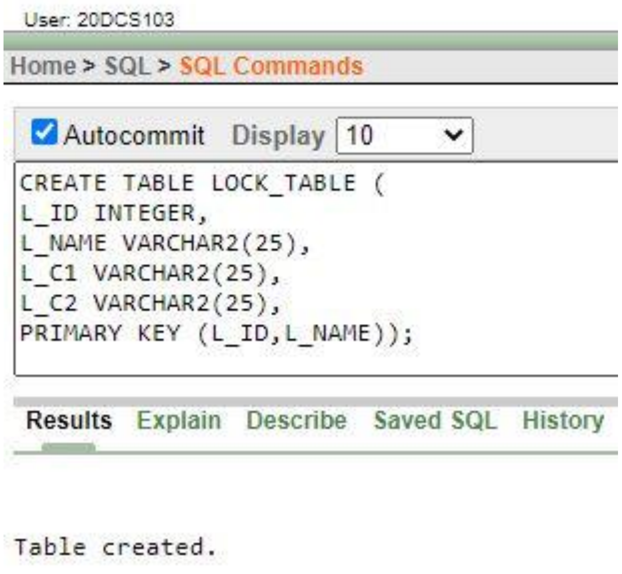
```
ALTER TABLE EMPLOYEE
ADD CONSTRAINT FK_KEY
FOREIGN KEY (JOB_ID) REFERENCES JOB(JOB_ID);
```

```
Table altered.
```

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

Answer:

CREATE TABLE LOCK_TABLE (
L_ID INTEGER,
L_NAME VARCHAR2(25),
L_C1 VARCHAR2(25),
L_C2 VARCHAR2(25),
PRIMARY KEY (L_ID,L_NAME));


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

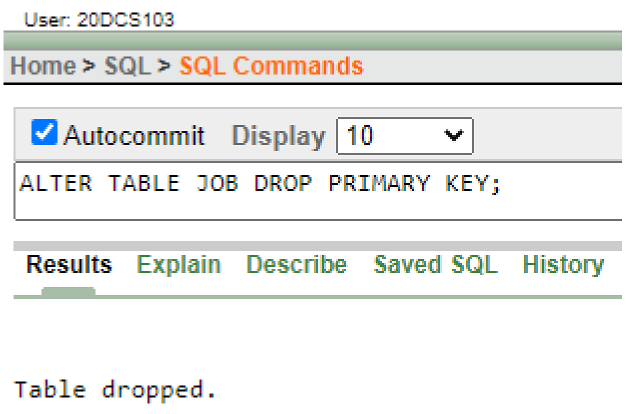
```
CREATE TABLE LOCK_TABLE (
L_ID INTEGER,
L_NAME VARCHAR2(25),
L_C1 VARCHAR2(25),
L_C2 VARCHAR2(25),
PRIMARY KEY (L_ID,L_NAME));
```

```
Table created.
```

4) Remove primary key constraint on job_id.

Answer:

ALTER TABLE DROP PRIMARY KEY;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
ALTER TABLE JOB DROP PRIMARY KEY;
```

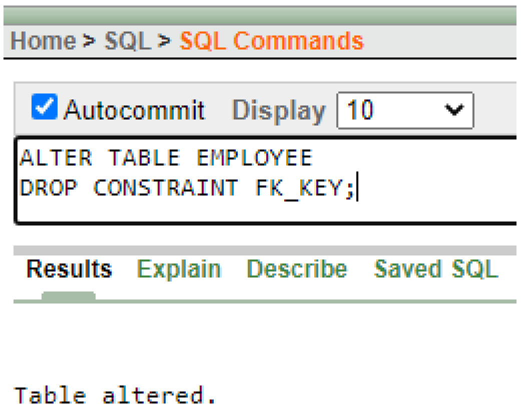
```
Table dropped.
```

5) Remove foreign key constraint on employee table.

Answer:

ALTER TABLE EMPLOYEE

DROP CONSTRAINT FK_KEY;



```
ALTER TABLE EMPLOYEE
DROP CONSTRAINT FK_KEY;
```

```
Table altered.
```

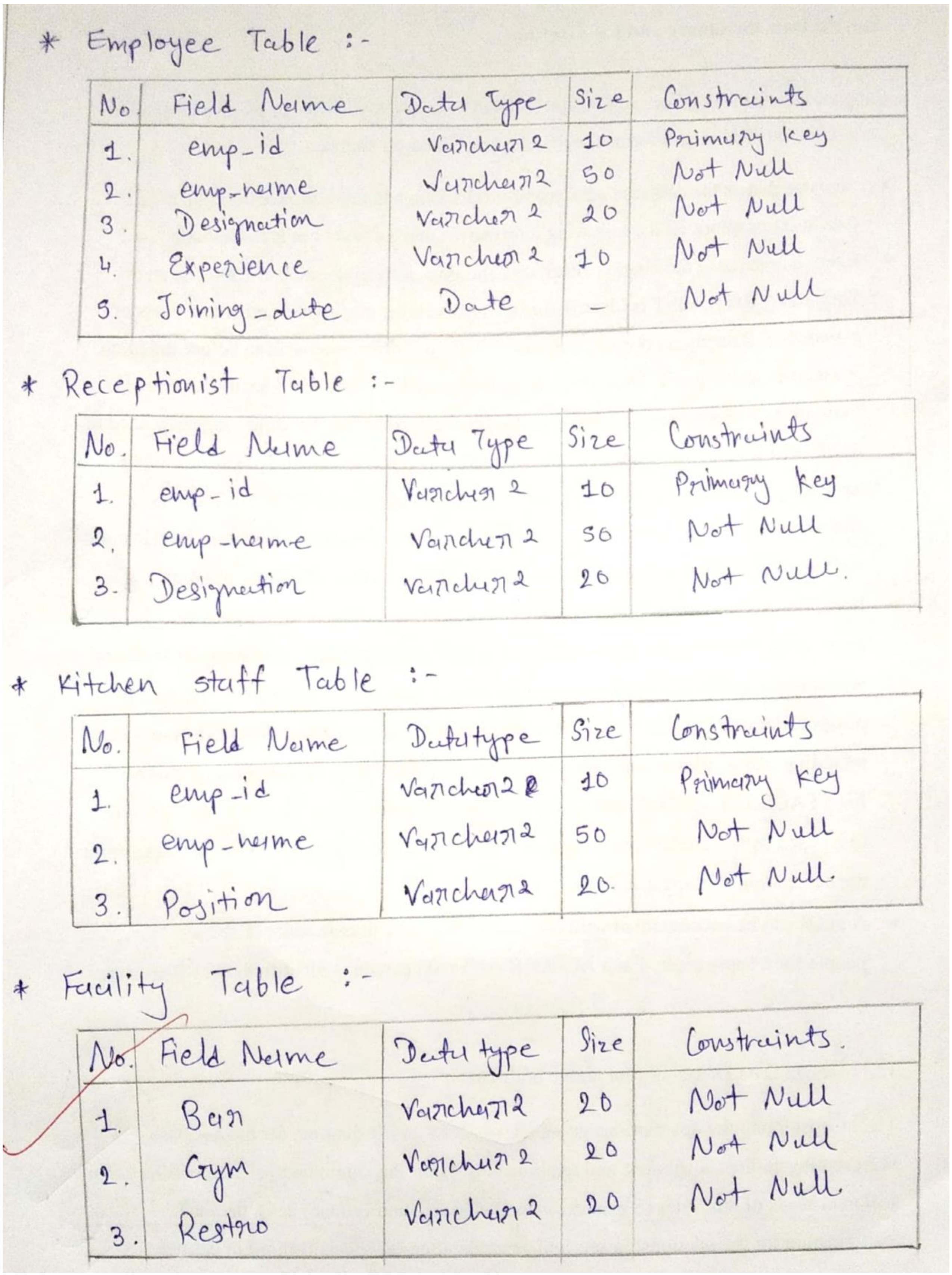
Aim 12: Data dictionary and ER diagram.

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

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

12.1) Design Data Dictionary for above problem.

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



* Employee Table :-

No	Field Name	Data Type	Size	Constraints
1.	emp_id	Varchar2	10	Primary key
2.	emp_name	Varchar2	50	Not Null
3.	Designation	Varchar2	20	Not Null
4.	Experience	Varchar2	10	Not Null
5.	Joining_date	Date	-	Not Null

* Receptionist Table :-

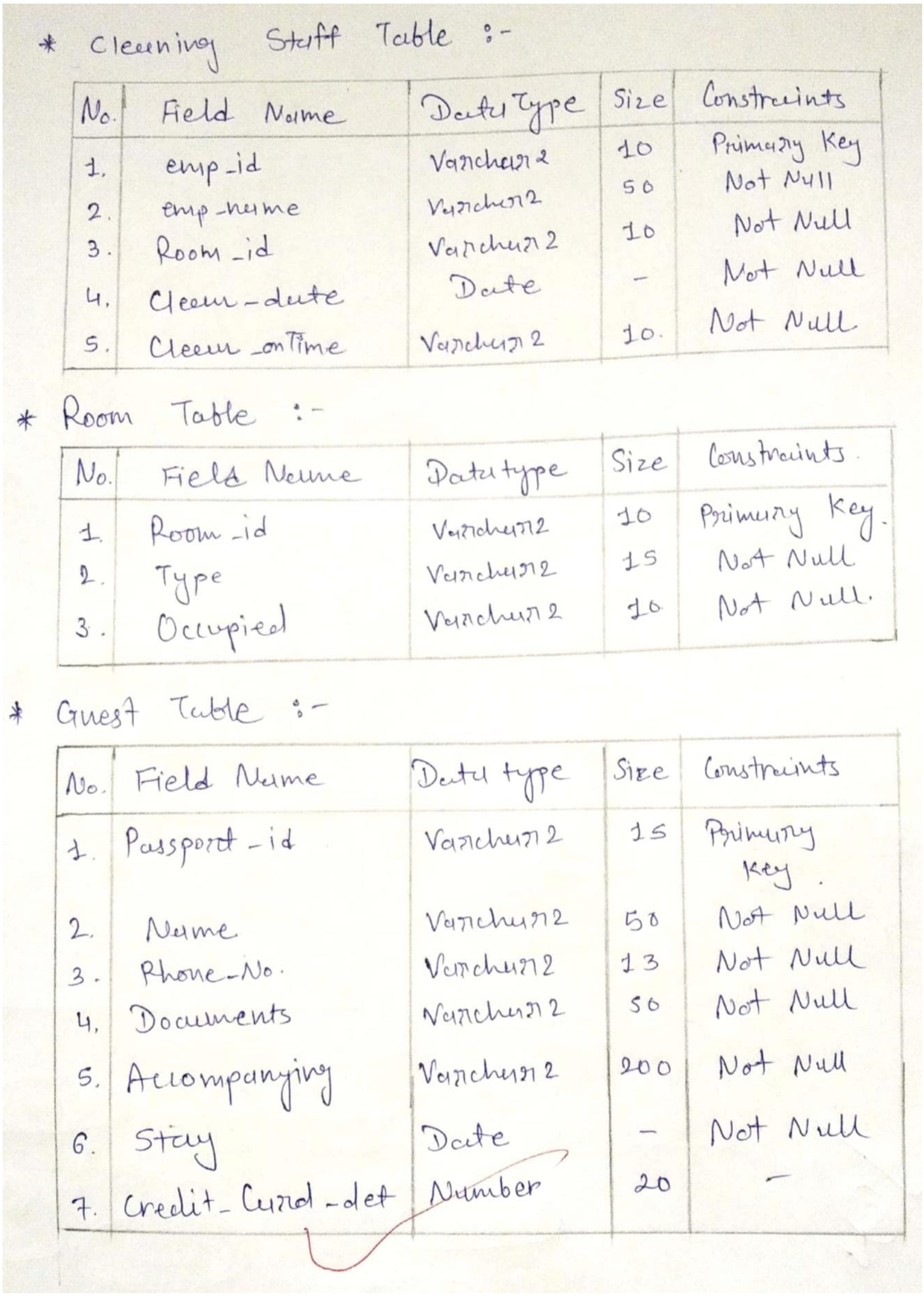
No.	Field Name	Data Type	Size	Constraints
1.	emp_id	Varchar2	10	Primary key
2.	emp_name	Varchar2	50	Not Null
3.	Designation	Varchar2	20	Not Null.

* Kitchen staff Table :-

No.	Field Name	Datatype	Size	Constraints
1.	emp_id	Varchar2	10	Primary key
2.	emp_name	Varchar2	50	Not Null
3.	Position	Varchar2	20.	Not Null.

* Facility Table :-

No.	Field Name	Data type	Size	Constraints.
1.	Bar	Varchar2	20	Not Null
2.	Gym	Varchar2	20	Not Null
3.	Restro	Varchar2	20	Not Null.



* Cleaning Staff Table :-

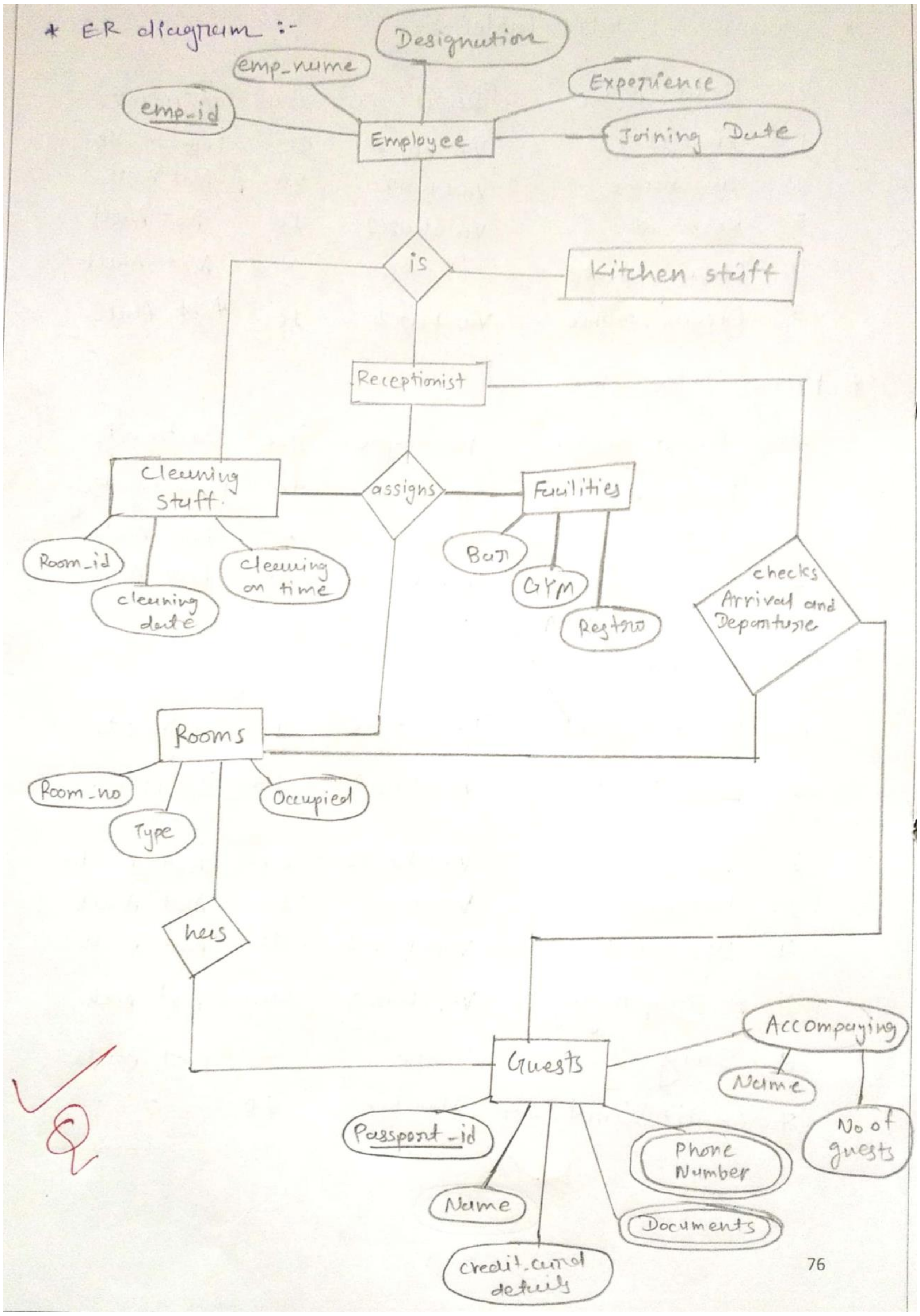
No.	Field Name	Data Type	Size	Constraints
1.	emp_id	Varchar2	10	Primary Key
2.	emp_name	Varchar2	50	Not Null
3.	Room_id	Varchar2	10	Not Null
4.	Clean_date	Date	-	Not Null
5.	Clean_onTime	Varchar2	10.	Not Null

* Room Table :-

No.	Field Name	Datatype	Size	Constraints.
1.	Room_id	Varchar2	10	Primary Key.
2.	Type	Varchar2	15	Not Null
3.	Occupied	Varchar2	10	Not Null.

* Guest Table :-

No.	Field Name	Data type	Size	Constraints
1.	Passport_id	Varchar2	15	Primary Key.
2.	Name	Varchar2	50	Not Null
3.	Phone_No.	Varchar2	13	Not Null
4.	Documents	Varchar2	50	Not Null
5.	Accompanying	Varchar2	200	Not Null
6.	Stay	Date	-	Not Null
7.	Credit_Card_det	Number	20	-


* ER diagram :-
Designation
emp_name
emp_id
Experience
Employee
Joining Date
is
Kitchen staff
Receptionist
Cleaning staff
assigns
Facilities
Room_id
cleaning date
cleaning on time
Bar
GYM
Restro
checks Arrival and Departure
Rooms
Room_no
Type
Occupied
has
Guests
Accompanying
Name
No of guests
Passport_id
Phone Number
Name
Documents
credit card details
76

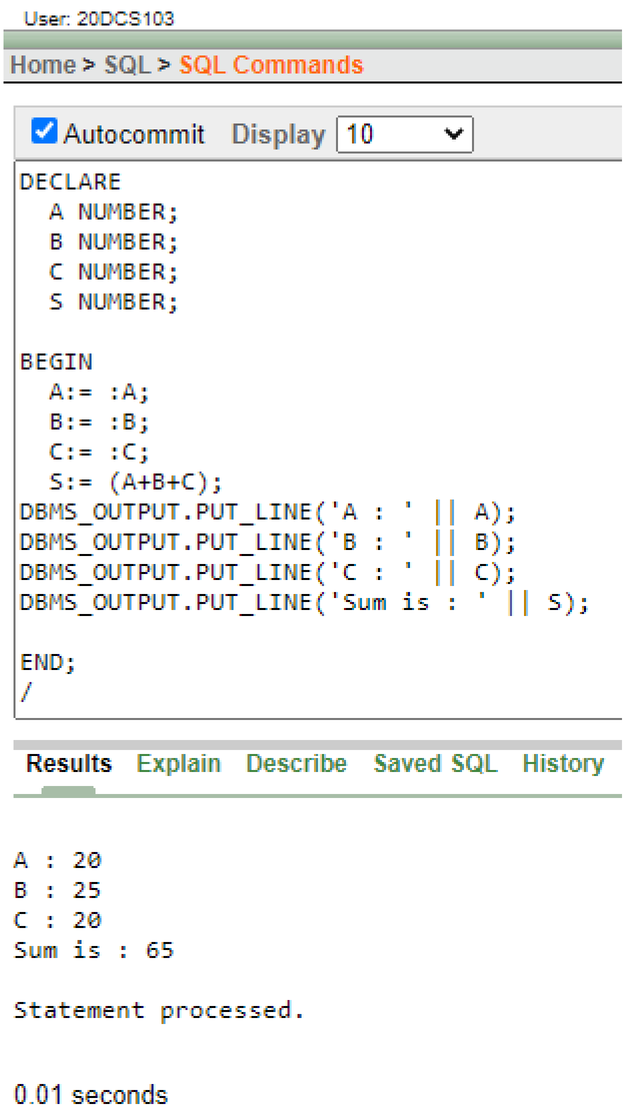
Aim 13: To perform basic PL/SQL blocks.

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

Answer:

Sum of three numbers:

