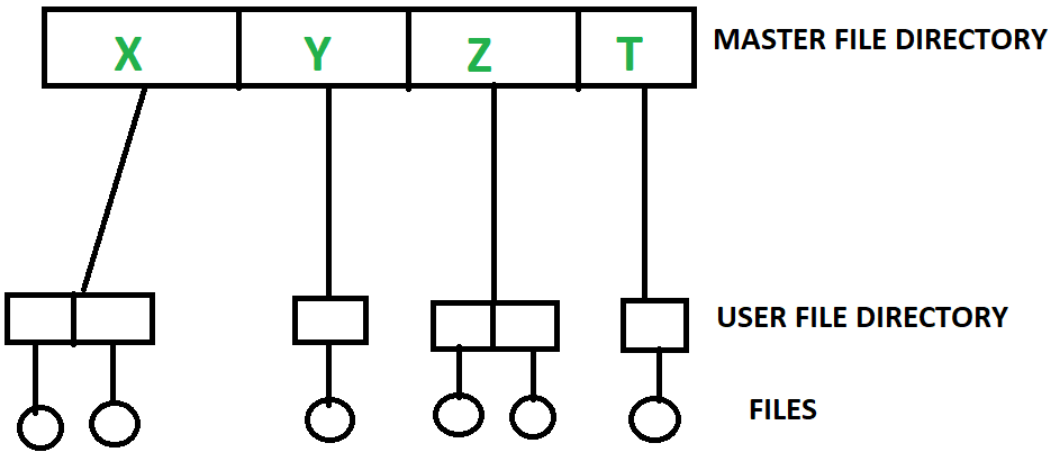
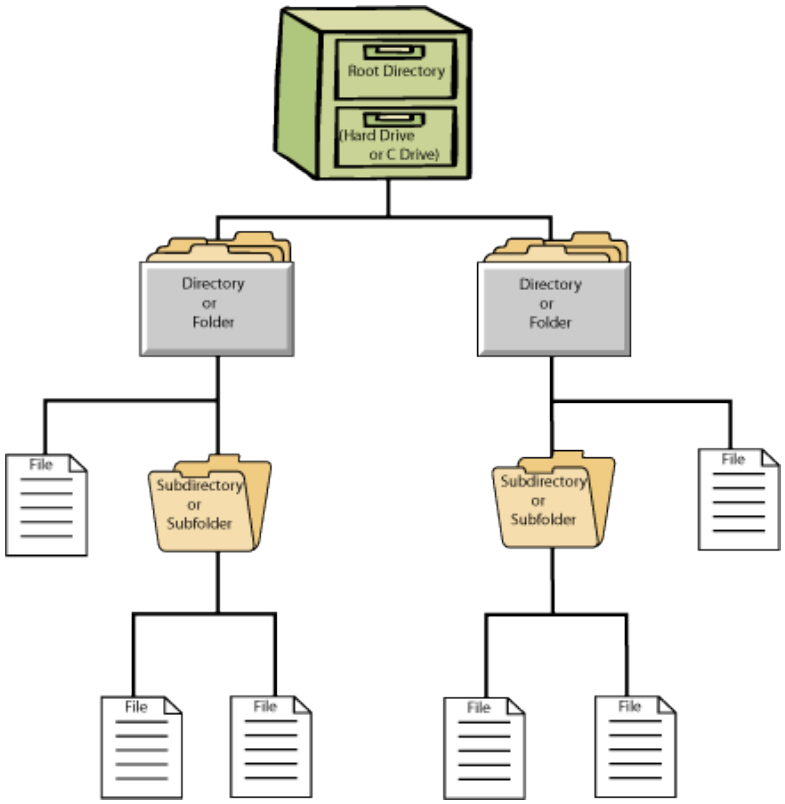
Practical 1

Aim: Explain the following terms: File system, DBMS, RDBMS, DDBMS.

File System:

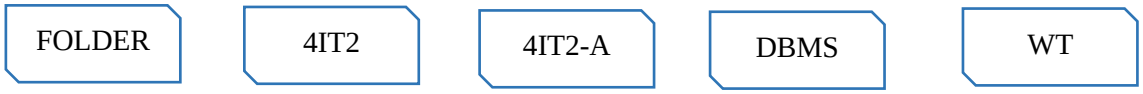
- file system or filesystem is a method and data structure that the operating system uses to control how data is stored and retrieved.
- Without a file system, data placed in a storage medium would be one large body of data with no way to tell where one piece of data stopped and the next began, or where any piece of data was located when it was time to retrieve it.
- File system is a software that manages and organizes the files in a storage medium within a computer.
- File system is basically a way of arranging the files in a storage medium like hard disk.
- File system organizes the files and helps in retrieval of files when they are required.
- File systems consists of different files which are grouped into directories. The directories further contain other folders and files.
- File system performs basic operations like management, file naming, giving access rules etc.
- Example:
  NTFS (New Technology File System), EXT (Extended File System).





EXAMPLE:

If I create the different sub folders in the given drive with different name, I can access the data or information and perform the certain operations in the system and this system is known as the file system.



ADVANTAGES:

Cost Effective : File management system is extremely cost effective since it is a digital filing system. Whatever the type of document that needs to be stored as a paper can be in the form of digital. Therefore, there is no cost involved in building rents, purchasing cabinets and physical papers.

Security : The documents stored in the file management system is protected using authentication methods like username and password. Additionally, all the data is encrypted which makes sure a document is confidential.

Data Sharing : FMS allows very efficient way of sharing data with each and every person. The same data that is stored on files can be shared with multiple users simultaneously.

Reliability : The data that is stored in a file management system is far more reliable than physically storing it using papers and files. Unlike traditional methods of storing data, files here is very less likely to undergo damage or destruction. Any damages from nature or handling can be completely avoided in a file management system.

Data Backup : In case of a failure, file management system provides a easiest way for backing up data. For this purpose, computers on default offer functionalities.

DISADVANTAGES:

Redundancy: Redundancy is a kind of duplication that occurs if the same type of information exists in different locations. In this event, there is a possibility of memory wastage to take place resulting in higher storage costs.

Inconsistency : The same copies of data located in different places contain different values. For preventing this, there should be paper listing among different files.

Data isolation : If the data is stored in different locations, this could essentially mean that they are isolated in file management system. Under these circumstances, the formats of each file can vary significantly. As a result, extracting data from files can be difficult as it requires complex programming.

Accessibility : Accessing data in file management system is not an easy process. It is not convenient as it should be. Whenever a user needs to access an information using different approaches, they must execute a special program.

Data duplication : Since data is stored in more than one location, there is a possibility of data duplication to take place. If file management system undergoes data duplication it will cause problems in the storage space. These duplications are difficult to correct due to the fact that they are independent to each other. Hence, it requires manual correction which can take time and effort.

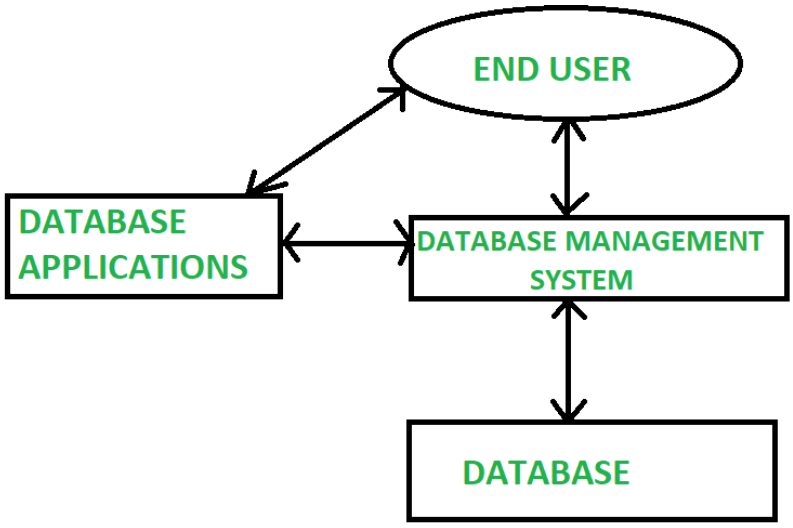
Applications :

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

- Helps you to update the metadata.
- Managing directories.

Database Management System (DBMS):

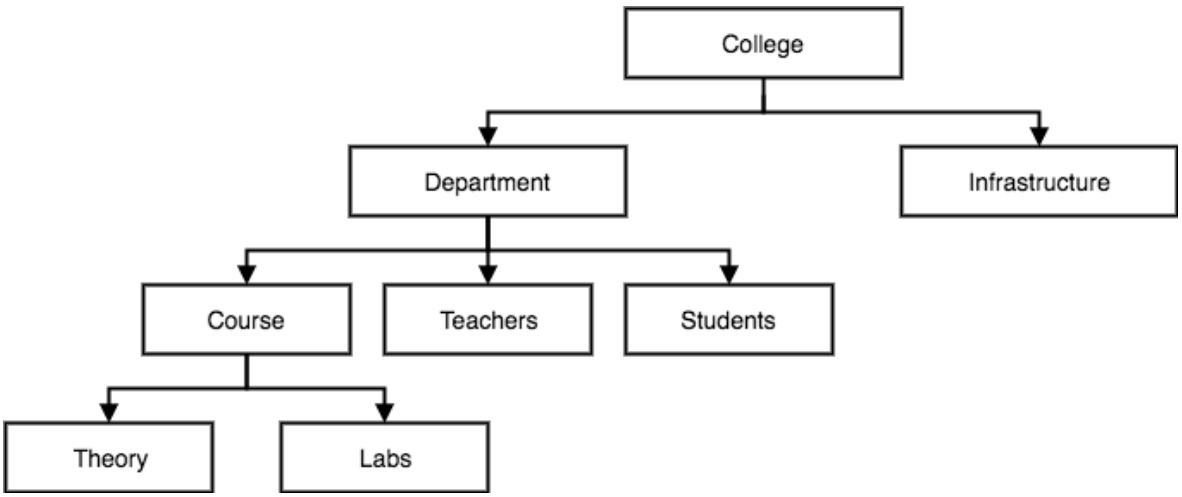
- Data - Fact that can be recorded or stored.
- Database - Collection of logically related data.
- Management - Manipulation, Searching and Security of data.
- System - Programs or tools used to manage database.
- DBMS - A Database Management System is a software for creating and managing databases.
- Database Management System (DBMS) is a software designed to define, manipulate, retrieve and manage data in a database.
- Database Management System is used for storing data and retrieving the data effectively when it is needed.
- Database Management System also provides proper security measures for protecting the data from unauthorized access.
- In Database Management System the data can be fetched by SQL queries and relational algebra.
- Database Management System also provides mechanisms for data recovery and data backup.
- Example:
  Oracle, MySQL, MS SQL server.



EXAMPLE:

If I am making any websites or application, use the “database” for storing and managing the information such as MYSQL, HTML, FIREBASE.

If the system of the class is to be managed then I will be using the ID and name and using this two I will be able to get all the other information such as result, attendance, standard and fees etc.



ADVANTAGES:

Controls database redundancy : It can control data redundancy because it stores all the data in one single database file and that recorded data is placed in the database.

Data sharing : In DBMS, the authorized users of an organization can share the data among multiple users.

Easily Maintenance : It can be easily maintainable due to the centralized nature of the database system.

Reduce time : It reduces development time and maintenance need.

Backup : It provides backup and recovery subsystems which create automatic backup of data from hardware and software failures and restores the data if required.

Multiple user interface : It provides different types of user interfaces like graphical user interfaces, application program interfaces

DISADVANTAGES:

Cost of Hardware and Software : It requires a high speed of data processor and large memory size to run DBMS software.

Size : It occupies a large space of disks and large memory to run them efficiently.

Complexity : Database system creates additional complexity and requirements.

Higher impact of failure : Failure is highly impacted the database because in most of the organization, all the data stored in a single database and if the database is damaged due to electric failure or database corruption then the data may be lost forever.

Popular DBMS Software :

- MySQL
- Microsoft Access
- Oracle
- SQLite
- IBM DB2
- Microsoft SQL Server

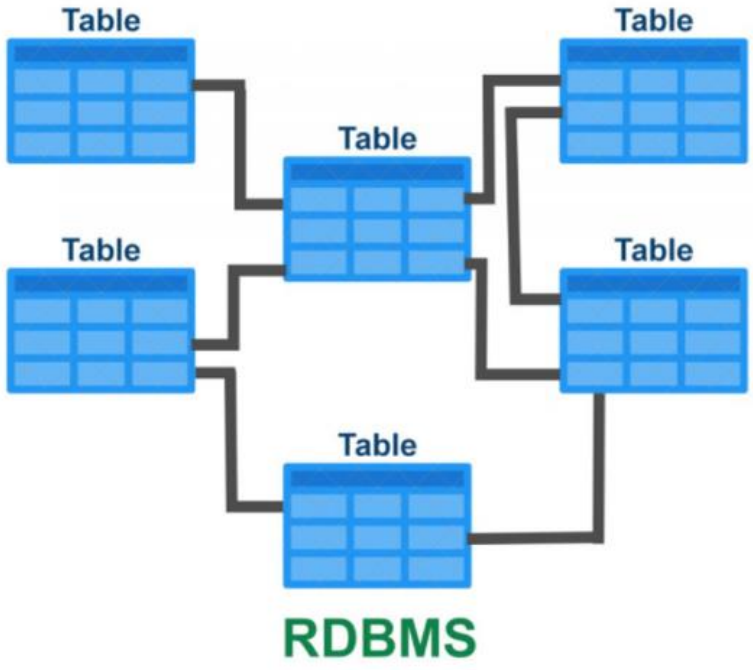
Applications :

Banking : For customer information, account activities, payments, deposits, loans, etc.
Airlines : For reservations and schedule information.
Universities : For student information, course registrations, colleges and grades.
Telecommunication : It helps to keep call records, monthly bills, maintaining balances, etc.

Relational Database Management System (RDBMS):

It is the system which defines the relationships among the available data appearing in a group that is used to store, manage, retrieve and query the data. It performs the CRUD operations (Create, Read, Update and Delete).

Each and Every table will be holding a relationship with the contents of other tables in the system and all the tables in the database system will be associated to the tables with similar properties.



Example :

If I want to fetch the data or information of the relationship among the two tables I can do it with the help of relational database system. I can fetch the details of the given data using the particular information in the table.

ADVANTAGES:

Speed : Even though a relational database is poor in terms of performance, still its speed is considerably higher because of its ease and simplicity. And also, various optimizations that is included in a relational database further increases its speed. So, all the applications will run with appropriate speed when used in a relational database.

Security : Since there are several tables in a relational database, certain tables can be made to be confidential. These tables are protected with username and password such that only authorized users will be able to access them. The users are only allowed to work on that specific table.

Accuracy : As mentioned earlier, relational database uses primary keys and foreign keys to make the tables interrelated to each other. Thus, all the data which is stored is non-repetitive. Which means that the data does not duplicate. Therefore, the data stored can be guaranteed to be accurate.

Multi User : Multiple users will be able to access a relational database at the same time. Even if the data is updated, the users can access them conveniently. Hence, the crashes happening from multi access is possibly prevented.

Simplicity : Compared to other types of network models, a relational database model is much simpler. It is free from query processing and complex structuring. As a result, it does not require any complex queries. A simple SQL query is sufficient enough for handling.

DISADVANTAGES:

Cost : The underlaying cost involved in a relational database is quite expensive. For setting up a relational database, there must be separate software which needs to be purchased.

Performance : Always the performance of the relational database depends on the number of tables. If there are a greater number of tables, the response given to the queries will be slower. Additionally, more data presence not only slows down the machine, it eventually makes it complex to find information. Thus, a relational database is known to be a slower database.

Physical Storage : A relational database also requires tremendous amount of physical memory since it is with rows and columns. Each of the operations depend on separate physical storage. The targeted applications can be made to have maximum physical memory.

Information Loss : Large organizations tends to use more number of number of database systems with more tables. This information can be used to be transferred from one system to another. This could pose a risk of data loss.

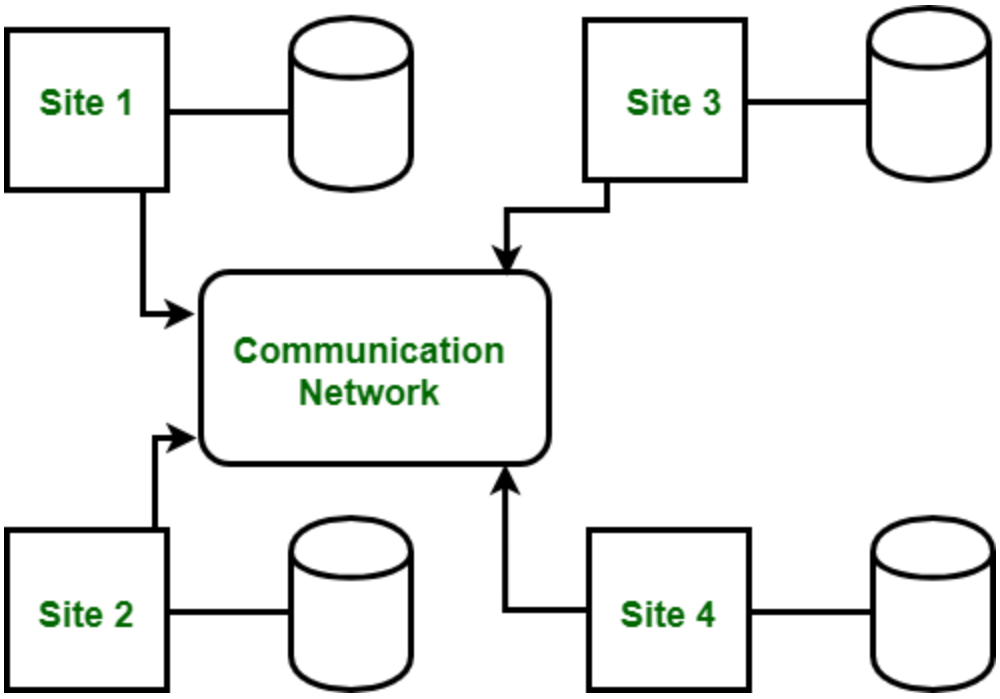
Complexity : Although a relational database is free from complex structuring, occasionally it may become complex too. When the amount of data in a relational database increase, it eventually makes the system more complicated. Each and every data is complex since the data is arranged using common characteristics.

Distributed Database Management System

It is used to manage thew data that are to be distributed among the different computer networks or the different locations. DDBMS is a centralized application that manages a distributed database. This database system synchronizes data periodically and ensures that any change in data made by users is universally updated in the database. 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.

- A Distributed Database Management System (DDBMS) consists of a single logical database that is split into a number of fragments.
- Each fragment is stored on one or more computers under the control of a separate DBMS, with the computers connected by a communications network.
- Each site is capable of independently processing user requests that require access to local data (that is, each site has some degree of local autonomy) and is also capable of processing data stored on other computers in the network.
- Users access the distributed database via applications.

- Applications are classified as those that do not require data from other sites and those that do require data from other sites.
- We require a DDBMS to have at least one global application.
- DDBMS is collection of logically related shared data.
- The data is split into a number of fragments.
- In DDBMS fragments may be replicated. Fragments/replicas are allocated to sites.
- The sites are linked by a communications network.
- The data at each site is under the control of a DBMS.
- The DBMS at each site can handle local applications, autonomously.



ADVANTAGES:

Modular Development : If the system needs to be expanded to new locations or new units, in centralized database systems, the action requires substantial efforts and disruption in the existing functioning. However, in distributed databases, the work simply requires adding new computers and local data to the new site and finally connecting them to the distributed system, with no interruption in current functions.

More Reliable : In case of database failures, the total system of centralized databases comes to a halt. However, in distributed systems, when a component fails, the functioning of the system continues may be at a reduced performance. Hence DDBMS is more reliable.

Better Response : If data communication costs for data manipulation can be minimized. This is not feasible in centralized systems.is distributed in an efficient manner, then user requests can be met from local data itself, thus providing faster response. On the other hand, in centralized systems, all queries have to pass through the central computer for processing, which increases the response time.

Lower Communication Cost : In distributed database systems, if data is located locally where it is mostly used then the cost of the system will be much lesser.

Easier Expandable : The database is easier to expand as it is already spread across multiple systems and it is not too complicated to add a system.

DISADVANTAGES:

Complexity : 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. Complicated and it is difficult to find people with the necessary experience who can manage and maintain it.

Security : 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.

Integrity : 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.

Increased training cost : Training costs are generally higher in a distributed model than they would be in a centralized model, sometimes even to the extent of offsetting

Technological difficulty : Data integrity, transaction management, concurrency control, security, backup, recovery, query optimization, access path selection, and so on, must all be addressed and resolved.

Conclusion :

In this practical, I have learned about the basic of database management system and its type, File System, Data Base Management System, Relational Data Base Management System, Distributed Data Base Management System.

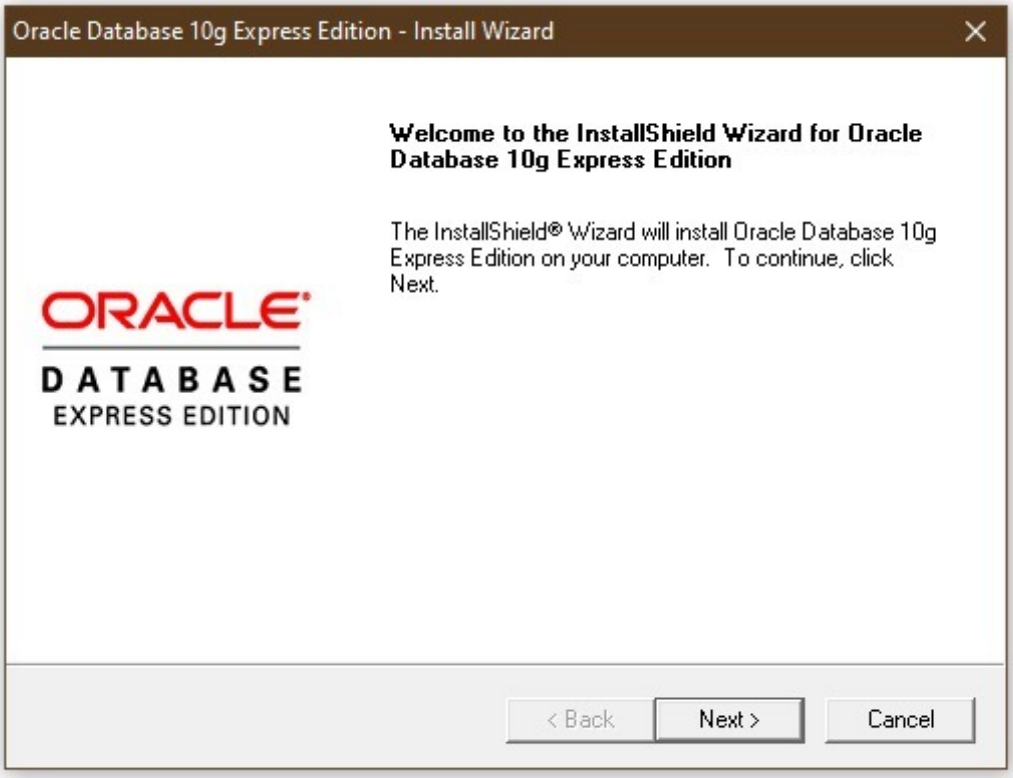
Practical 2

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

Steps:

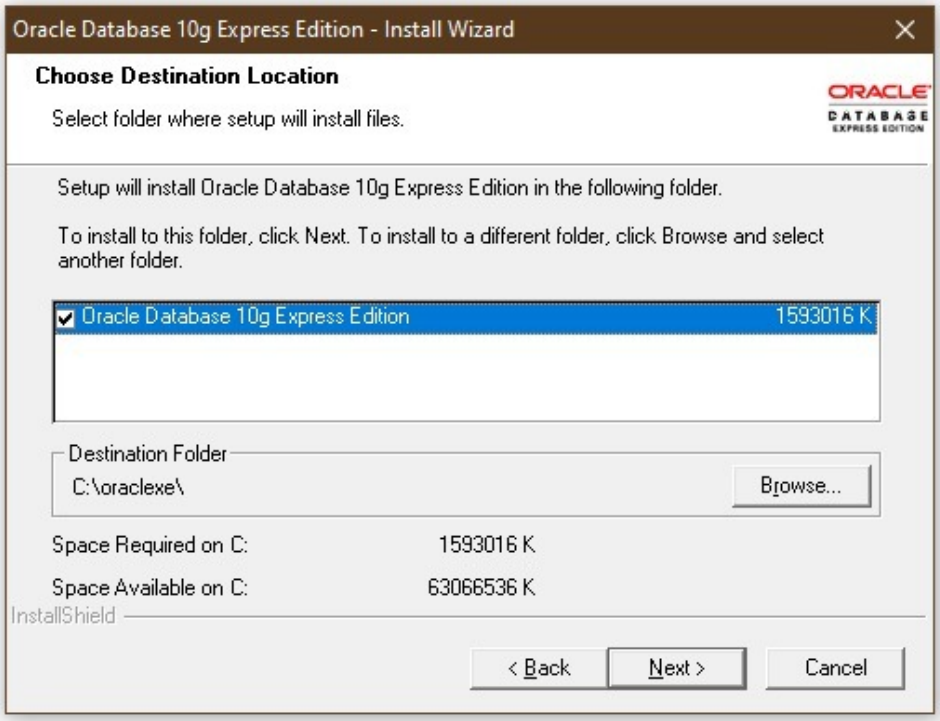
1. Download Oracle 10g from below link:
   https://drive.google.com/file/d/1Y6ghDOEfVorTNrzWgF1UKaENHjeGgrHG/view

2. Start installation by opening the .exe file.



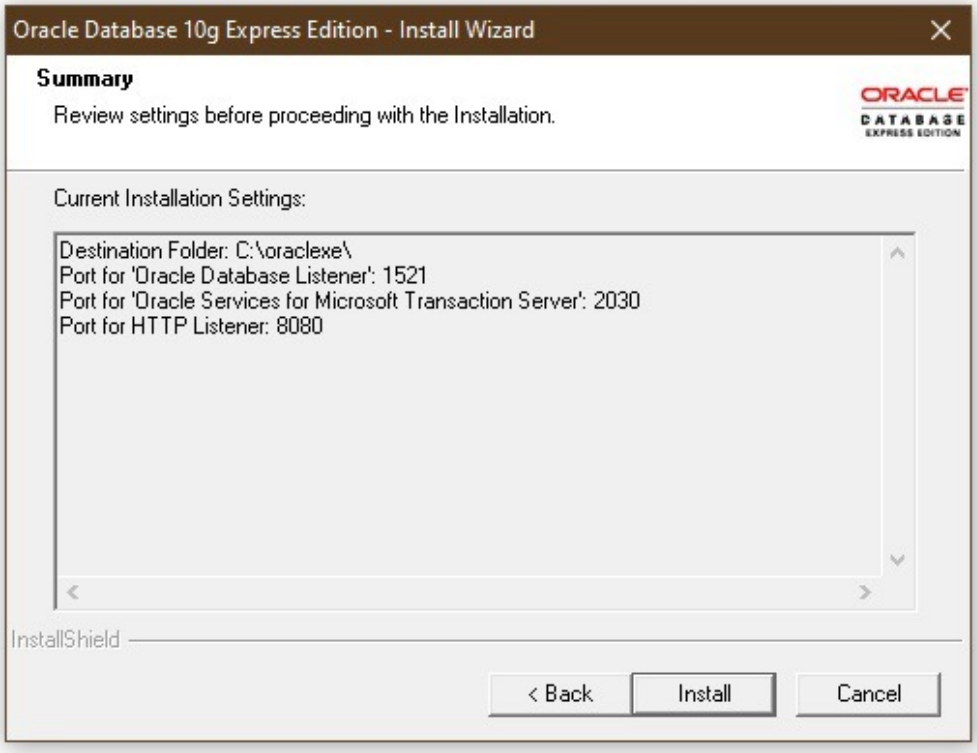
3. Click on next.

4. Select appropriate folder location and click Next.

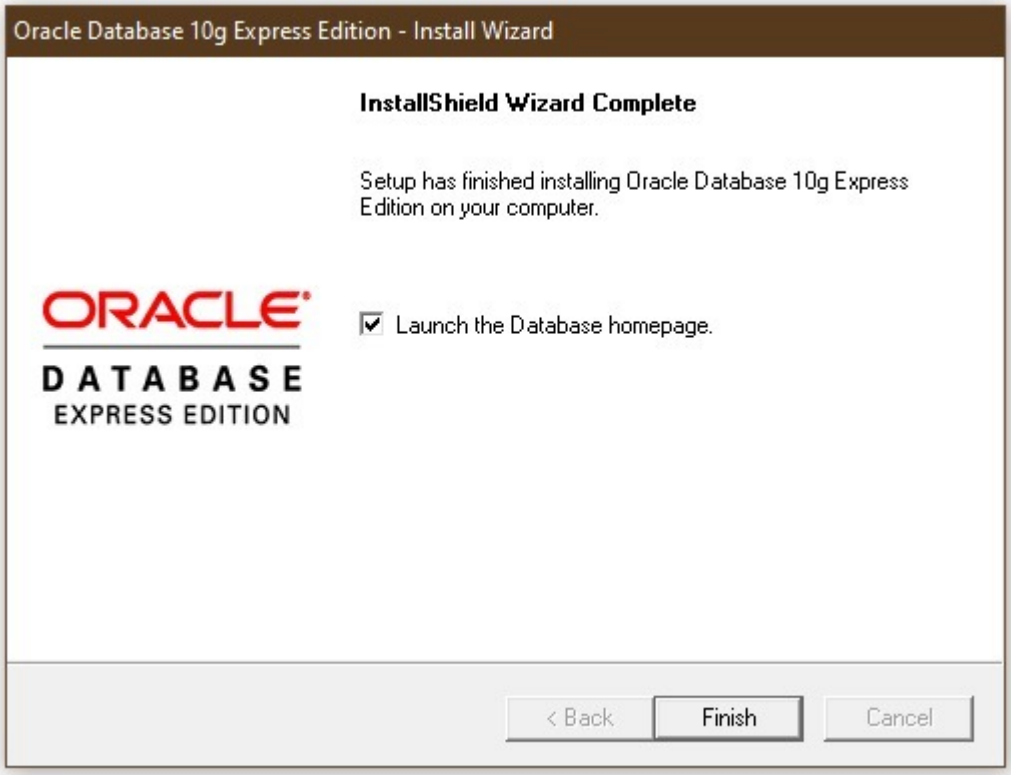


5. Enter a password of your choice and click Next.

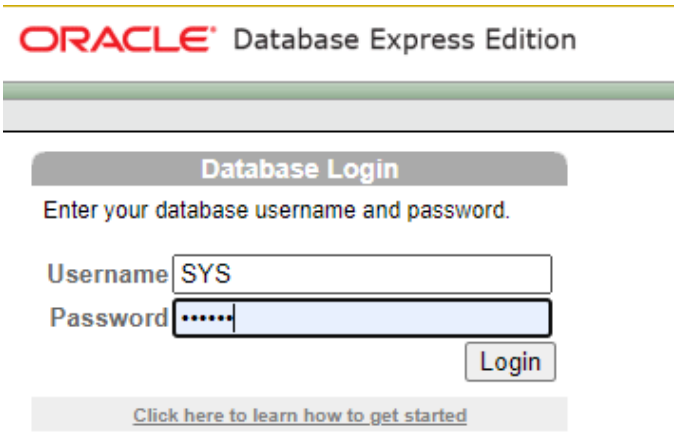
6. Click on Install.



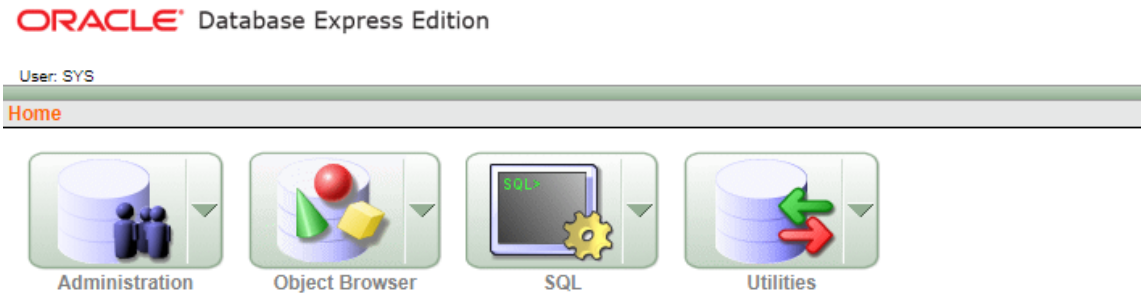
7. Click on finish.



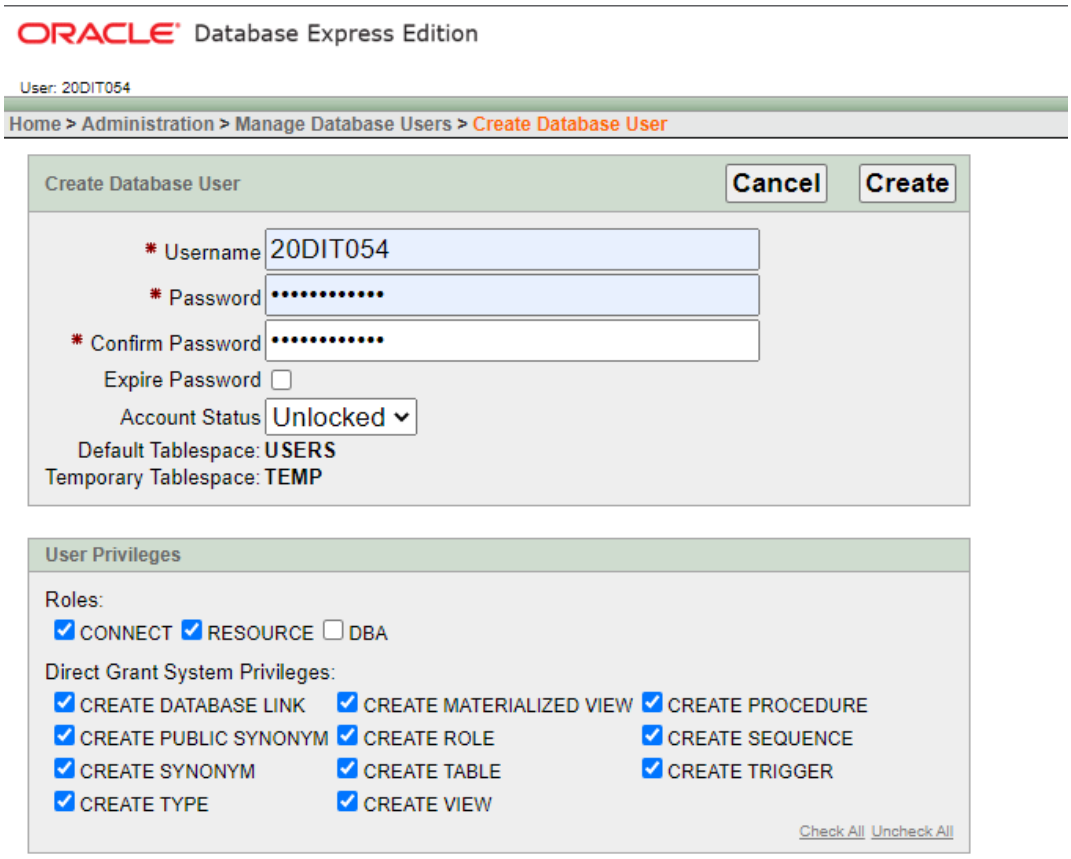
8. Enter username “sys” or “system” and enter password from step 6.



9. Click on Administration.

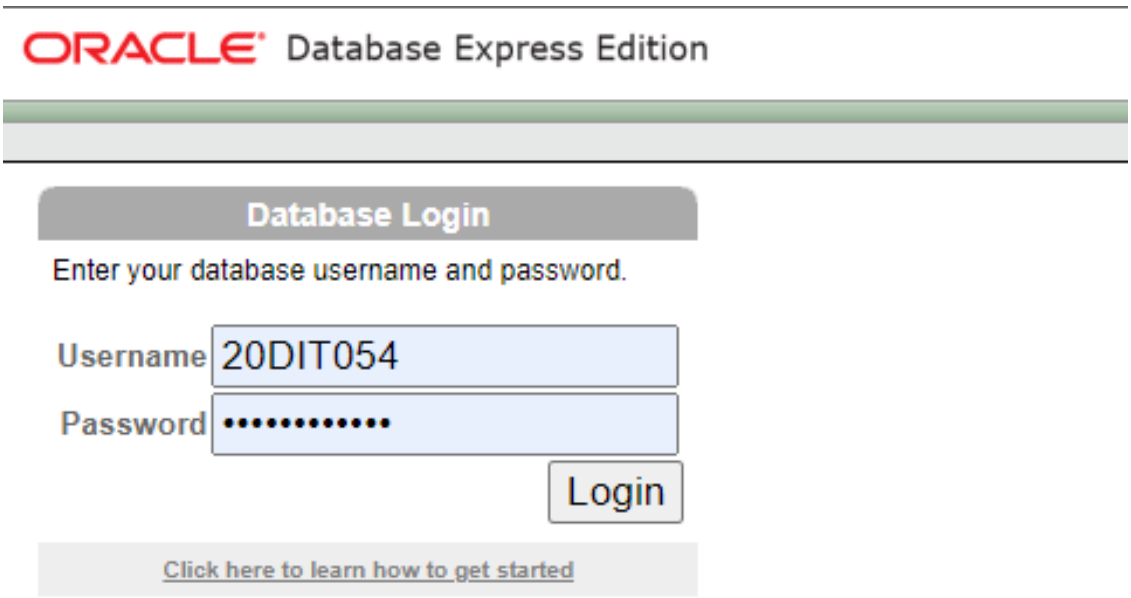


10. Click on “database user dropdown button” and click “create user”. Enter college roll number in username and give password (NEW) and confirm password. Leave expire password unchecked, make account status unblocked if it, give all privileges to user and click on “create” button.

