ADVANCED DATABASE MANAGEMENT SYSTEMS

LAB ASSIGNMENT FILE

SUBMITTED TO:
MS VIMMI KOCHHER
AMITY SCHOOL OF ENGINEERING & TECHNOLOGY

SUBMITTED BY:
HARSH - A501144824003
M.TECH(AI)
M.TECH 2024-2026

INDEX

Date – 11/09/2024

Practical No : 1

DDL Commands

Data Definition Language(DDL) : It is a subset of SQL and a part of DBMS(Database Management System). DDL consist of Commands to commands like CREATE, ALTER, TRUNCATE and DROP. These commands are used to create or modify the tables in SQL.

DDL Commands :

1. Create
   This command is used to create a new table in SQL. The user has to give information like table name, column names, and their datatypes.

   Syntax –

   ```
   CREATE TABLE table_name
   (
   column_1 datatype,
   column_2 datatype,
   column_3 datatype,
   ....
   );
   ```

2. Alter
   This command is used to add, delete or change columns in the existing table. The user needs to know the existing table name and can do add, delete or modify tasks easily.

   Syntax –
   Syntax to add a column to an existing table.

   ```
   ALTER TABLE table_name
   ADD column_name datatype;
   ```

3. Truncate
   This command is used to remove all rows from the table, but the structure of the table still exists.

   Syntax –
   Syntax to remove an existing table.

   ```
   TRUNCATE TABLE table_name;
   ```

4. Drop
   This command is used to remove an existing table along with its structure from the Database.

   Syntax –
   Syntax to drop an existing table.

```
DROP TABLE table_name;
```

5. Rename
   It is possible to change name of table with or without data in it using simple RENAME command.

   We can rename any table object at any point of time.

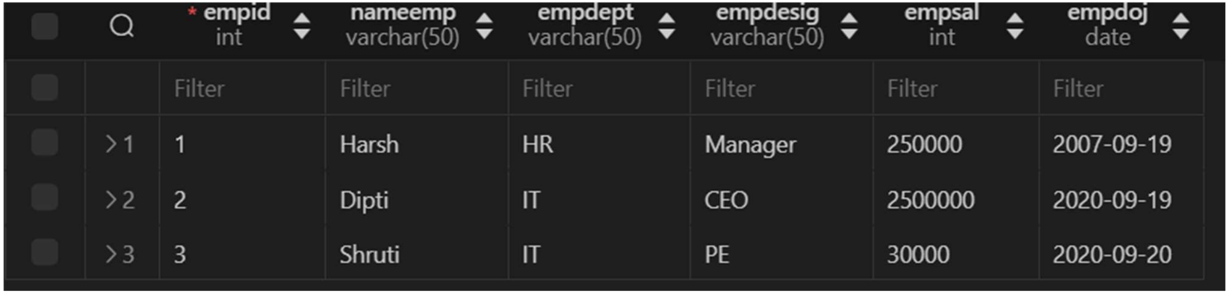
   Syntax –

   ```
   RENAME TABLE <Table Name> To <New_Table_Name>;
   ```

CODE for DDL Commands

```
1)  -- create employee table
create table employee (
   empid INT,
   empname VARCHAR(50),
   empdept VARCHAR(50),
   empdesig VARCHAR(50),
   empsal INT,
   empdoj DATE
);

2)  -- alter table to empid as PRIMARY
alter table amity.employee add primary key (empid);

3)  --Truncate
TRUNCATE employee;

4)  --DROP Table
DROP TABLE employee;

5)  --Rename attribute
alter table employee RENAME COLUMN empname TO nameemp;
```

Output ->



	* empid int	nameemp varchar(50)	empdept varchar(50)	empdesig varchar(50)	empsal int	empdoj date
	Filter	Filter	Filter	Filter	Filter	Filter
>1	1	Harsh	HR	Manager	250000	2007-09-19
>2	2	Dipti	IT	CEO	2500000	2020-09-19
>3	3	Shruti	IT	PE	30000	2020-09-20

Date – 11/09/2024

Practical No : 2

DML Commands

Data Manipulation Language (DML) : The SQL commands that deal with the manipulation of data present in the database belong to DML or Data Manipulation Language and this includes most of the SQL statements. It is the component of the SQL statement that controls access to data and to the database. Basically, DCL statements are grouped with DML statements.