```
DECLARE
  A NUMBER;
  B NUMBER;
  C NUMBER;
  S NUMBER;

BEGIN
  A:= :A;
  B:= :B;
  C:= :C;
  S:= (A+B+C);
DBMS_OUTPUT.PUT_LINE('A : ' || A);
DBMS_OUTPUT.PUT_LINE('B : ' || B);
DBMS_OUTPUT.PUT_LINE('C : ' || C);
‘DBMS_OUTPUT.PUT_LINE('Sum is : ' || S);

END;
/
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

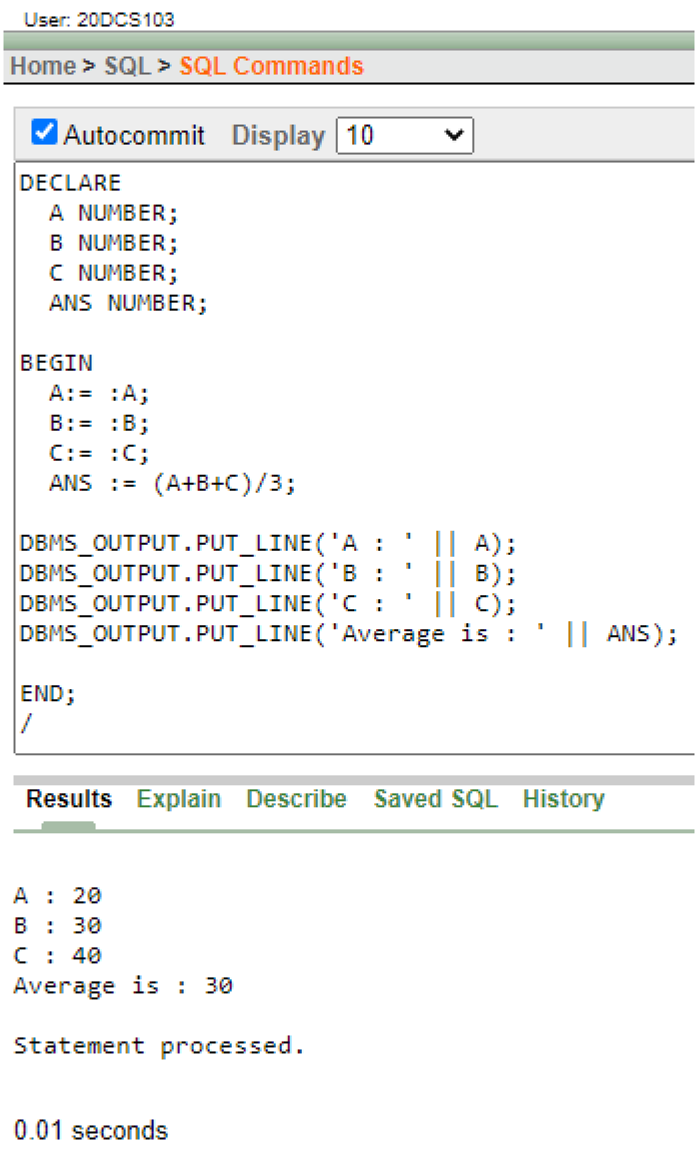
```
DECLARE
  A NUMBER;
  B NUMBER;
  C NUMBER;
  S NUMBER;

BEGIN
  A:= :A;
  B:= :B;
  C:= :C;
  S:= (A+B+C);
DBMS_OUTPUT.PUT_LINE('A : ' || A);
DBMS_OUTPUT.PUT_LINE('B : ' || B);
DBMS_OUTPUT.PUT_LINE('C : ' || C);
DBMS_OUTPUT.PUT_LINE('Sum is : ' || S);

END;
/
```

```
A : 20
B : 25
C : 20
Sum is : 65

Statement processed.
```