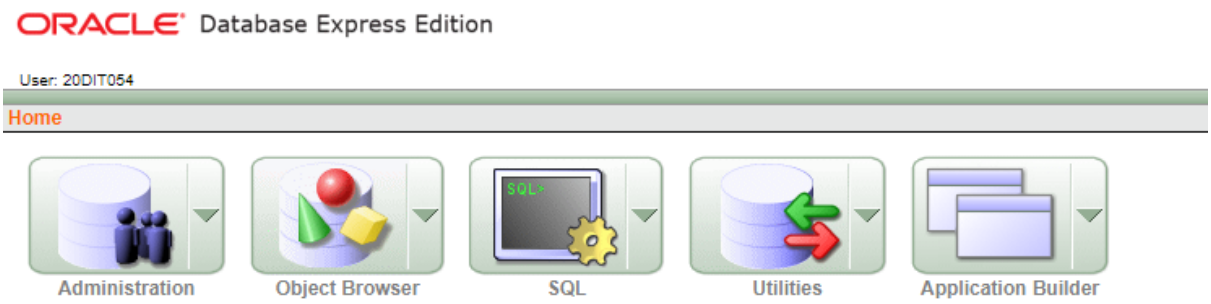


11. Click Logout button.

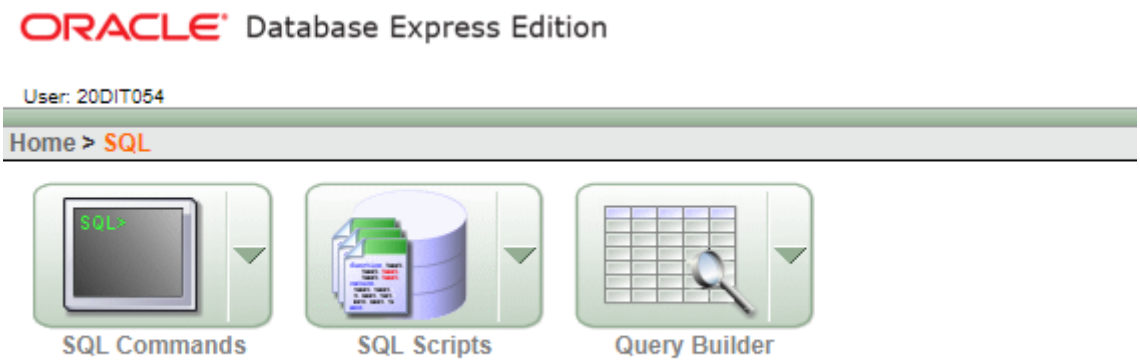
12. Enter username and password that you just created and click on “login” button.



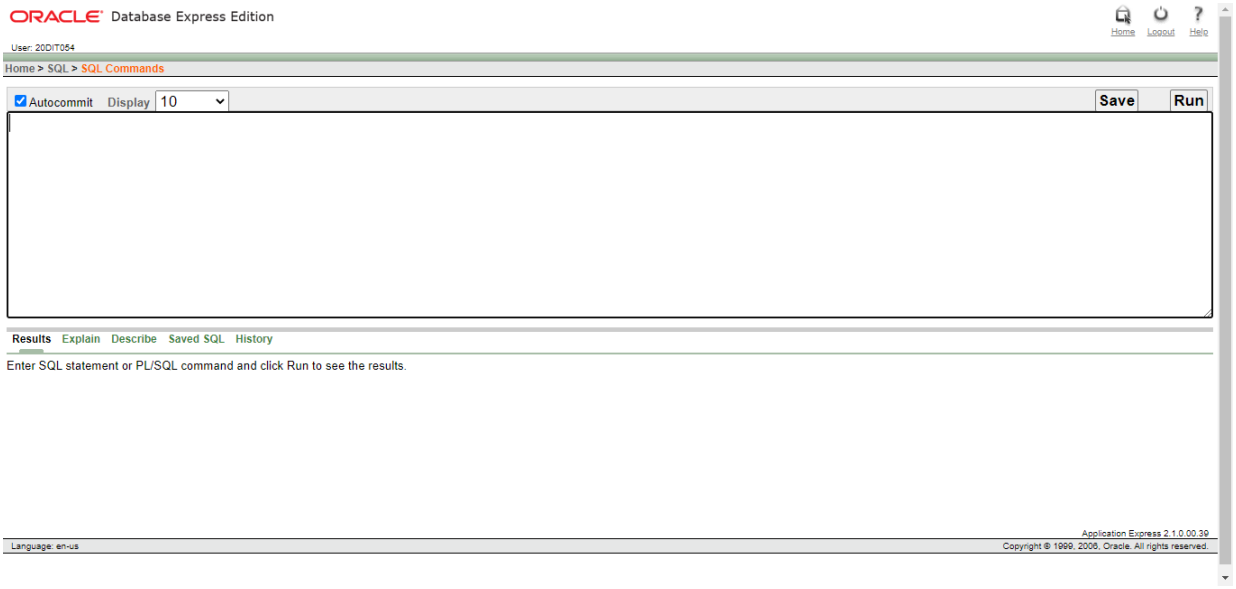
13. Click on SQL.



14. Click on SQL commands.



15. Oracle 10g is now set up and ready to use.



SQL:

- SQL stands for Structured Query Language. It helps in accessing and manipulating databases.
- SQL is a domain-specific language used in programming and designed for managing data held in a relational database management system.
- SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Sybase, Informix, Postgres and SQL Server use SQL as their standard database language.

PL/SQL:

- PL/SQL is a combination of SQL along with the procedural features of programming languages.
- PL/SQL was developed by Oracle Corporation in the early 90's to enhance the capabilities of SQL.
- PL/SQL is one of three key programming languages embedded in the Oracle Database, along with SQL itself and Java.

Conclusion:

From this practical, I learnt how to install Oracle 10g and create a user on it.

Practical-3

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

(i) Create tables according to the following definition.

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

(ii) Insert the data as shown below

DEPOSIT

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

BRANCH

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

CUSTOMERS

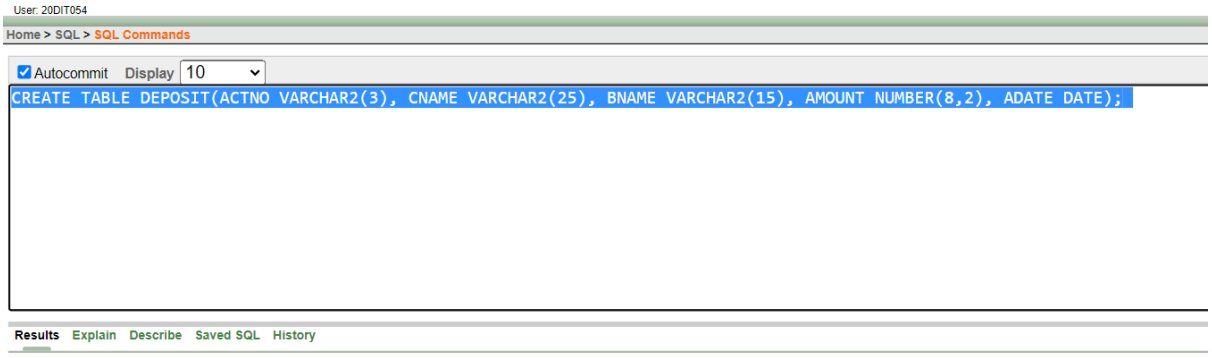
CNAME	CITY
ANIL	CALCUTTA
SUNIL	DELHI
MEHUL	BARODA
MANDAR	PATNA
MADHURI	NAGPUR
PRAMOD	NAGPUR
SANDIP	SURAT
SHIVANI	BOMBAY
KRANTI	BOMBAY
NAREN	BOMBAY

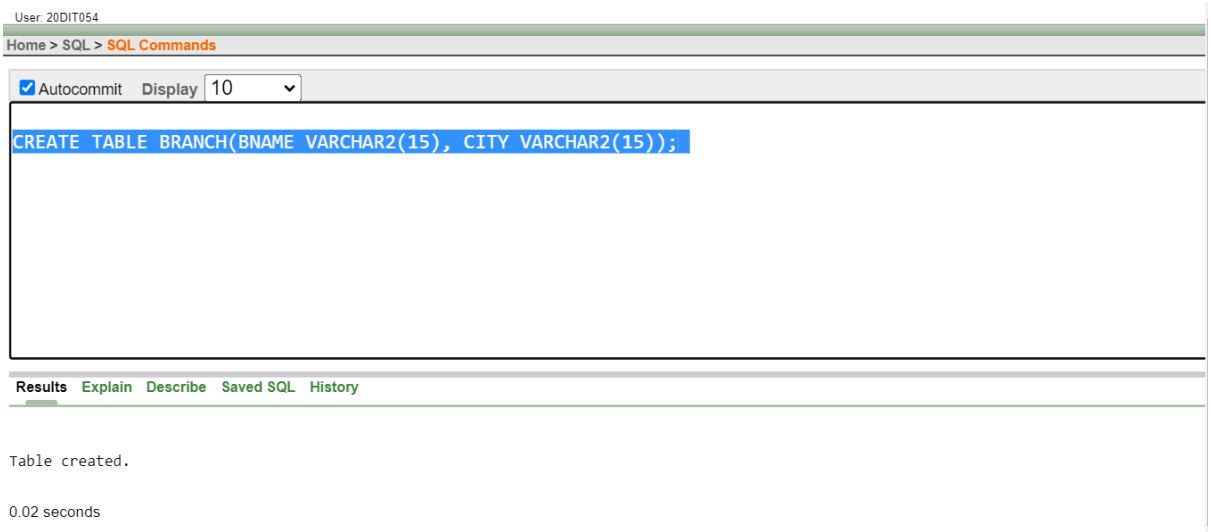
BORROW

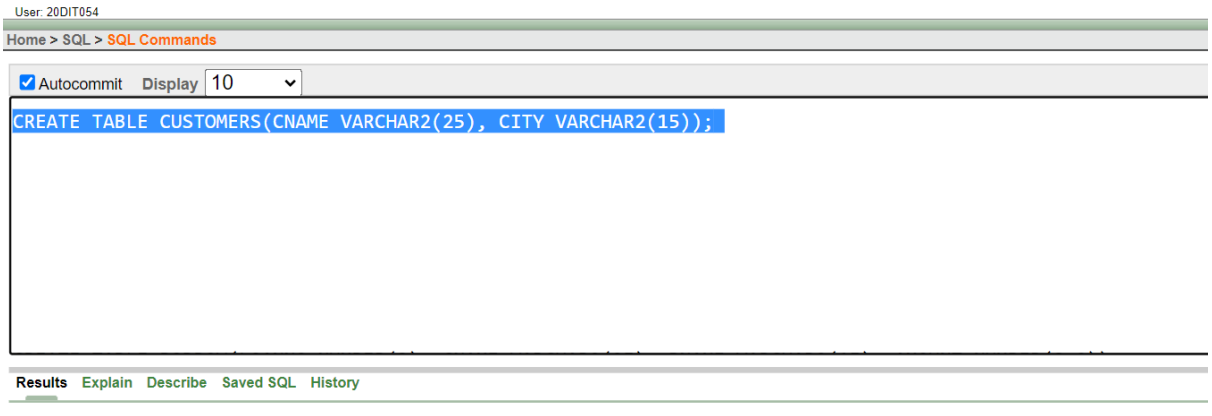
LOANNO	CNAME	BNAME	AMOUNT
201	ANIL	VRCE	1000.00
206	MEHUL	AJNI	5000.00
311	SUNIL	DHARAMPETH	3000.00
321	MADHURI	ANDHERI	2000.00
375	PRMOD	VIRAR	8000.00
481	KRANTI	NEHRU PLACE	3000.00

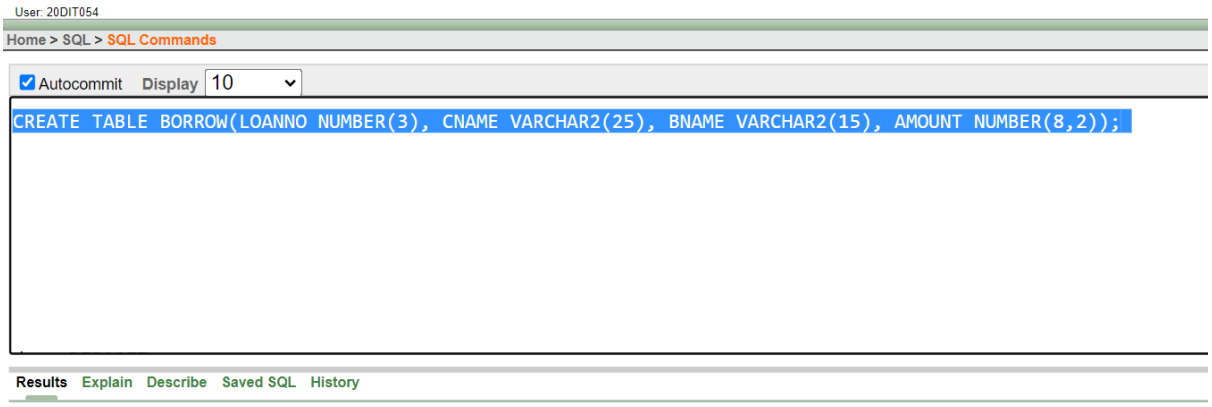
Code:

CREATE TABLE:



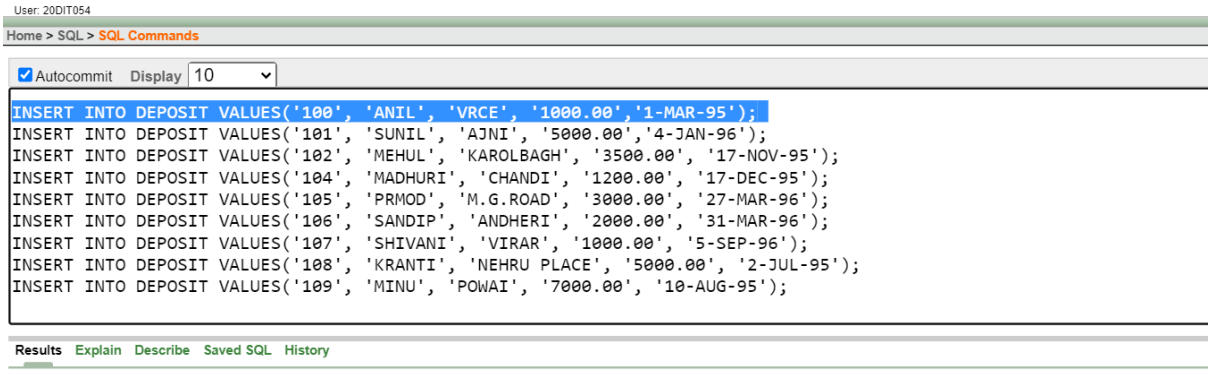




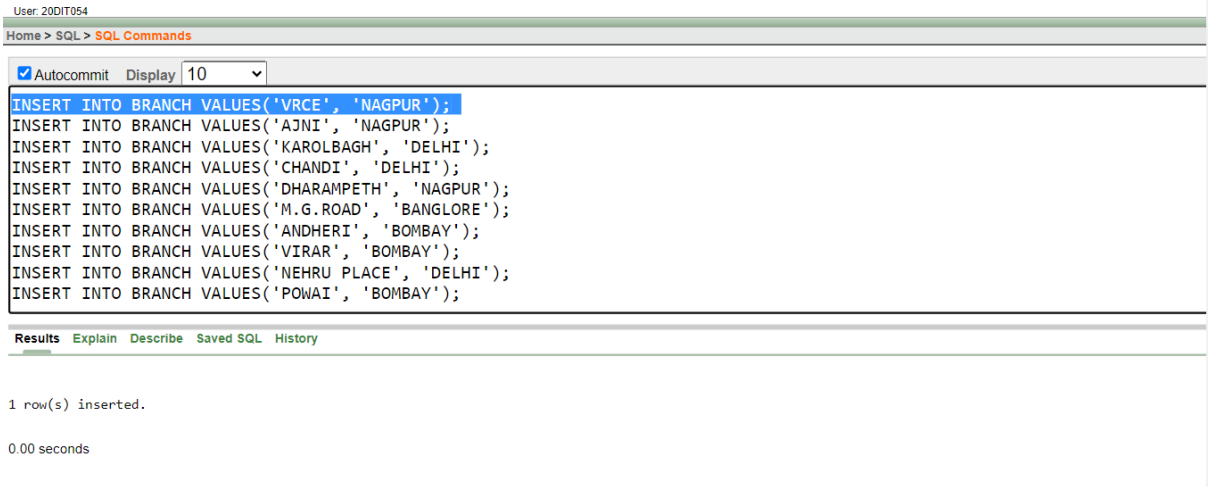


INSERT VALUES INTO TABLE:

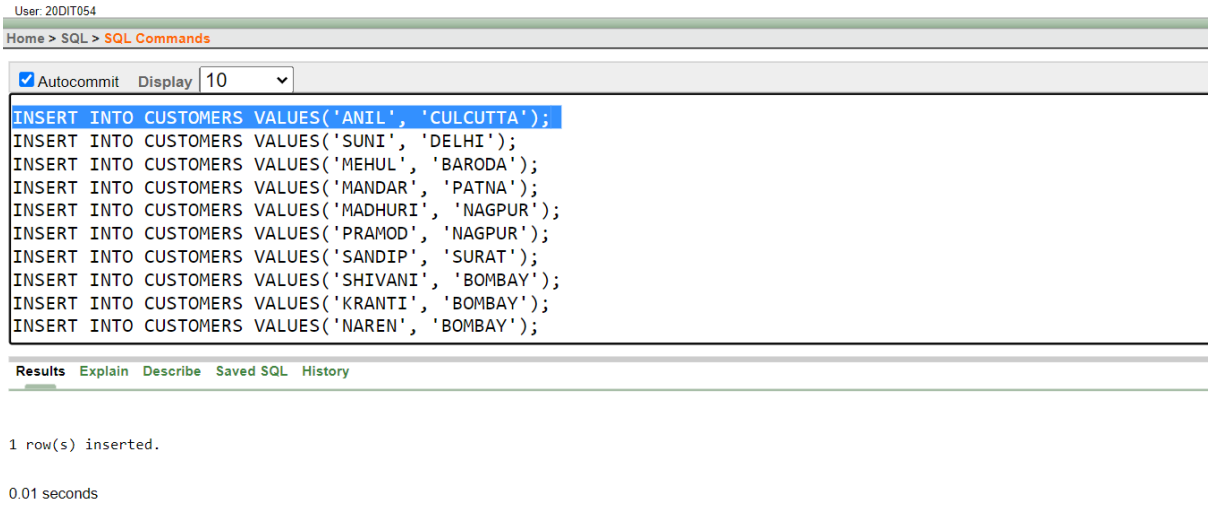
Insert values into DEPOSIT table :



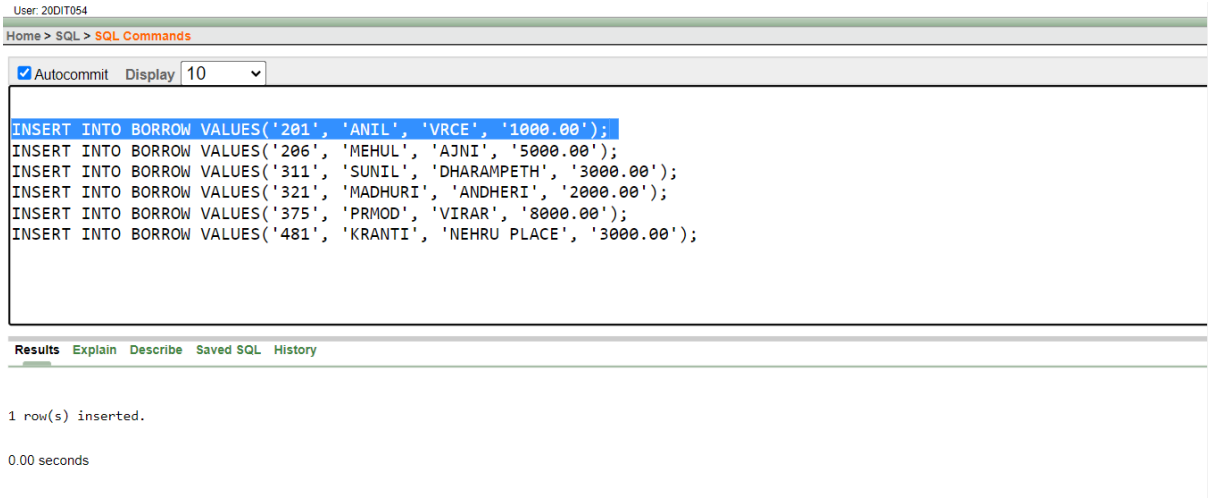
Insert values into BRANCH table :



Insert values into CUSTOMERS table :



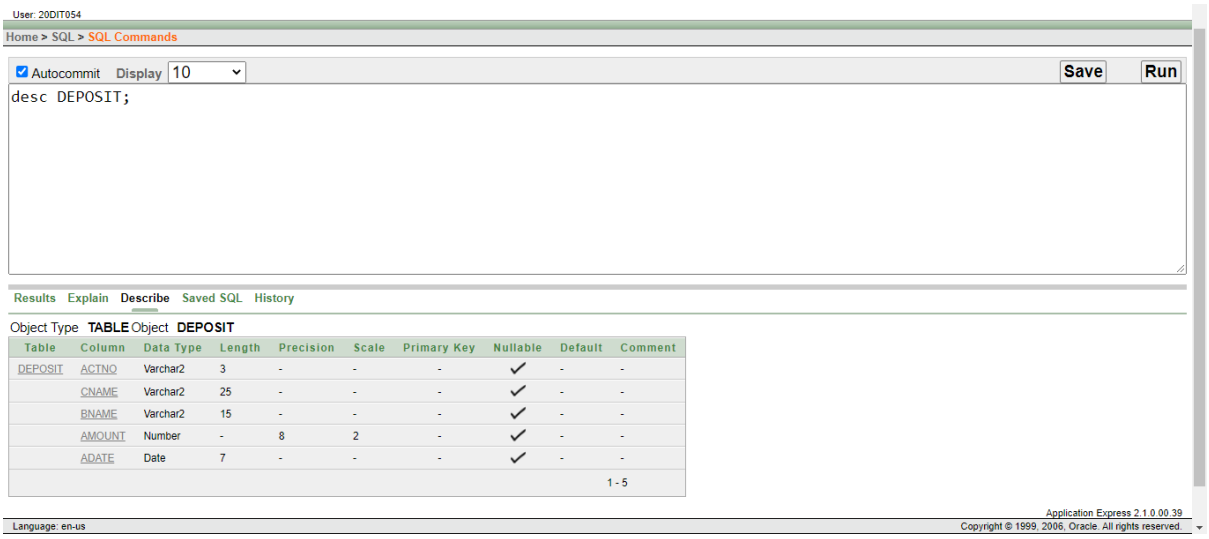
Insert values into BORROW table :



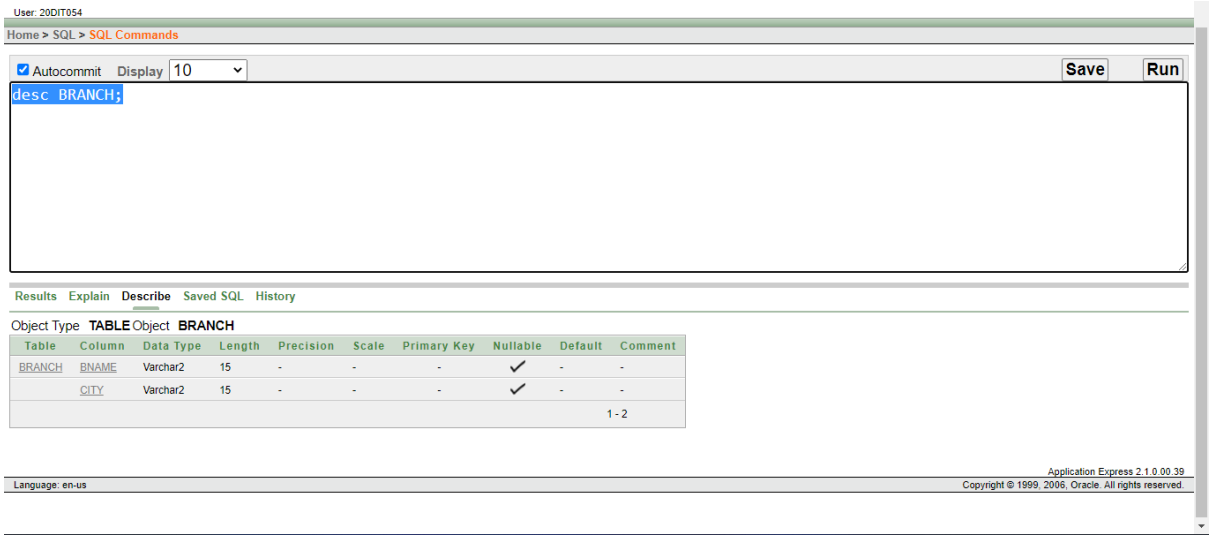
Queries :

(1) Describe deposit, branch.

DEPOSIT :

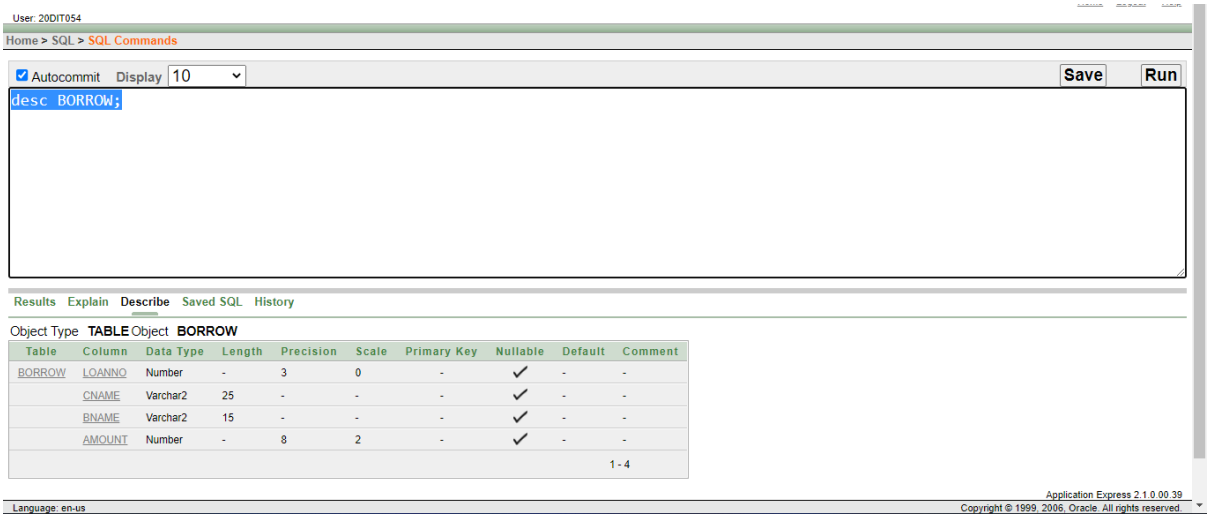


BRANCH :

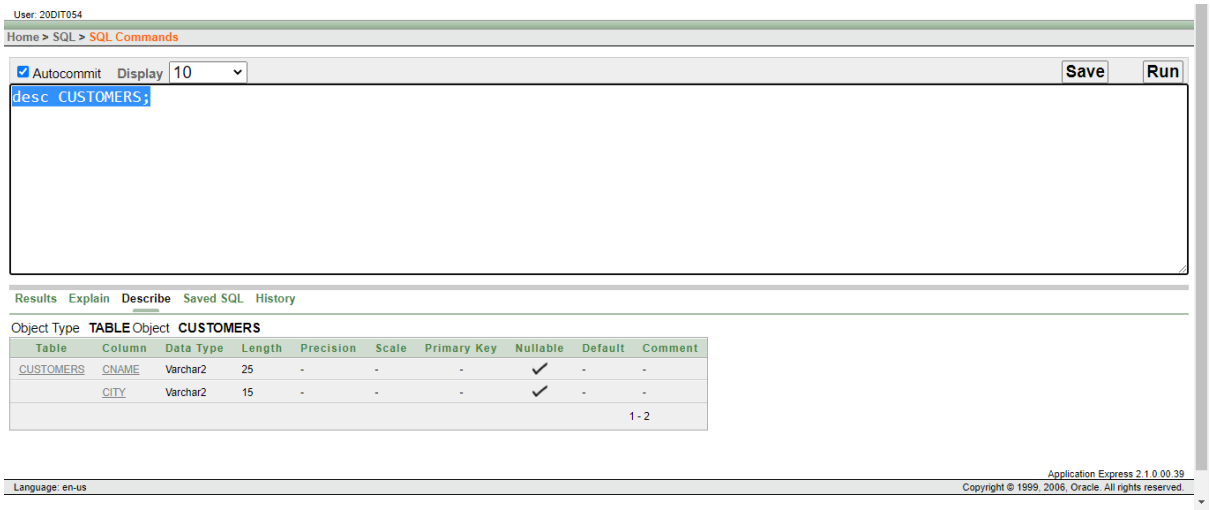


(2) Describe borrow, customers.

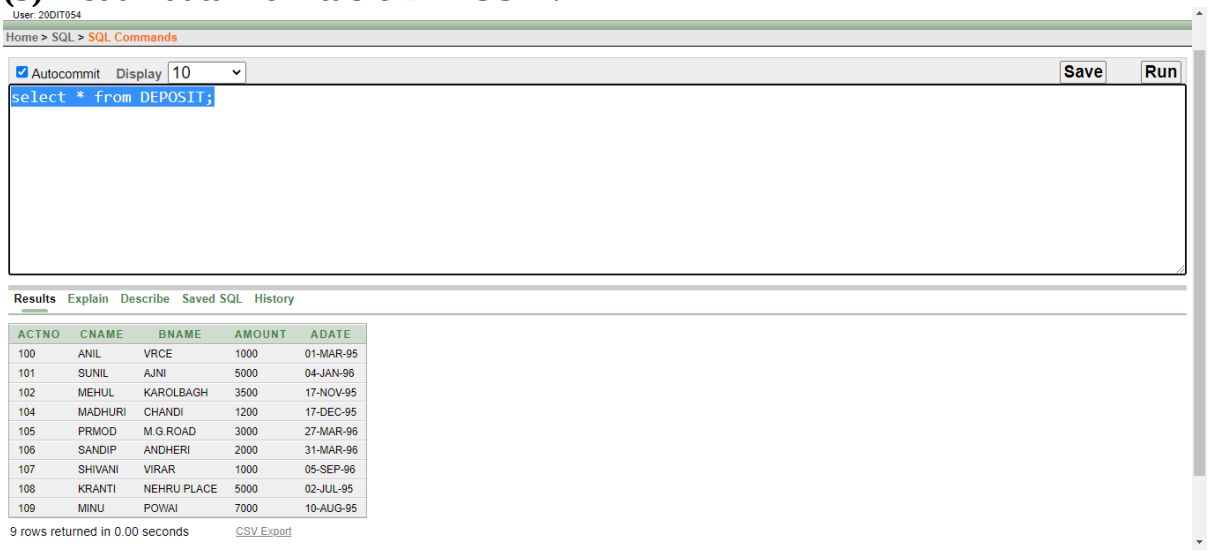
BORROW :



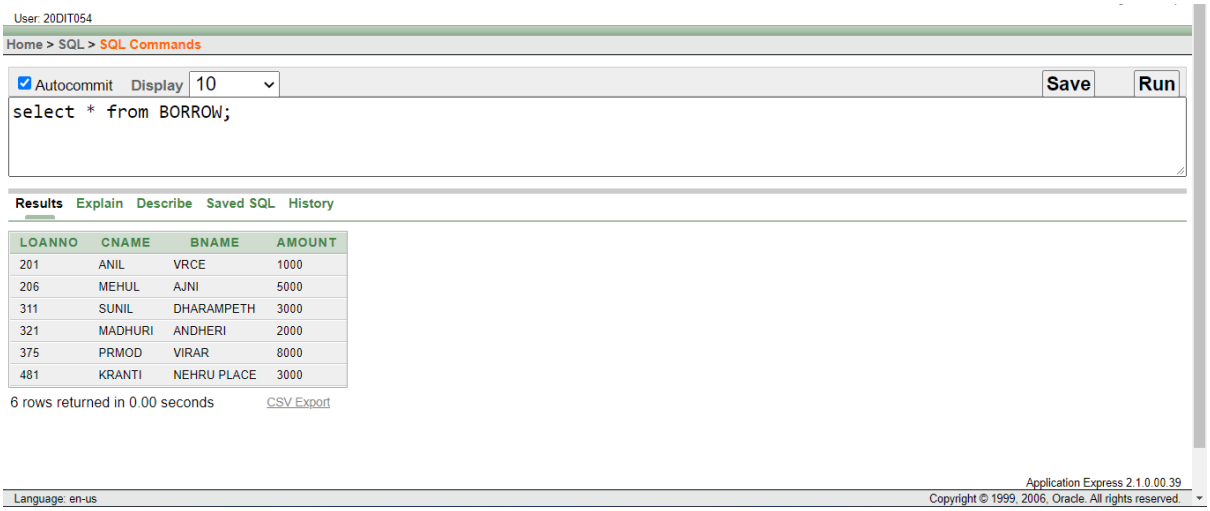
CUSTOMERS :



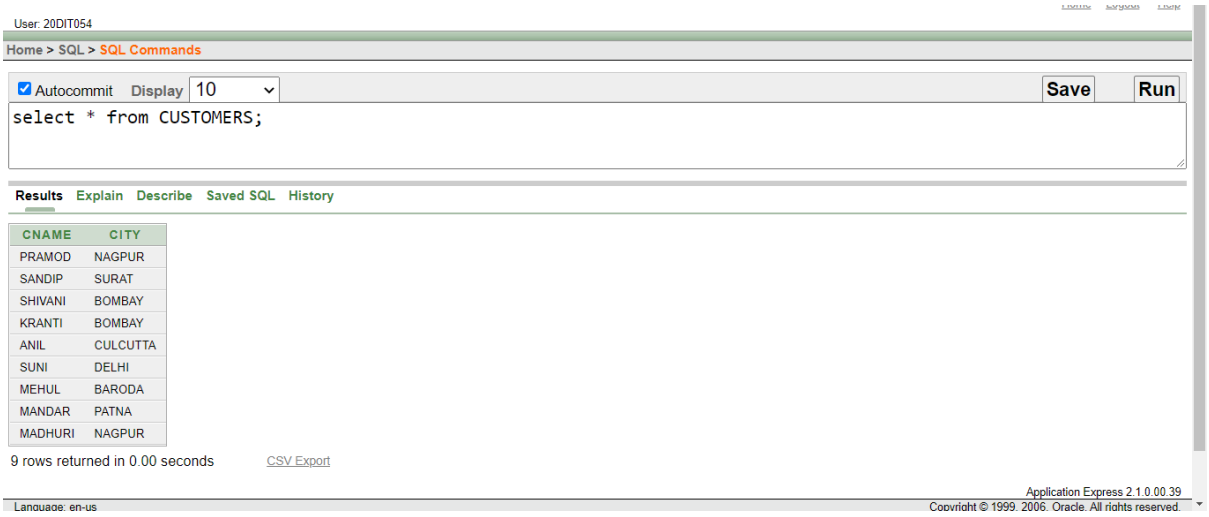
(3) List all data from table DEPOSIT.