DDL Commands :

1. Insert
   Used to insert new data (rows) into a table.
   - **Syntax to insert data into all columns**:

   ```
   INSERT INTO table_name (column1, column2, ...)
   VALUES (value1, value2, ...);
   ```

2. Update
   Used to modify existing records in a table.
   - **Syntax to update specific columns**:

   ```
   UPDATE table_name
   SET column1 = value1, column2 = value2
   WHERE condition;
   ```

3. Delete
   Used to remove existing records from a table.
   - **Syntax to delete specific rows**:

   ```
   DELETE FROM table_name
   WHERE condition;
   ```

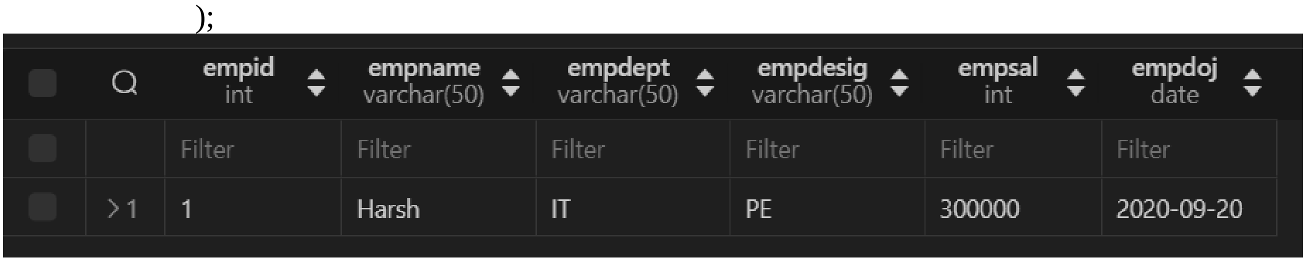
4. Select
   Used to retrieve data from one or more tables.
   - **Syntax to select specific columns**:

   ```
   SELECT column1, column2
   FROM table_name
   WHERE condition;
   ```

CODE for DDL Commands

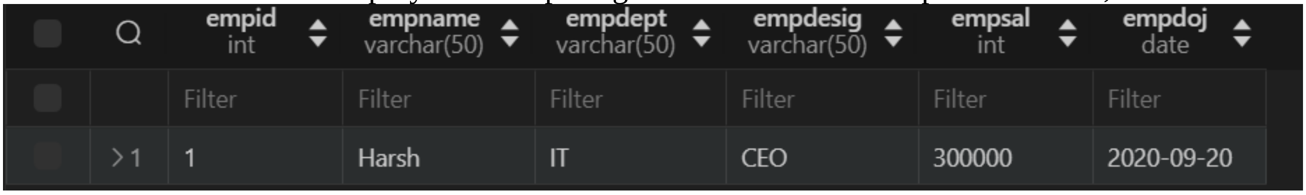
1)
```
-- Input data in table
INSERT INTO
   employee (
      empid,
      empname,
      empdept,
      empdesig,
      empsal,
      empdoj
   )
VALUES (
      3,
```

```
            'Shruti',
            'IT',
            'HR',
            300000,
            '2020-09-21'
        );
```



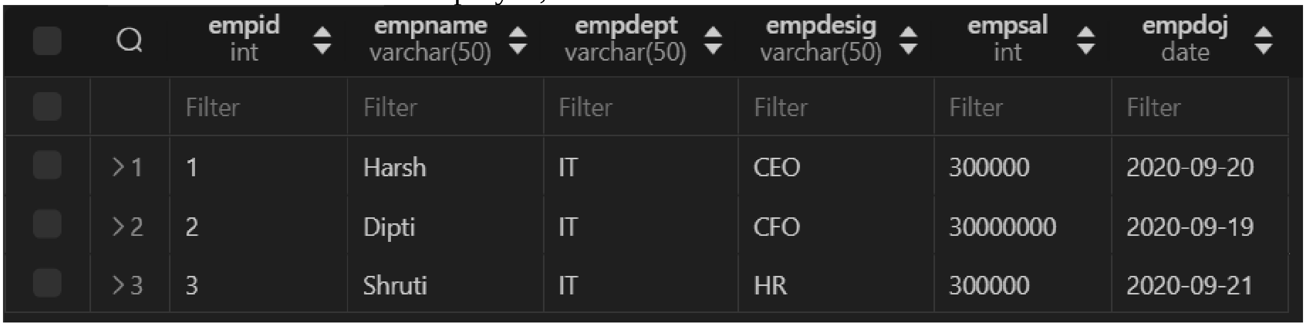
		empid int	empname varchar(50)	empdept varchar(50)	empdesig varchar(50)	empsal int	empdoj date
		Filter	Filter	Filter	Filter	Filter	Filter
	› 1	1	Harsh	IT	PE	300000	2020-09-20