0.01 seconds

Average of three numbers:

```
DECLARE
  A NUMBER;
  B NUMBER;
  C NUMBER;
  ANS NUMBER;

BEGIN
  A:= :A;
  B:= :B;
  C:= :C;
  ANS := (A+B+C)/3;

DBMS_OUTPUT.PUT_LINE('A : ' || A);
DBMS_OUTPUT.PUT_LINE('B : ' || B);
DBMS_OUTPUT.PUT_LINE('C : ' || C);
DBMS_OUTPUT.PUT_LINE('Average is : ' || ANS);

END;
/
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
  A NUMBER;
  B NUMBER;
  C NUMBER;
  ANS NUMBER;

BEGIN
  A:= :A;
  B:= :B;
  C:= :C;
  ANS := (A+B+C)/3;

DBMS_OUTPUT.PUT_LINE('A : ' || A);
DBMS_OUTPUT.PUT_LINE('B : ' || B);
DBMS_OUTPUT.PUT_LINE('C : ' || C);
DBMS_OUTPUT.PUT_LINE('Average is : ' || ANS);

END;
/
```

```
A : 20
B : 30
C : 40
Average is : 30

Statement processed.
```

0.01 seconds

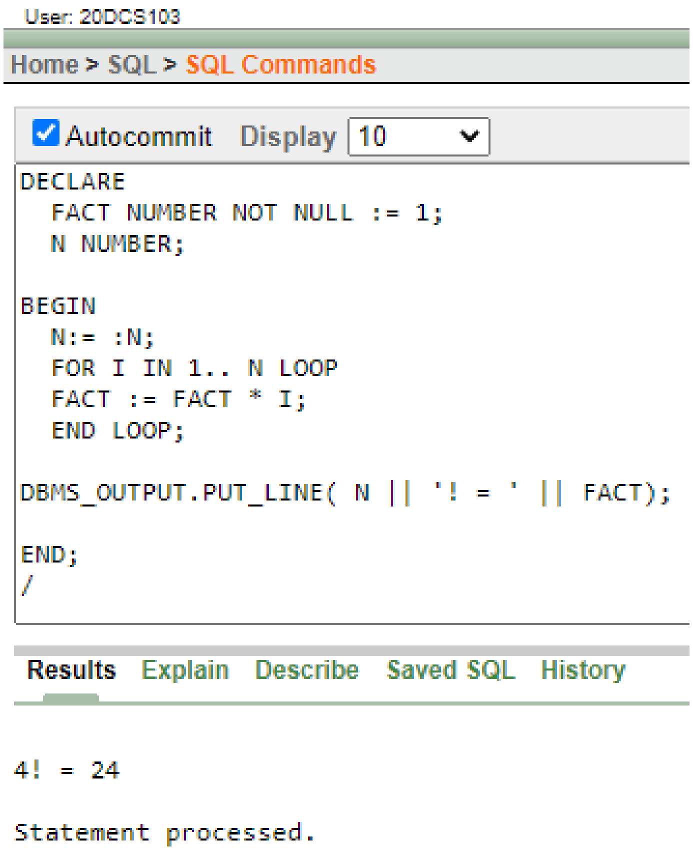
Aim 14: To perform the concept of loop.

Find the factorial of a number in pl/sql using for, while and simple loop.

Answer:

FOR loop:

```
DECLARE
  FACT NUMBER NOT NULL := 1;
  N NUMBER;

BEGIN
  N:= :N;
  FOR I IN 1.. N LOOP
  FACT := FACT * I;
  END LOOP;
DBMS_OUTPUT.PUT_LINE( N || '! = ' || FACT);
END;
/
```

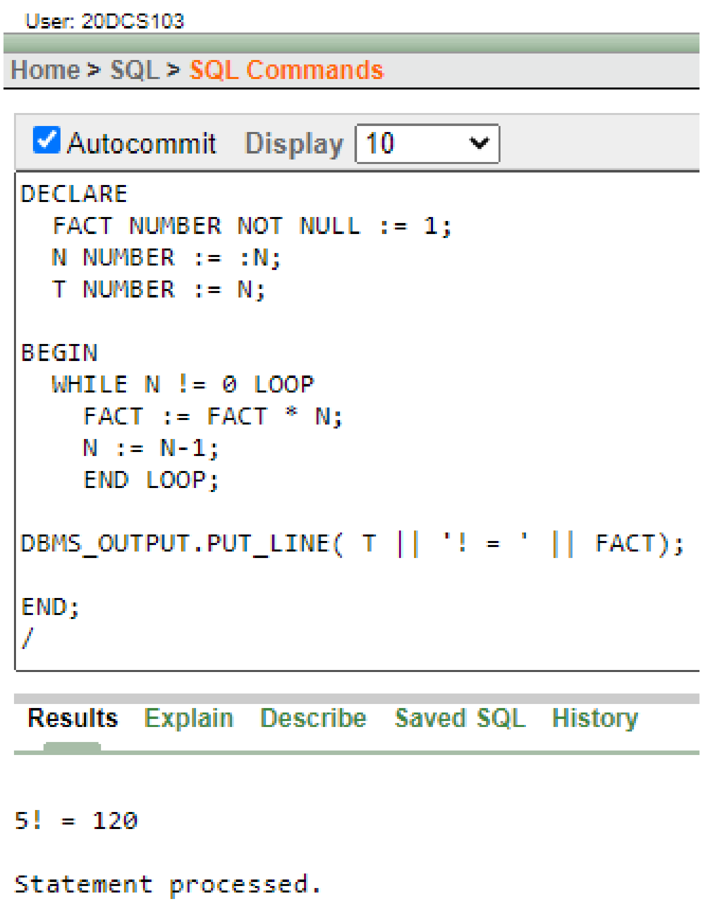


```
DECLARE
   FACT NUMBER NOT NULL := 1;
   N NUMBER;

BEGIN
   N:= :N;
   FOR I IN 1.. N LOOP
   FACT := FACT * I;
   END LOOP;

DBMS_OUTPUT.PUT_LINE( N || '! = ' || FACT);

END;
/
```

```
4! = 24

Statement processed.
```

WHILE loop:

```
DECLARE
  FACT NUMBER NOT NULL := 1;
  N NUMBER := :N;
  T NUMBER := N;

BEGIN
  WHILE N != 0 LOOP
    FACT := FACT * N;
    N := N-1;
    END LOOP;
DBMS_OUTPUT.PUT_LINE( T || '! = ' || FACT);

END;
/
```

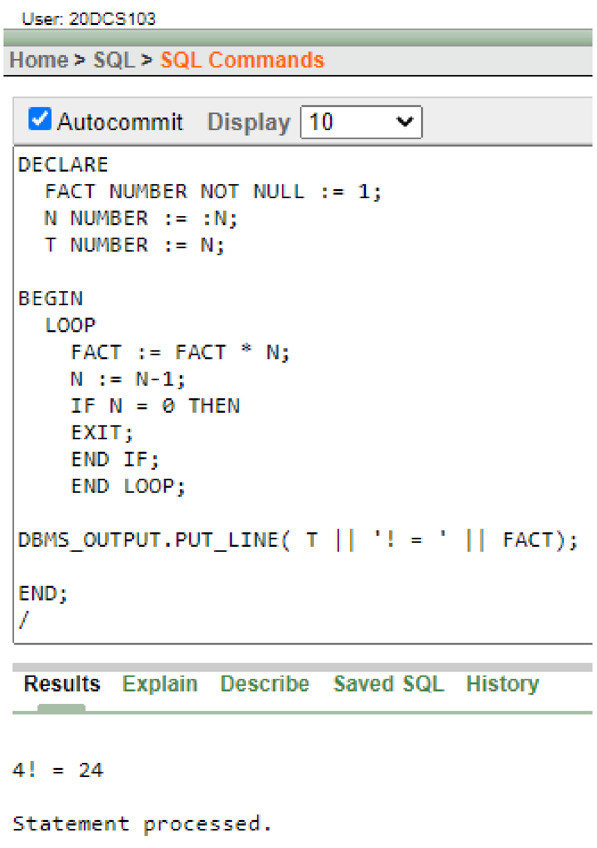

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
  FACT NUMBER NOT NULL := 1;
  N NUMBER := :N;
  T NUMBER := N;

BEGIN
  WHILE N != 0 LOOP
    FACT := FACT * N;
    N := N-1;
    END LOOP;

DBMS_OUTPUT.PUT_LINE( T || '! = ' || FACT);

END;
/
```

```
5! = 120

Statement processed.
```

Simple loop:

```
DECLARE
  FACT NUMBER NOT NULL := 1;
  N NUMBER := :N;
  T NUMBER := N;
BEGIN
  LOOP
    FACT := FACT * N;
    N := N-1;
    IF N = 0 THEN
    EXIT;
    END IF;
    END LOOP;
DBMS_OUTPUT.PUT_LINE( T || '! = ' || FACT);
END;
/
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
  FACT NUMBER NOT NULL := 1;
  N NUMBER := :N;
  T NUMBER := N;

BEGIN
  LOOP
    FACT := FACT * N;
    N := N-1;
    IF N = 0 THEN
    EXIT;
    END IF;
    END LOOP;

DBMS_OUTPUT.PUT_LINE( T || '! = ' || FACT);

END;
/
```

```
4! = 24

Statement processed.
```

Aim 15: To understand the concept of "select into" and "% type" attribute.

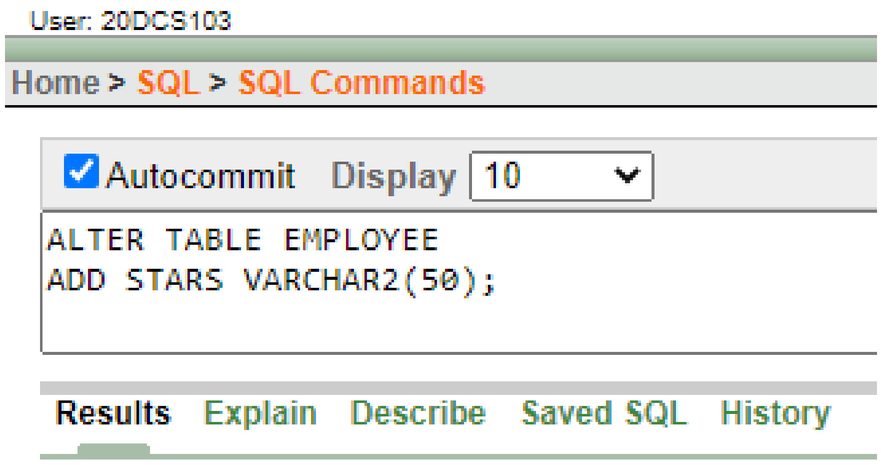
Create an EMPLOYEES table that is a replica of the EMP table. Add a new column, STARS, of VARCHAR2 data type and length of 50 to the EMPLOYEES table for storing asterisk (*).

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.

Answer:

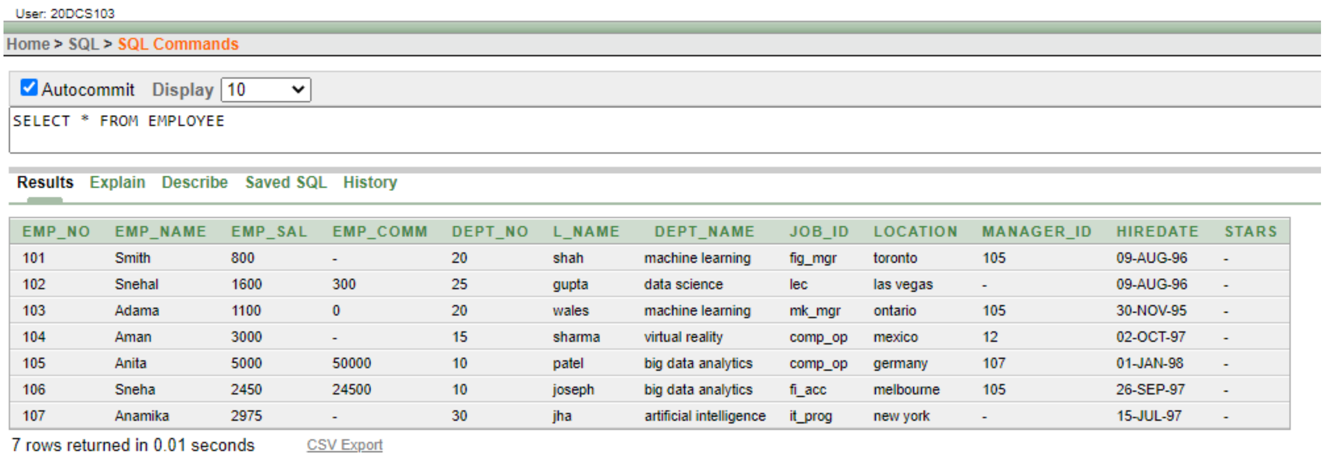
```
DECLARE
ESAL NUMBER;
ENO NUMBER := 101;
ST VARCHAR(20);
ITR NUMBER;