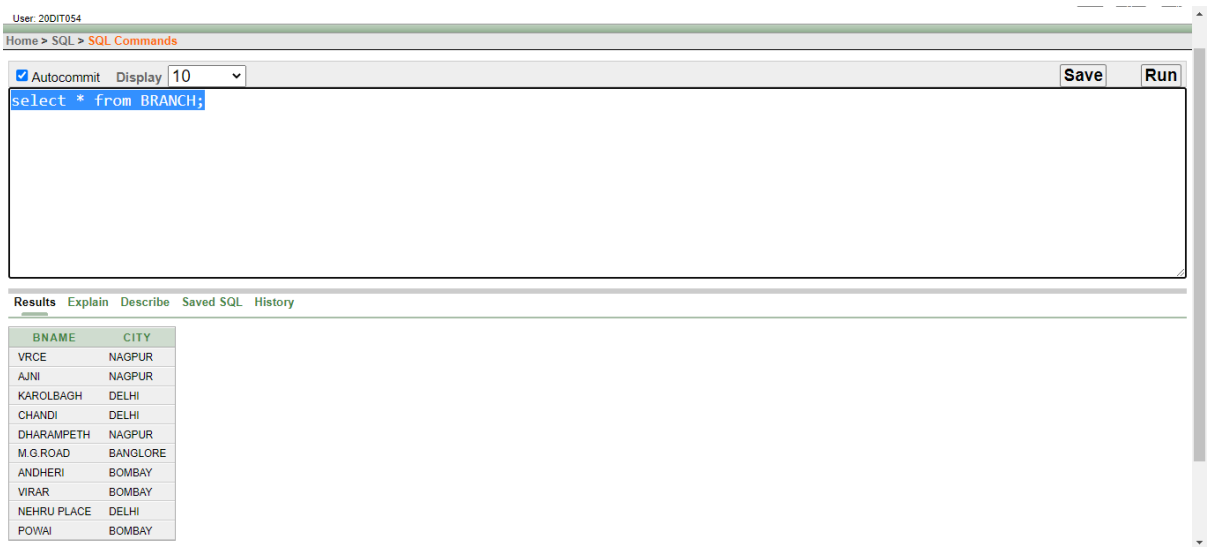
(4) List all data from table BORROW.



(5) List all data from table CUSTOMERS.

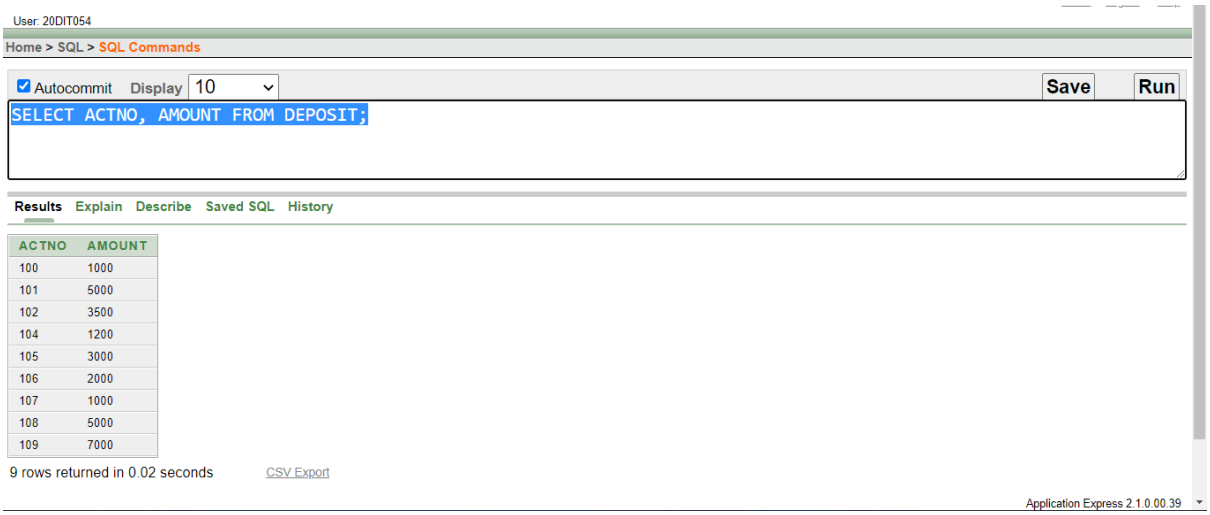


(6) List all data from table BRANCH.



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

(7) Give account no and amount of depositors

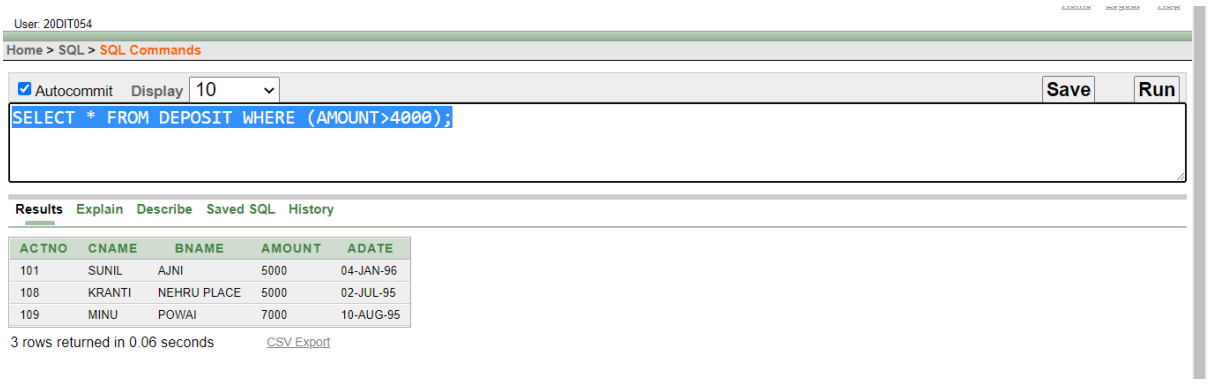


ACTNO	AMOUNT
100	1000
101	5000
102	3500
104	1200
105	3000
106	2000
107	1000
108	5000
109	7000

9 rows returned in 0.02 seconds CSV Export

Application Express 2.1.0.00.39

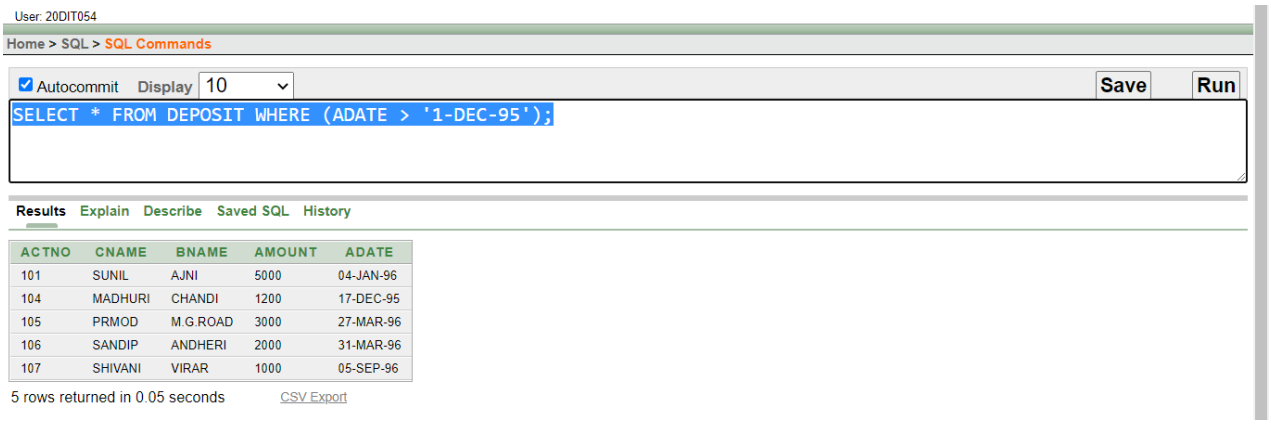
(8) Give name of depositors having amount greater than 4000.



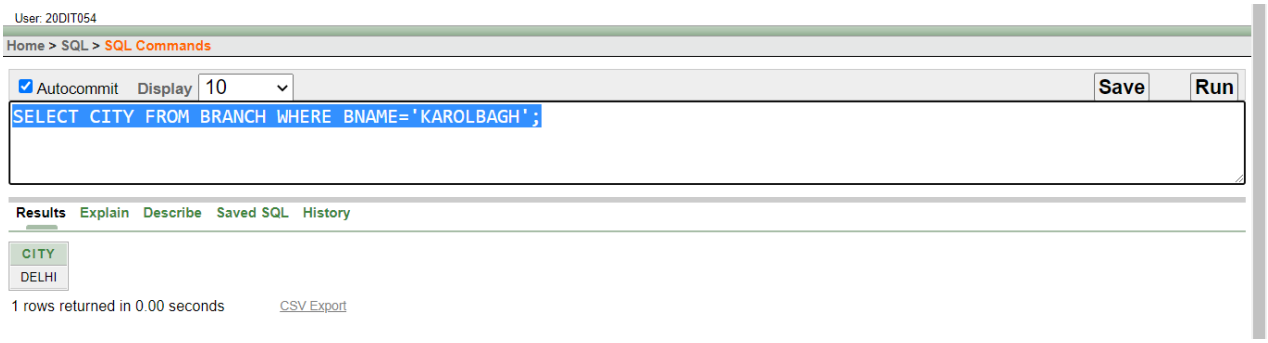
ACTNO	CNAME	BNAME	AMOUNT	ADATE
101	SUNIL	AJNI	5000	04-JAN-96
108	KRANTI	NEHRU PLACE	5000	02-JUL-95
109	MINU	POWAI	7000	10-AUG-95

3 rows returned in 0.06 seconds CSV Export

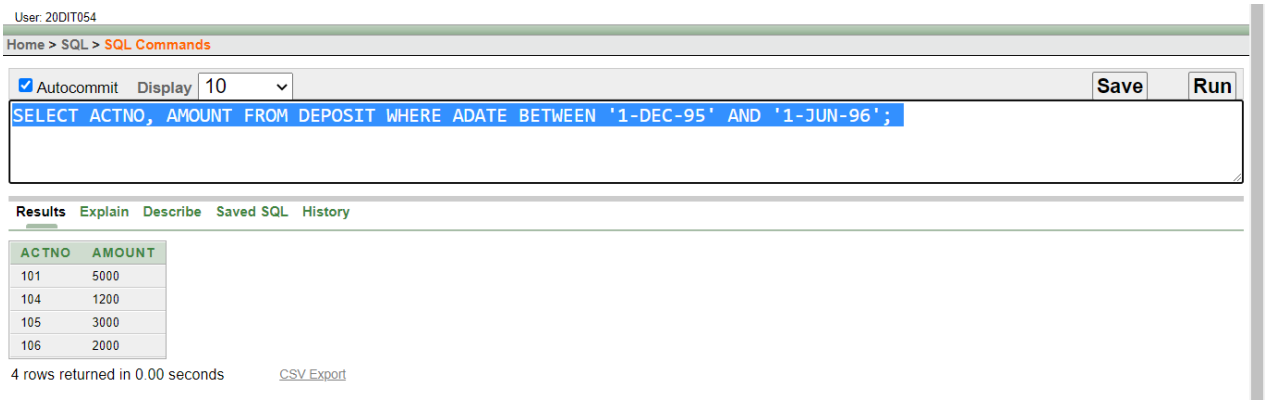
(9) Give name of customers who opened account after date '1-12-95'.



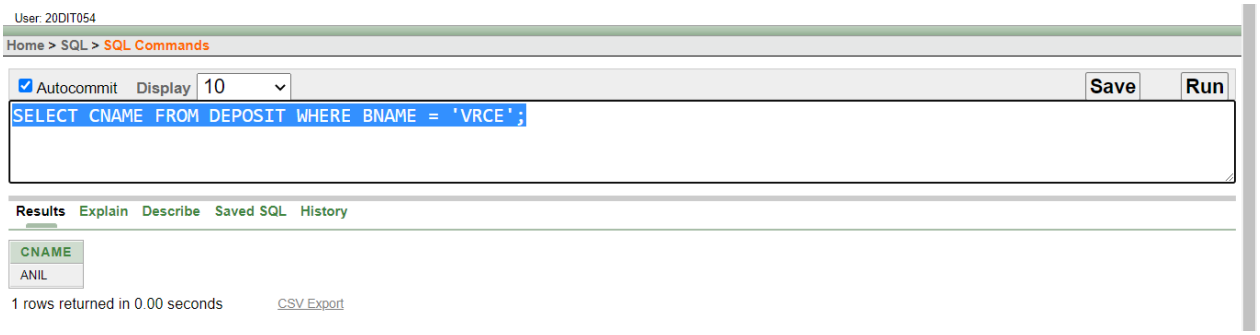
(10) Give name of city where branch karolbagh is located.



(11) Give account no and amount of customer having account opened between date 1-12-95 and 1-6-96.



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



Conclusion:

From this practical I learned how to fetch data from table and learned different type of DDL and DML commands.

Practical-4

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

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

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

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

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

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

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

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

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

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

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

Insert following values in the table **Job**.

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

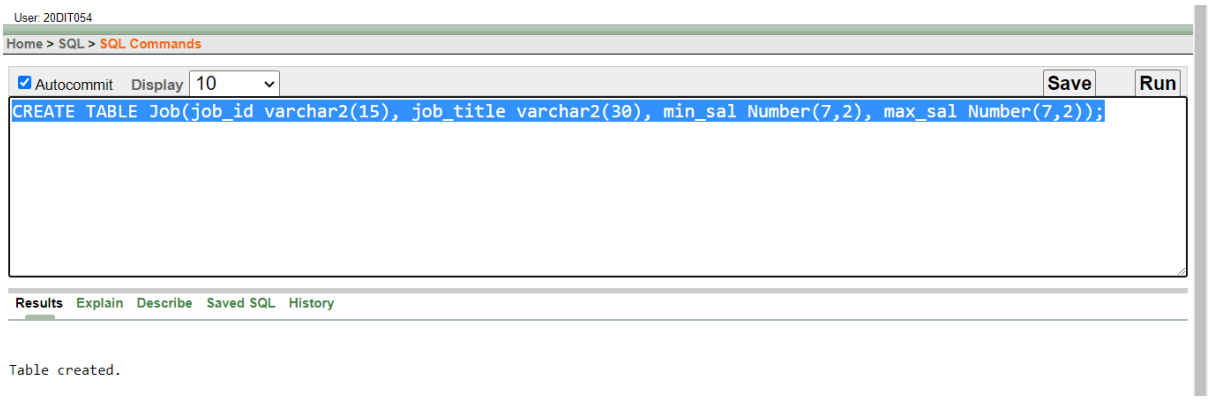
Insert following values in the table **deposit**.

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

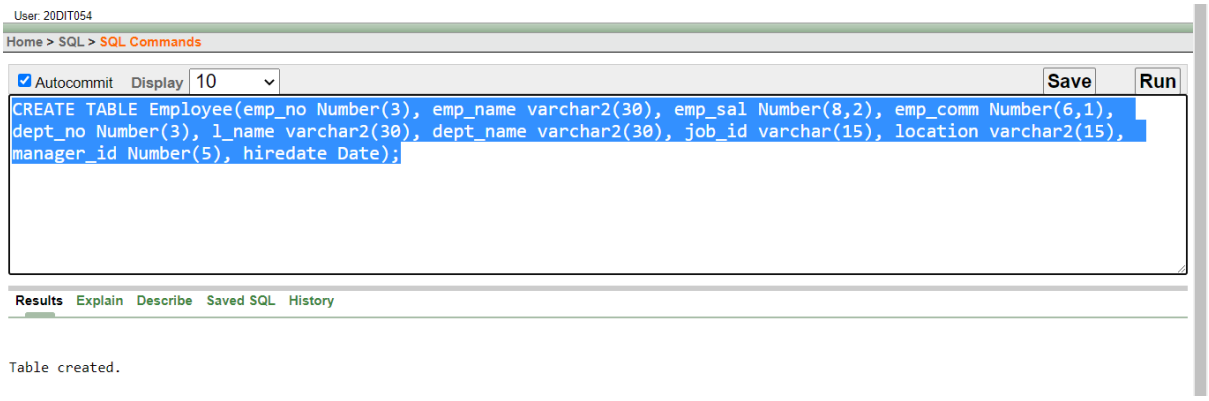
Code:

CREATE TABLE:

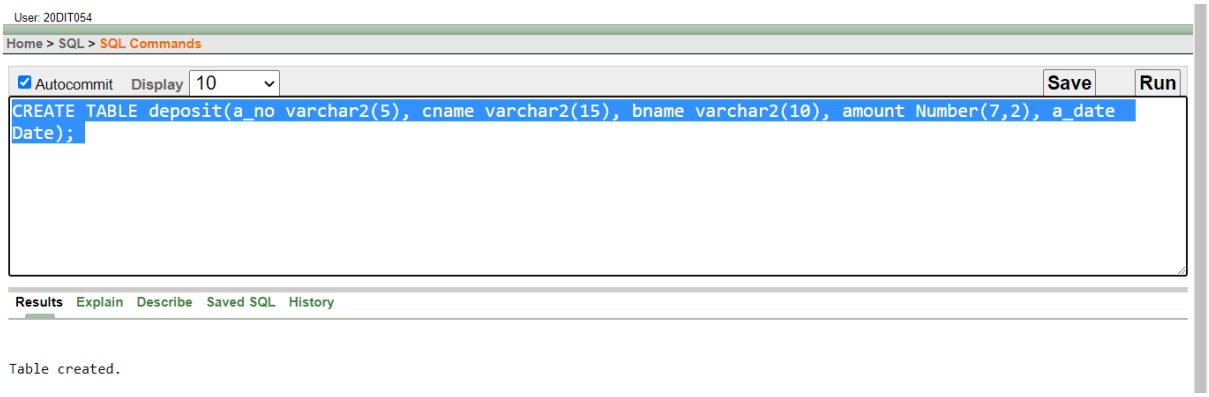
Creating Job table:



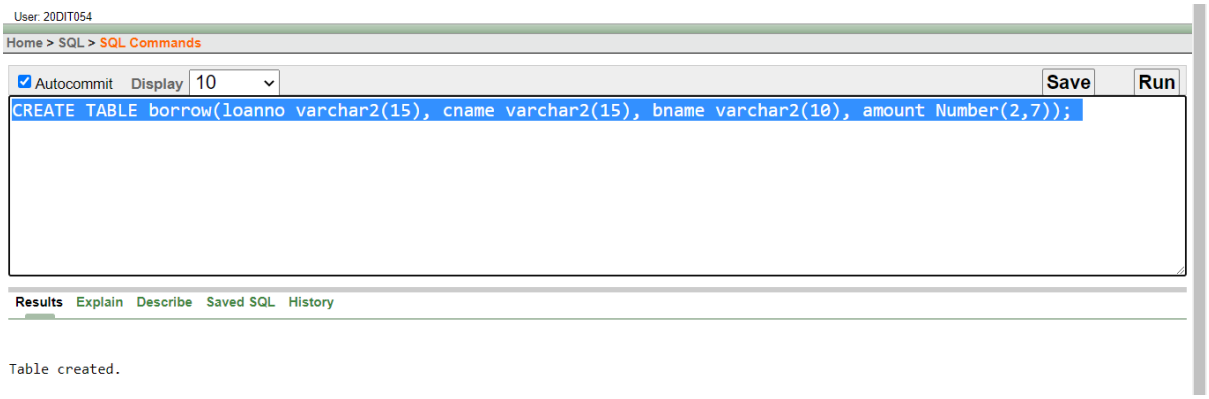
Creating Employee table:



Creating deposit table:

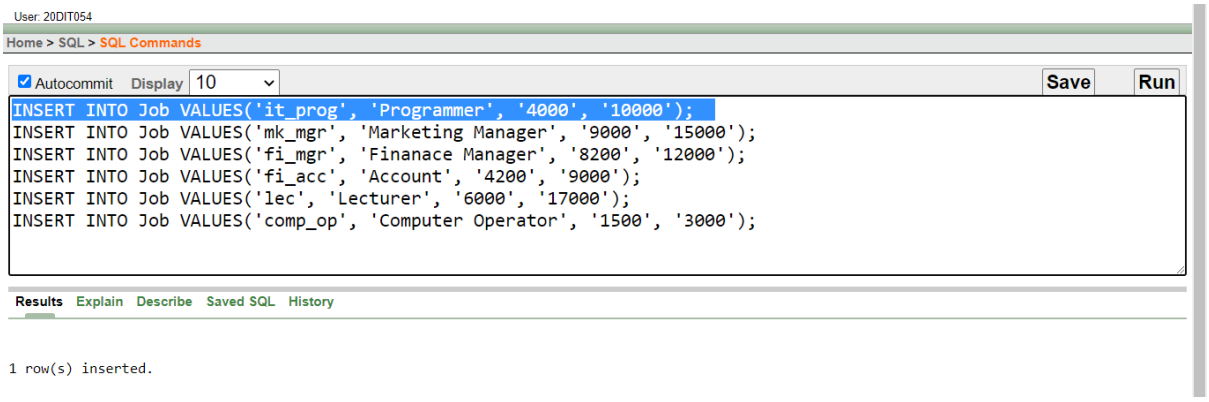


Creating borrow table:

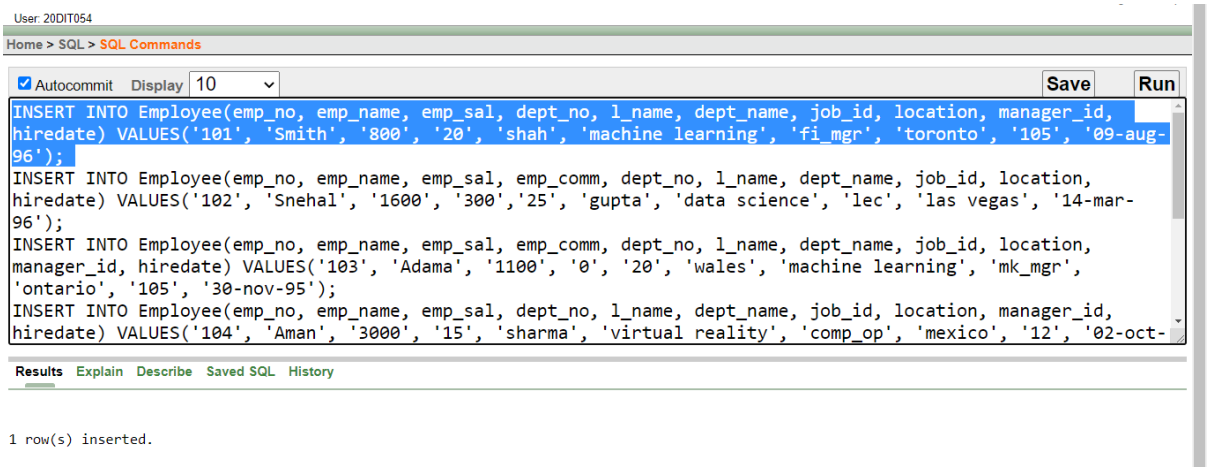


INSERT VALUES INTO TABLE:

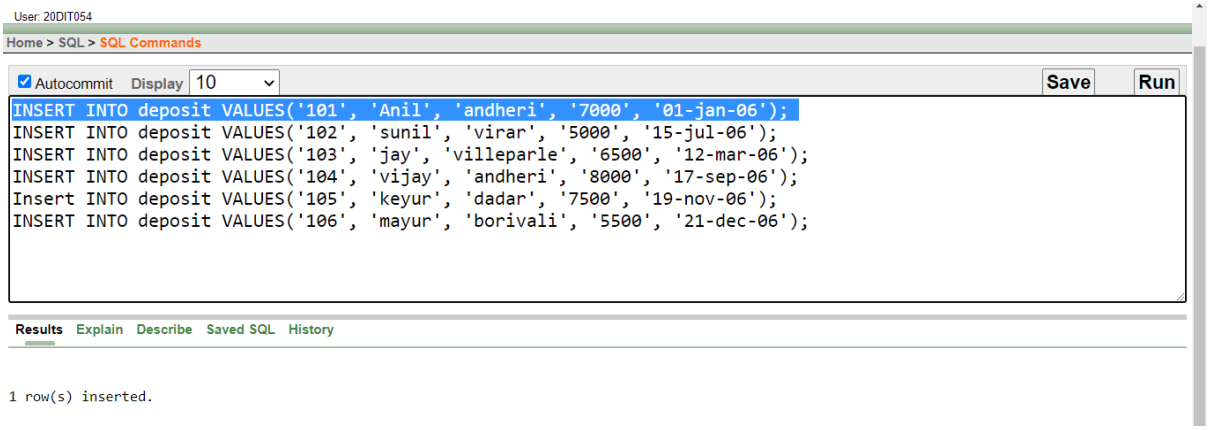
Insert values into Job table :



Insert values into Employee table :



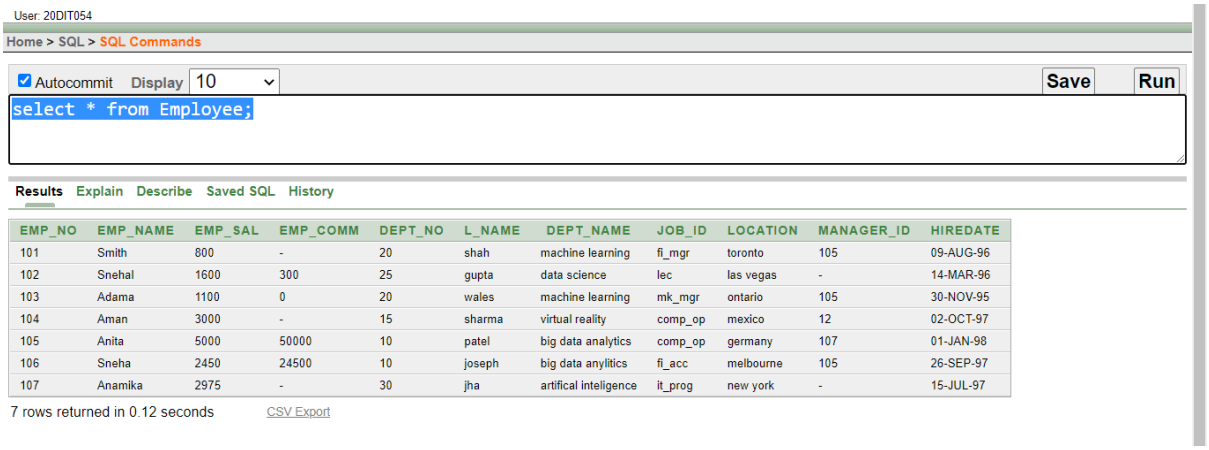
Insert values into deposit table :



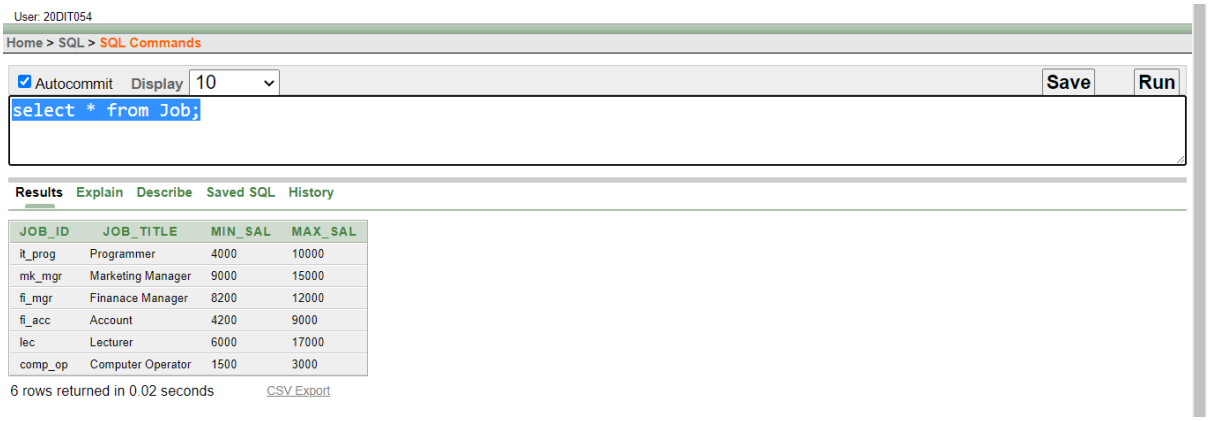
Perform following queries

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

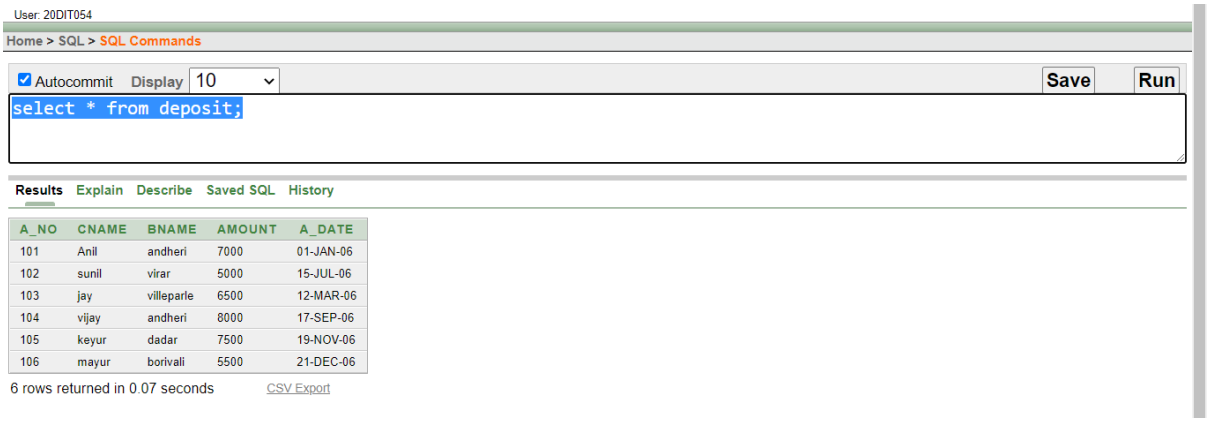
Employee :



Job :



Deposit :


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

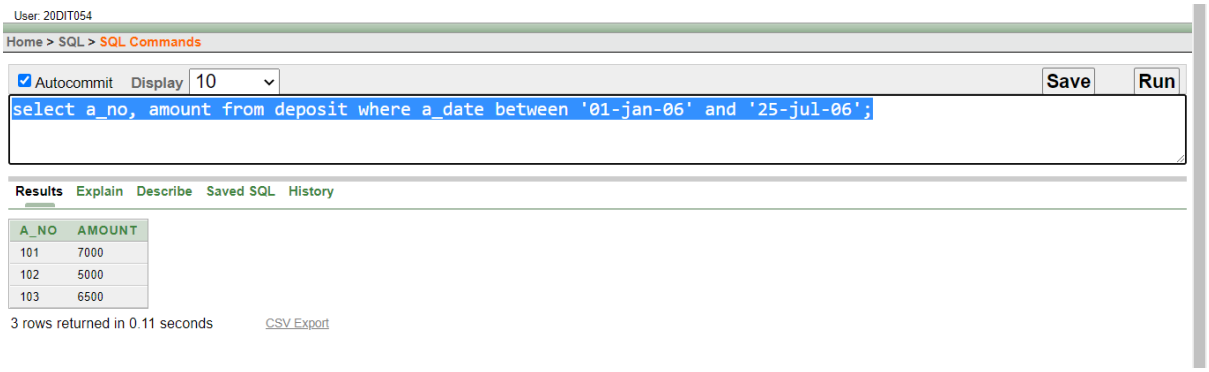
```
select * from deposit;
```

Results Explain Describe Saved SQL History

A_NO	CNAME	BNAME	AMOUNT	A_DATE
101	Anil	andheri	7000	01-JAN-06
102	sunil	virar	5000	15-JUL-06
103	jay	villeparle	6500	12-MAR-06
104	vijay	andheri	8000	17-SEP-06
105	keyur	dadar	7500	19-NOV-06
106	mayur	borivali	5500	21-DEC-06

6 rows returned in 0.07 seconds CSV Export

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


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

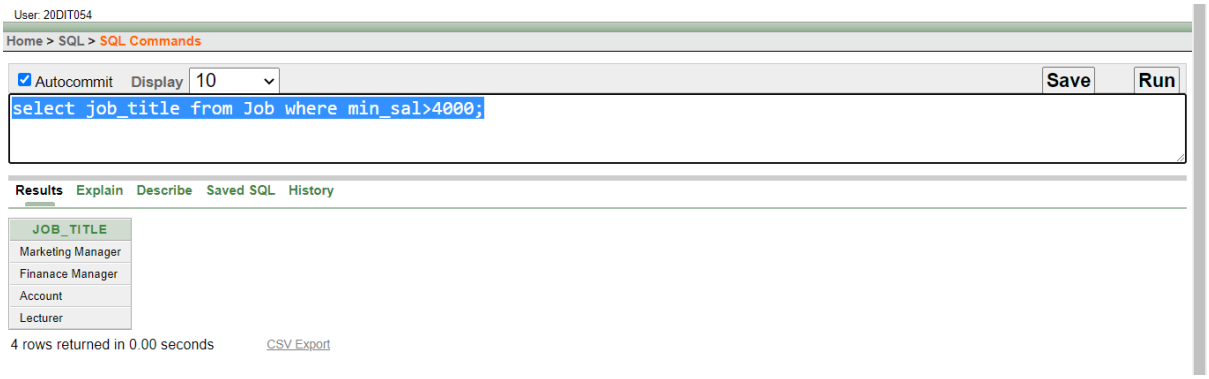
```
select a_no, amount from deposit where a_date between '01-jan-06' and '25-jul-06';
```

Results Explain Describe Saved SQL History

A_NO	AMOUNT
101	7000
102	5000
103	6500

3 rows returned in 0.11 seconds CSV Export

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


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

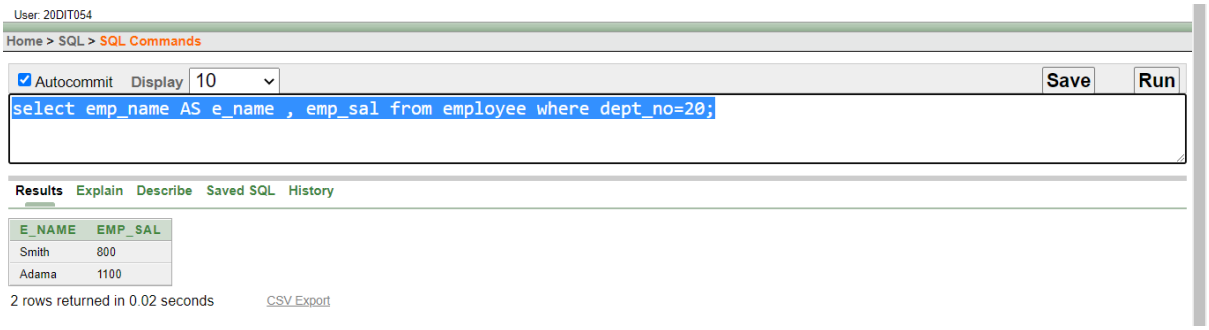
```
select job_title from Job where min_sal>4000;
```

Results Explain Describe Saved SQL History

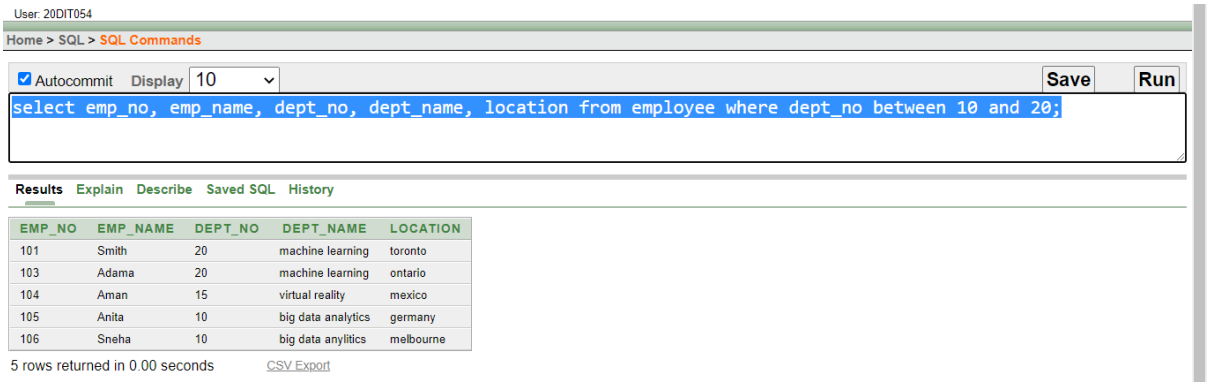
JOB_TITLE
Marketing Manager
Finanace Manager
Account
Lecturer

4 rows returned in 0.00 seconds CSV Export

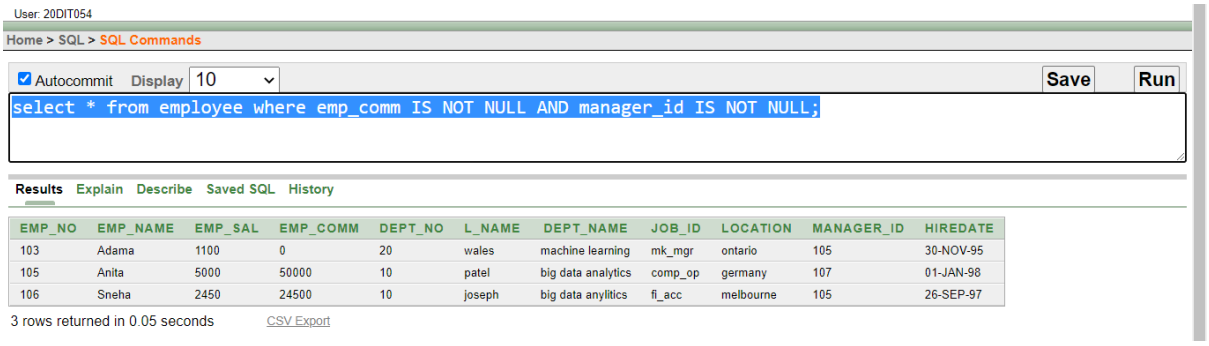
(4) Display name and salary of employee whose department no is 20. Give alias name to name of employee.