2) --Update Table
UPDATE employee set empdesig = "CEO" WHERE empsal = 300000;



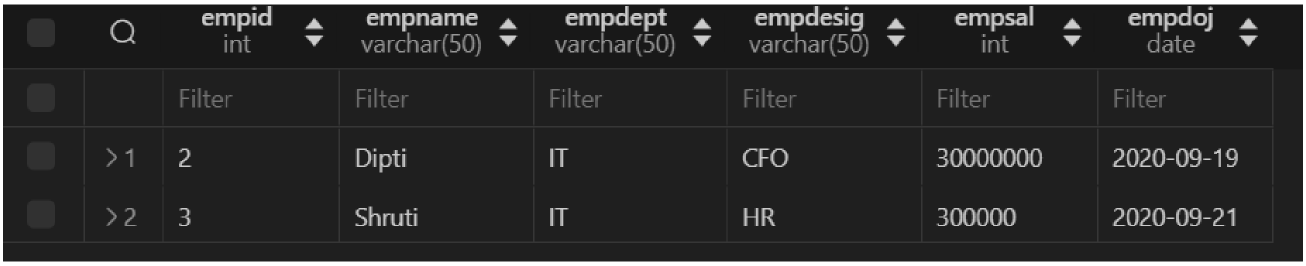
		empid int	empname varchar(50)	empdept varchar(50)	empdesig varchar(50)	empsal int	empdoj date
		Filter	Filter	Filter	Filter	Filter	Filter
	› 1	1	Harsh	IT	CEO	300000	2020-09-20

3) --Select from table;
SELECT * from employee;



		empid int	empname varchar(50)	empdept varchar(50)	empdesig varchar(50)	empsal int	empdoj date
		Filter	Filter	Filter	Filter	Filter	Filter
	› 1	1	Harsh	IT	CEO	300000	2020-09-20
	› 2	2	Dipti	IT	CFO	30000000	2020-09-19
	› 3	3	Shruti	IT	HR	300000	2020-09-21

4) --Delete from table
DELETE FROM employee WHERE empid = 1;



		empid int	empname varchar(50)	empdept varchar(50)	empdesig varchar(50)	empsal int	empdoj date
		Filter	Filter	Filter	Filter	Filter	Filter
	› 1	2	Dipti	IT	CFO	30000000	2020-09-19
	› 2	3	Shruti	IT	HR	300000	2020-09-21

Date – 11/09/2024

Practical No : 3

AIM - Using Relational, Logical Operators

Create Table Employeepr3

```
create table employeepr3 (
   empNo INT,
   empname VARCHAR(50),
   empJob VARCHAR(50),
   empManagerName VARCHAR(50),
   emphiredate DATE,
   empsal INT,
   empcommission INT,
   empdept VARCHAR(50)
)
```

Create Table Departmentpr3

```
create table Departmentpr3 (
   deptNo INT,
   deptName VARCHAR(50),
   deptLoc VARCHAR(50)
)
```