BEGIN
WHILE ENO < 108 LOOP
  SELECT EMP_SAL INTO ESAL FROM EMP WHERE EMP_NO = ENO;
  ITR := CEIL(ESAL/1000);

  FOR I IN 1 .. ITR LOOP
    ST := ST || '*';
  END LOOP;

  UPDATE EMPLOYEE SET STARS = ST WHERE EMP_NO = ENO;
  ST := NULL;
  ENO := ENO+1;

END LOOP;
END;
```



```
ALTER TABLE EMPLOYEE
ADD STARS VARCHAR2(50);
```

```
Table altered.
```

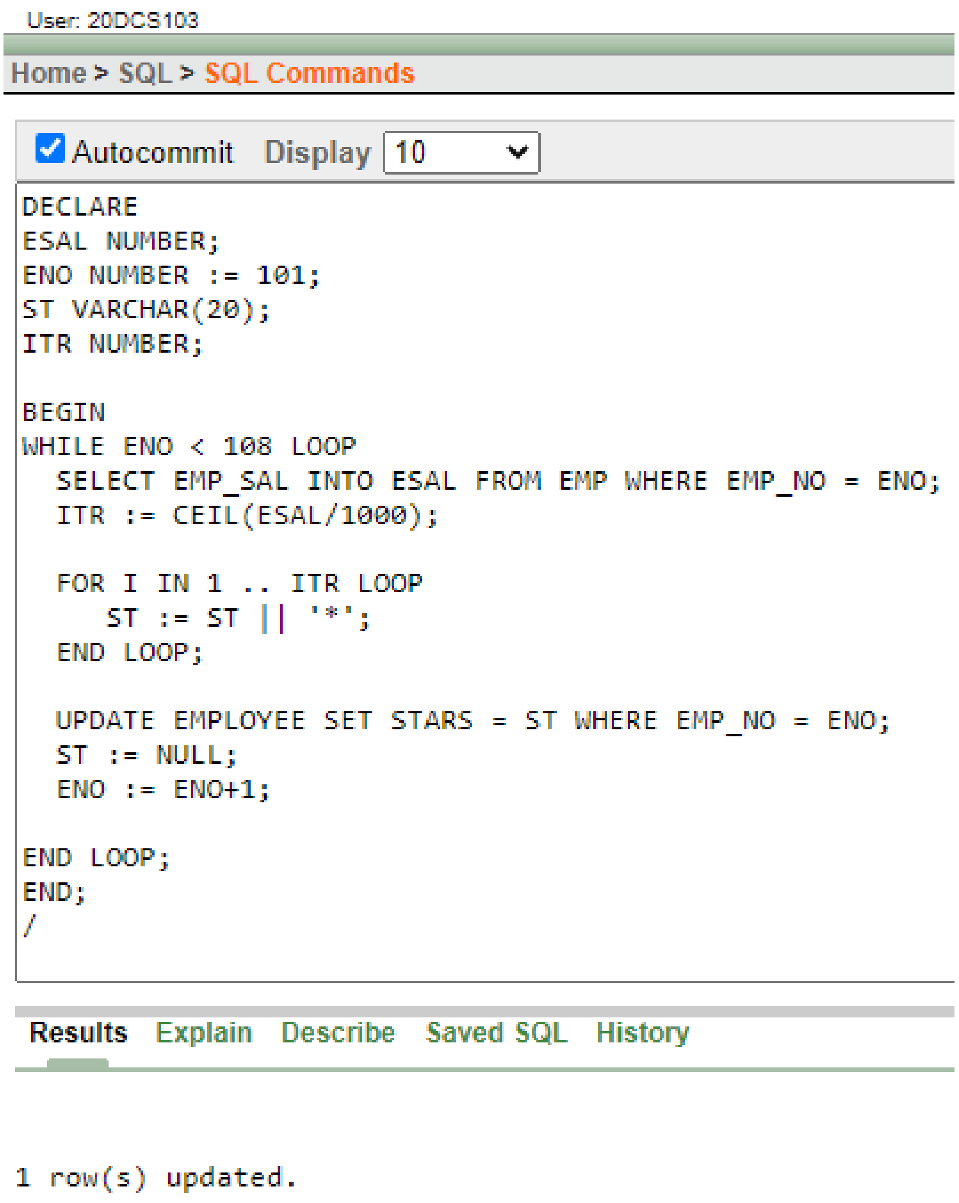
/



```
SELECT * FROM EMPLOYEE
```

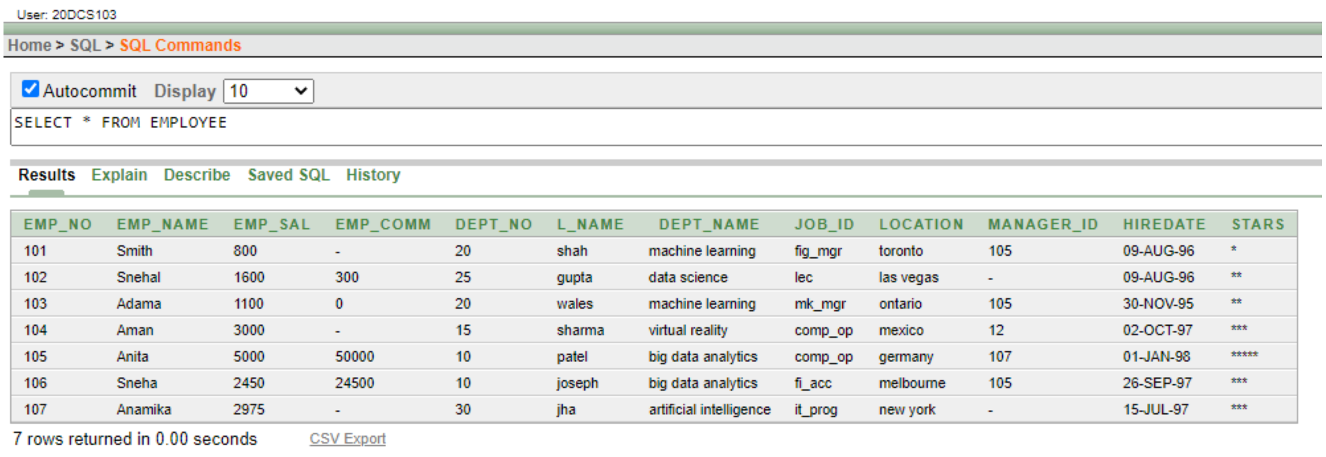
EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	STARS
101	Smith	800	-	20	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	-
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	09-AUG-96	-
103	Adama	1100	0	20	wales	machine learning	mk_mgr	ontario	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	-

7 rows returned in 0.01 seconds


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
ESAL NUMBER;
ENO NUMBER := 101;
ST VARCHAR(20);
ITR NUMBER;

BEGIN
WHILE ENO < 108 LOOP
  SELECT EMP_SAL INTO ESAL FROM EMP WHERE EMP_NO = ENO;
  ITR := CEIL(ESAL/1000);

  FOR I IN 1 .. ITR LOOP
     ST := ST || '*';
  END LOOP;

  UPDATE EMPLOYEE SET STARS = ST WHERE EMP_NO = ENO;
  ST := NULL;
  ENO := ENO+1;

END LOOP;
END;
/
```

```
1 row(s) updated.
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM EMPLOYEE
```

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	STARS
101	Smith	800	-	20	shah	machine learning	fig_mgr	toronto	105	09-AUG-96	*
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	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	***

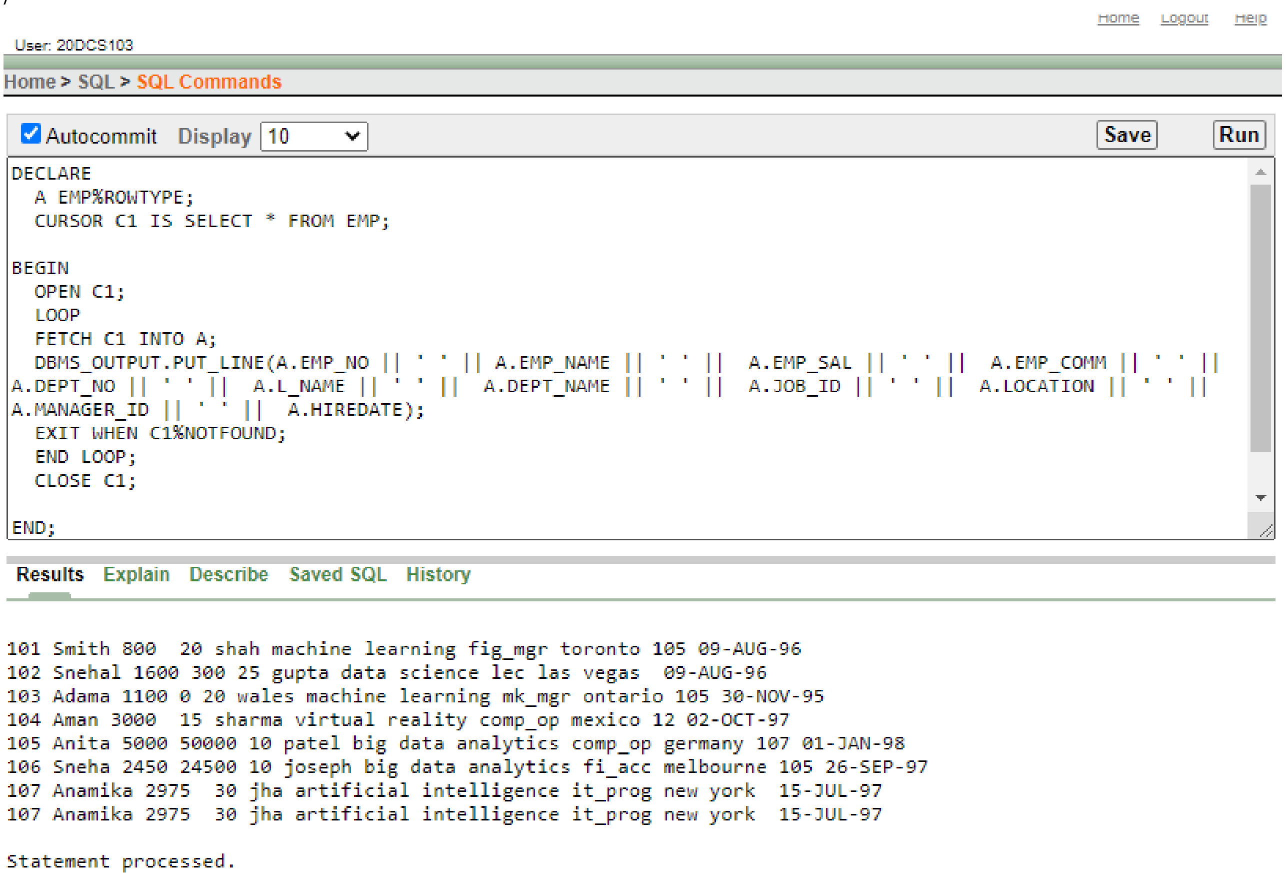
7 rows returned in 0.00 seconds

Aim 16: To perform the concept of cursor.

A) Display all the information of EMP table using %ROWTYPE.