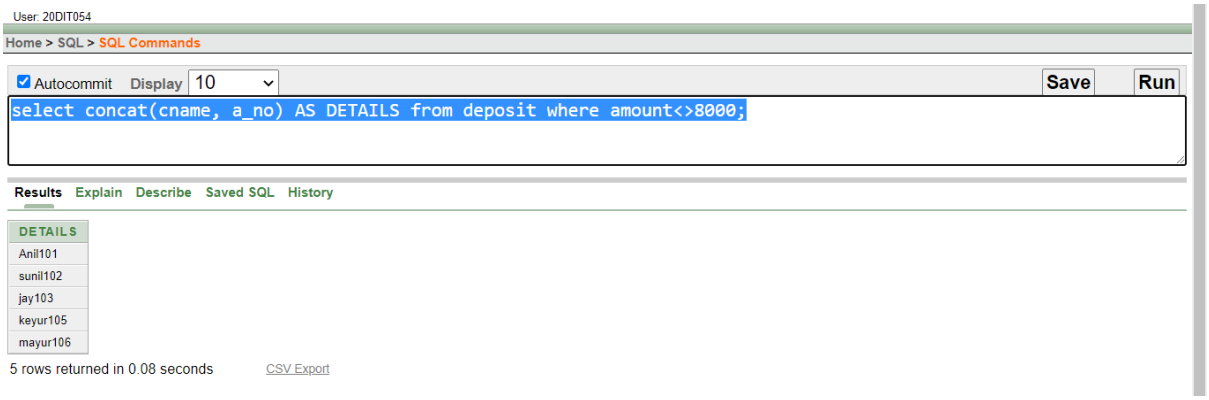
(5) Display employee no, name and department details of those employee whose department lies in (10,20).



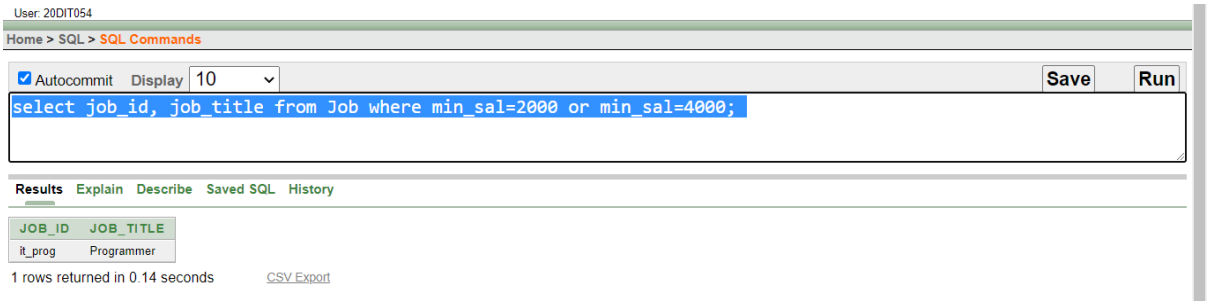
(6) Display the non-null values of employees.



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

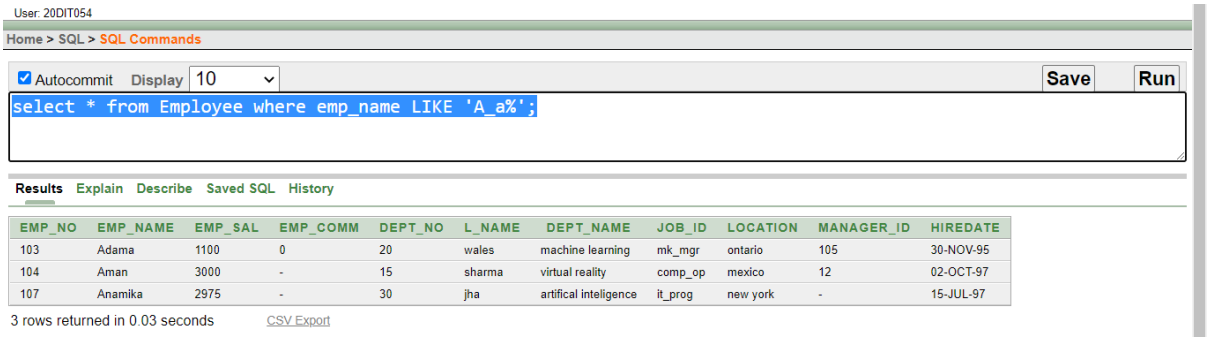


(8) Display the content of job details with minimum salary either 2000 or 4000.

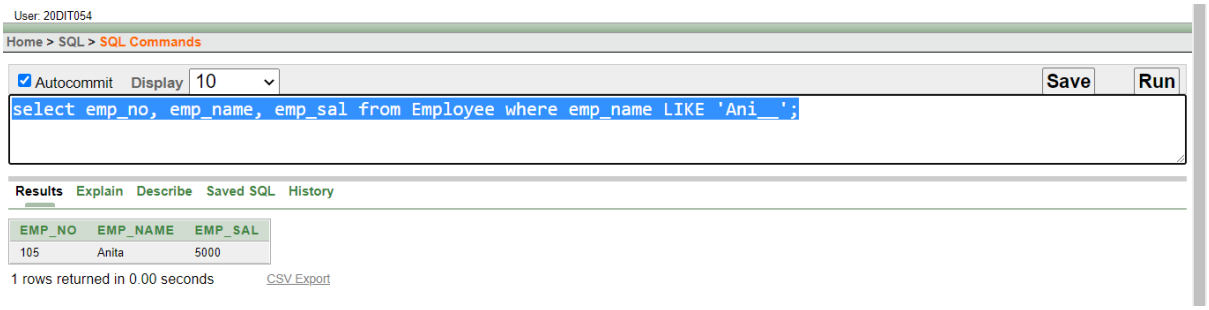


To study various options of LIKE predicate :

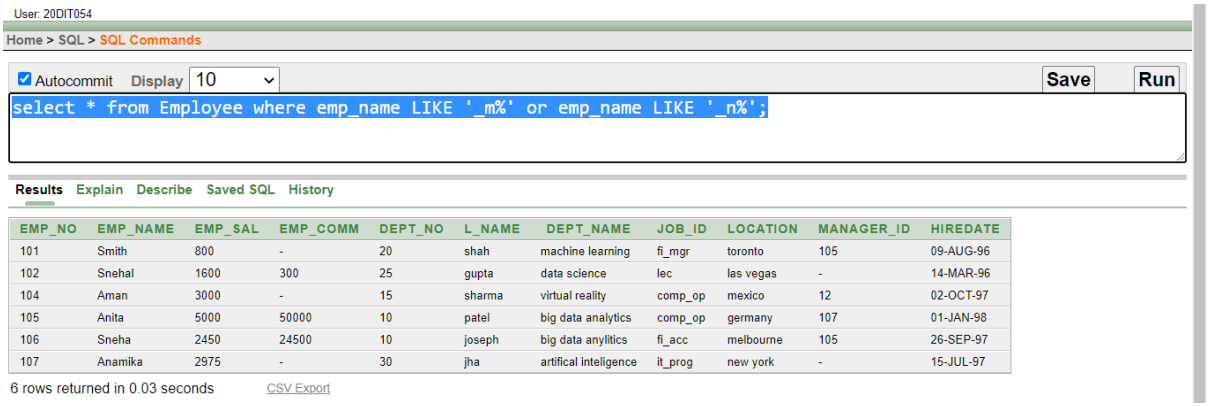
(1) Display all employee whose name start with 'A' and third character is 'a'.



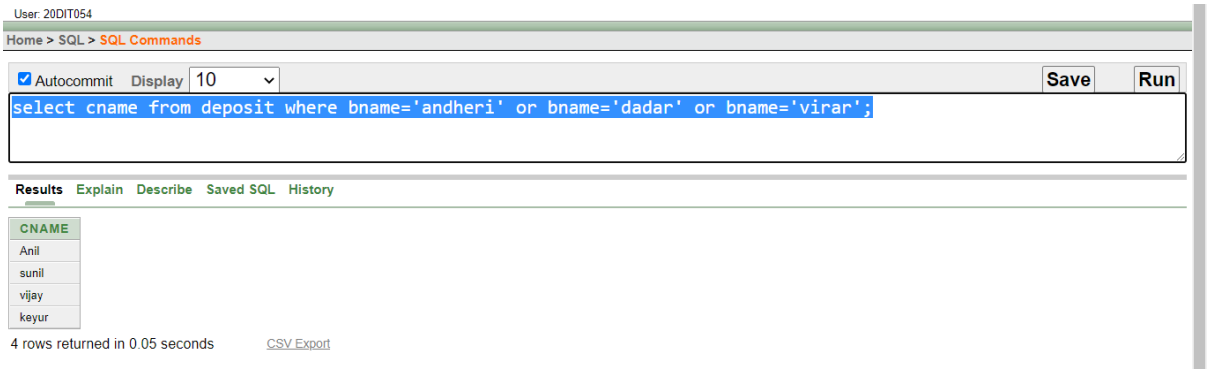
(2) Display name, number and salary of those employees whose name is 5 characters long and first three characters are 'Ani'.



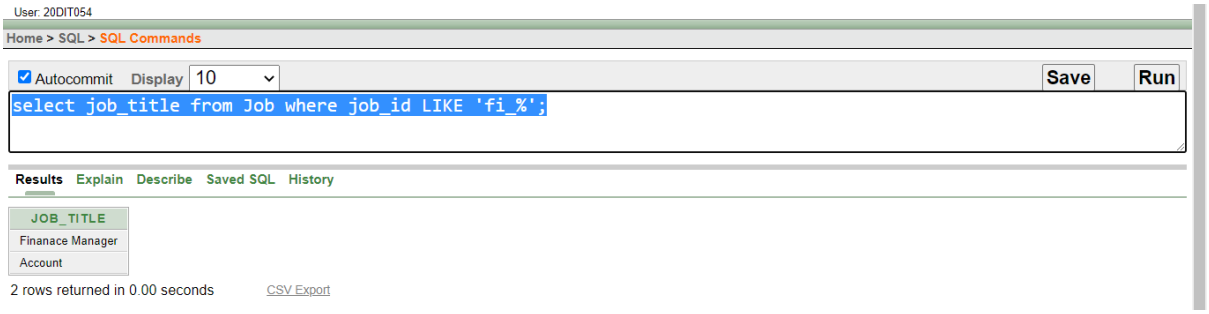
(3) Display all information of employee whose second character of name is either 'm' or 'n'.



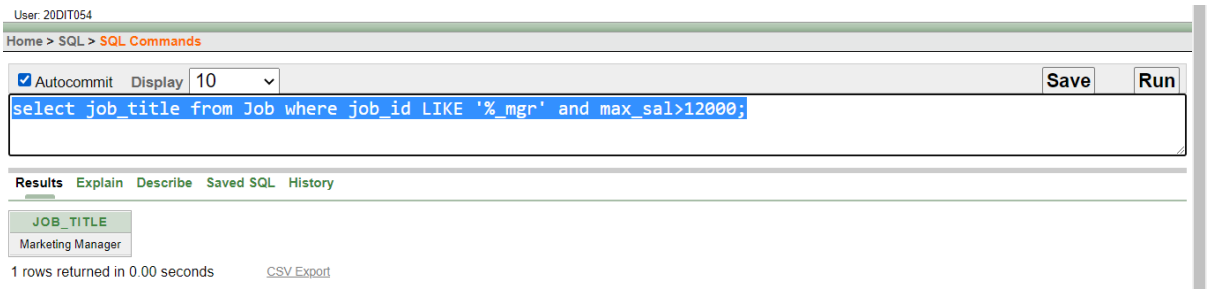
(4) Find the list of all customer name whose branch is in 'andheri' or 'dadar' or 'virar'.



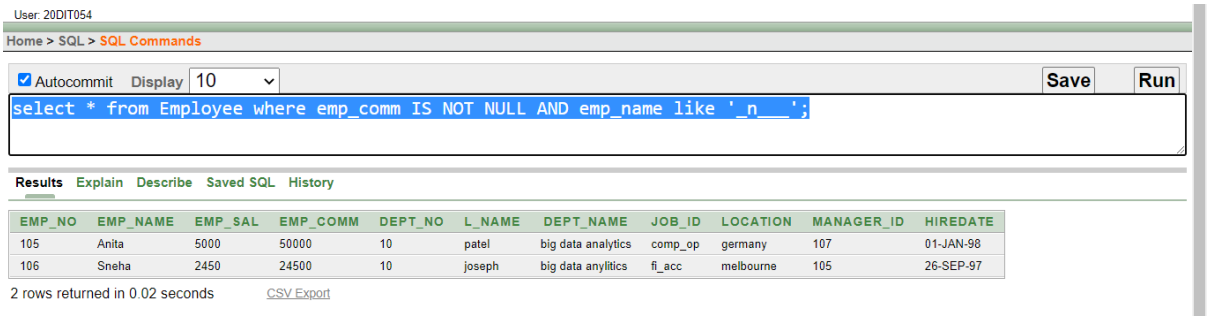
(5) Display the job name whose first three characters in job id field is 'FI_'.



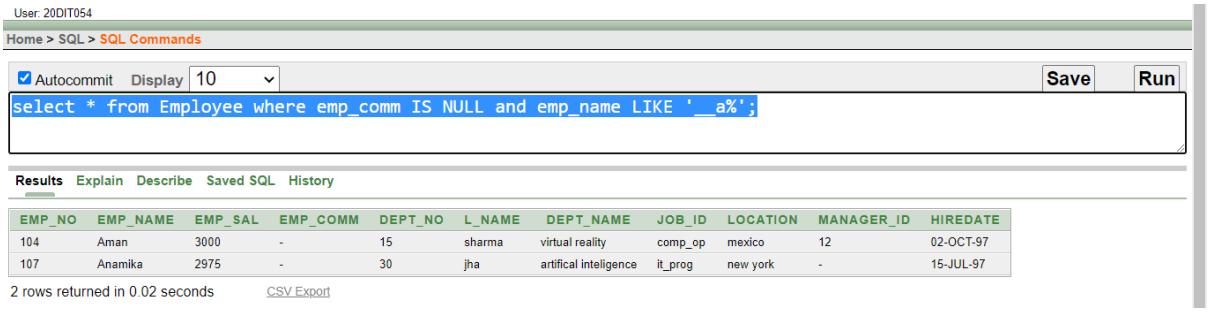
(6) Display the title/name of job who's last three character are '_MGR' and their maximum salary is greater than Rs 12000.



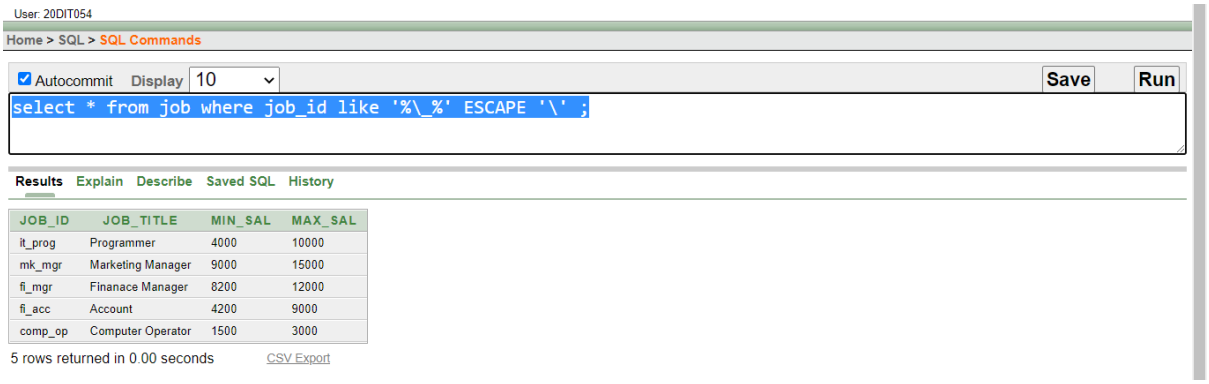
(7) Display the non-null values of employees and also employee name second character should be 'n' and string should be 5-character long.



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



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



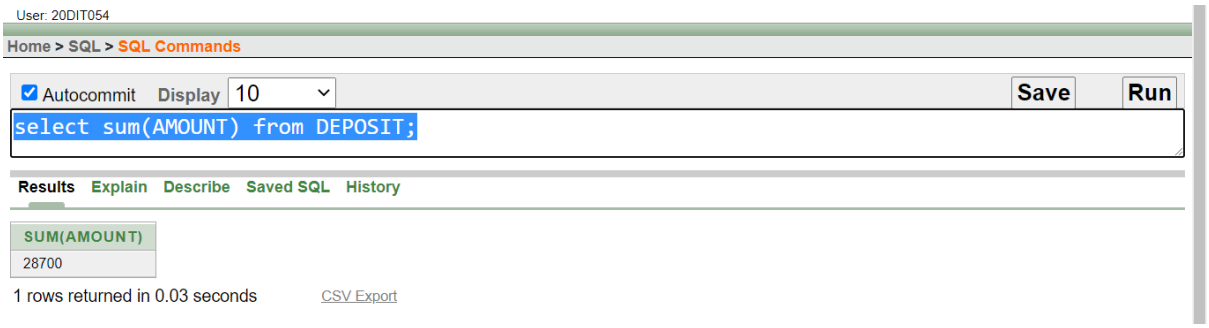
Conclusion :

In this practical we perform some basic queries and learn about various options of LIKE predicate and perform those queries.

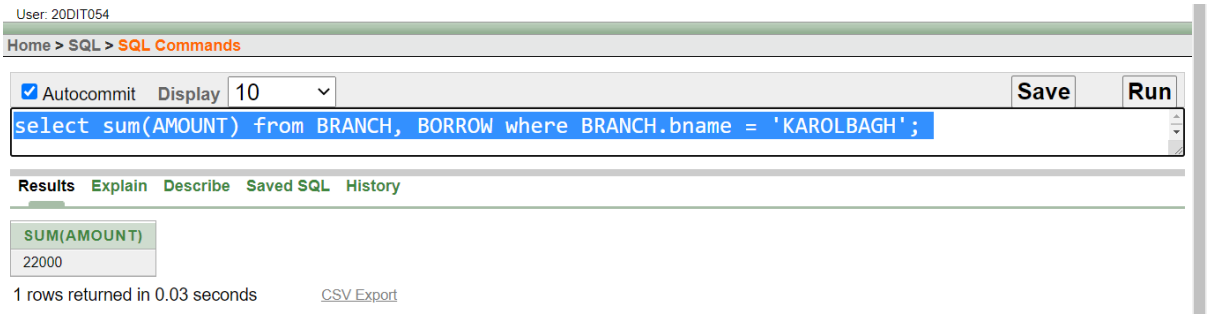
Practical-5

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

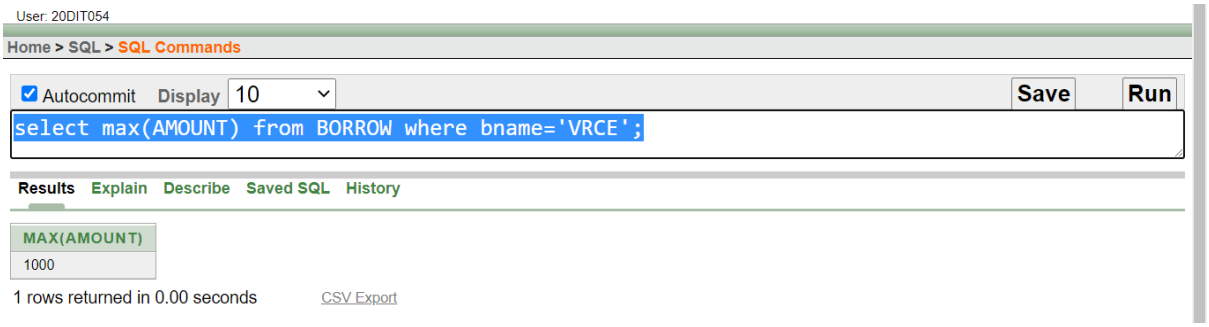
(1) List total deposit from deposit.



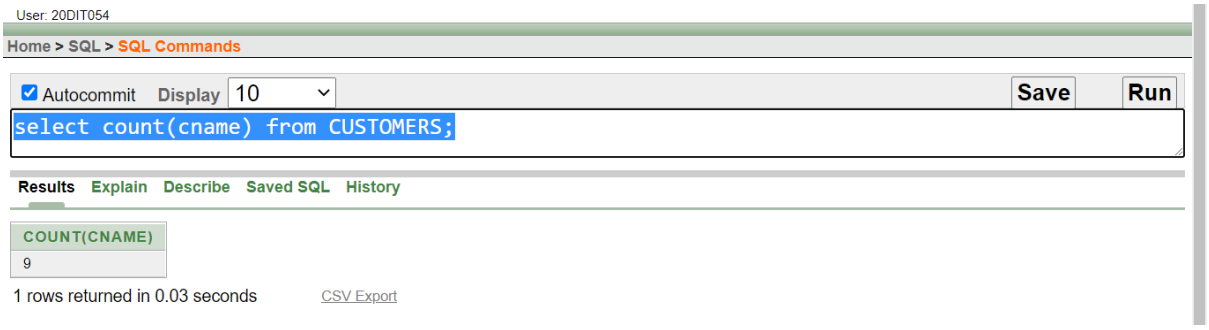
(2) List total loan from karolbagh branch



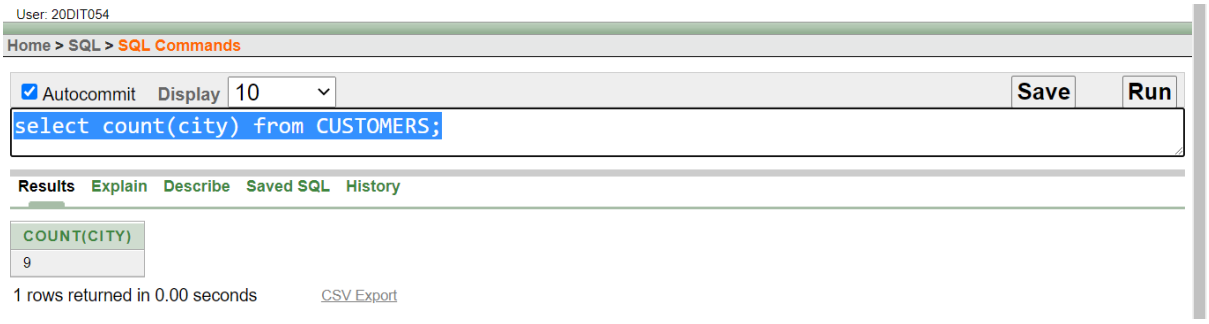
(3) Give maximum loan from branch vrce.



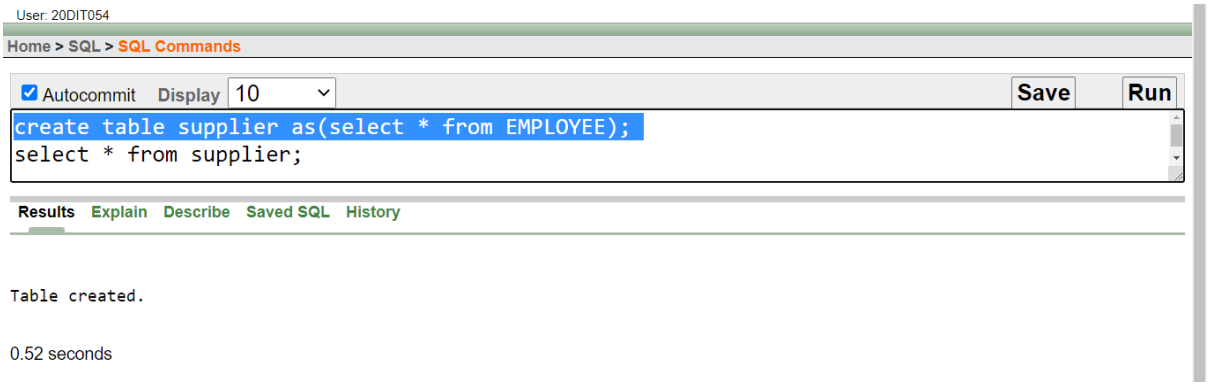
(4) Count total number of customers

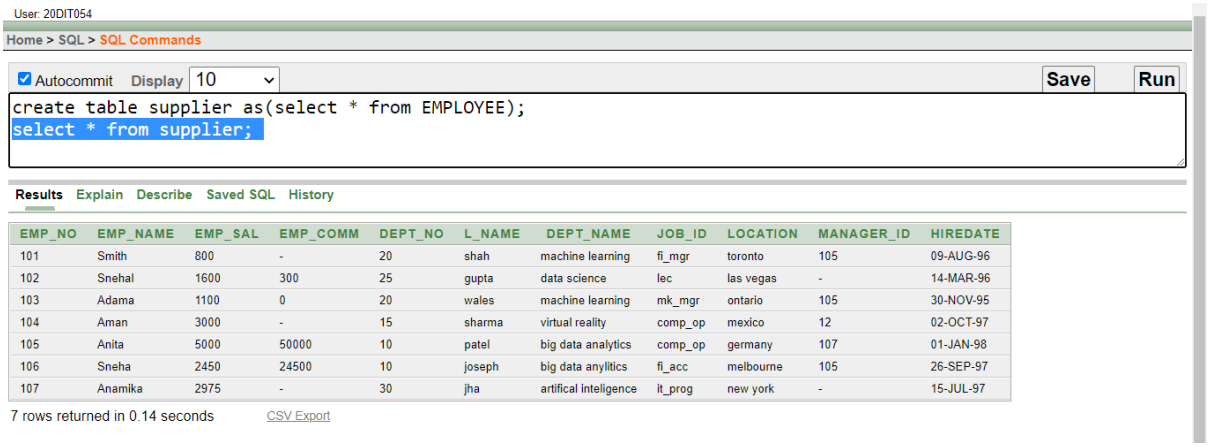


(5) Count total number of customer's cities.

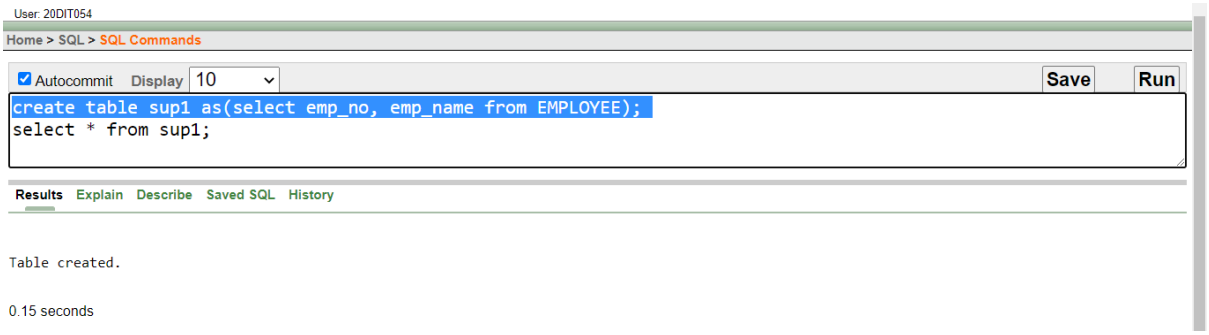


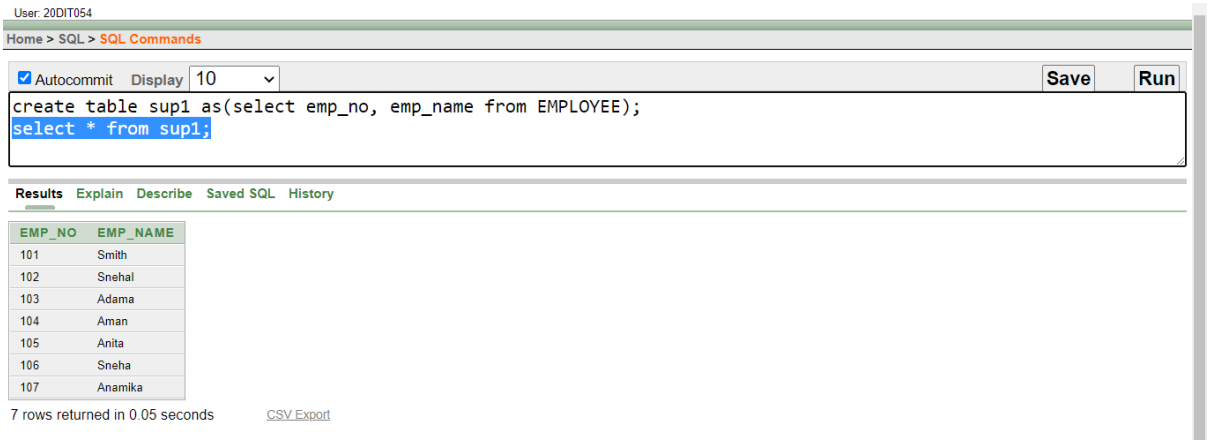
(6) Create table supplier from employee with all the columns.



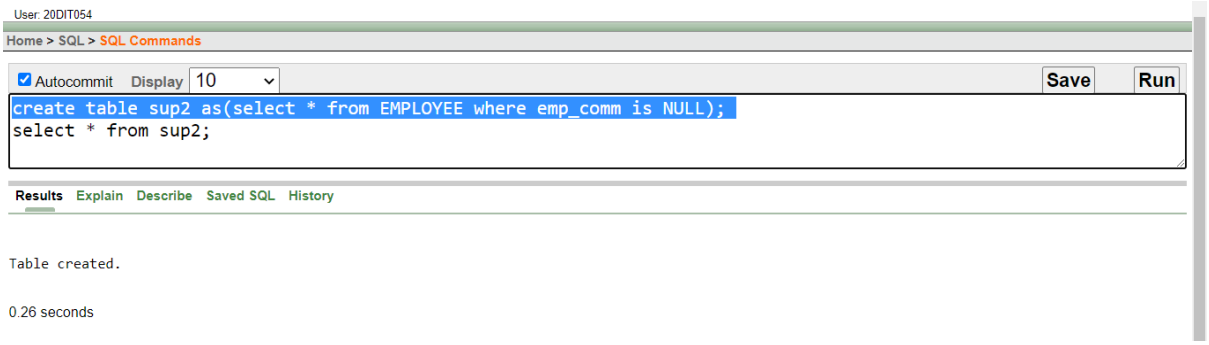


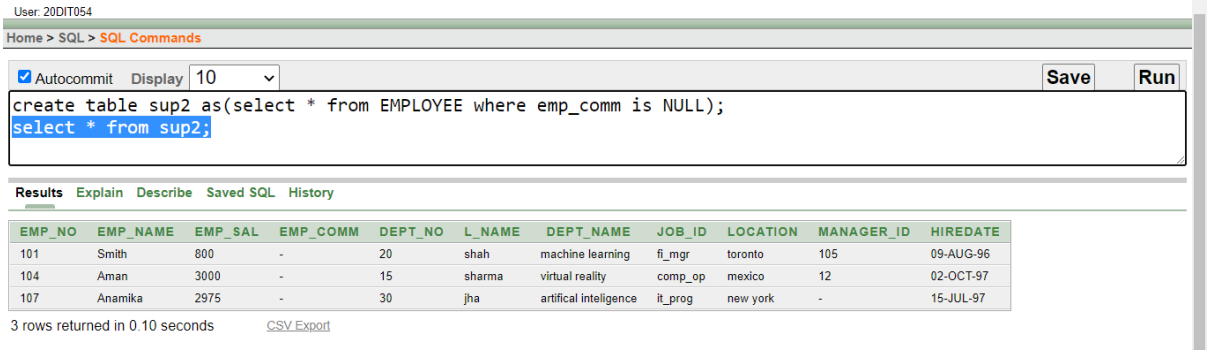
(7) Create table sup1 from employee with first two columns.



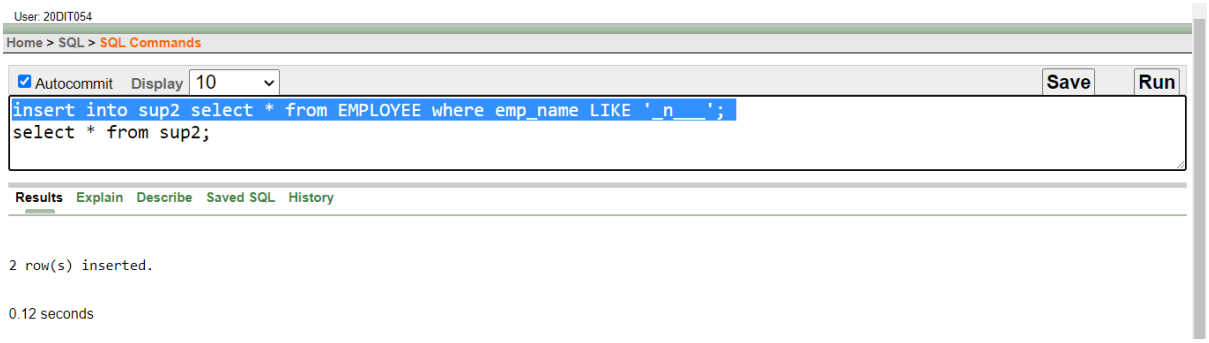


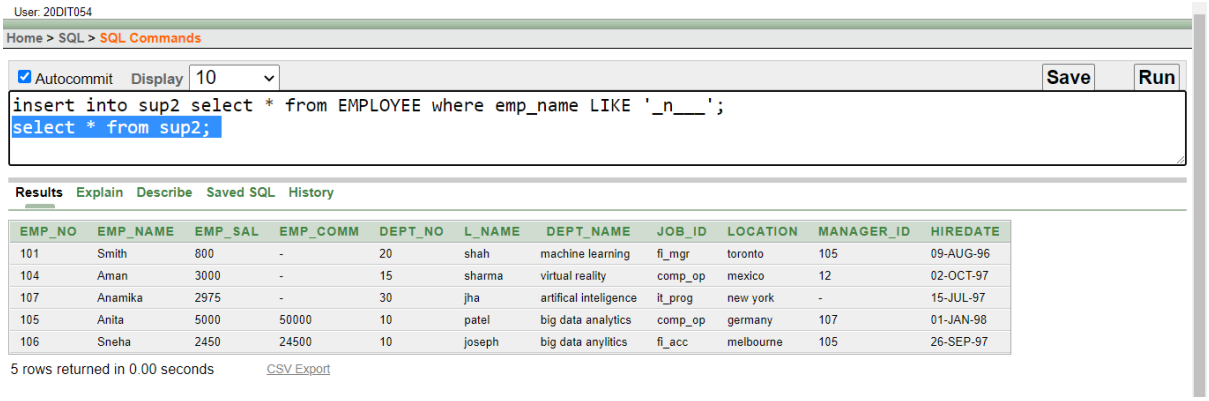
(8) Create table sup2 from employee with no data





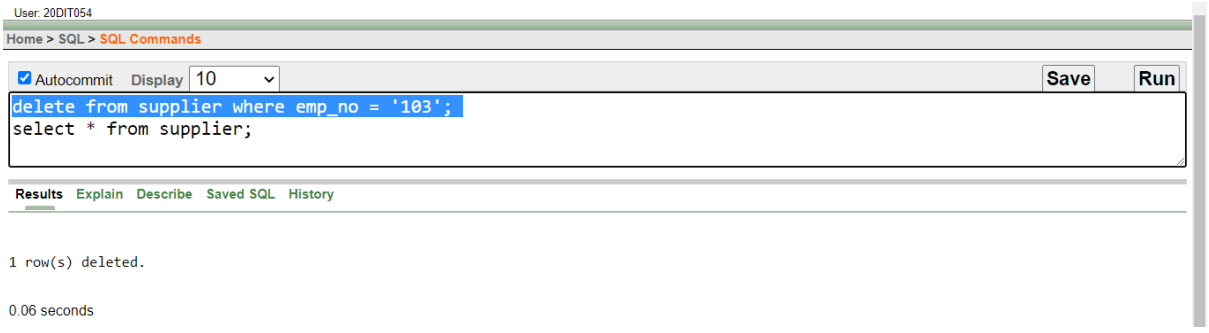
(9) Insert the data into sup2 from employee whose second character should be ‘n’ and string should be 5 characters long in employee name field.