Answer:

```
DECLARE
  A EMP%ROWTYPE;
  CURSOR C1 IS SELECT * FROM EMP;

BEGIN
  OPEN C1;
  LOOP
  FETCH C1 INTO A;
  DBMS_OUTPUT.PUT_LINE(A.EMP_NO || ' ' || A.EMP_NAME || ' ' ||
A.EMP_SAL || ' ' ||  A.EMP_COMM || ' ' ||  A.DEPT_NO || ' ' ||  A.L_NAME || ' ' ||
A.DEPT_NAME || ' ' ||  A.JOB_ID || ' ' ||  A.LOCATION || ' ' ||  A.MANAGER_ID
|| ' ' ||  A.HIREDATE);
  EXIT WHEN C1%NOTFOUND;
  END LOOP;
  CLOSE C1;

END;
/
```


Home Logout Help
User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Save Run
Results Explain Describe Saved SQL History

```
DECLARE
  A EMP%ROWTYPE;
  CURSOR C1 IS SELECT * FROM EMP;

BEGIN
  OPEN C1;
  LOOP
  FETCH C1 INTO A;
  DBMS_OUTPUT.PUT_LINE(A.EMP_NO || ' ' || A.EMP_NAME || ' ' ||  A.EMP_SAL || ' ' ||  A.EMP_COMM || ' ' ||
A.DEPT_NO || ' ' ||  A.L_NAME || ' ' ||  A.DEPT_NAME || ' ' ||  A.JOB_ID || ' ' ||  A.LOCATION || ' ' ||
A.MANAGER_ID || ' ' ||  A.HIREDATE);
  EXIT WHEN C1%NOTFOUND;
  END LOOP;
  CLOSE C1;

END;
```

```
101 Smith 800  20 shah machine learning fig_mgr toronto 105 09-AUG-96
102 Snehal 1600 300 25 gupta data science lec las vegas  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
107 Anamika 2975  30 jha artificial intelligence it_prog new york  15-JUL-97

Statement processed.
```

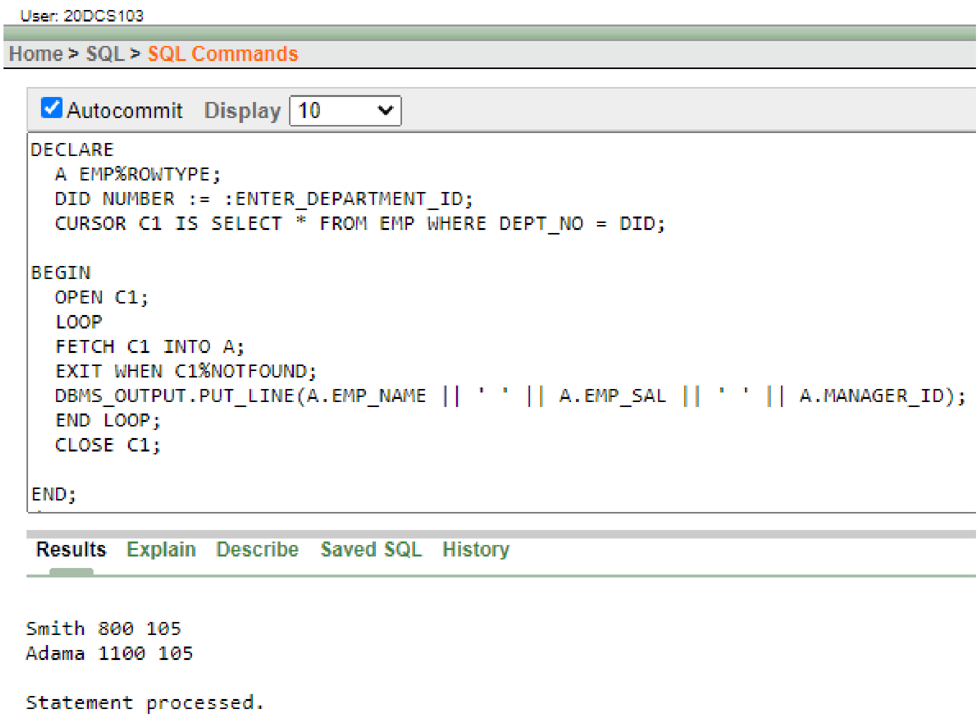
(b) Create a PL/SQL block that does the following:

In a PL/SQL block, retrieve the name, salary, and MANAGER ID of the employees working in the particular department. Take Department Id from user.

If the salary of the employee is less than 1000 and if the manager ID is either 7902 or 7839, display the message <> Due for a raise. Otherwise, display the message <> Not due for a raise.

Answer:

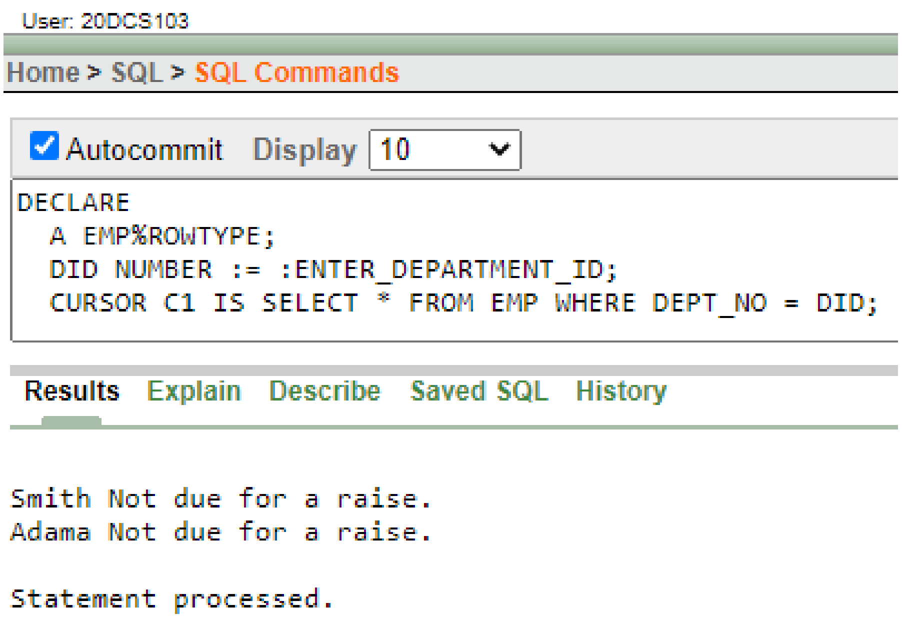
```
DECLARE
  A EMP%ROWTYPE;
  DID NUMBER := :ENTER_DEPARTMENT_ID;
  CURSOR C1 IS SELECT * FROM EMP WHERE DEPT_NO = DID;

BEGIN
  OPEN C1;
  LOOP
  FETCH C1 INTO A;
  EXIT WHEN C1%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(A.EMP_NAME || ' ' || A.EMP_SAL || ' ' ||
A.MANAGER_ID);
  END LOOP;
  CLOSE C1;

END;
/
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
DECLARE
  A EMP%ROWTYPE;
  DID NUMBER := :ENTER_DEPARTMENT_ID;
  CURSOR C1 IS SELECT * FROM EMP WHERE DEPT_NO = DID;

BEGIN
  OPEN C1;
  LOOP
  FETCH C1 INTO A;
  EXIT WHEN C1%NOTFOUND;
  DBMS_OUTPUT.PUT_LINE(A.EMP_NAME || ' ' || A.EMP_SAL || ' ' || A.MANAGER_ID);
  END LOOP;
  CLOSE C1;

END;
```

```
Smith 800 105
Adama 1100 105

Statement processed.
```

```
DECLARE
 A EMP%ROWTYPE;
 DID NUMBER := :ENTER_DEPARTMENT_ID;
 CURSOR C1 IS SELECT * FROM EMP WHERE DEPT_NO = DID;
BEGIN
 OPEN C1;
 LOOP
  FETCH C1 INTO A;
  EXIT WHEN C1%NOTFOUND;
  IF (A.EMP_SAL <1000 AND (A.MANAGER_ID = 7902 OR A.MANAGER_ID = 7839))
THEN
    DBMS_OUTPUT.PUT_LINE(A.EMP_NAME || ' Due for a raise.');
  ELSE
    DBMS_OUTPUT.PUT_LINE(A.EMP_NAME || ' Not due for a raise.');
  END IF;
  END LOOP;
  CLOSE C1;


END;
/
```



```
DECLARE
  A EMP%ROWTYPE;
  DID NUMBER := :ENTER_DEPARTMENT_ID;
  CURSOR C1 IS SELECT * FROM EMP WHERE DEPT_NO = DID;
```

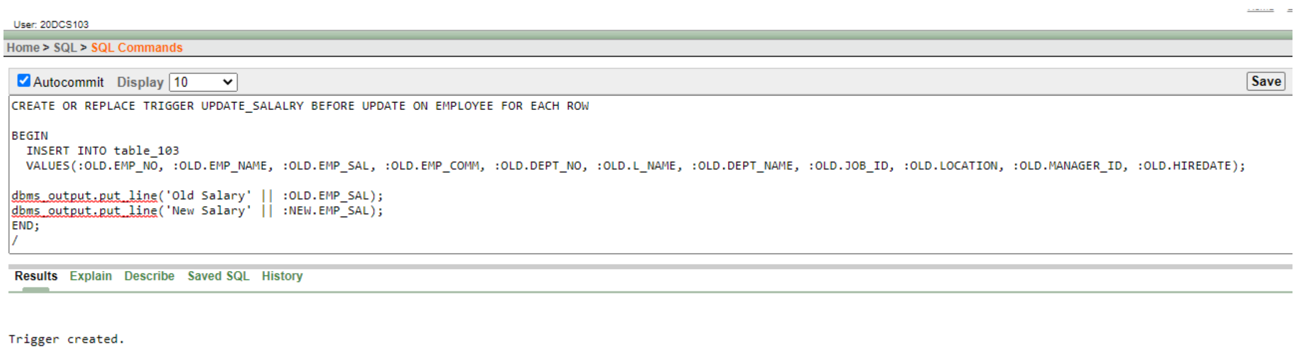
```
Smith Not due for a raise.
Adama Not due for a raise.

Statement processed.
```

Aim 17: To perform the concept of trigger.

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.

Answer:

```
CREATE OR REPLACE TRIGGER UPDATE_SALALRY BEFORE UPDATE ON
EMPLOYEE FOR EACH ROW


BEGIN
  INSERT INTO table_103
  VALUES(:OLD.EMP_NO, :OLD.EMP_NAME, :OLD.EMP_SAL, :OLD.EMP_COMM,
:OLD.DEPT_NO, :OLD.L_NAME, :OLD.DEPT_NAME, :OLD.JOB_ID,
:OLD.LOCATION, :OLD.MANAGER_ID, :OLD.HIREDATE);


dbms_output.put_line('Old Salary' || :OLD.EMP_SAL);
dbms_output.put_line('New Salary' || :NEW.EMP_SAL);
END;
/
```

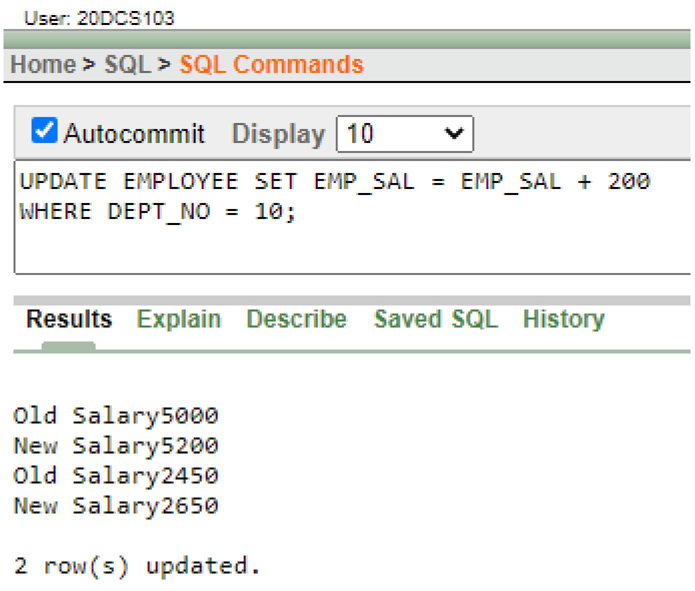

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Save
Results Explain Describe Saved SQL History

```
CREATE OR REPLACE TRIGGER UPDATE_SALALRY BEFORE UPDATE ON EMPLOYEE FOR EACH ROW

BEGIN
  INSERT INTO table_103
  VALUES(:OLD.EMP_NO, :OLD.EMP_NAME, :OLD.EMP_SAL, :OLD.EMP_COMM, :OLD.DEPT_NO, :OLD.L_NAME, :OLD.DEPT_NAME, :OLD.JOB_ID, :OLD.LOCATION, :OLD.MANAGER_ID, :OLD.HIREDATE);

dbms_output.put_line('Old Salary' || :OLD.EMP_SAL);
dbms_output.put_line('New Salary' || :NEW.EMP_SAL);
END;
/
```

```
Trigger created.
```

UPDATE EMPLOYEE SET EMP_SAL = EMP_SAL + 200

WHERE DEPT_NO = 10;



```
UPDATE EMPLOYEE SET EMP_SAL = EMP_SAL + 200
WHERE DEPT_NO = 10;
```

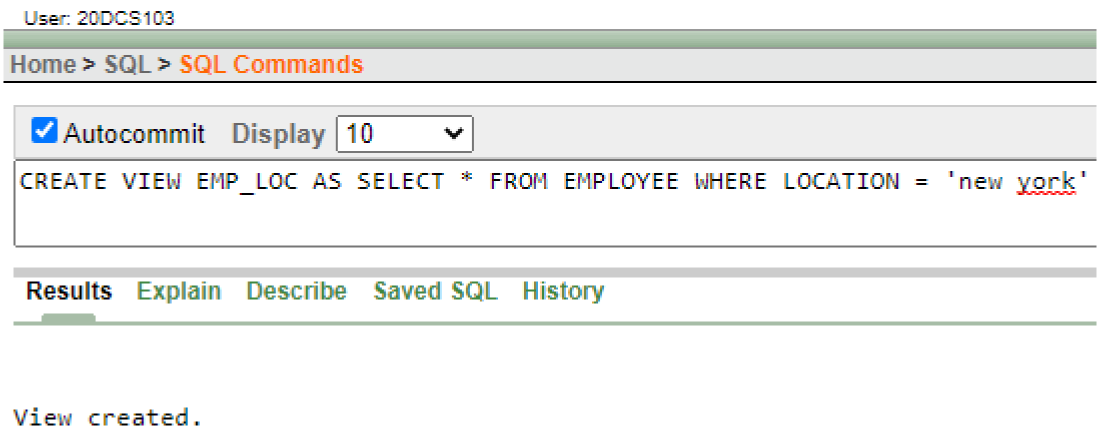
```
Old Salary5000
New Salary5200
Old Salary2450
New Salary2650

2 row(s) updated.
```

Aim 18: To solve the queries using the concept of View.