(10) Delete all the rows from sup1.

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



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

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

Results Explain Describe Saved SQL History

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

6 rows returned in 0.00 seconds CSV Export

(12) Rename the table sup2.

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
rename sup2 to supplier2;
select * from supplier2;
```

Results Explain Describe Saved SQL History

```
Statement processed.
```

0.06 seconds

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
rename sup2 to supplier2;
select * from supplier2;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	20	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
104	Aman	3000	-	15	sharma	virtual reality	comp_op	mexico	12	02-OCT-97
107	Anamika	2975	-	30	jha	artifical inteligence	it_prog	new york	-	15-JUL-97
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98
106	Sneha	2450	24500	10	joseph	big data anylitics	fi_acc	melbourne	105	26-SEP-97

5 rows returned in 0.02 seconds CSV Export

(13) Destroy table sup1 with all the data.

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

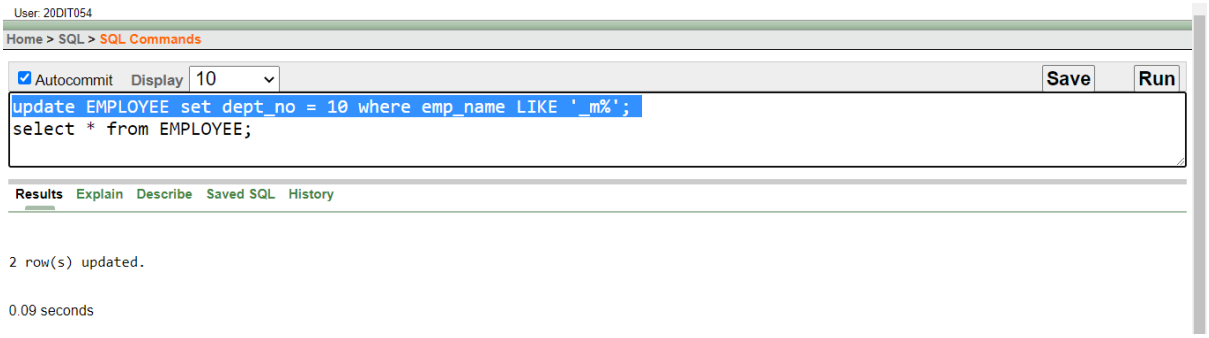
```
drop table sup1;
```

Results Explain Describe Saved SQL History

```
Table dropped.
```

0.19 seconds

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



User: 20DIT054

Home > SQL > SQL Commands

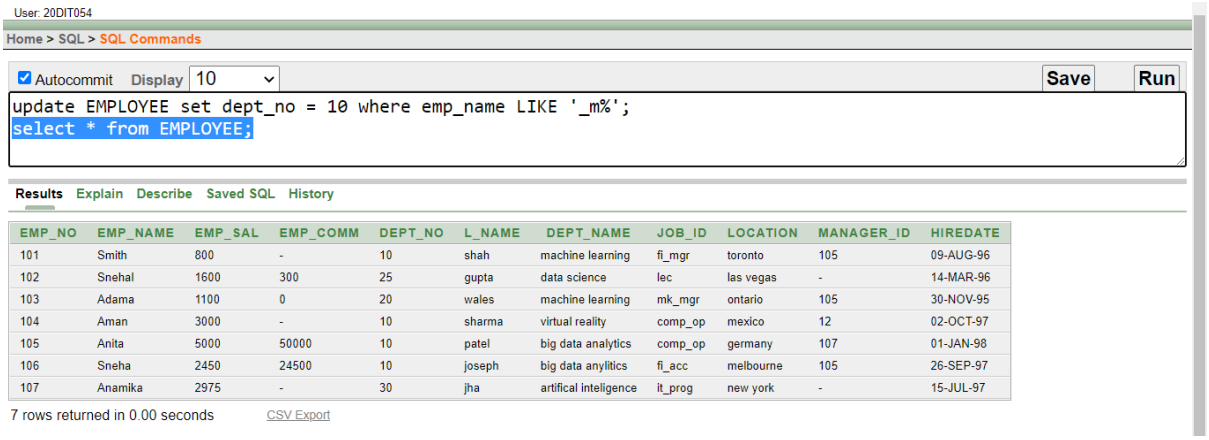
Autocommit Display 10 Save Run

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

Results Explain Describe Saved SQL History

2 row(s) updated.

0.09 seconds



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

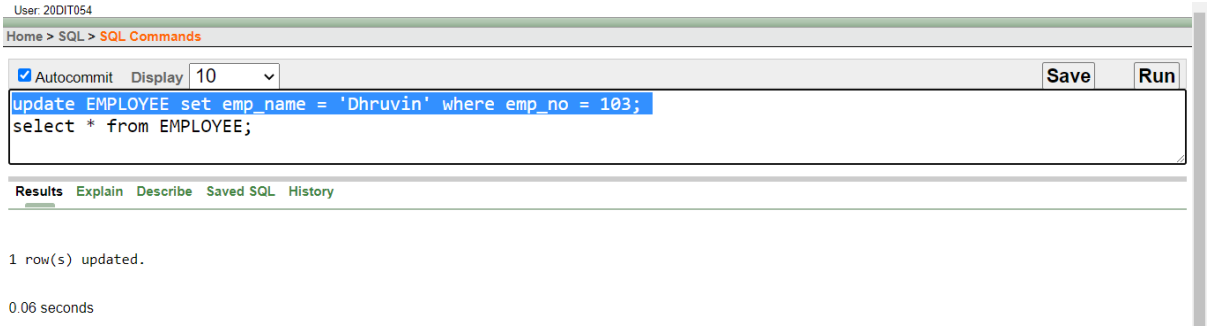
```
update EMPLOYEE set dept_no = 10 where emp_name LIKE '_m%';
select * from EMPLOYEE;
```

Results Explain Describe Saved SQL History

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

7 rows returned in 0.00 seconds CSV Export

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



User: 20DIT054

Home > SQL > SQL Commands

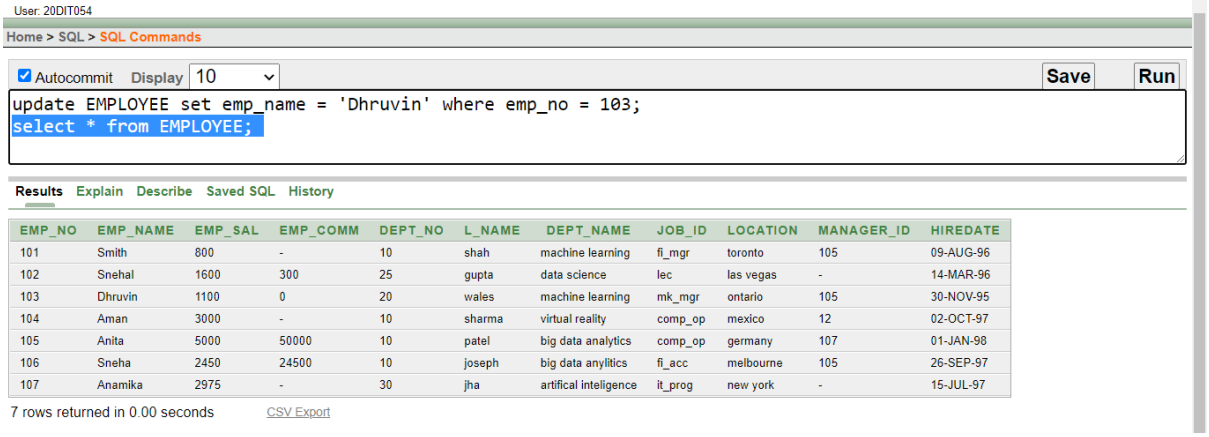
Autocommit Display 10 Save Run

```
update EMPLOYEE set emp_name = 'Dhruvin' where emp_no = 103;
select * from EMPLOYEE;
```

Results Explain Describe Saved SQL History

1 row(s) updated.

0.06 seconds



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

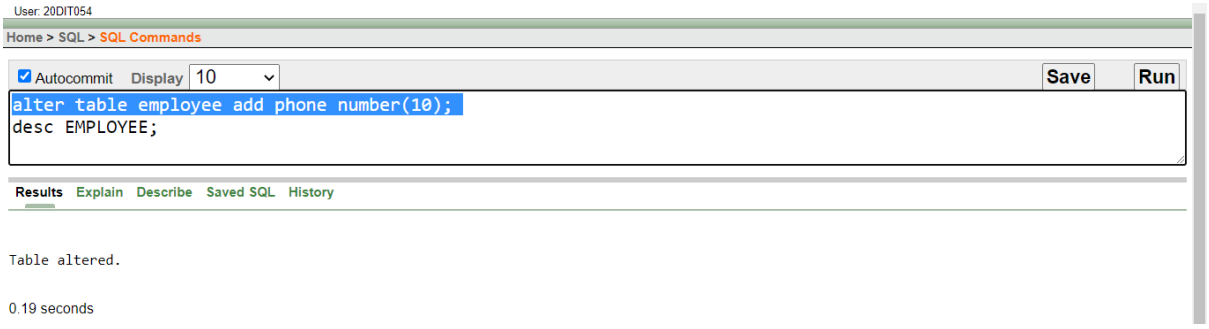
```
update EMPLOYEE set emp_name = 'Dhruvin' where emp_no = 103;
select * from EMPLOYEE;
```

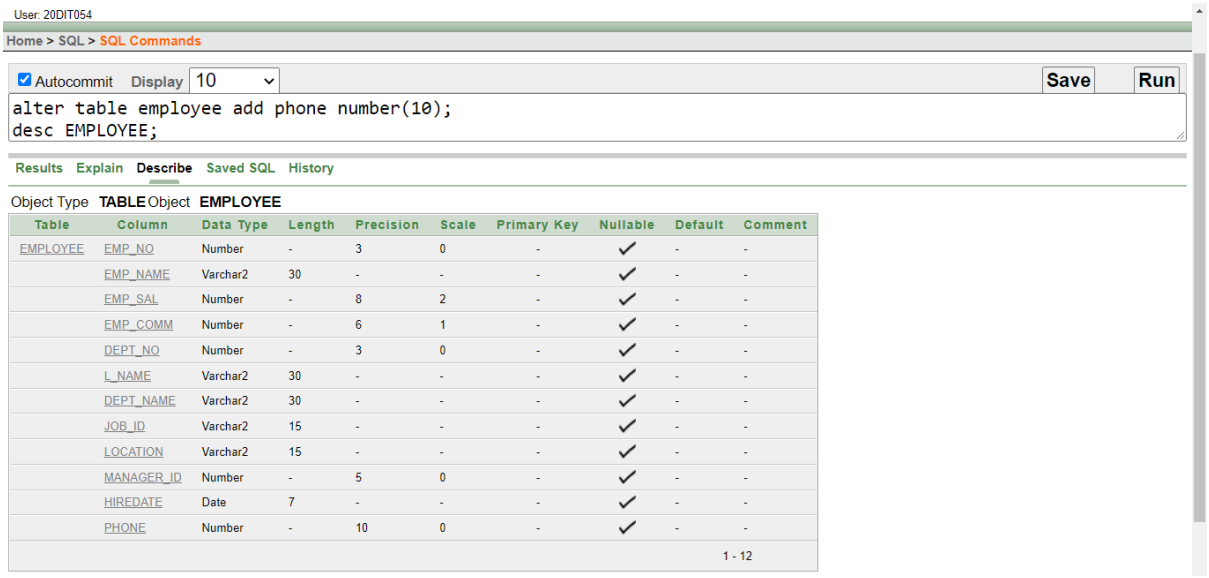
Results Explain Describe Saved SQL History

EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96
103	Dhruvin	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 anylitics	fi_acc	melbourne	105	26-SEP-97
107	Anamika	2975	-	30	jha	artifical inteligence	it_prog	new york	-	15-JUL-97

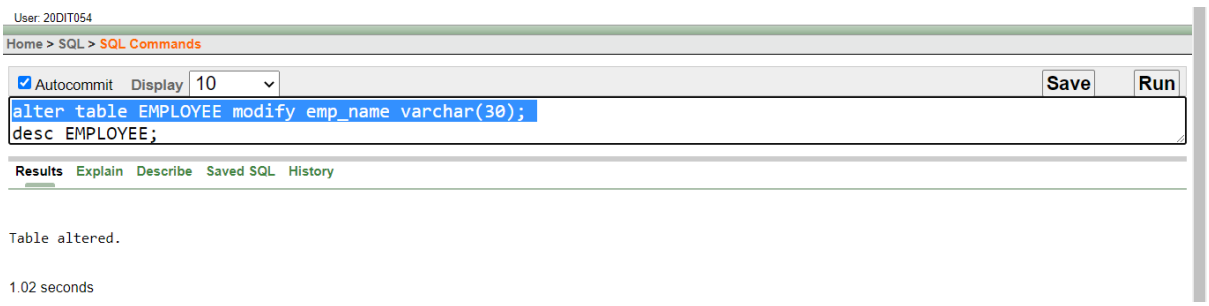
7 rows returned in 0.00 seconds CSV Export

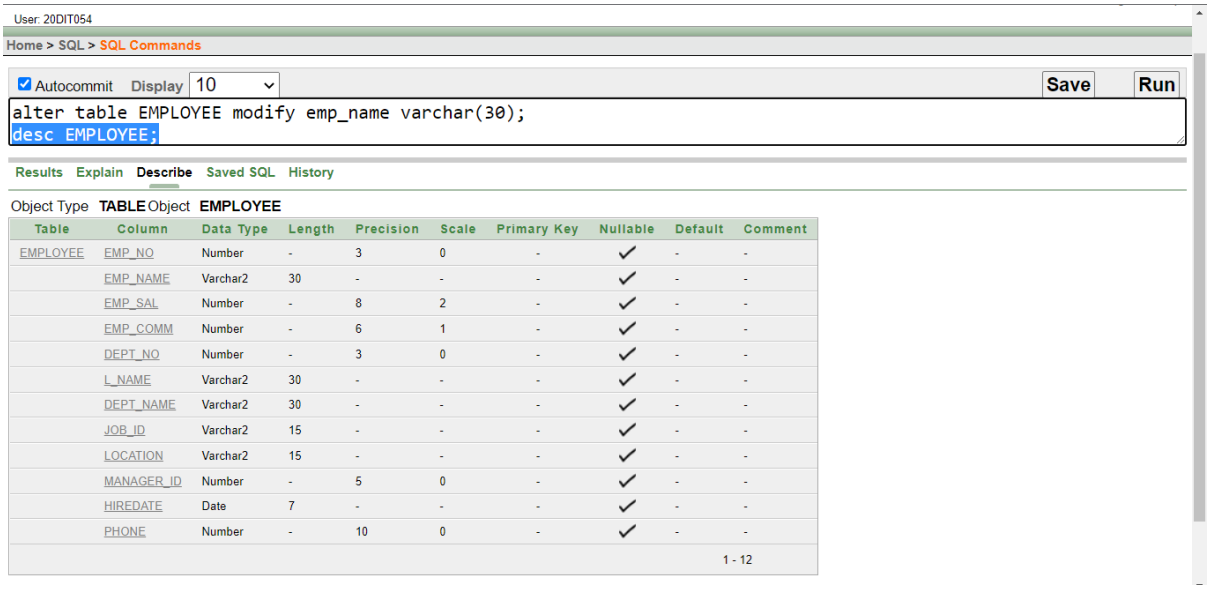
(16) Add one column phone to employee with size of column is 10.





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





User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
alter table EMPLOYEE modify emp_name varchar(30);
desc EMPLOYEE;
```

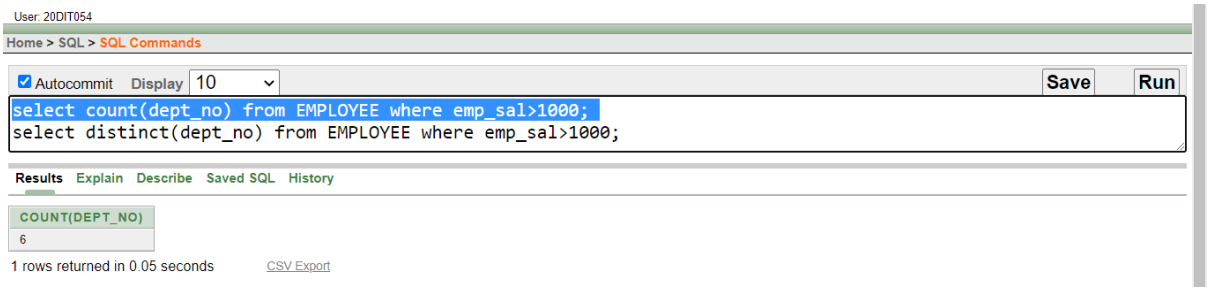
Results Explain Describe Saved SQL History

Object Type TABLE Object EMPLOYEE

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

1 - 12

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



User: 20DIT054

Home > SQL > SQL Commands

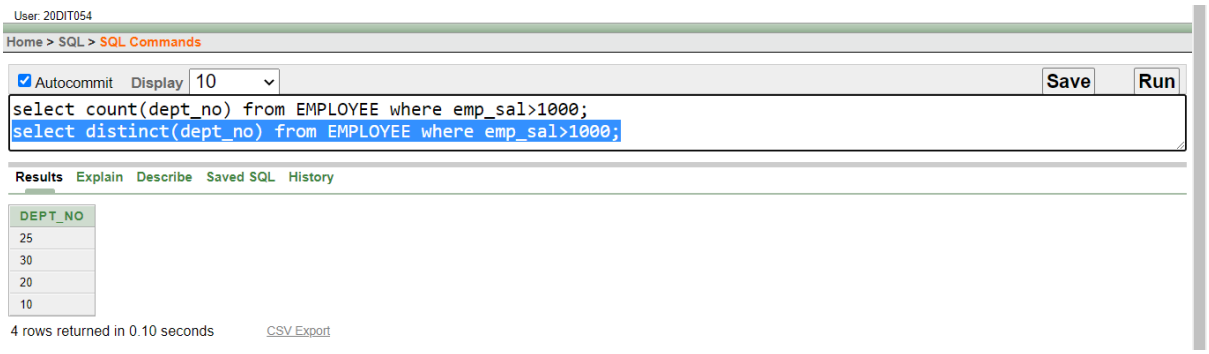
Autocommit Display 10 Save Run

```
select count(dept_no) from EMPLOYEE where emp_sal>1000;
select distinct(dept_no) from EMPLOYEE where emp_sal>1000;
```

Results Explain Describe Saved SQL History

COUNT(DEPT_NO)
6

1 rows returned in 0.05 seconds CSV Export



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

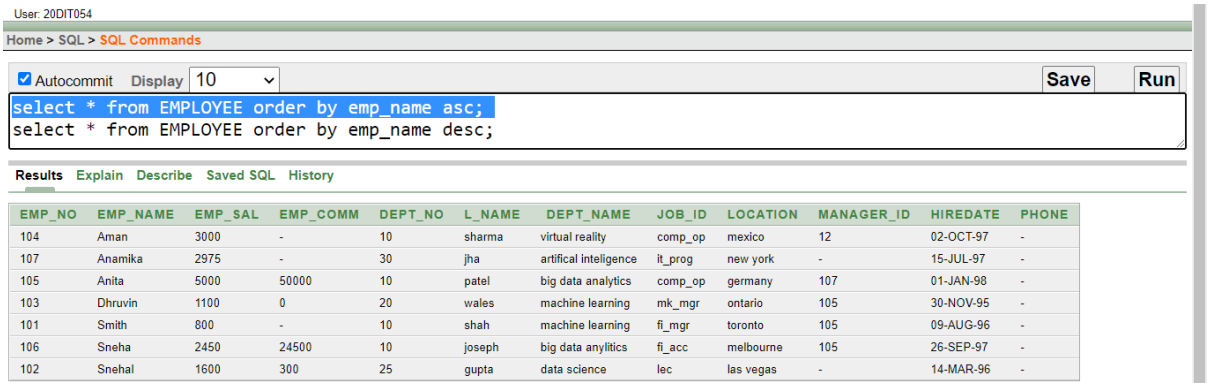
```
select count(dept_no) from EMPLOYEE where emp_sal>1000;
select distinct(dept_no) from EMPLOYEE where emp_sal>1000;
```

Results Explain Describe Saved SQL History

DEPT_NO
25
30
20
10

4 rows returned in 0.10 seconds CSV Export

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



User: 20DIT054

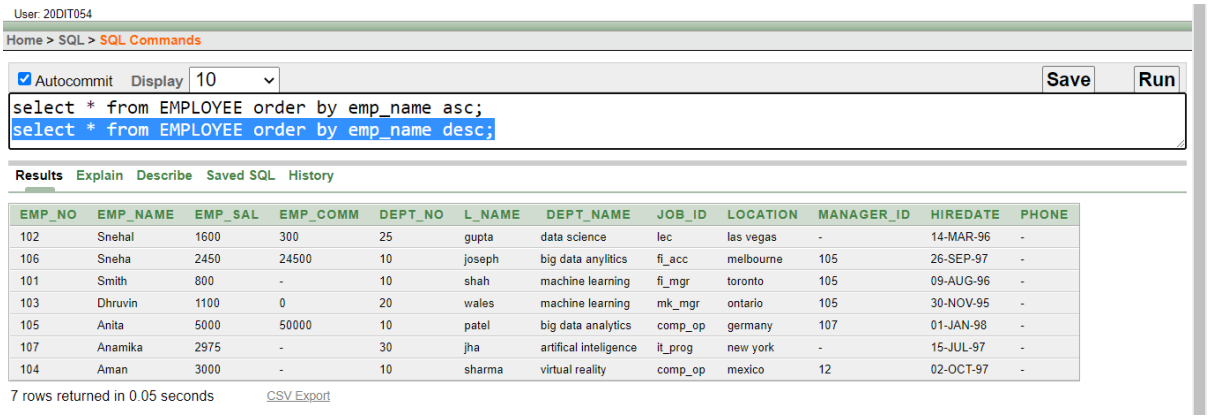
Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select * from EMPLOYEE order by emp_name asc;
select * from EMPLOYEE order by emp_name desc;
```

Results Explain Describe Saved SQL History

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



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

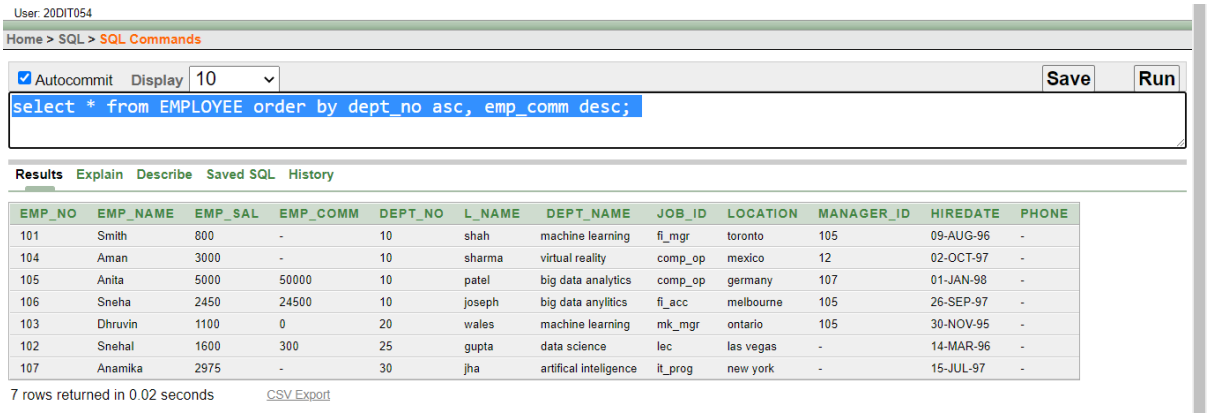
```
select * from EMPLOYEE order by emp_name asc;
select * from EMPLOYEE order by emp_name desc;
```

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
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
106	Sneha	2450	24500	10	joseph	big data anylitics	fi_acc	melbourne	105	26-SEP-97	-
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-
103	Dhruvin	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	-
107	Anamika	2975	-	30	jha	artifical inteligence	it_prog	new york	-	15-JUL-97	-
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-

7 rows returned in 0.05 seconds CSV Export

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



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

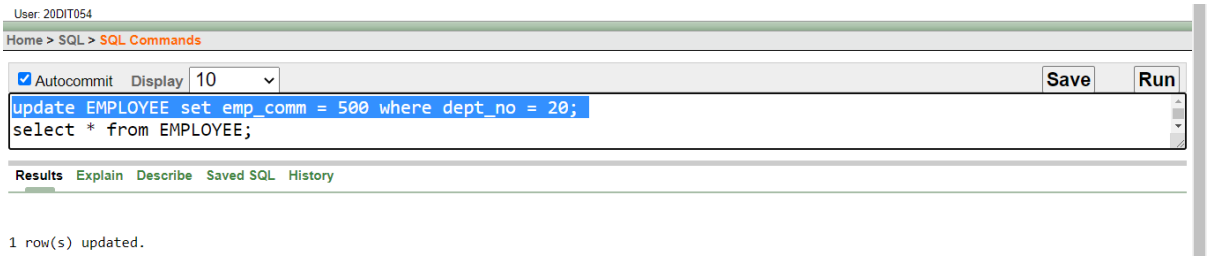
```
select * from EMPLOYEE order by dept_no asc, emp_comm desc;
```

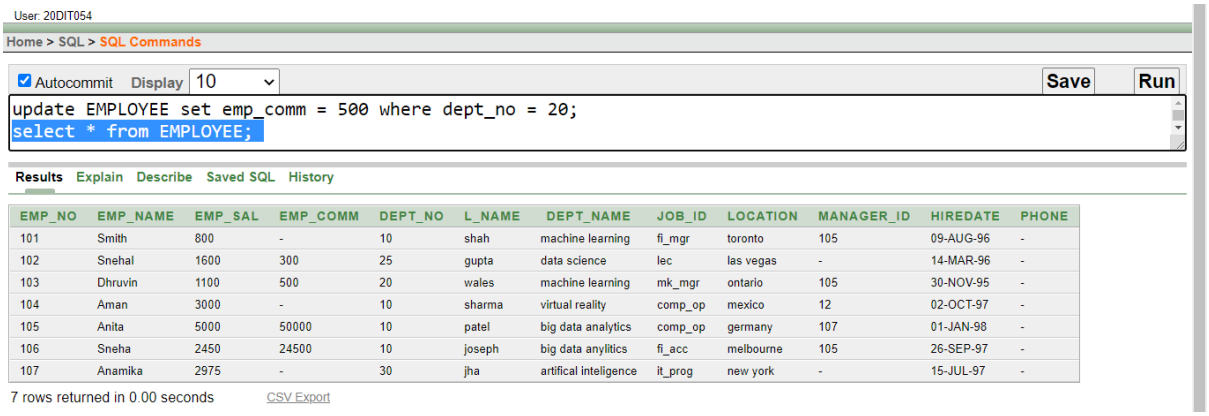
Results Explain Describe Saved SQL History

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

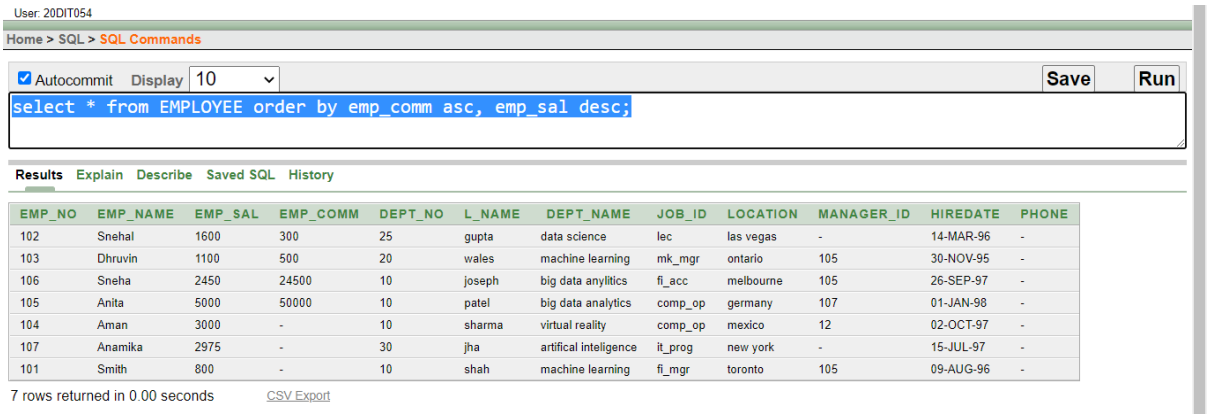
7 rows returned in 0.02 seconds CSV Export

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

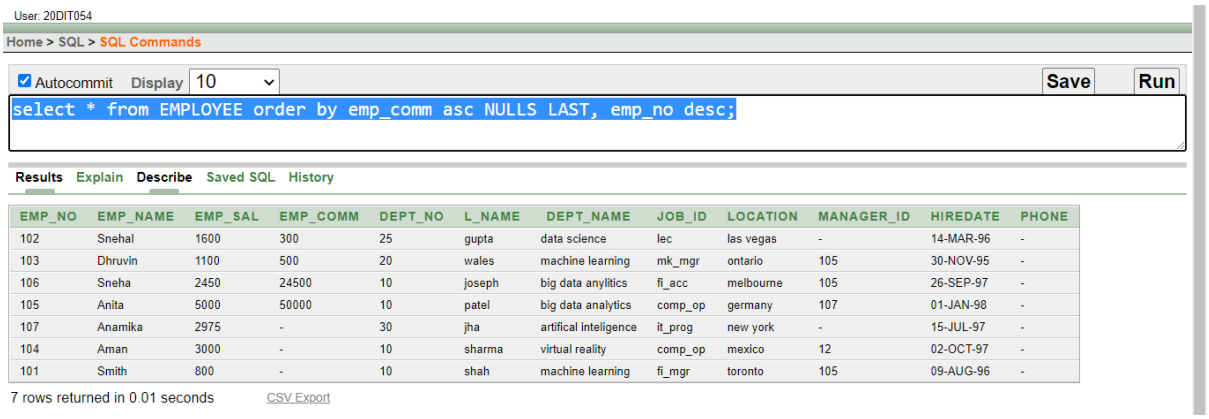




(22) Display the emp_comm in ascending order with null value first and accordingly sort employee salary in descending order.



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



User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select * from EMPLOYEE order by emp_comm asc 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
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-
103	Dhruvin	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-
106	Sneha	2450	24500	10	joseph	big data anylitics	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	artifical inteligence	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	fi_mgr	toronto	105	09-AUG-96	-

7 rows returned in 0.01 seconds CSV Export

Conclusion :

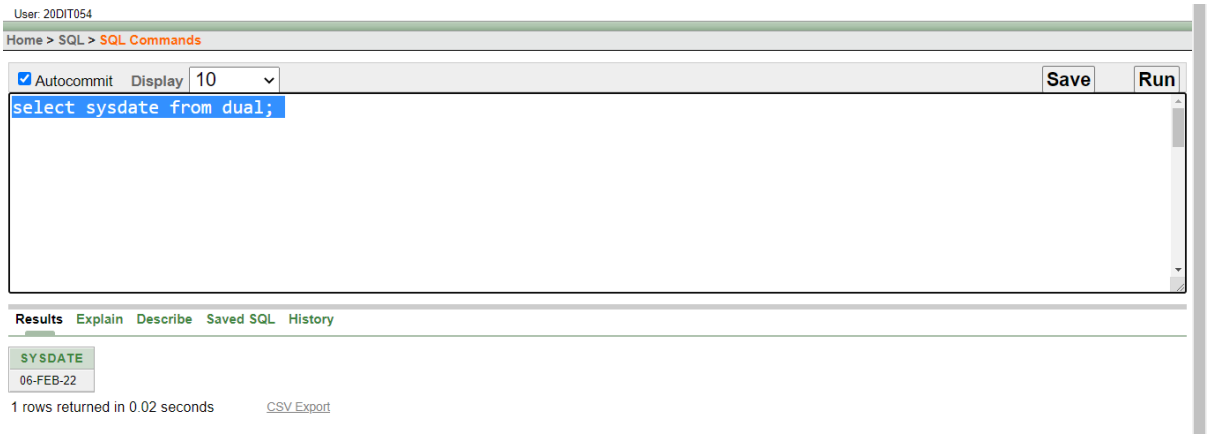
We learn about various data manipulation commands and sorting concept on all created tables.

Practical-6

Aim: To study Single-row functions.

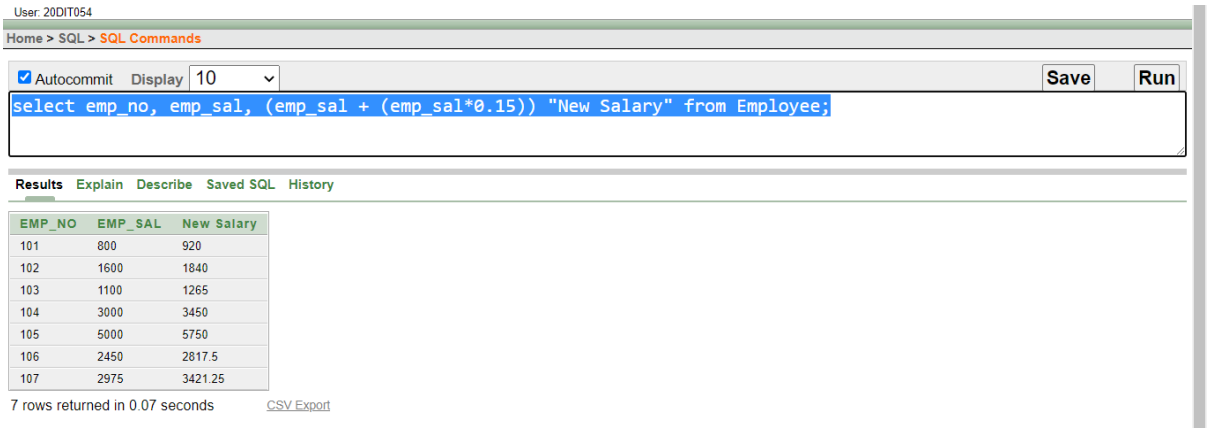
Queries :

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

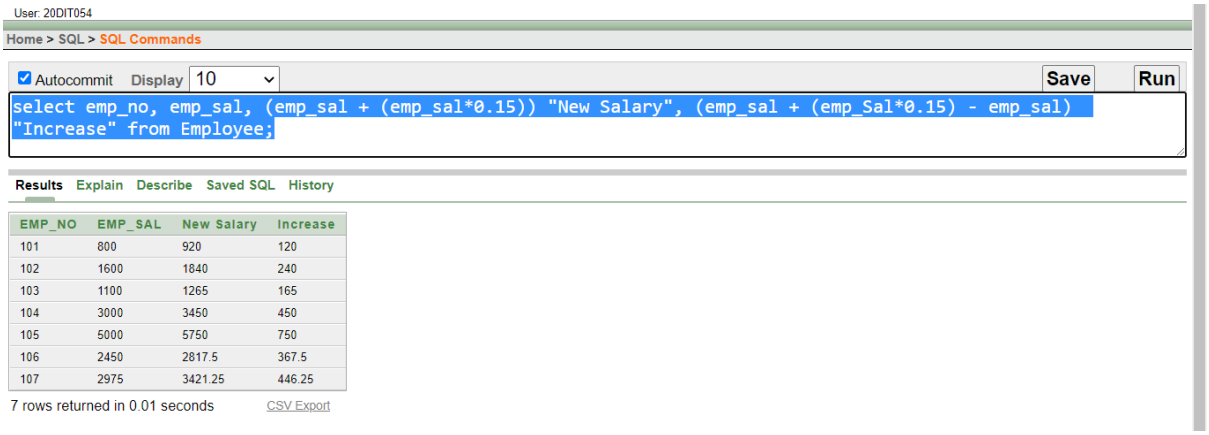


(2) For each employee, display the employee number, salary, and salary increased by 15% and expressed as a whole number.

Label the column New Salary



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


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

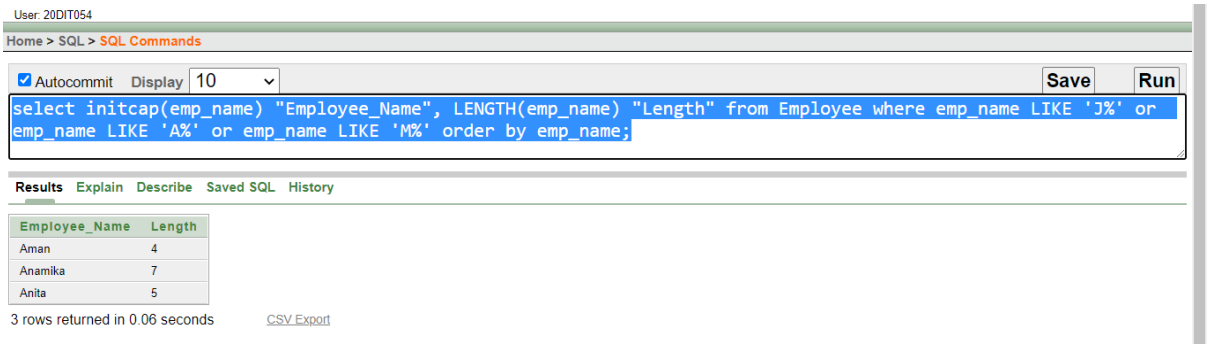
```
select emp_no, emp_sal, (emp_sal + (emp_sal*0.15)) "New Salary", (emp_sal + (emp_Sal*0.15) - emp_sal)
"Increase" from Employee;
```

Results Explain Describe Saved SQL History

EMP_NO	EMP_SAL	New Salary	Increase
101	800	920	120
102	1600	1840	240
103	1100	1265	165
104	3000	3450	450
105	5000	5750	750
106	2450	2817.5	367.5
107	2975	3421.25	446.25

7 rows returned in 0.01 seconds CSV Export

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


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
select initcap(emp_name) "Employee_Name", LENGTH(emp_name) "Length" from Employee where emp_name LIKE 'J%' or
emp_name LIKE 'A%' or emp_name LIKE 'M%' order by emp_name;
```

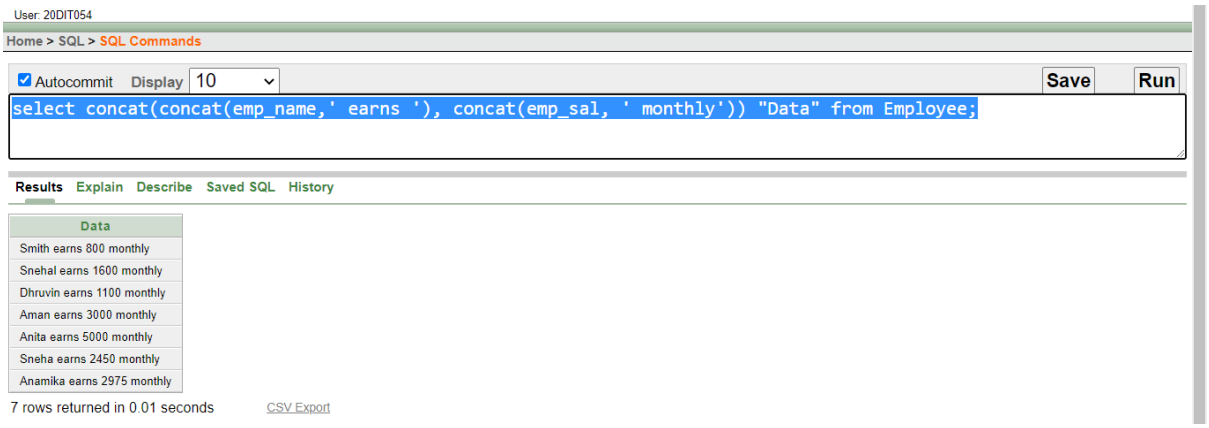
Results Explain Describe Saved SQL History

Employee_Name	Length
Aman	4
Anamika	7
Anita	5

3 rows returned in 0.06 seconds CSV Export

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

<employee last name> earns <salary> monthly


User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

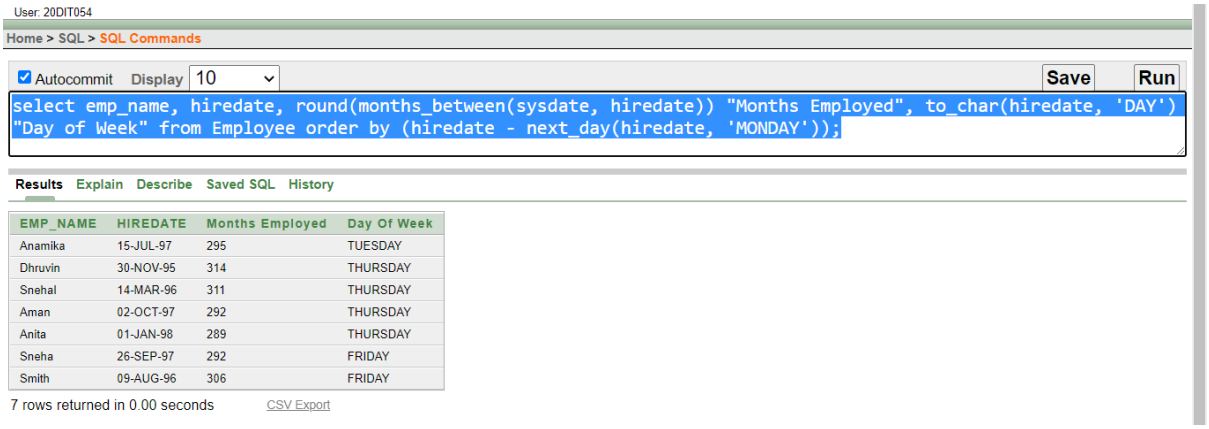
```
select concat(concat(emp_name,' earns '), concat(emp_sal, ' monthly')) "Data" from Employee;
```

Results Explain Describe Saved SQL History

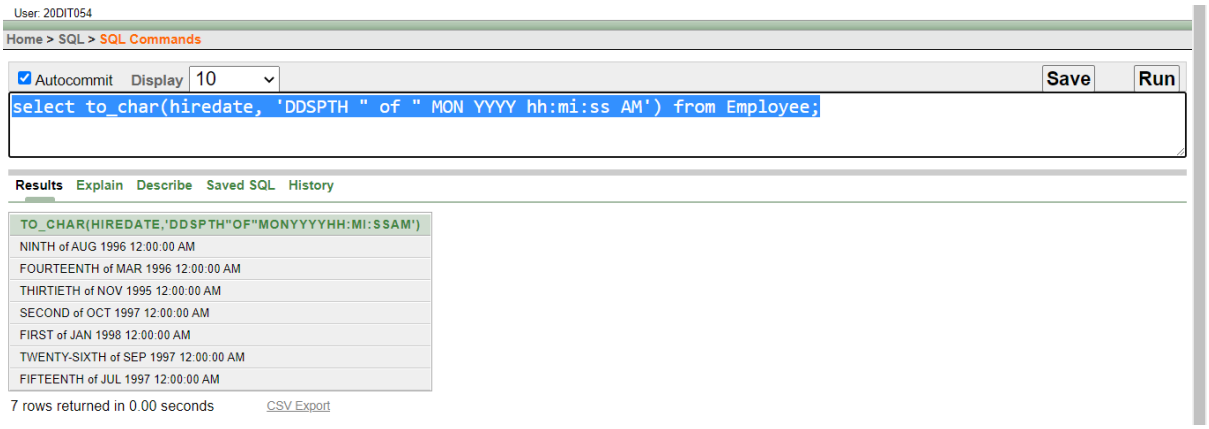
Data
Smith earns 800 monthly
Snehal earns 1600 monthly
Dhruvin earns 1100 monthly
Aman earns 3000 monthly
Anita earns 5000 monthly
Sneha earns 2450 monthly
Anamika earns 2975 monthly

7 rows returned in 0.01 seconds CSV Export

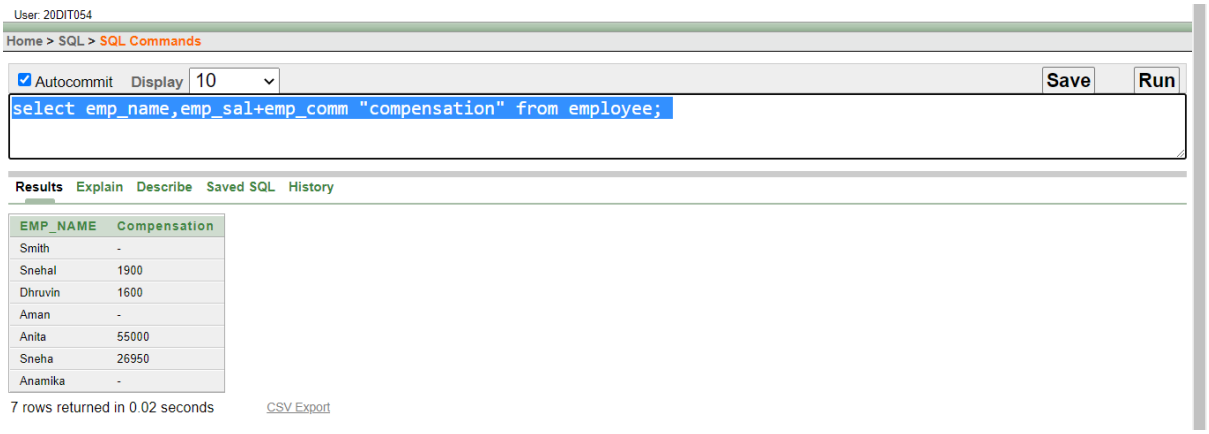
(6) Display the name, date, number of months employed and day of the week on which the employee has started. Order the results by the day of the week starting with Monday.



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



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



Conclusion :

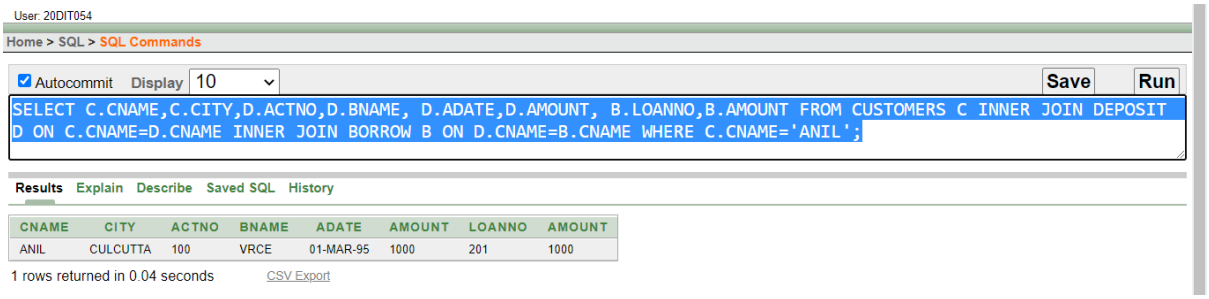
In this practical I learned about different types of single row functions.

Practical-7

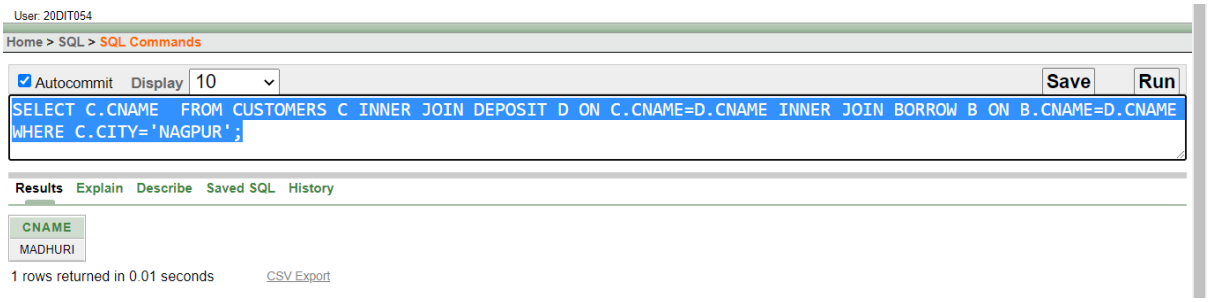
Aim: Displaying data from Multiple Tables (join)

Queries :

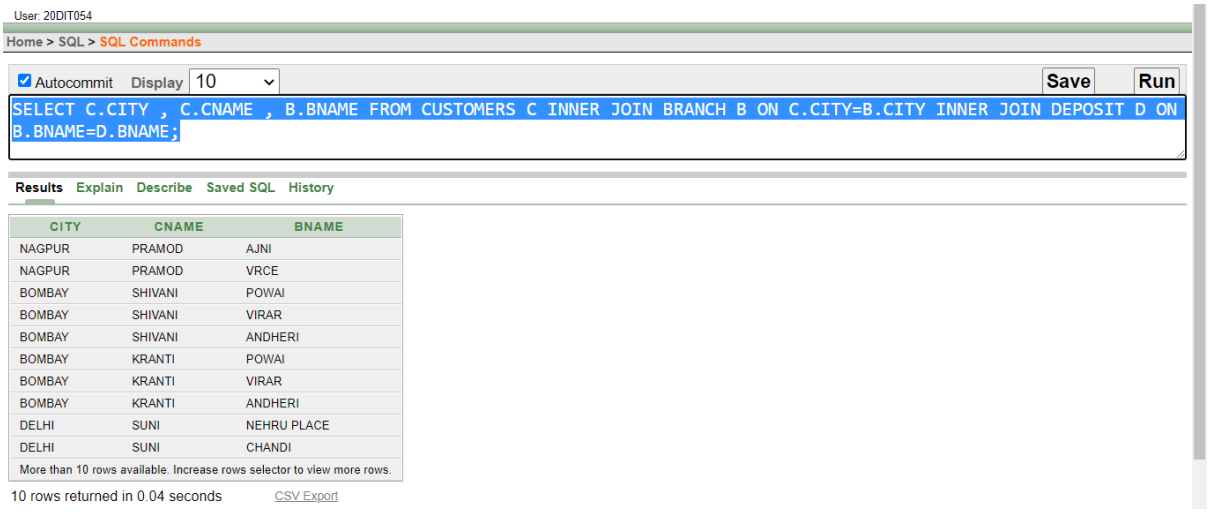
(1) Give details of customers ANIL.



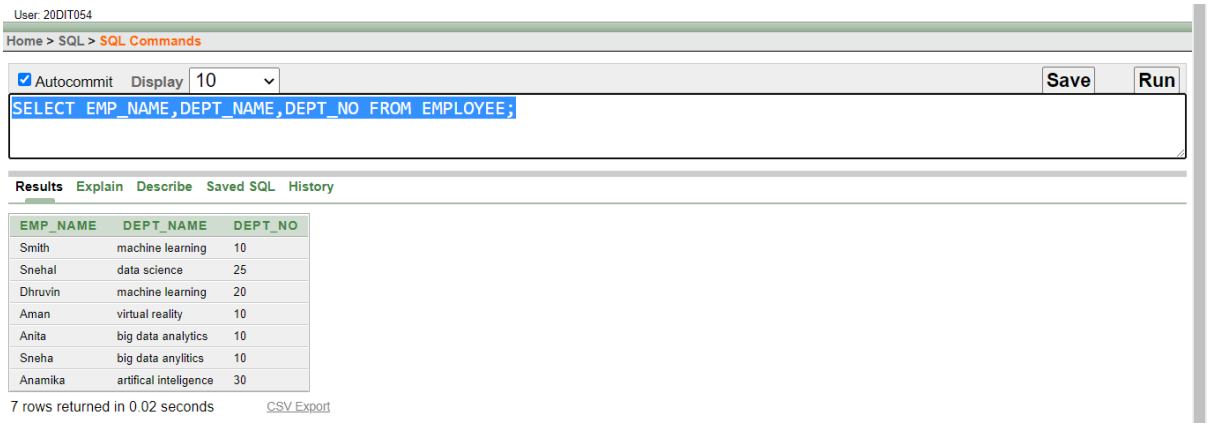
(2) Give name of customer who are borrowers and depositors and having living city Nagpur



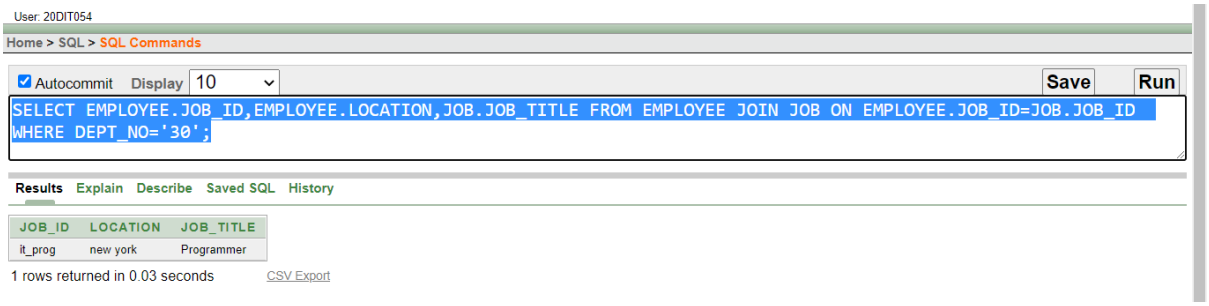
(3) Give city as their city name of customers having same living branch.



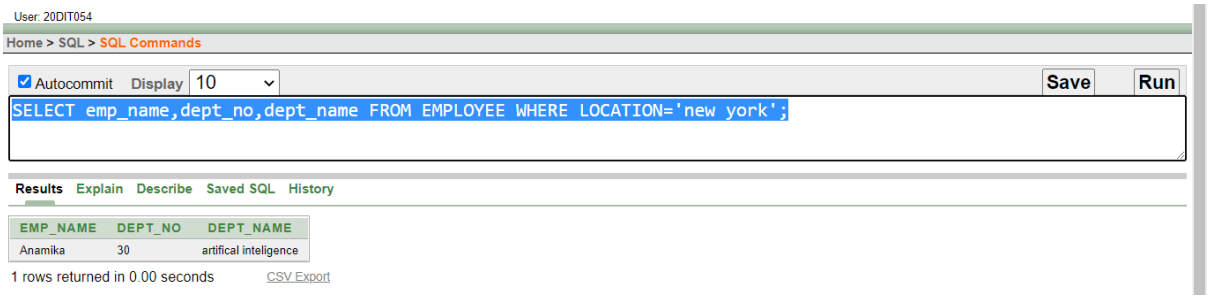
(4) Write a query to display the last name, department number, and department name for all employees.



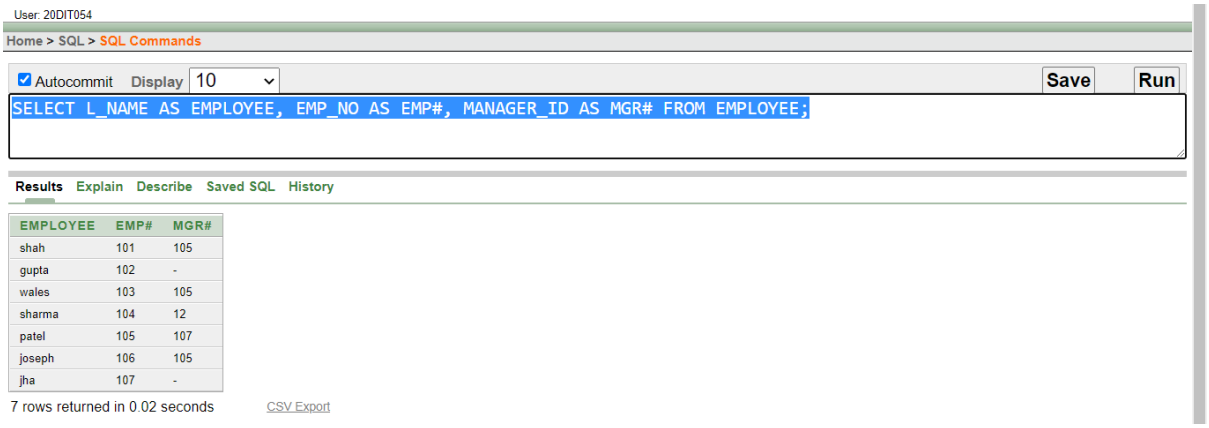
(5) Create a unique listing of all jobs that are in department 30. Include the location of the department in the output



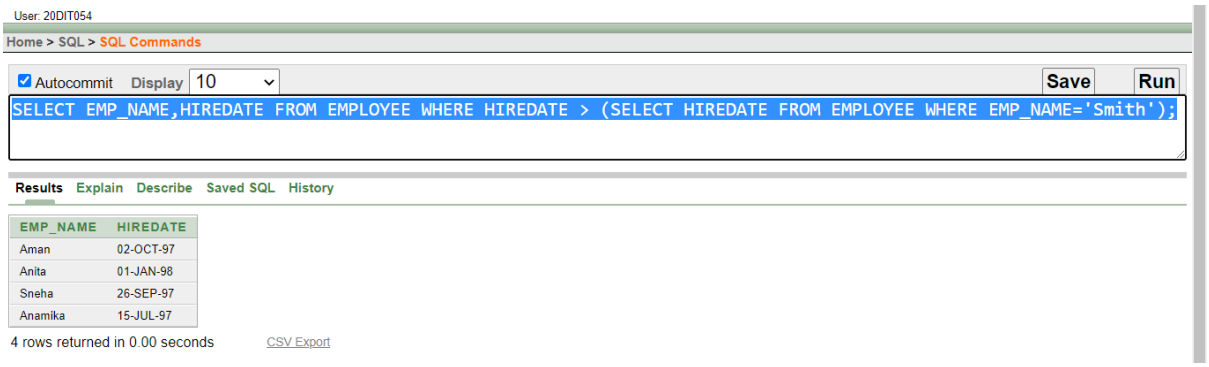
(6) Write a query to display the employee name, department number, and department name for all employees who work in NEW YORK.



(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.



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



Conclusion :

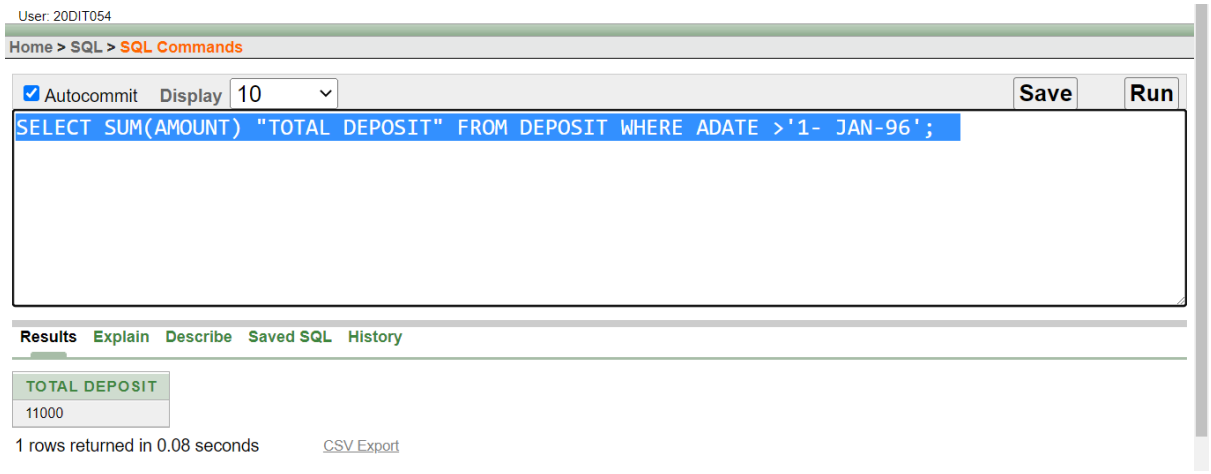
In this practical I learned about how to fetch data from multiple tables.

Practical-8

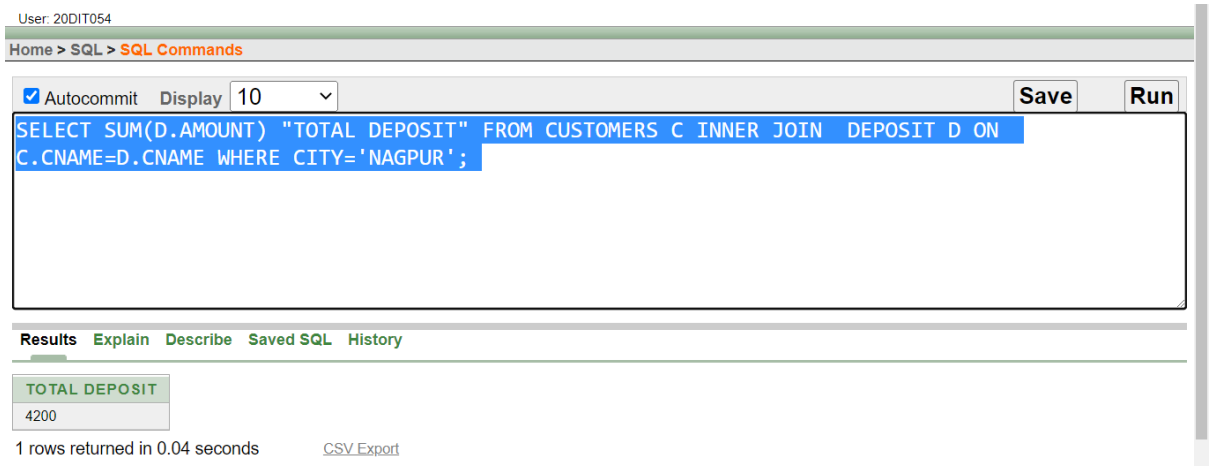
Aim: To apply the concept of Aggregating Data using Group functions.

Queries :

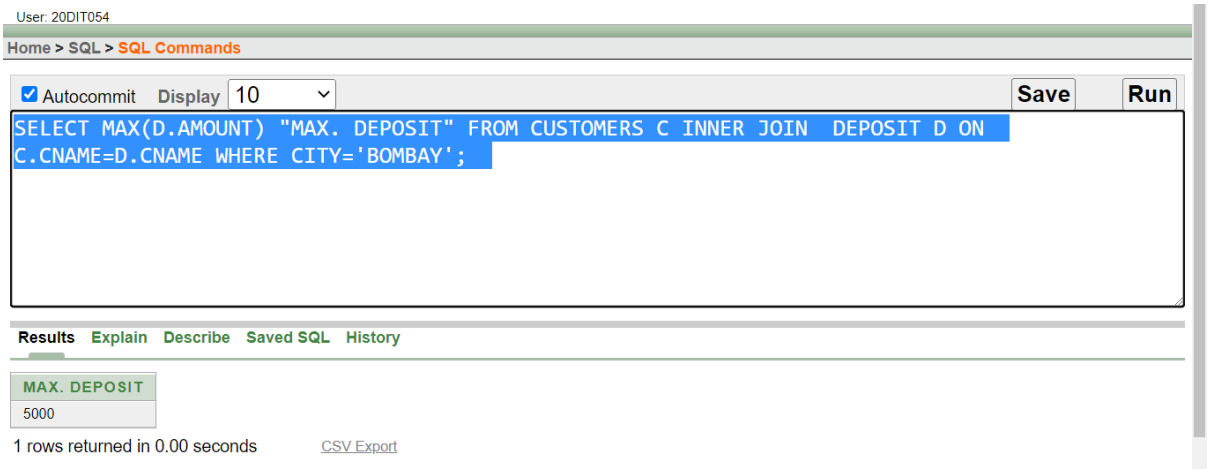
(1) List total deposit of customer having account date after 1-jan-96.



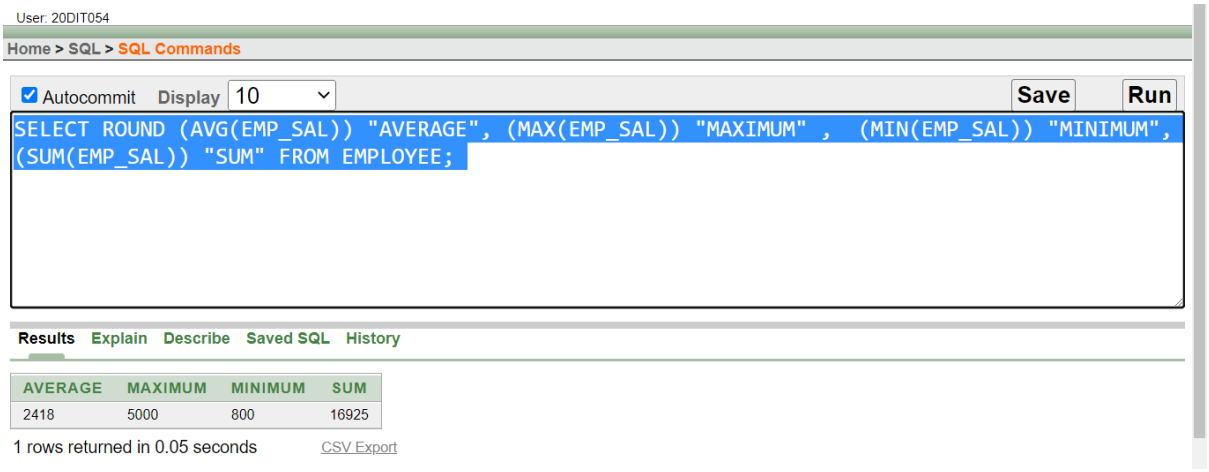
(2) List total deposit of customers living in city Nagpur.



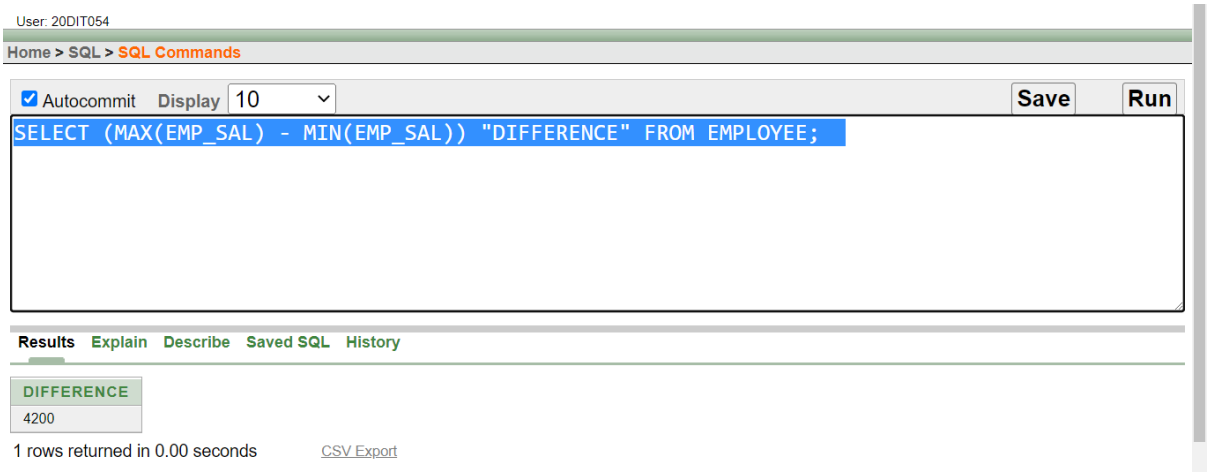
(3) List maximum deposit of customers living in bombay.



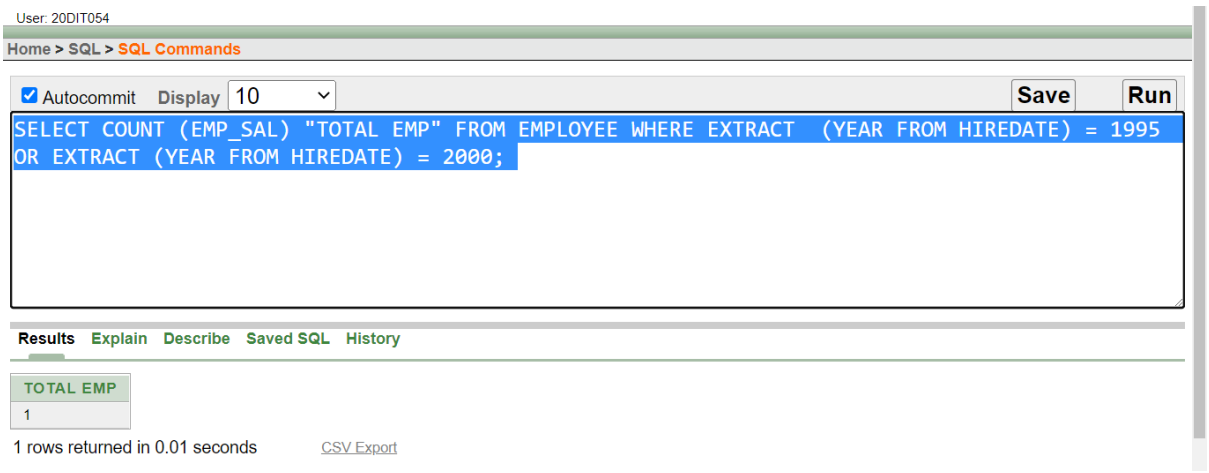
(4) Display the highest, lowest, sum, and average salary of all employees. Label the columns Maximum, Minimum, Sum, and Average, respectively. Round your results to the nearest whole number.



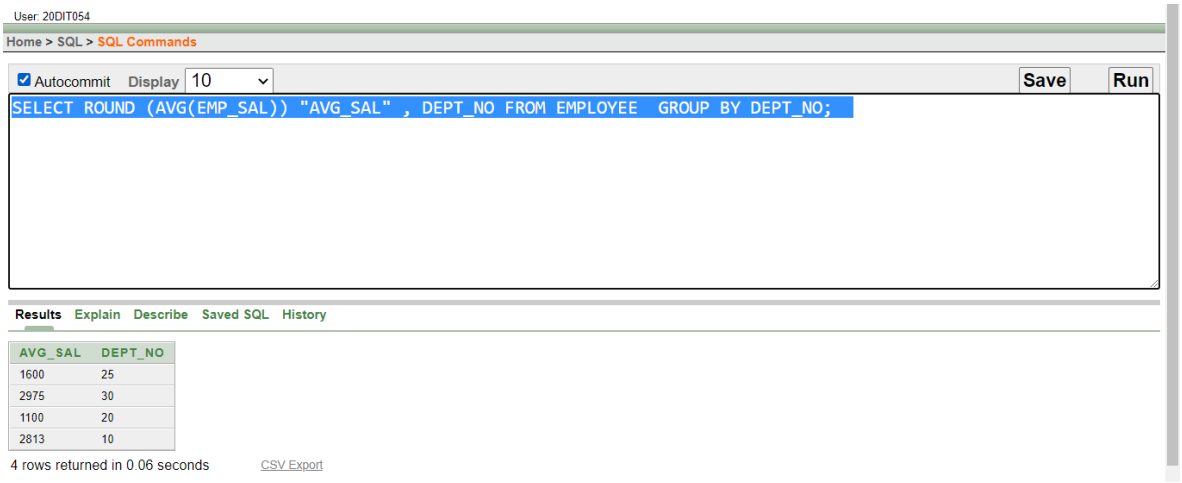
(5) Write a query that displays the difference between the highest and lowest salaries. Label the column DIFFERENCE.