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

Answer:

CREATE VIEW EMP_LOC AS SELECT * FROM EMPLOYEE WHERE LOCATION = 'new york'


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

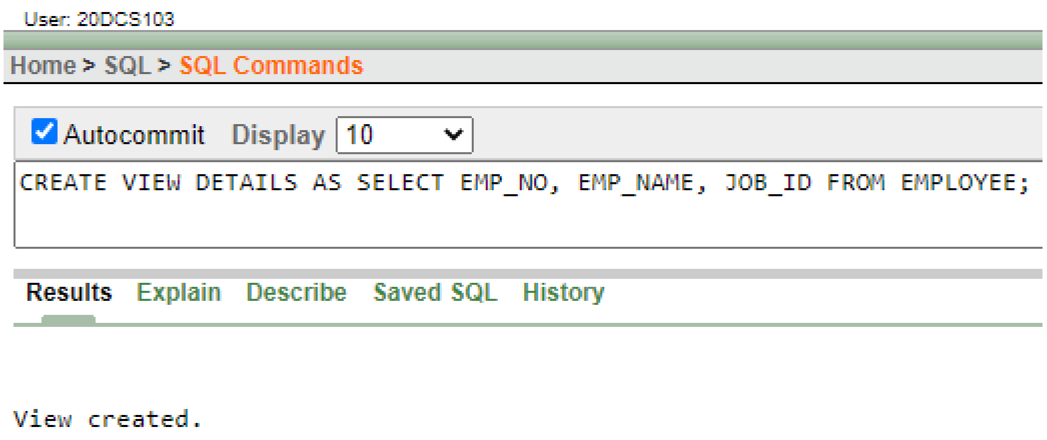
```
CREATE VIEW EMP_LOC AS SELECT * FROM EMPLOYEE WHERE LOCATION = 'new york'
```

```
View created.
```

2) Write a query to create a view for all employee with columns emp_id, emp_name, and job_id.

Answer:

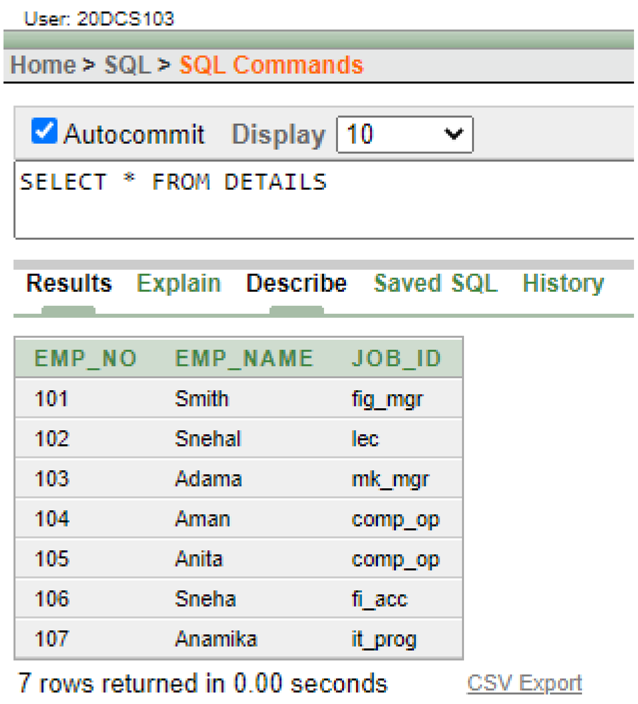
CREATE VIEW DETAILS AS SELECT EMP_NO, EMP_NAME, JOB_ID FROM EMPLOYEE;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE VIEW DETAILS AS SELECT EMP_NO, EMP_NAME, JOB_ID FROM EMPLOYEE;
```

```
View created.
```

SELECT * FROM DETAILS


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM DETAILS
```

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

7 rows returned in 0.00 seconds

3) Write a query to find the salesmen of the location New York who having salary more than 3000.

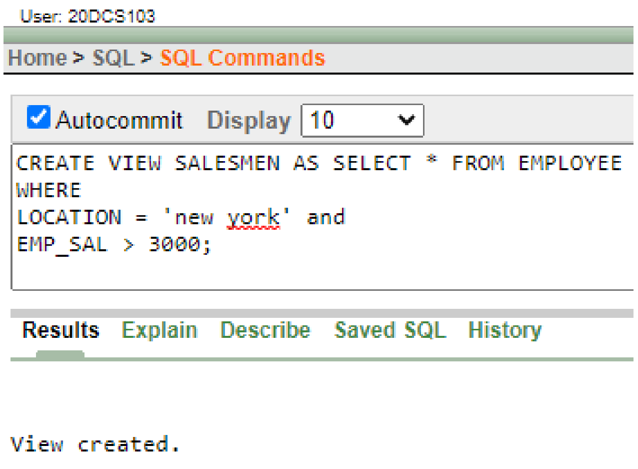
Answer:

CREATE VIEW SALESMEN AS SELECT * FROM EMPLOYEE

WHERE

LOCATION = 'new york' and

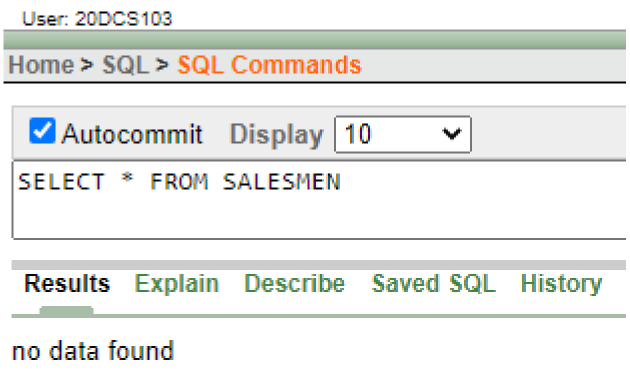
EMP_SAL > 3000;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE VIEW SALESMEN AS SELECT * FROM EMPLOYEE
WHERE
LOCATION = 'new york' and
EMP_SAL > 3000;
```

```
View created.
```

SELECT * FROM SALESMEN


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
SELECT * FROM SALESMEN
```

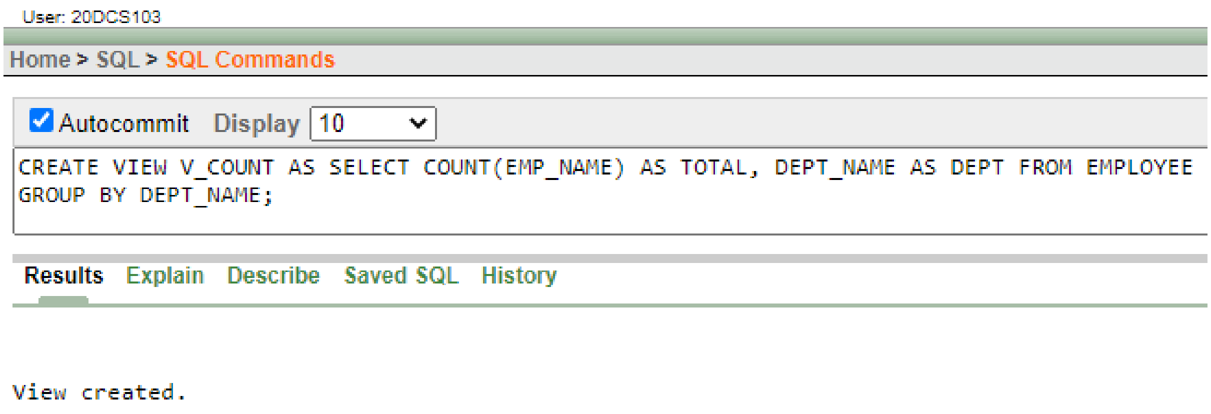
```
no data found
```

4) Write a query to create a view to getting a count of how many employee we have at each department.

Answer:

CREATE VIEW V_COUNT AS SELECT COUNT(EMP_NAME) AS TOTAL, DEPT_NAME AS DEPT FROM EMPLOYEE

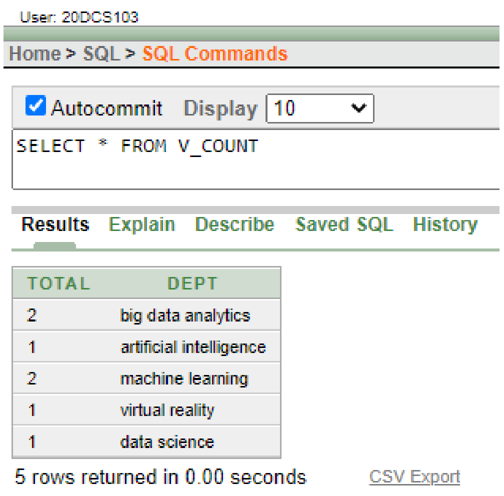
GROUP BY DEPT_NAME;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
CREATE VIEW V_COUNT AS SELECT COUNT(EMP_NAME) AS TOTAL, DEPT_NAME AS DEPT FROM EMPLOYEE
GROUP BY DEPT_NAME;
```

```
View created.
```

SELECT * FROM V_COUNT


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History
CSV Export