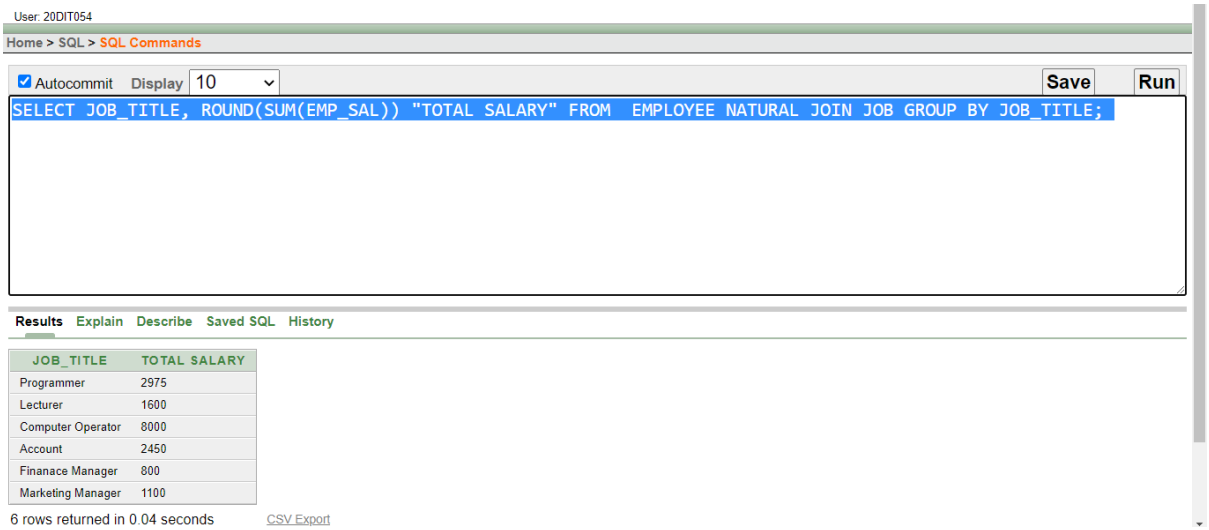
(6) Create a query that will display the total number of employees and, of that total, the number of employees hired in 1995, 1996, 1997, and 1998



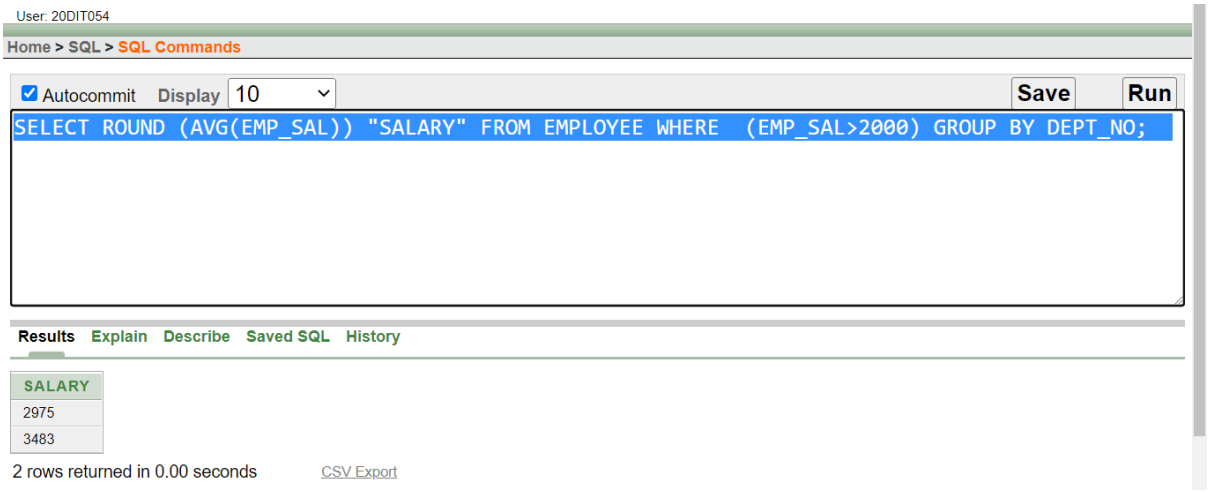
(7) Find the average salaries for each department without displaying the respective department numbers.



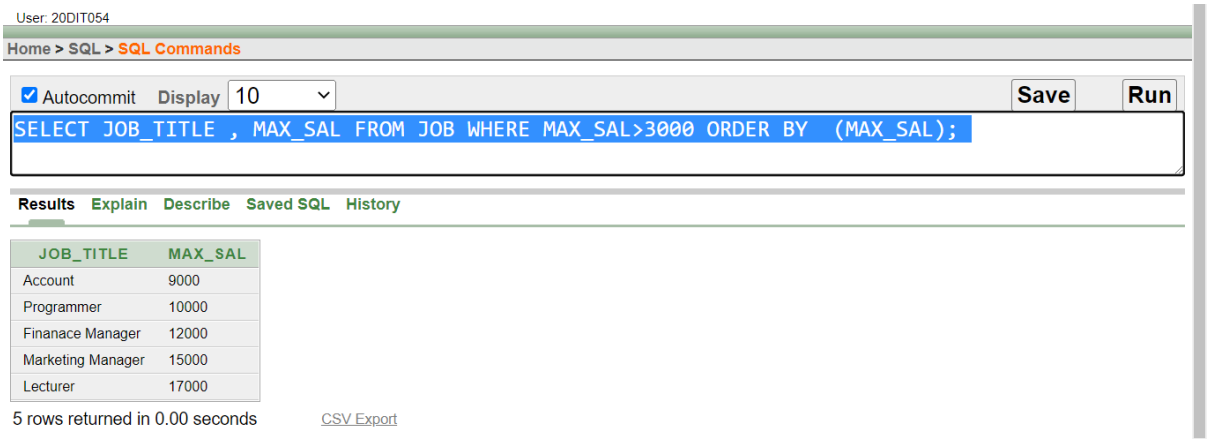
(8) Write a query to display the total salary being paid to each job title, within each department.



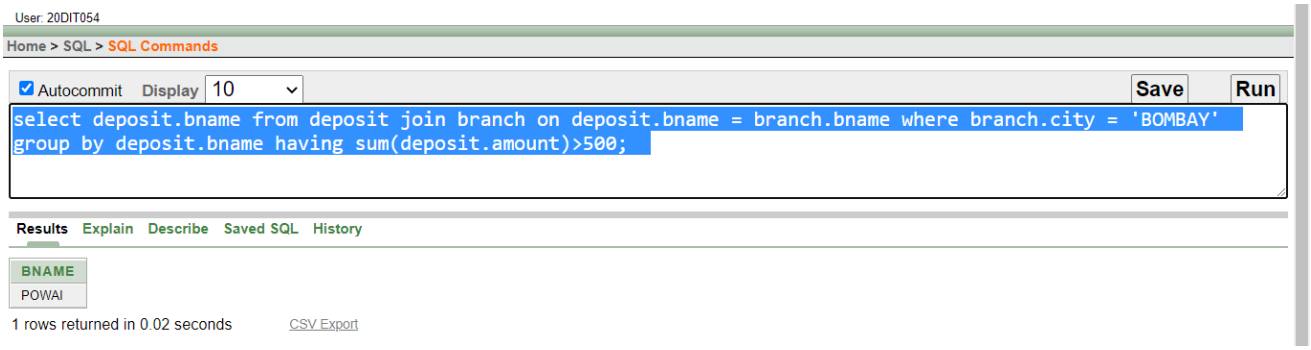
(9) Find the average salaries > 2000 for each department without displaying the respective department numbers.



(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.



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



Conclusion :

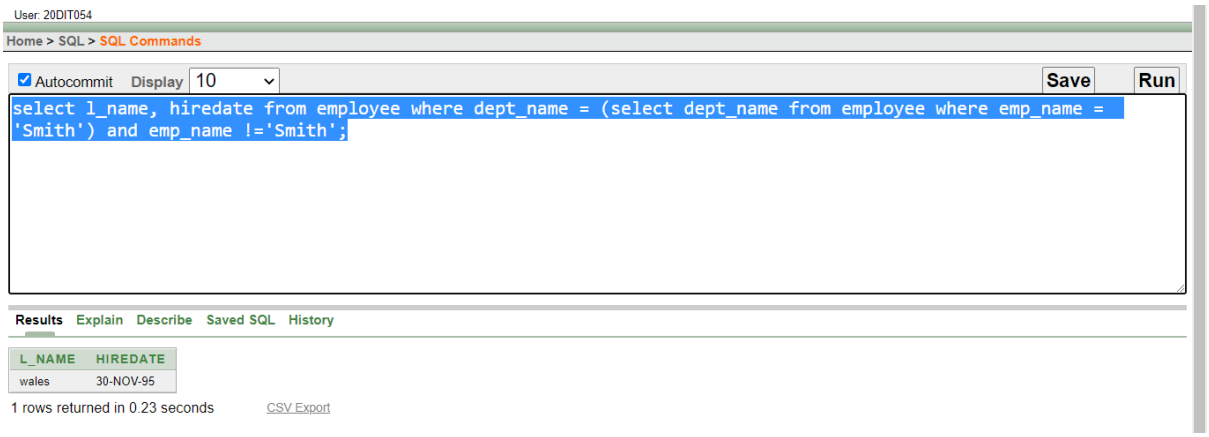
In this practical I learned about concept of Aggregating Data using Group functions.

Practical-9

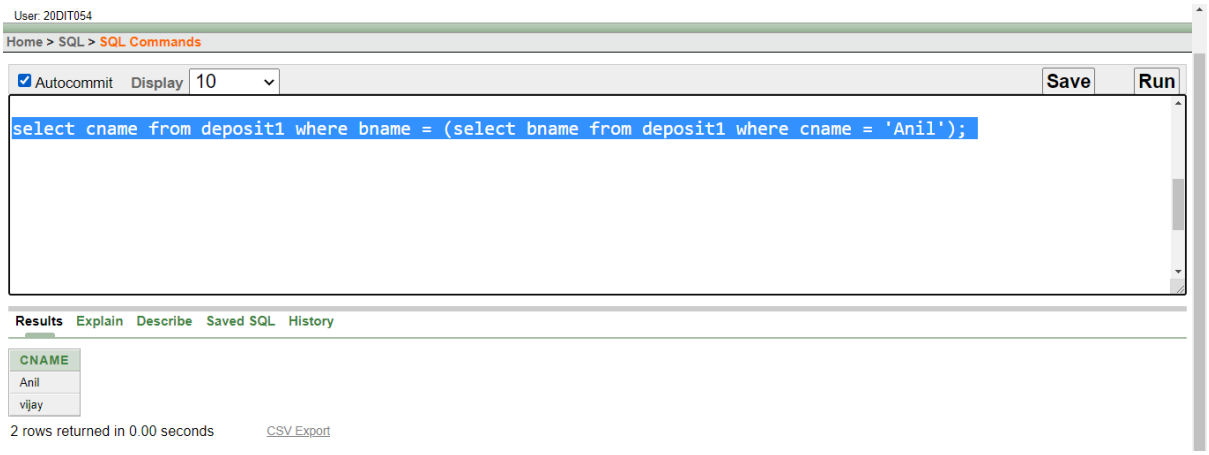
Aim: To solve queries using the concept of sub query.

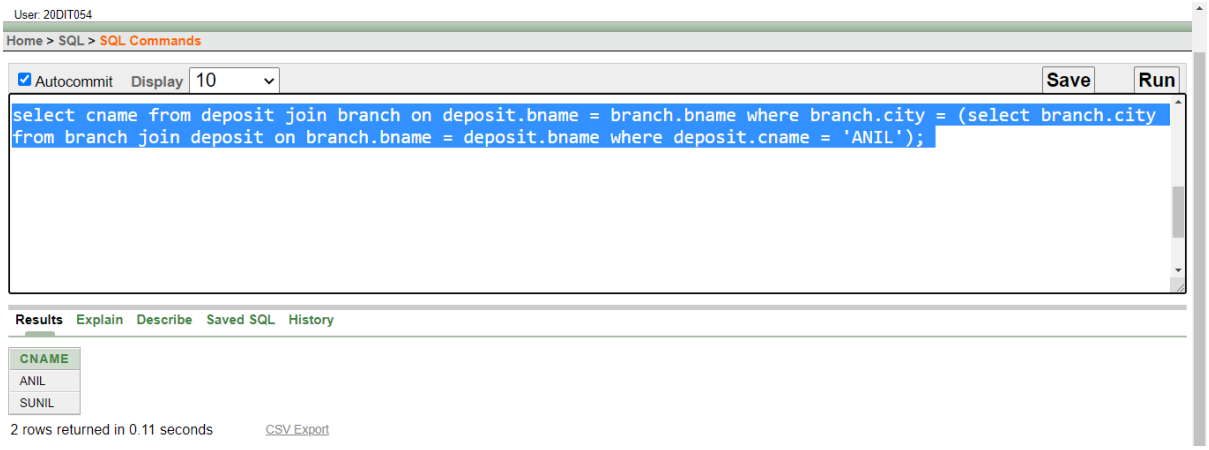
Queries :

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

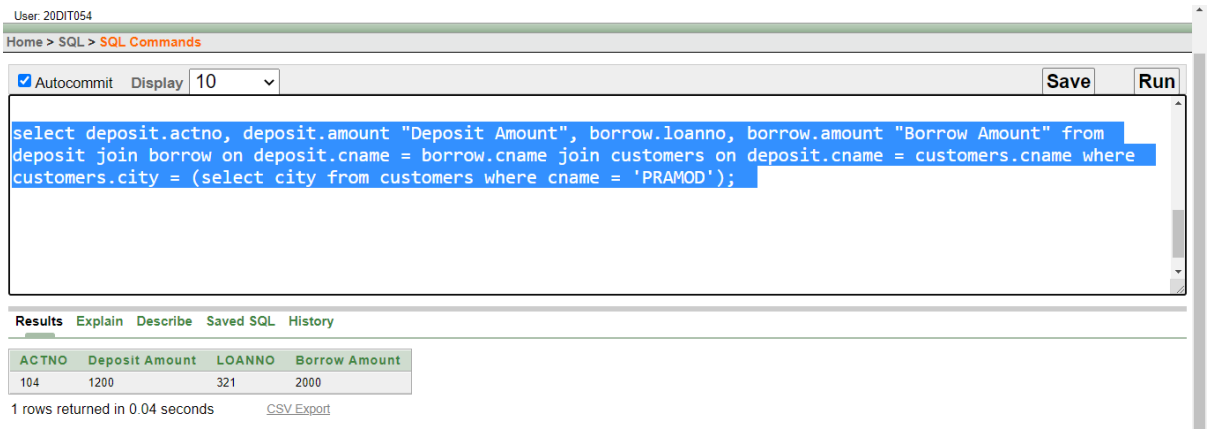


(2) Give name of customers who are depositors having same branch city of mr. sunil.

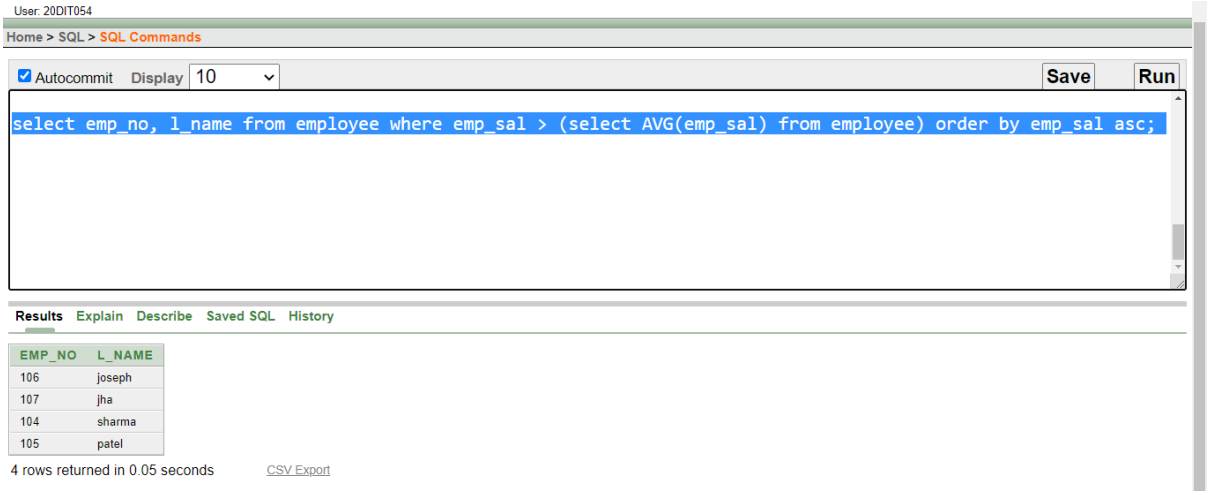




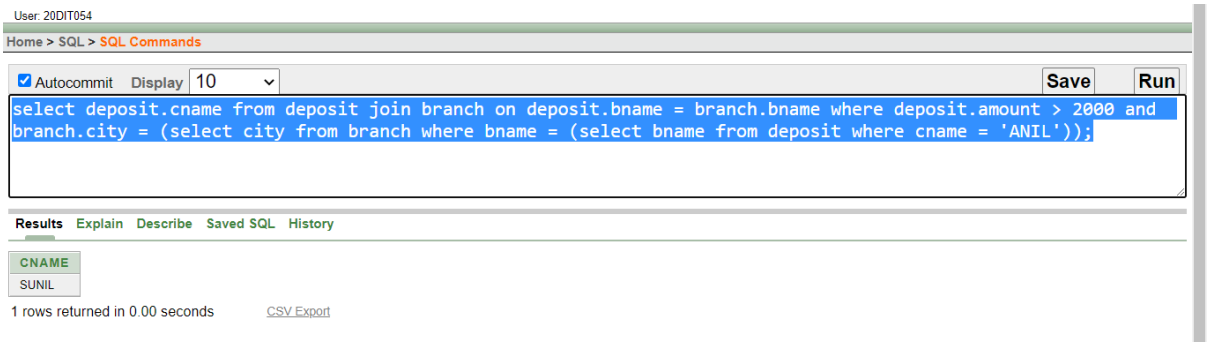
(3) Give deposit details and loan details of customer in same city where pramod is living.



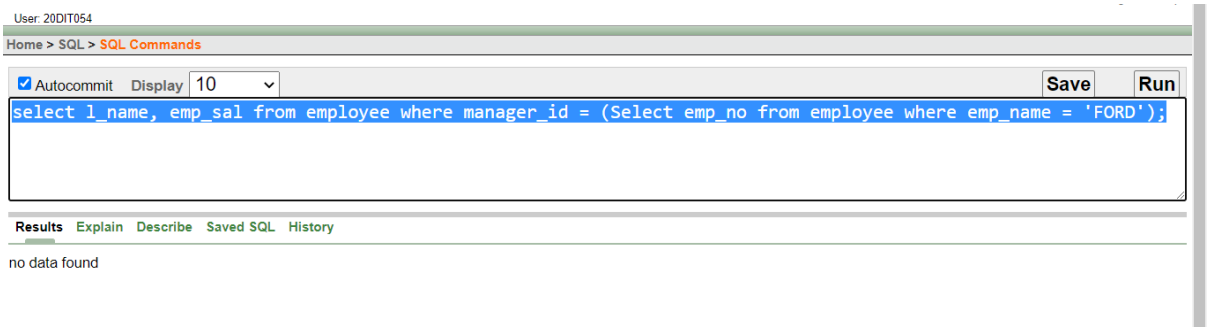
(4) Create a query to display the employee numbers and last names of all employees who earn more than the average salary. Sort the results in ascending order of salary.



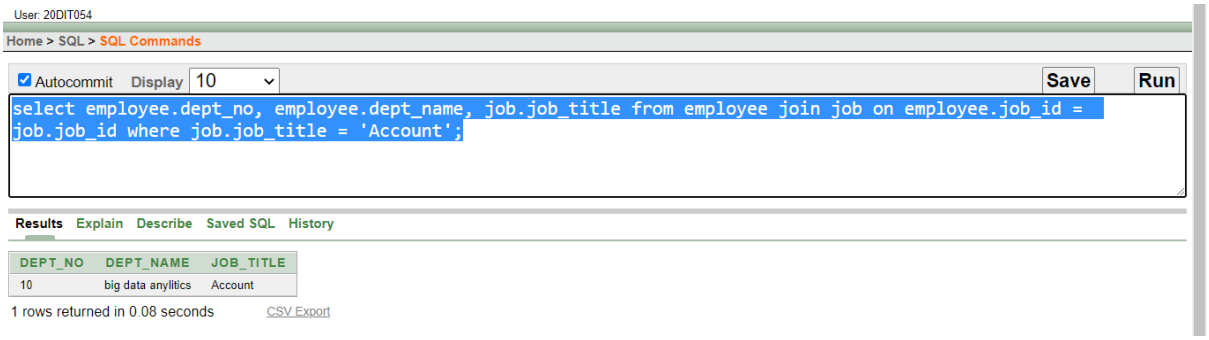
(5) Give names of depositors having same living city as mr. anil and having deposit amount greater than 2000



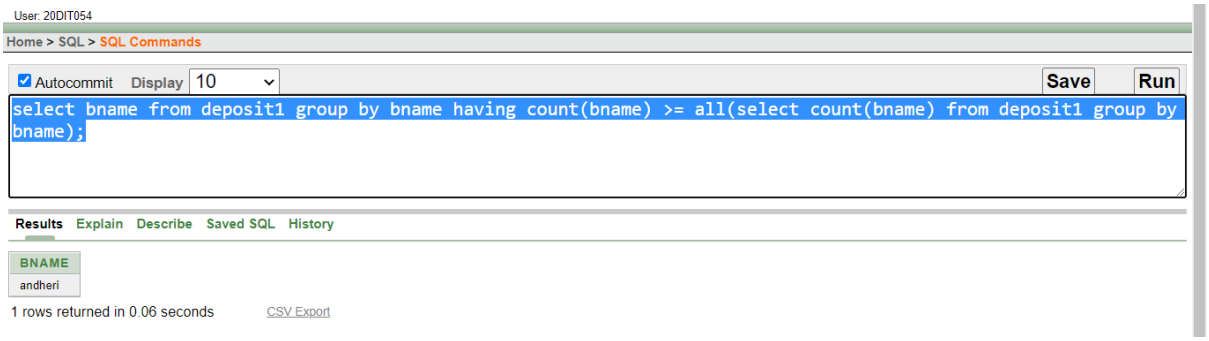
(6) Display the last name and salary of every employee who reports to ford.



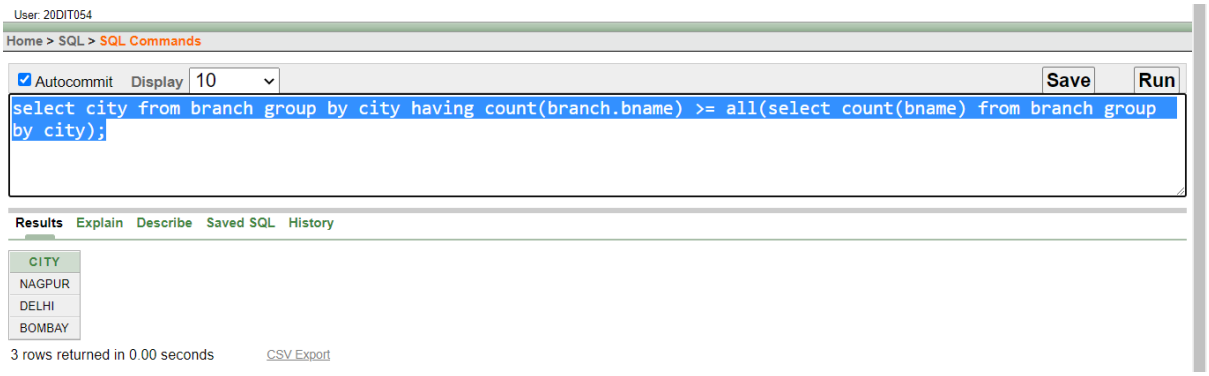
(7) Display the department number, name, and job for every employee in the Accounting department.



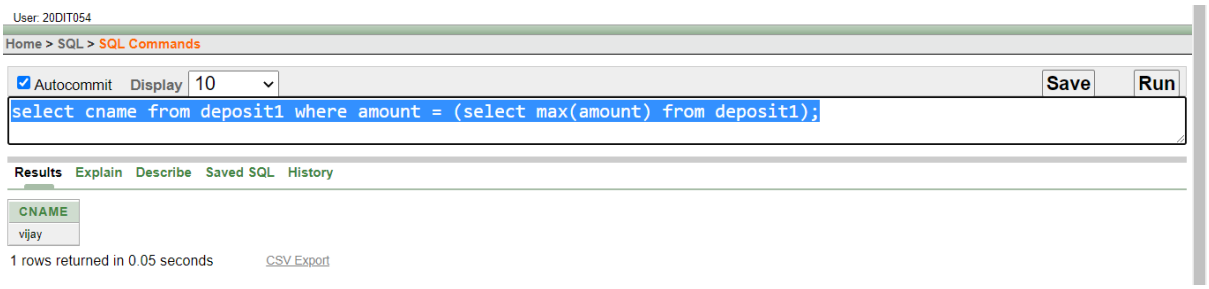
(8) List the name of branch having highest number of depositors.



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



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



Conclusion :

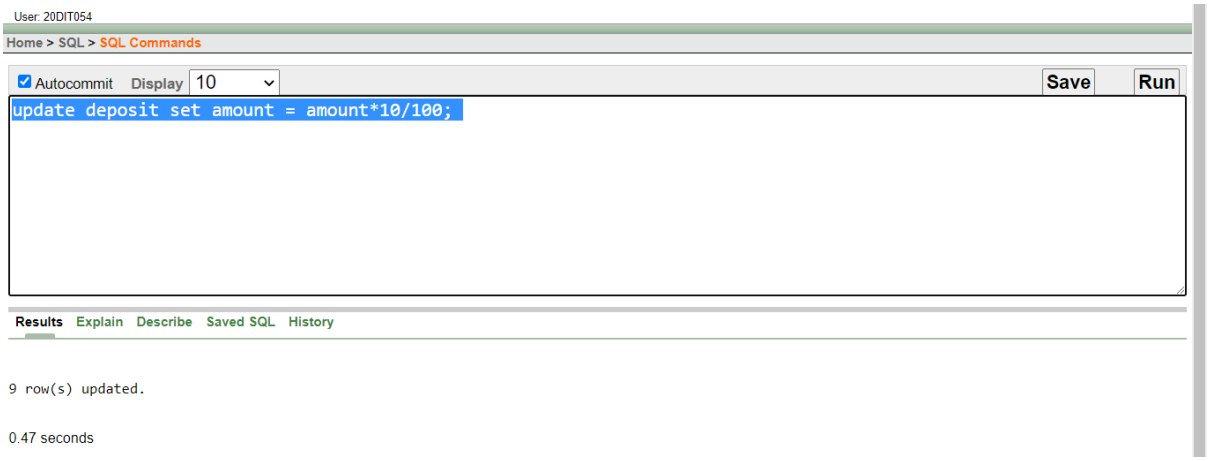
From this practical we learn about sub queries.

Practical-10

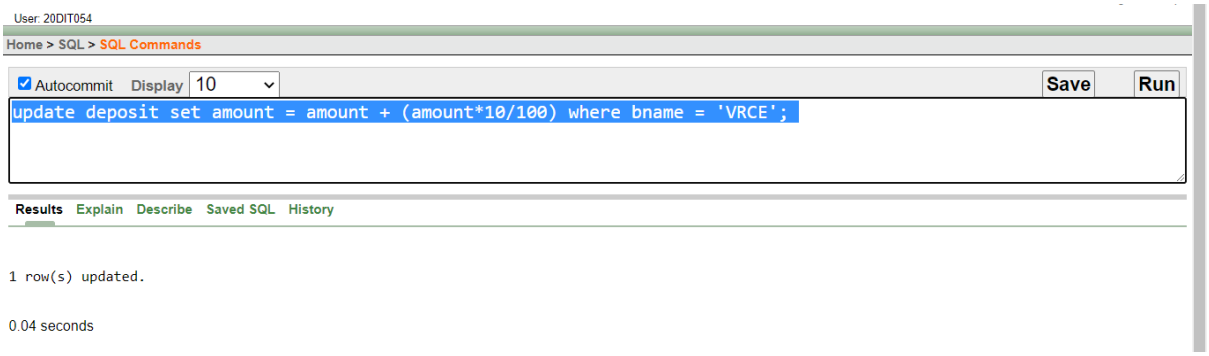
Aim: Manipulating Data

Queries :

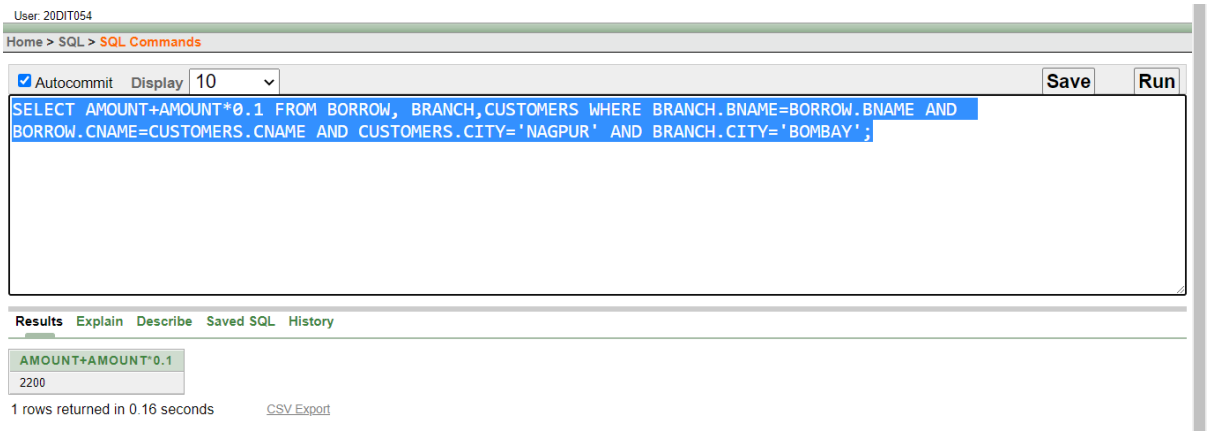
(1) Give 10% interest to all depositors.



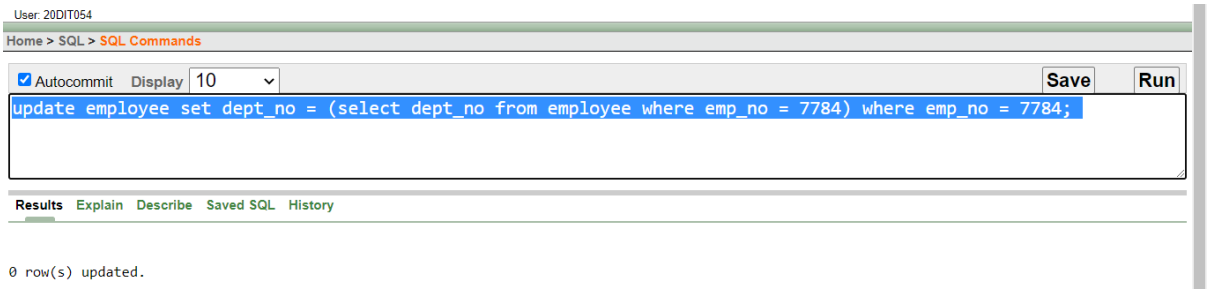
(2) Give 10% interest to all depositors having branch vrce



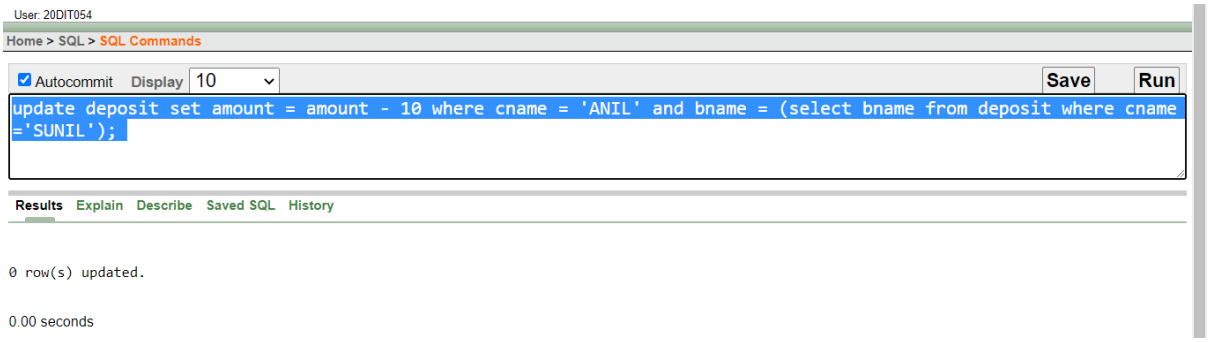
(3) Give 10% interest to all depositors living in nagpur and having branch city bombay.

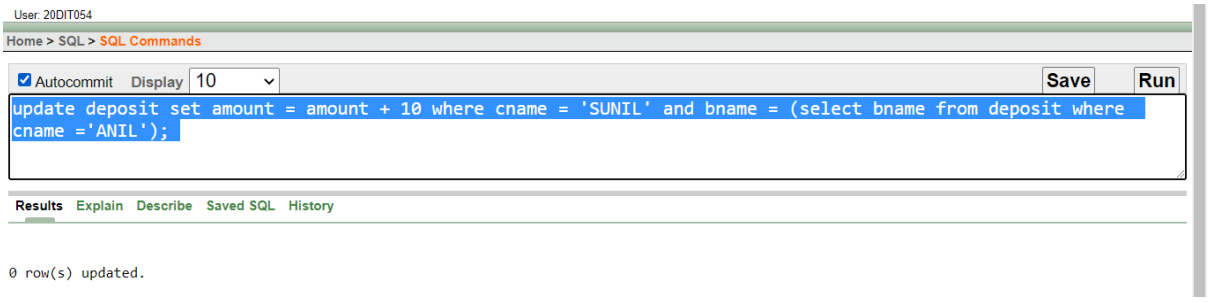


(4) Write a query which changes the department number of all employees with empno 7788's job to employee 7844'current department number.

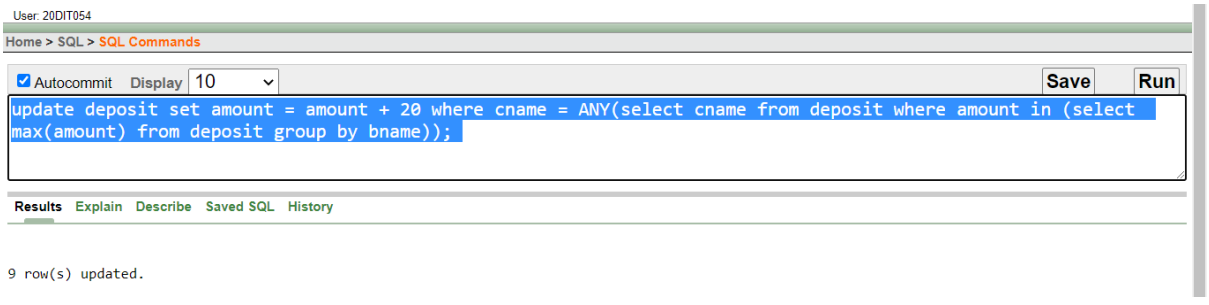


(5) Transfer 10 Rs from account of anil to sunil if both are having same branch.

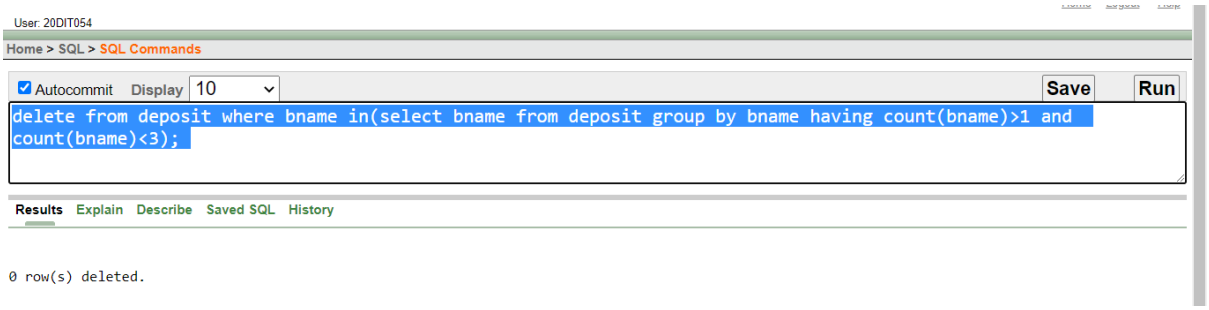




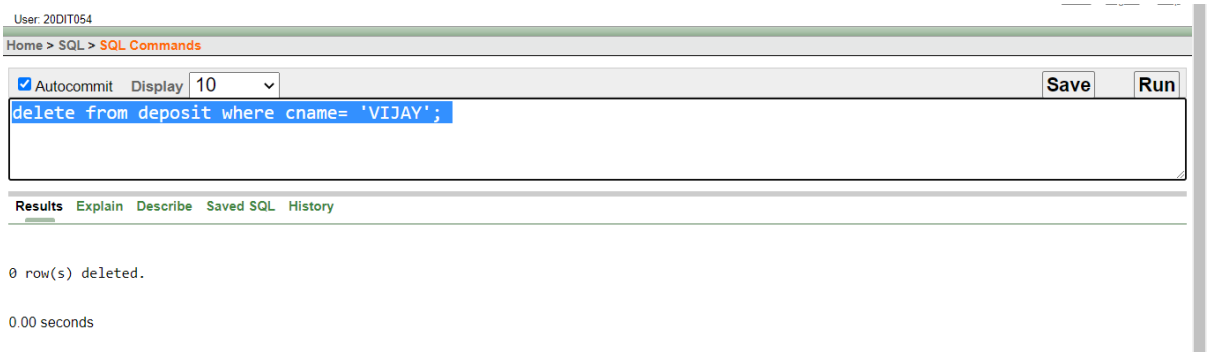
(6) Give 100 Rs more to all depositors if they are maximum depositors in their respective branch.



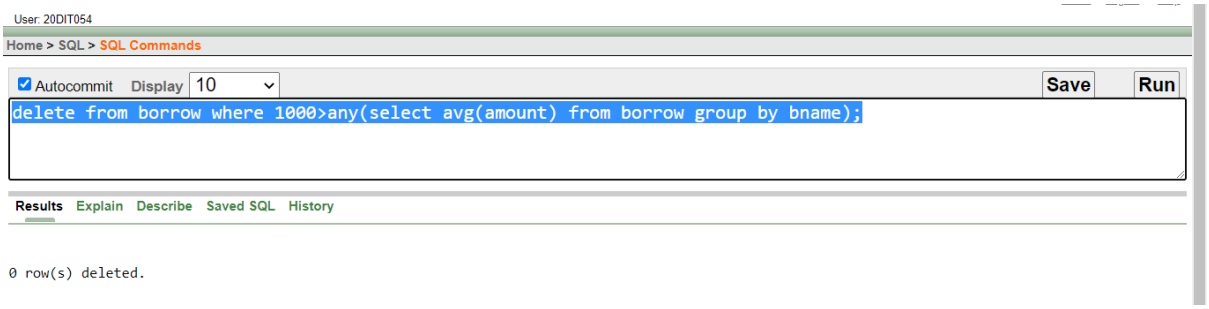
(7) Delete depositors of branches having number of customers between 1 to 3.



(8) Delete deposit of vijay.



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



Conclusion :

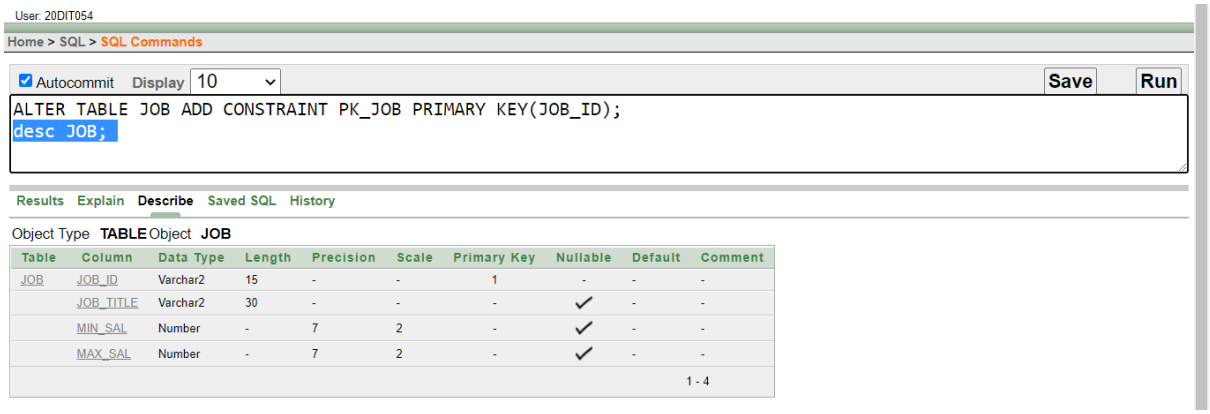
From this practical we learn about manipulating of data.

Practical-11

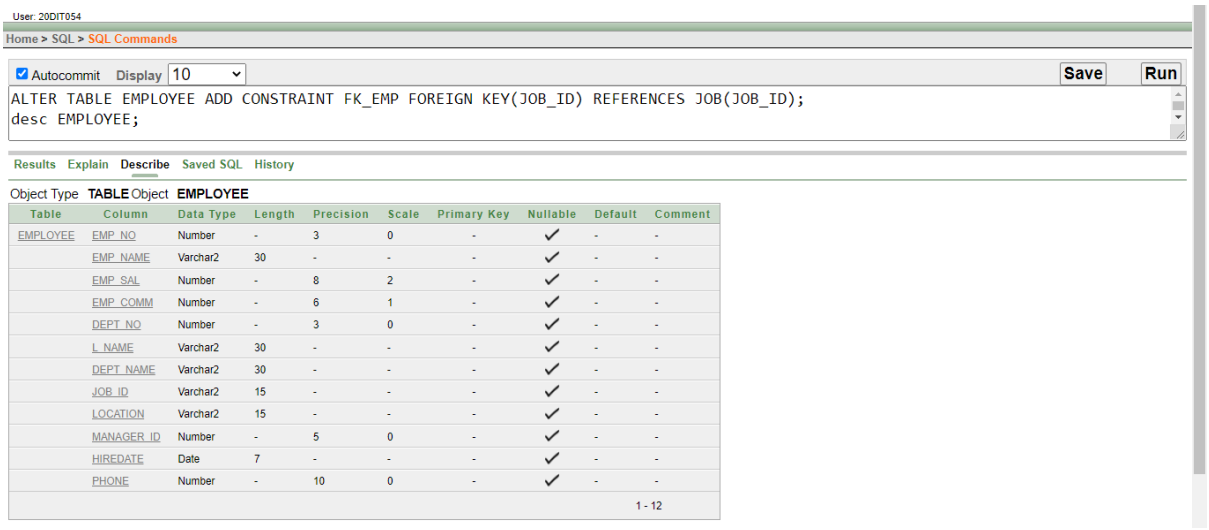
Aim: Add and Remove constraint

Queries :

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



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



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

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
CREATE TABLE LOCK1(L_ID NUMBER,L_NAME VARCHAR2(50),L_ADDRESS VARCHAR2(50),AGE NUMBER,CONSTRAINT PK_LOCK1 PRIMARY
KEY(L_ID,L_NAME));
desc LOCK1;
```

Results Explain Describe Saved SQL History

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

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

1 - 4

(4) Remove primary key constraint on job_id

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

```
ALTER TABLE JOB DROP CONSTRAINT PK_JOB;
DESC JOB;
```

Results Explain Describe Saved SQL History

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

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

1 - 4

(5) Remove foreign key constraint on employee table

User: 20DIT054

Home > SQL > SQL Commands

Autocommit Display 10 Save Run

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

Results Explain Describe Saved SQL History

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

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

1 - 12

Conclusion :

From this practical, I have learnt about Add and Remove constraint.

Practical-12

Aim: Data Dictionary and E-R Diagram

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

- The hotel offers three types of ROOMS, including single room, double room, and triple room. Every room is Identified by its unique number.

- Every employee at the hotel is either a receptionist, a cleaning staff, or a kitchen staff. Each 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.

- Receptionist 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.

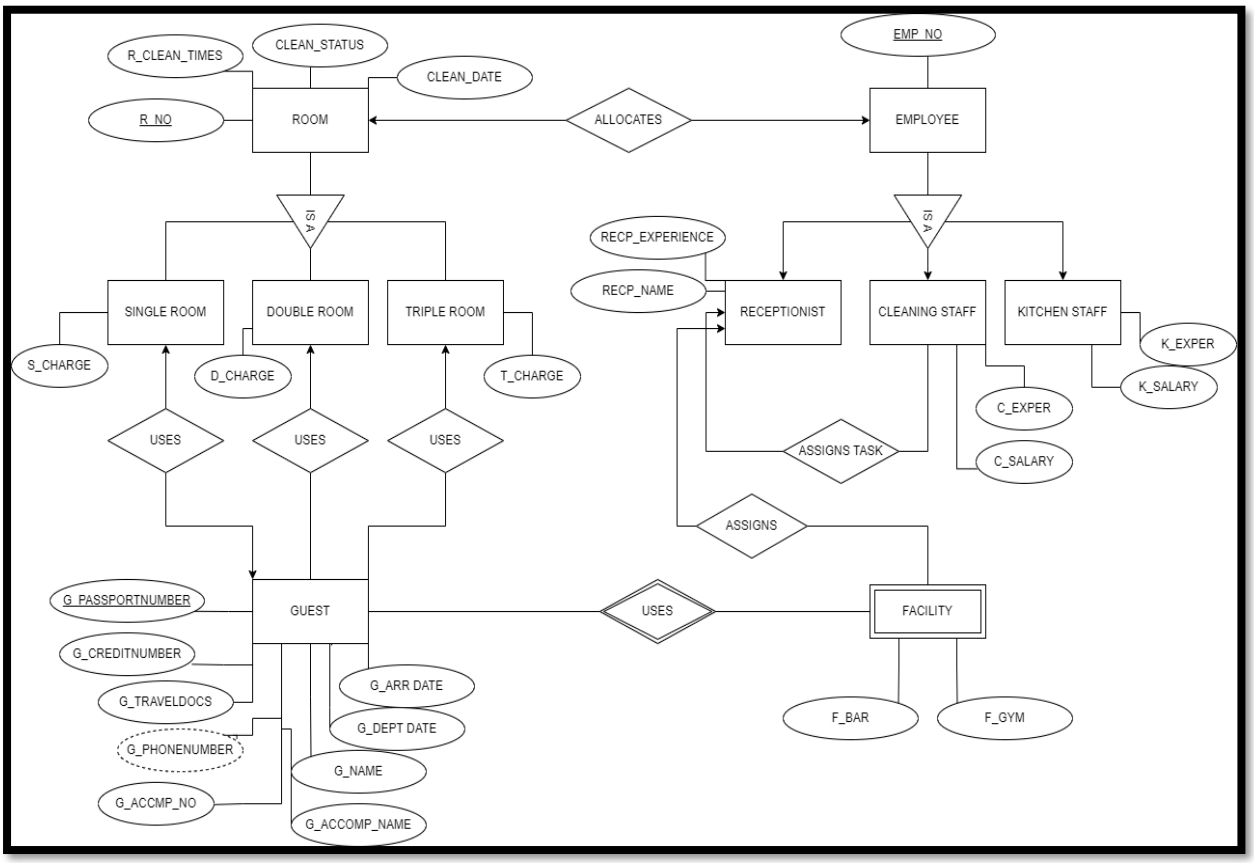
Hint: Pay attention to clear identification of different kinds of attributes (e.g., multi-valued, derived, and Primary key), the total participation for the relationship sets and generalization (or specialization) of entities.

Output :

ENTITY	ATTRIBUTE	DATATYPE	LENGTH	CONSTRAINT
ROOM	CLEAN_DATE	DATE	-	NOT NULL
	CLEAN_STATUS	VARCHAR2	15	NOT NULL
	R_NO	NUMBER	8	PRIMARY KEY
	R_CLEAN_TIME	NUMBER	20	-
SINGLE ROOM	S_CHARGE	NUMBER	8	NOT NULL
DOUBLE ROOM	D_CHARGE	NUMBER	8	NOT NULL
TRIPLE ROOM	T_CHARGE	NUMBER	8	NOT NULL
EMPLOYEE	EMP_NO	VARCHAR	10	PRIMARY KEY
RECEPTIONIST	RECP_EXPERIENCE	NUMBER	2	-
	RECP_NAME	VARCHAR2	20	NOT NULL
CLEANING STAFF	C_EXPER	NUMBER	2	-
	C_SALARY	NUMBER	-	NOT NULL
KITCHEN STAFF	K_SALARY	NUMBER	-	NOT NULL
	K_EXPER	NUMBER	2	-
GUEST	G_PASSPORTNUMBER	NUMBER	20	PRIMARY KEY
	G_CREDITNUMBER	NUMBER	20	NOT NULL
	G_TRAVELDOCS	NUMBER	20	-
	G_PHONENUMBER	NUMBER	10	NOT NULL
	G_ACCMP_NO	NUMBER	-	-
	G_ARRDATE	DATE	-	NOT NULL

	G_DEPTDATE	DATE	-	-
	G_NAME	VARCHAR2	20	NOT NULL
	G_ACCOMPNAME	VARCHAR2	20	NOT NULL
FACILITY	F_BAR	VARCHAR2	20	-
	F_GYM	VARCHAR2	20	-

E-R Diagram:



Conclusion :

From this practical we learned about how ER diagram is designed and how data dictionary is designed.

Practical-13

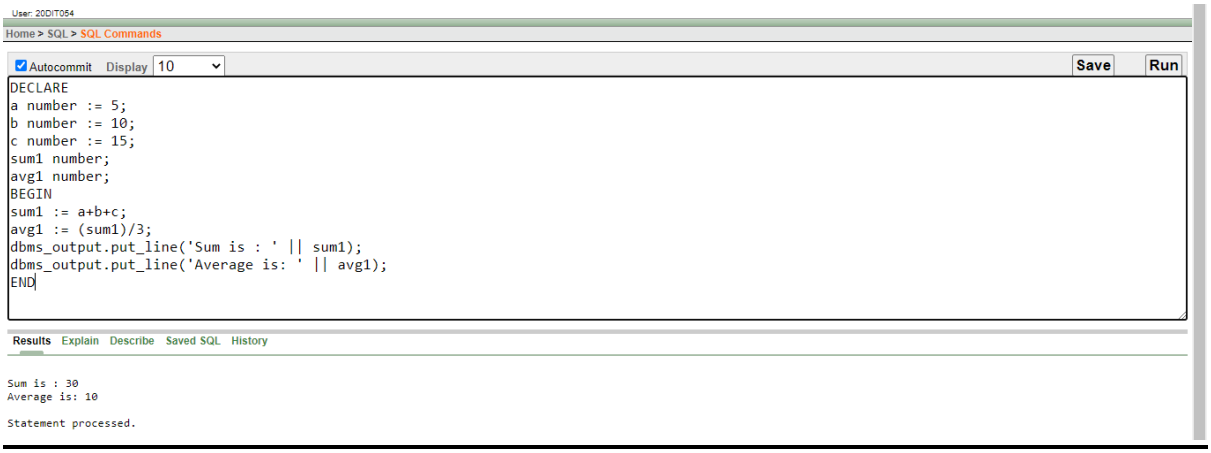
Aim: To perform basic PL/SQL blocks

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

Program :

```
DECLARE
a number := 5;
b number := 10;
c number := 15;
sum1 number;
avg1 number;
BEGIN
sum1 := a+b+c;
avg1 := (sum1)/3;
dbms_output.put_line('Sum is : ' || sum1);
dbms_output.put_line('Average is: ' || avg1);
END
```

Output :



Conclusion :

From this practical we learned about how to add and remove the constraints on any table while on creation as well as after table is created.

Practical-14

Aim: To perform the concept of loop

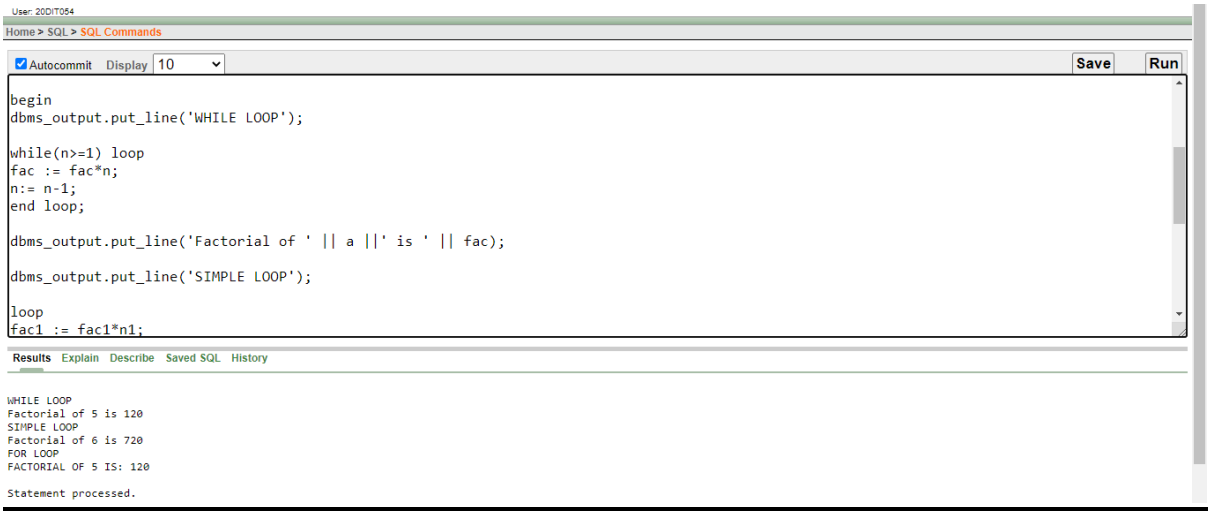
Find the factorial of a number in pl/sql using for, While and Simple Loop.

Program :

```
declare
n number:=5;
a number :=n;
fac number :=1;

n1 number:=6;
a1 number :=n1;
fac1 number :=1;

A2 NUMBER:=5;
FACT NUMBER:=1;

begin
dbms_output.put_line('WHILE LOOP');

while(n>=1) loop
fac := fac*n;
n:= n-1;
end loop;

dbms_output.put_line('Factorial of ' || a ||' is ' || fac);

dbms_output.put_line('SIMPLE LOOP');

loop
fac1 := fac1*n1;
n1:= n1-1;
if(n1=1) then
exit;
end if;
end loop;

dbms_output.put_line('Factorial of ' || a1 ||' is ' || fac1);
```

```
DBMS_OUTPUT.PUT_LINE('FOR LOOP');
FOR I IN 1..A2 LOOP
FACT:=FACT*I;
END LOOP;
DBMS_OUTPUT.PUT_LINE('FACTORIAL OF 5 IS: '|| FACT);

end;
```

Output :



Conclusion :

From this practical we learned how can we implement the factorial program using the PL/SQL.

Practical-15

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

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

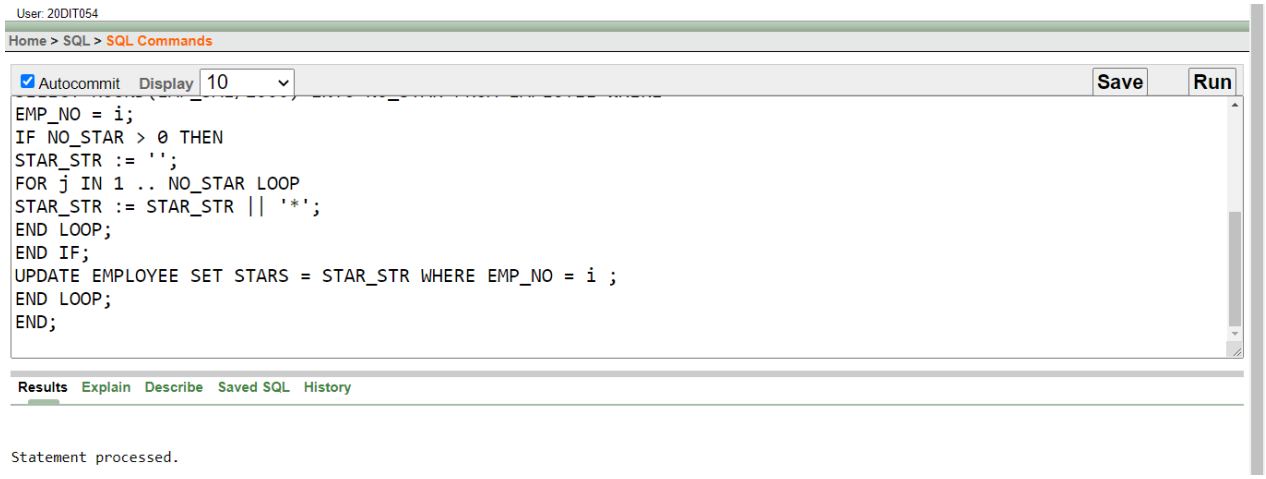
Create a PL/SQL block that rewards an employee by appending an asterisk in the STARS column for every Rs1000/- of the employee's salary. For example, if the employee has a salary amount of Rs8000/-, the string of asterisks should contain eight asterisks. If the employee has a salary amount of Rs12500/-, the string of asterisks should contain 13 asterisks.

Update the STARS column for the employee with the string of asterisks.

Program :

```
DECLARE
MIN_EMP_NO EMPLOYEE.EMP_NO%TYPE;
MAX_EMP_NO EMPLOYEE.EMP_NO%TYPE;
NO_STAR NUMBER;
STAR_STR VARCHAR2(20);
BEGIN
SELECT MAX(EMP_NO) INTO MAX_EMP_NO FROM EMPLOYEE;
SELECT MIN(EMP_NO) INTO MIN_EMP_NO FROM EMPLOYEE;
FOR i IN MIN_EMP_NO .. MAX_EMP_NO LOOP
SELECT ROUND(EMP_SAL/1000) INTO NO_STAR FROM EMPLOYEE WHERE
EMP_NO = i;
IF NO_STAR > 0 THEN
STAR_STR := '';
FOR j IN 1 .. NO_STAR LOOP
STAR_STR := STAR_STR || '*';
END LOOP;
END IF;
UPDATE EMPLOYEE SET STARS = STAR_STR WHERE EMP_NO = i ;
END LOOP;
END;
```

Output :



EMP_NO	EMP_NAME	EMP_SAL	EMP_COMM	DEPT_NO	L_NAME	DEPT_NAME	JOB_ID	LOCATION	MANAGER_ID	HIREDATE	PHONE	STARS
101	Smith	800	-	10	shah	machine learning	fi_mgr	toronto	105	09-AUG-96	-	*
102	Snehal	1600	300	25	gupta	data science	lec	las vegas	-	14-MAR-96	-	**
103	Dhruvin	1100	500	20	wales	machine learning	mk_mgr	ontario	105	30-NOV-95	-	*
104	Aman	3000	-	10	sharma	virtual reality	comp_op	mexico	12	02-OCT-97	-	***
105	Anita	5000	50000	10	patel	big data analytics	comp_op	germany	107	01-JAN-98	-	*****
106	Sneha	2450	24500	10	joseph	big data anylitics	fi_acc	melbourne	105	26-SEP-97	-	**
107	Anamika	2975	-	30	jha	artifical inteligence	it_prog	new york	-	15-JUL-97	-	***

Conclusion :

After performing this practical, we understood the concept of select into and %type in PL/SQL.

Practical-16

Aim: To perform the concept of cursor

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

(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 <<last name>> Due for a raise. Otherwise, display the message <<last_name>> Not due for a raise.

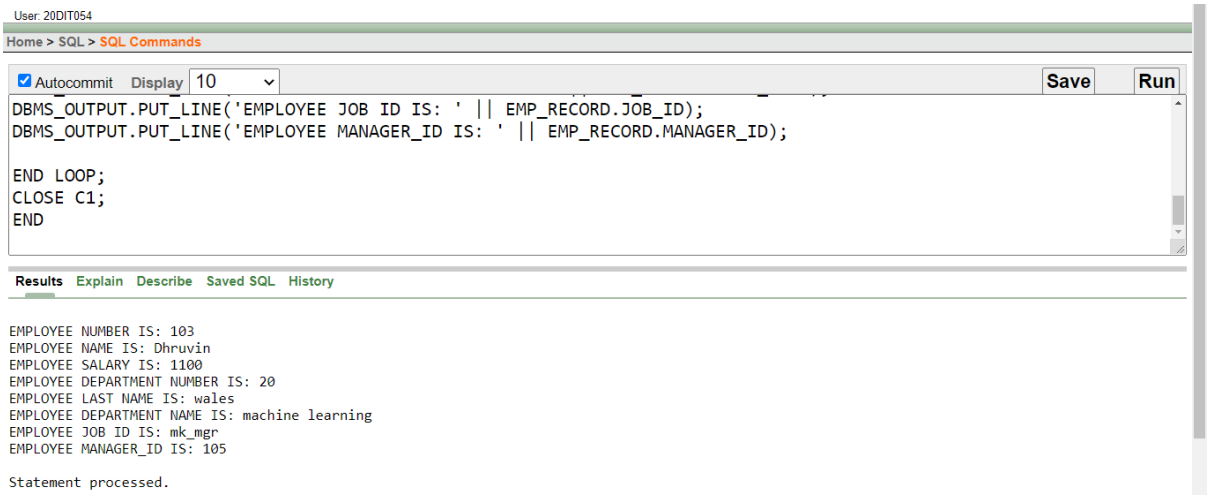
Program :

(a)

```
DECLARE

CURSOR C1 IS
SELECT
EMP_NO,EMP_NAME,EMP_SAL,DEPT_NO,L_NAME,DEPT_NAME,JOB_ID,MANAG
ER_ID FROM EMPLOYEE WHERE EMP_NO=103;

EMP_RECORD C1%ROWTYPE;

BEGIN

OPEN C1;
LOOP

FETCH C1 INTO EMP_RECORD;
EXIT WHEN C1%NOTFOUND;

DBMS_OUTPUT.PUT_LINE('EMPLOYEE NUMBER IS: ' || EMP_RECORD.EMP_NO);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE NAME IS: ' || EMP_RECORD.EMP_NAME);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE SALARY IS: ' || EMP_RECORD.EMP_SAL);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE DEPARTMENT NUMBER IS: ' ||
EMP_RECORD.DEPT_NO);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE LAST NAME IS: ' ||
EMP_RECORD.L_NAME);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE DEPARTMENT NAME IS: ' ||
EMP_RECORD.DEPT_NAME);
```

```
DBMS_OUTPUT.PUT_LINE('EMPLOYEE JOB ID IS: ' || EMP_RECORD.JOB_ID);
DBMS_OUTPUT.PUT_LINE('EMPLOYEE MANAGER_ID IS: ' ||
EMP_RECORD.MANAGER_ID);

END LOOP;
CLOSE C1;
END
```

Output :



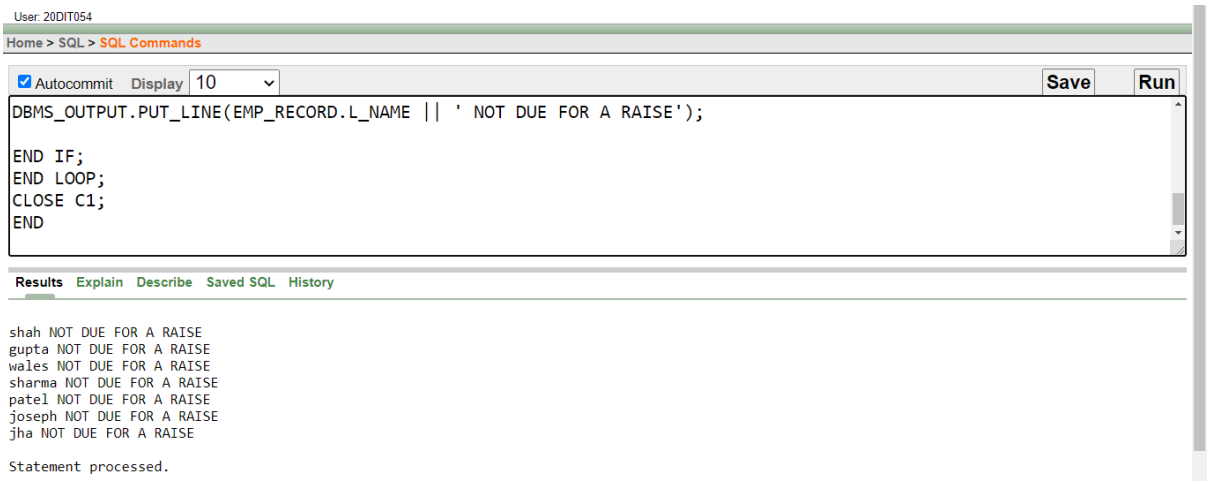
Program :

(b)

```
DECLARE

CURSOR C1 IS
SELECT * FROM EMPLOYEE;

EMP_RECORD C1%ROWTYPE;

BEGIN

OPEN C1;
LOOP

FETCH C1 INTO EMP_RECORD;
EXIT WHEN C1%NOTFOUND;

IF(EMP_RECORD.EMP_SAL<1000 AND (EMP_RECORD.MANAGER_ID=7902 OR
EMP_RECORD.MANAGER_ID=7839)) THEN
```

```
DBMS_OUTPUT.PUT_LINE(EMP_RECORD.L_NAME || ' DUE FOR A RAISE');

ELSE
DBMS_OUTPUT.PUT_LINE(EMP_RECORD.L_NAME || ' NOT DUE FOR A RAISE');

END IF;
END LOOP;
CLOSE C1;
END
```

Output :



Conclusion :

From this practical we learned that the implementation of cursor and %TYPE and %ROWTYPE

Practical-17

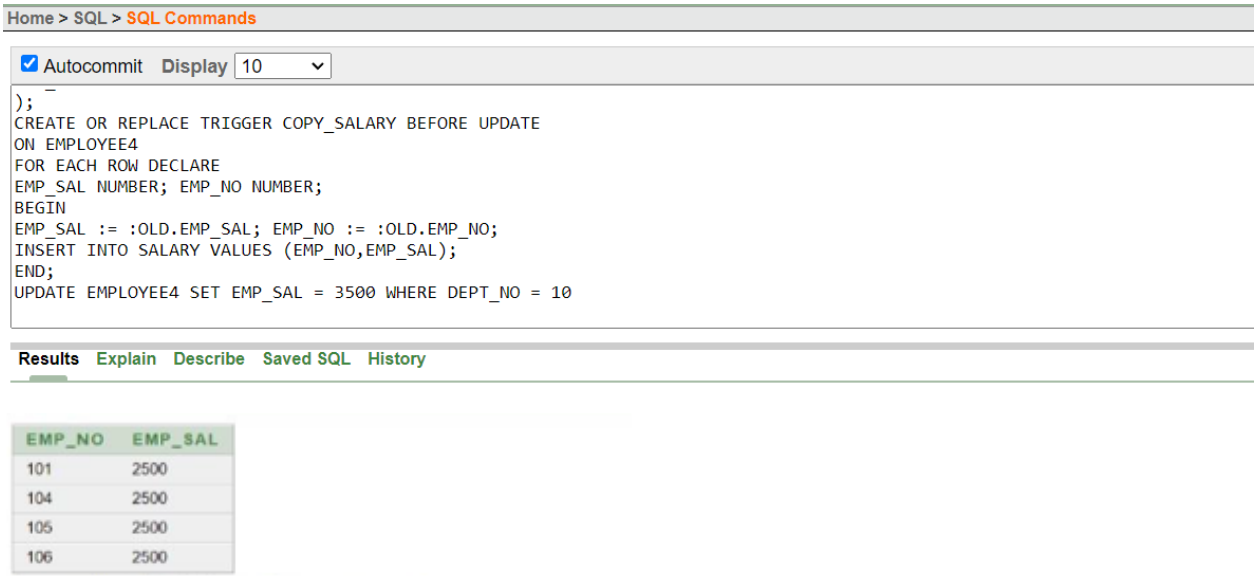
Aim: To perform the concept of trigger

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

Program :

```
EMP_NO NUMBER,
EMP_SAL NUMBER
);
CREATE OR REPLACE TRIGGER COPY_SALARY BEFORE UPDATE
ON EMPLOYEE4
FOR EACH ROW DECLARE
EMP_SAL NUMBER; EMP_NO NUMBER;
BEGIN
EMP_SAL := :OLD.EMP_SAL; EMP_NO := :OLD.EMP_NO;
INSERT INTO SALARY VALUES (EMP_NO,EMP_SAL);
END;
UPDATE EMPLOYEE4 SET EMP_SAL = 3500 WHERE DEPT_NO = 10
```

Output :



EMP_NO	EMP_SAL
101	2500
104	2500
105	2500
106	2500

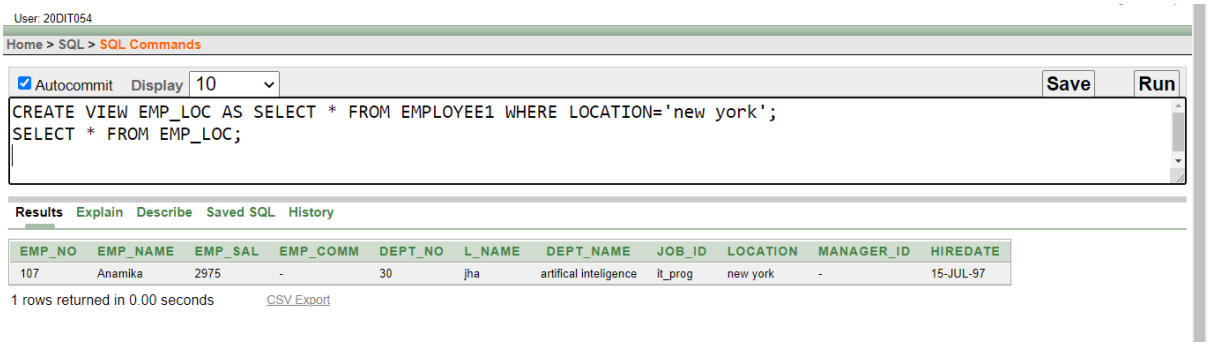
Conclusion :

From this practical we learned about how we can set the triggers and display the updation messages in output.

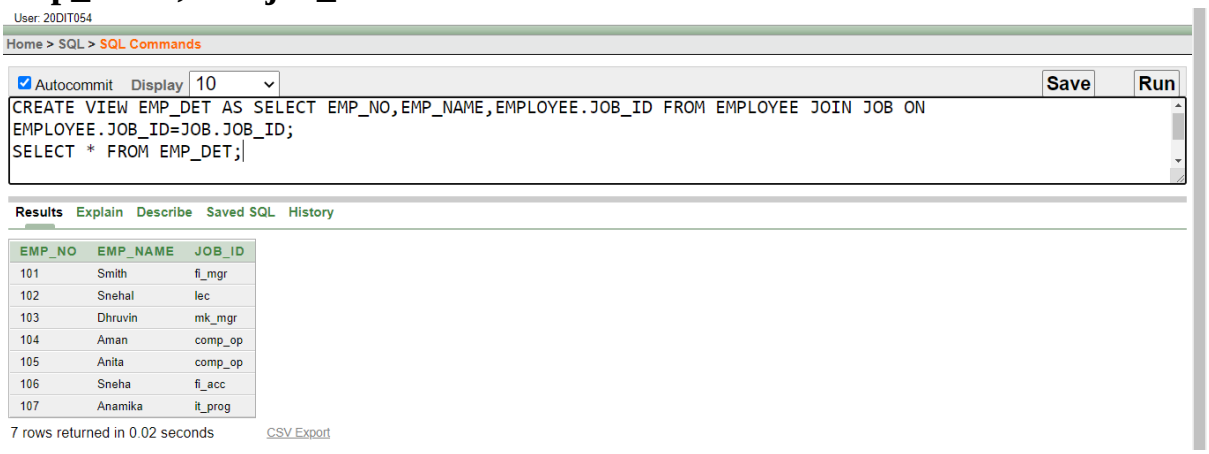
Practical-18

Aim: To solve queries using the concept of View.

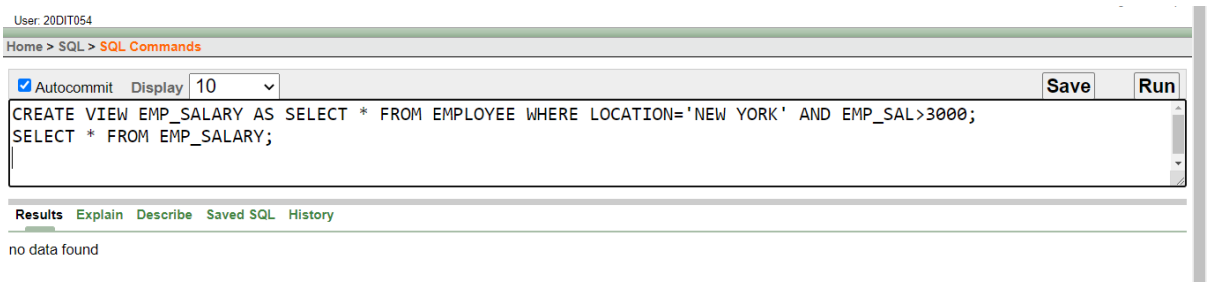
(1) Write a query to create a view for those employees belongs to the location New York.



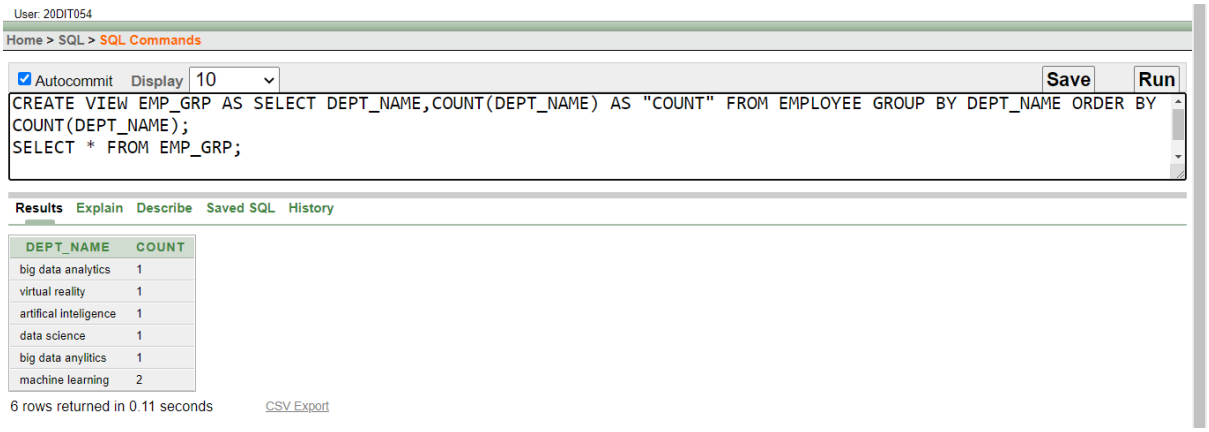
(2) Write a query to create a view for all employee with columns emp_id, emp_name, and job_id.



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



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



Conclusion :

From this practical we learned about creating a view and altering the data in main database would also update the data in the view.