```
SELECT * FROM V_COUNT
```

TOTAL	DEPT
2	big data analytics
1	artificial intelligence
2	machine learning
1	virtual reality
1	data science

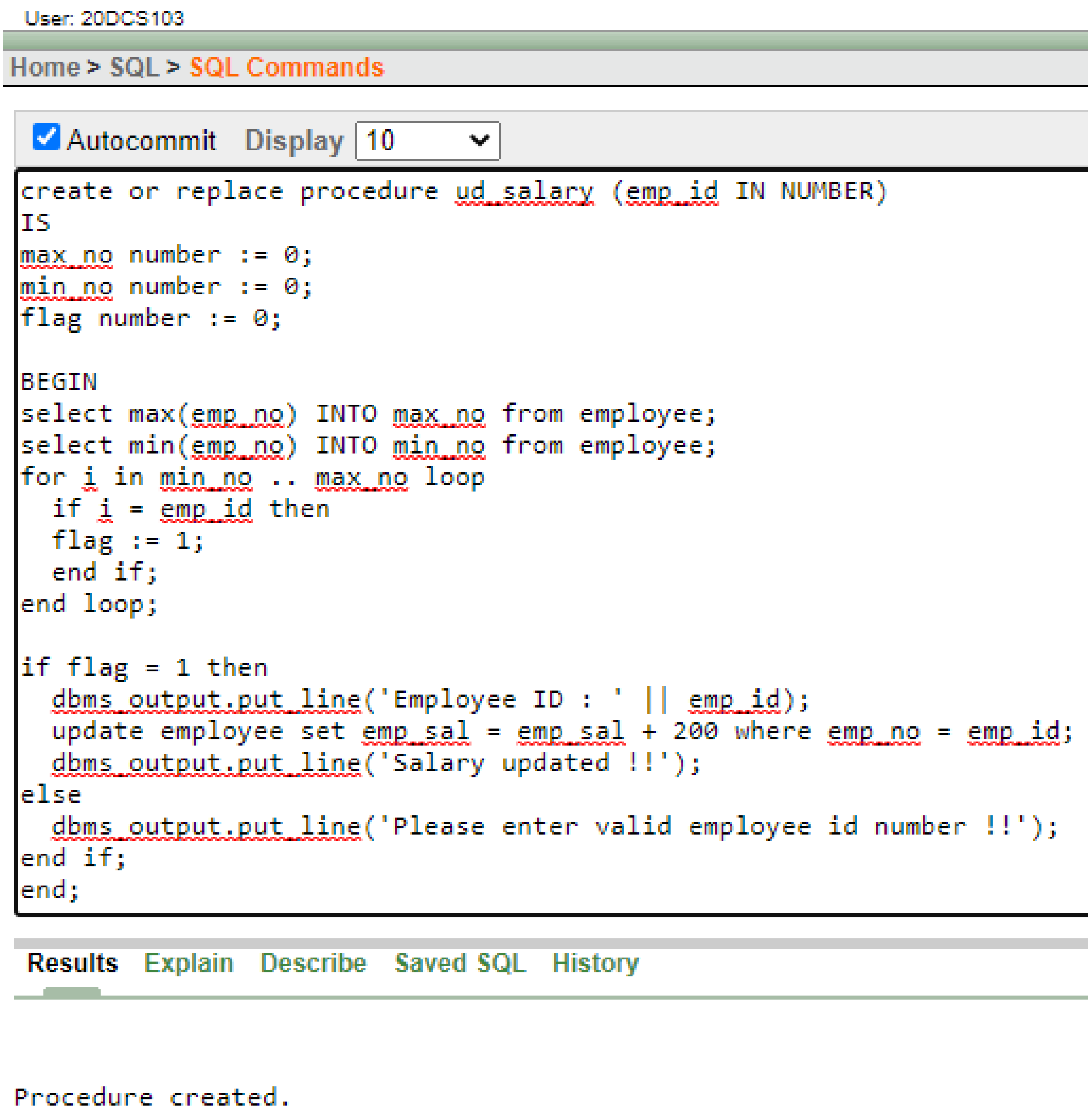
5 rows returned in 0.00 seconds

Aim 19: **To perform the concept of function and procedure.**

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

Answer:

```
create or replace procedure ud_salary (emp_id IN NUMBER)

IS

max_no number := 0;

min_no number := 0;

flag number := 0;


BEGIN

select max(emp_no) INTO max_no from employee;

select min(emp_no) INTO min_no from employee;

for i in min_no .. max_no loop

  if i = emp_id then

  flag := 1;

  end if;

end loop;


if flag = 1 then

  dbms_output.put_line('Employee ID : ' || emp_id);

  update employee set emp_sal = emp_sal + 200 where emp_no = emp_id;

  dbms_output.put_line('Salary updated !!');

else

  dbms_output.put_line('Please enter valid employee id number !!');

end if;

end;
```


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
create or replace procedure ud_salary (emp_id IN NUMBER)
IS
max_no number := 0;
min_no number := 0;
flag number := 0;

BEGIN
select max(emp_no) INTO max_no from employee;
select min(emp_no) INTO min_no from employee;
for i in min_no .. max_no loop
  if i = emp_id then
  flag := 1;
  end if;
end loop;

if flag = 1 then
  dbms_output.put_line('Employee ID : ' || emp_id);
  update employee set emp_sal = emp_sal + 200 where emp_no = emp_id;
  dbms_output.put_line('Salary updated !!');
else
  dbms_output.put_line('Please enter valid employee id number !!');
end if;
end;
```

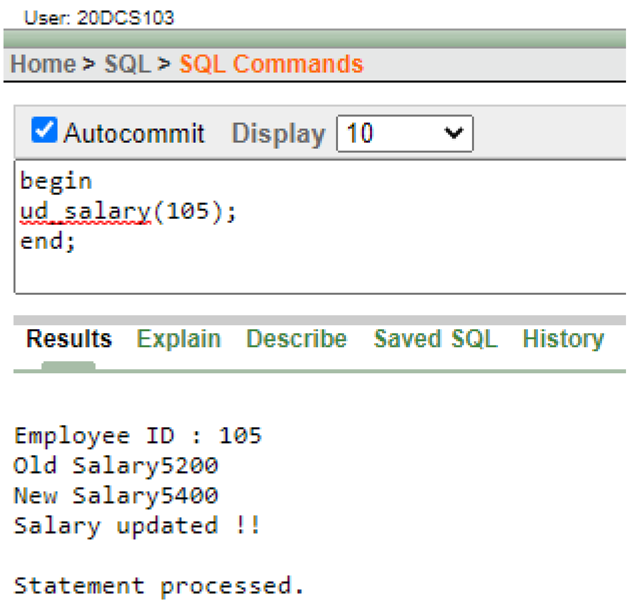
```
Procedure created.
```

Procedure:

begin

ud_salary(105);

end;


User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
begin
ud_salary(105);
end;
```

```
Employee ID : 105
Old Salary5200
New Salary5400
Salary updated !!

Statement processed.
```

Code using function:

create or replace function ud_sal (emp_id IN NUMBER) return number

IS

max_no number := 0;

min_no number := 0;

flag number := 0;

BEGIN

select max(emp_no) INTO max_no from employee;

select min(emp_no) INTO min_no from employee;

for i in min_no .. max_no loop

if i = emp_id then

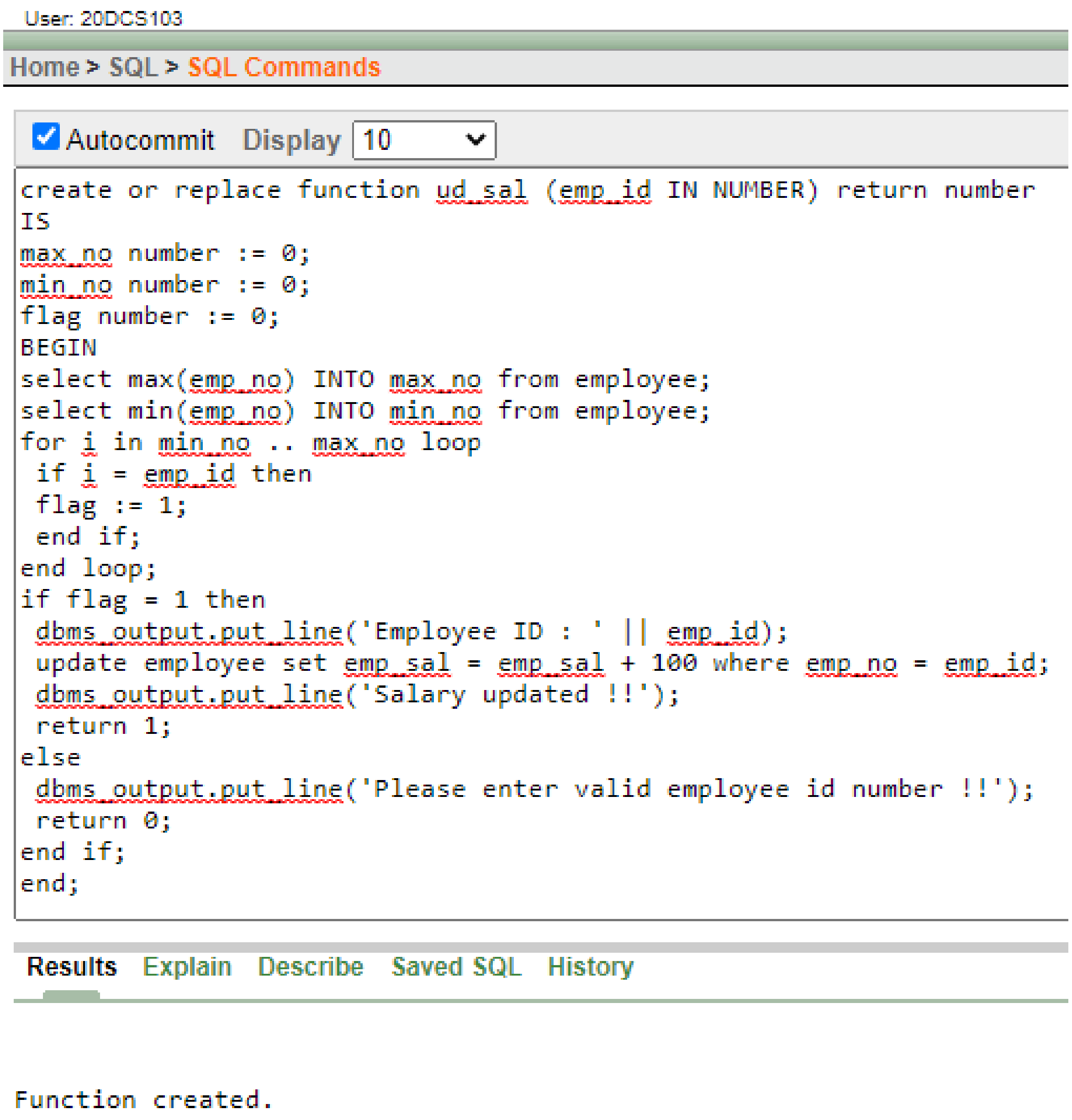
flag := 1;

end if;

end loop;

if flag = 1 then

```
 dbms_output.put_line('Employee ID : ' || emp_id);
 update employee set emp_sal = emp_sal + 100 where emp_no = emp_id;
 dbms_output.put_line('Salary updated !!');
 return 1;
else
 dbms_output.put_line('Please enter valid employee id number !!');
 return 0;
end if;
end;
```

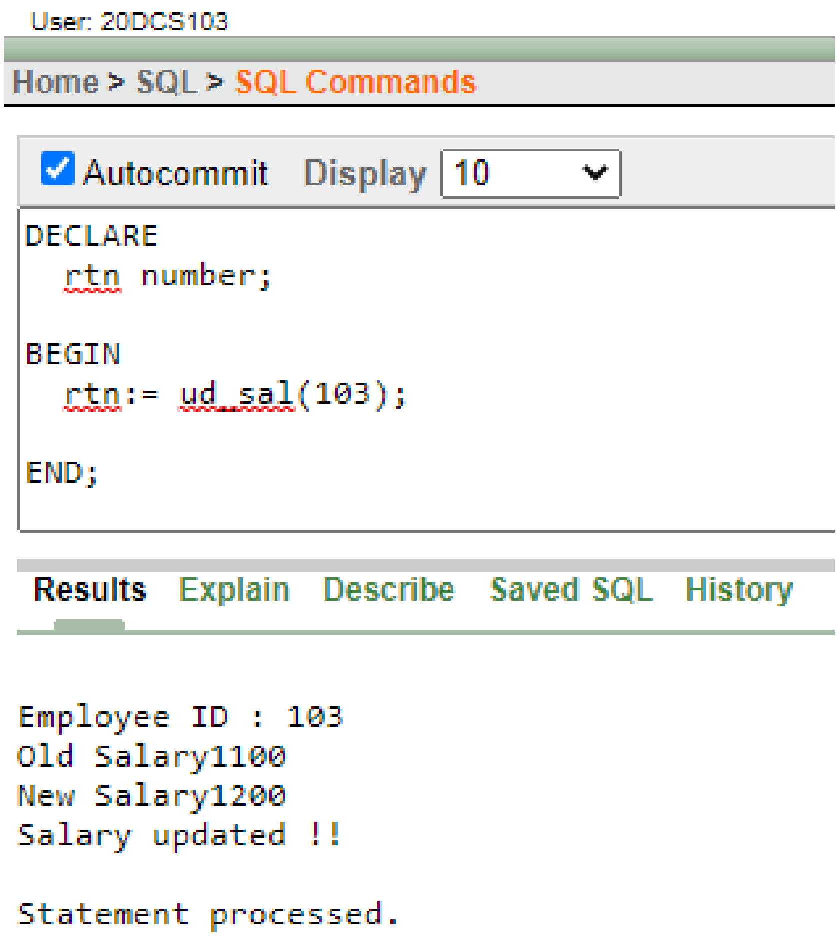


```
create or replace function ud_sal (emp_id IN NUMBER) return number
IS
max_no number := 0;
min_no number := 0;
flag number := 0;
BEGIN
select max(emp_no) INTO max_no from employee;
select min(emp_no) INTO min_no from employee;
for i in min_no .. max_no loop
 if i = emp_id then
 flag := 1;
 end if;
end loop;
if flag = 1 then
 dbms_output.put_line('Employee ID : ' || emp_id);
 update employee set emp_sal = emp_sal + 100 where emp_no = emp_id;
 dbms_output.put_line('Salary updated !!');
 return 1;
else
 dbms_output.put_line('Please enter valid employee id number !!');
 return 0;
end if;
end;
```

```
Function created.
```

Execution of function:

```
DECLARE
  rtn number;
BEGIN
  rtn:= ud_sal(103);
END;
```



```
DECLARE
  rtn number;

BEGIN
  rtn:= ud_sal(103);

END;
```

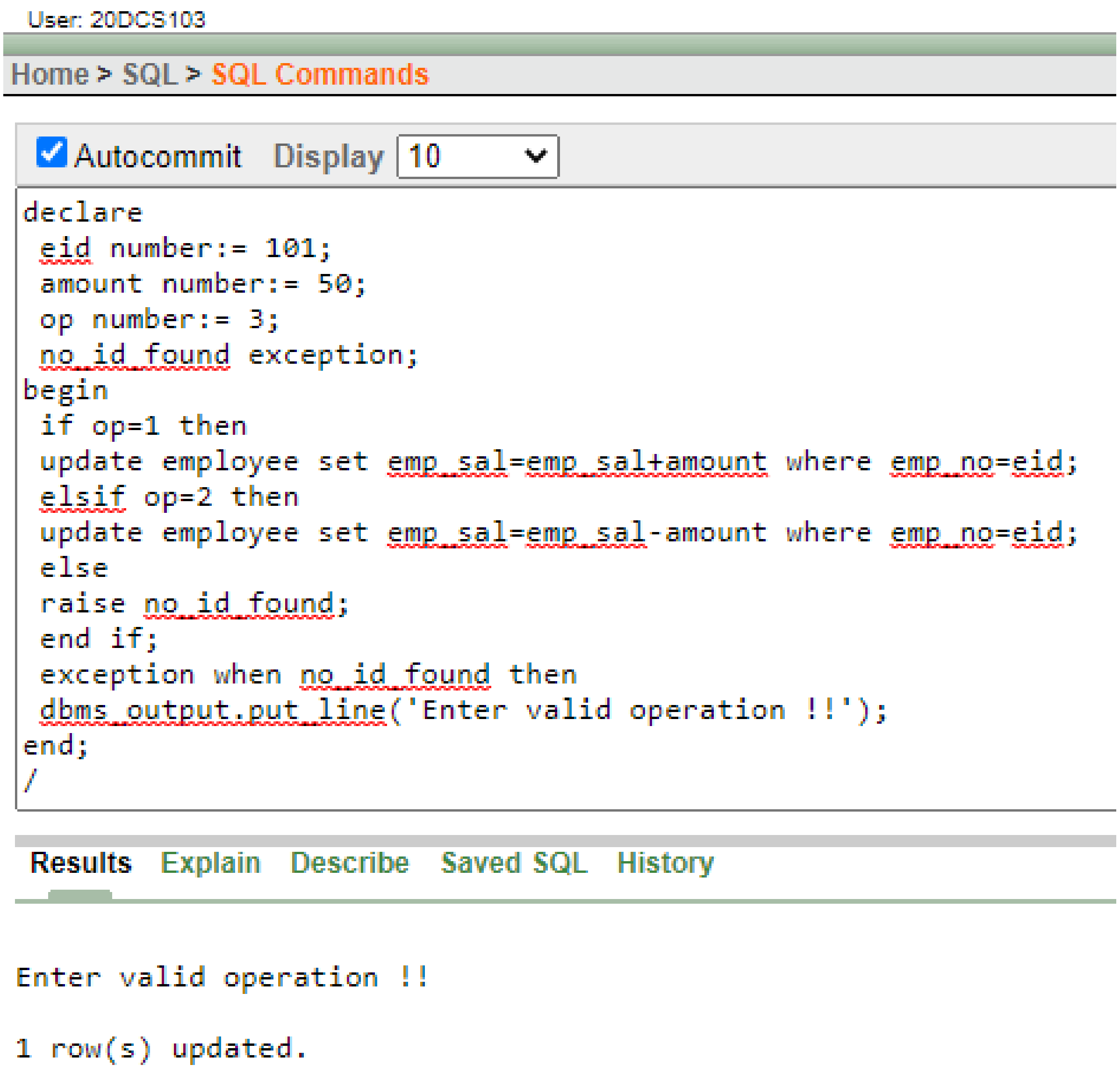
```
Employee ID : 103
Old Salary1100
New Salary1200
Salary updated !!

Statement processed.
```

Aim 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.

Answer:

```
declare

 eid number:= 101;

 amount number:= 50;

 op number:= 3;

 no_id_found exception;

begin

 if op=1 then

 update employee set emp_sal=emp_sal+amount where emp_no=eid;

 elsif op=2 then

 update employee set emp_sal=emp_sal-amount where emp_no=eid;

 else

 raise no_id_found;

 end if;

 exception when no_id_found then

 dbms_output.put_line('Enter valid operation !!');

end;

/
```



```
declare
 eid number:= 101;
 amount number:= 50;
 op number:= 3;
 no_id_found exception;
begin
 if op=1 then
 update employee set emp_sal=emp_sal+amount where emp_no=eid;
 elsif op=2 then
 update employee set emp_sal=emp_sal-amount where emp_no=eid;
 else
 raise no_id_found;
 end if;
 exception when no_id_found then
 dbms_output.put_line('Enter valid operation !!');
end;
/
```

```
Enter valid operation !!

1 row(s) updated.
```

Aim 21: **To perform the concept of package.**

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

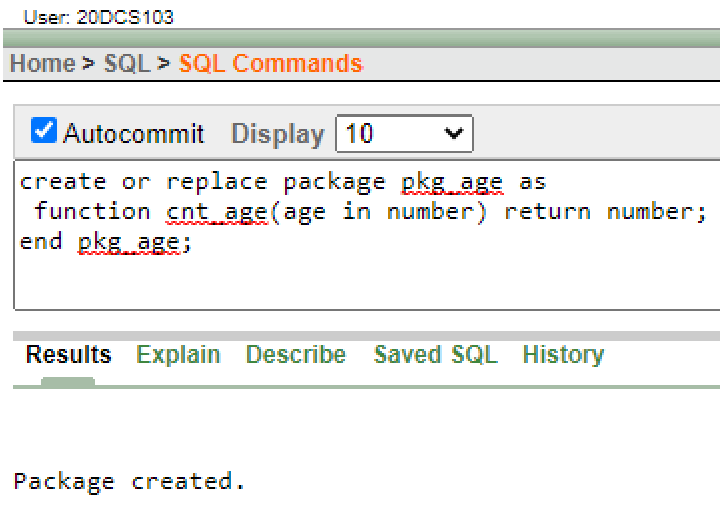
Answer:

Package creation

create or replace package pkg_age as

 function cnt_age(age in number) return number;

end pkg_age;

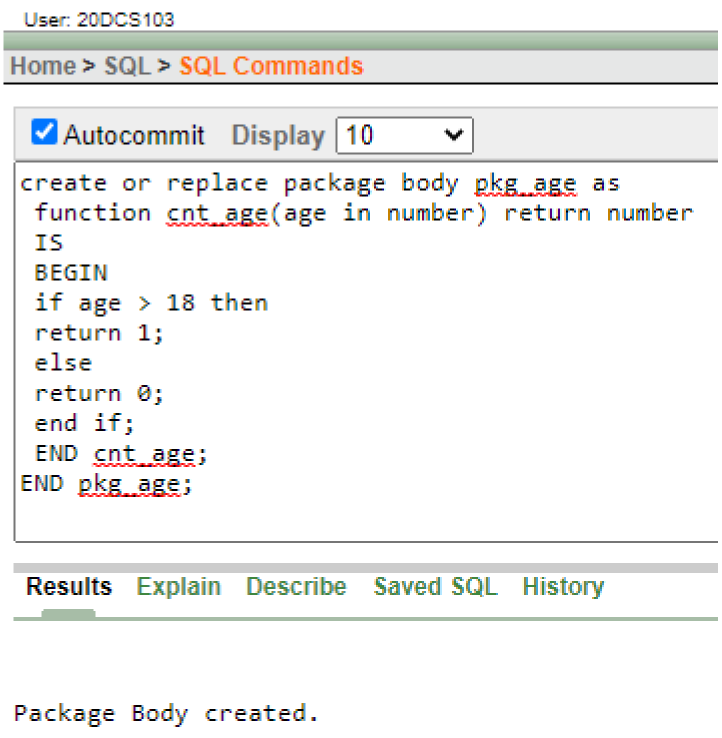


```
create or replace package pkg_age as
 function cnt_age(age in number) return number;
end pkg_age;
```

```
Package created.
```

Package body:

```
create or replace package body pkg_age as

 function cnt_age(age in number) return number

 IS

 BEGIN

 if age > 18 then

 return 1;

 else

 return 0;

 end if;

 END cnt_age;

END pkg_age;
```

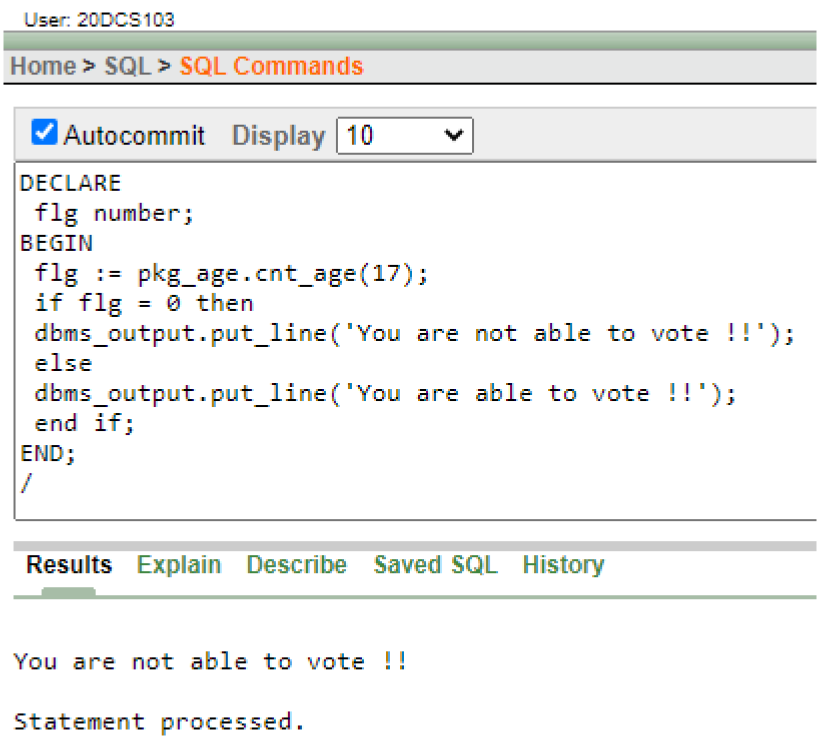

User: 20DCS103
Home > SQL > SQL Commands
Autocommit Display 10
Results Explain Describe Saved SQL History

```
create or replace package body pkg_age as
 function cnt_age(age in number) return number
 IS
 BEGIN
 if age > 18 then
 return 1;
 else
 return 0;
 end if;
 END cnt_age;
END pkg_age;
```

```
Package Body created.
```

Package execution:

```
DECLARE
 flg number;
BEGIN
 flg := pkg_age.cnt_age(17);
 if flg = 0 then
 dbms_output.put_line('You are not able to vote !!');
 else
 dbms_output.put_line('You are able to vote !!');
 end if;
END;
/
```



```
DECLARE
 flg number;
BEGIN
 flg := pkg_age.cnt_age(17);
 if flg = 0 then
 dbms_output.put_line('You are not able to vote !!');
 else
 dbms_output.put_line('You are able to vote !!');
 end if;
END;
/
```

```
You are not able to vote !!

Statement processed.
```