Insert into Employeepr3

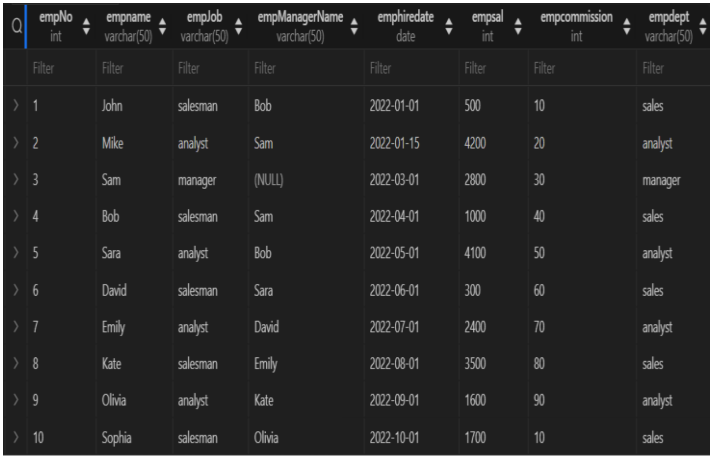
```
insert into employeepr3 (empNo, empname, empJob, empManagerName, emphiredate, empsal,
empcommission, empdept) values
   (1, 'John', 'salesman', 'Bob', '2022-01-01', 4500, 10, 'sales'),
   (2, 'Mike', 'analyst', 'Sam', '2022-01-15', 4200, 20, 'analyst'),
   (3, 'Sam', 'manager', null, '2022-03-01', 4800, 30, 'manager'),
   (4, 'Bob', 'salesman', 'Sam', '2022-04-01', 4000, 40, 'sales'),
   (5, 'Sara', 'analyst', 'Bob', '2022-05-01', 4100, 50, 'analyst'),
   (6, 'David', 'salesman', 'Sara', '2022-06-01', 4300, 60, 'sales'),
   (7, 'Emily', 'analyst', 'David', '2022-07-01', 4400, 70, 'analyst'),
   (8, 'Kate', 'salesman', 'Emily', '2022-08-01', 4500, 80, 'sales'),
   (9, 'Olivia', 'analyst', 'Kate', '2022-09-01', 4600, 90, 'analyst'),
   (10, 'Sophia', 'salesman', 'Olivia', '2022-10-01', 4700, 10, 'sales')
```

Insert into Departmentpr3

```
insert into Departmentpr3 (deptNo, deptName, deptLoc)
values (10, 'sales', 'New York'),
   (20, 'analyst', 'Chicago'),
   (30, 'manager', 'Los Angeles'),
   (40, 'sales', 'New York'),
   (50, 'analyst', 'Chicago'),
   (60, 'manager', 'Los Angeles'),
   (70, 'sales', 'New York'),
   (80, 'analyst', 'Chicago'),
   (90, 'manager', 'Los Angeles'),
   (100, 'sales', 'New York')
```

a) **List all the information about all employees from emp table.**

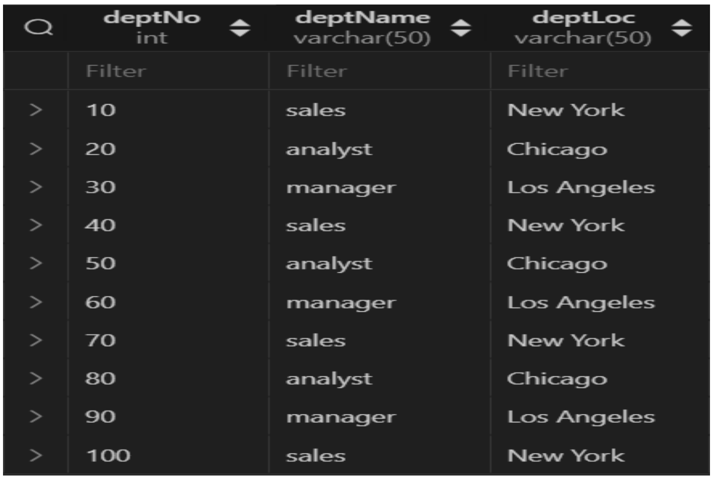
select * from employeepr3



	empNo int	empname varchar(50)	empJob varchar(50)	empManagerName varchar(50)	emphiredate date	empsal int	empcommission int	empdept varchar(50)
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
›	1	John	salesman	Bob	2022-01-01	500	10	sales
›	2	Mike	analyst	Sam	2022-01-15	4200	20	analyst
›	3	Sam	manager	(NULL)	2022-03-01	2800	30	manager
›	4	Bob	salesman	Sam	2022-04-01	1000	40	sales
›	5	Sara	analyst	Bob	2022-05-01	4100	50	analyst
›	6	David	salesman	Sara	2022-06-01	300	60	sales
›	7	Emily	analyst	David	2022-07-01	2400	70	analyst
›	8	Kate	salesman	Emily	2022-08-01	3500	80	sales
›	9	Olivia	analyst	Kate	2022-09-01	1600	90	analyst
›	10	Sophia	salesman	Olivia	2022-10-01	1700	10	sales

b) **List all the information about all departments from department table.**

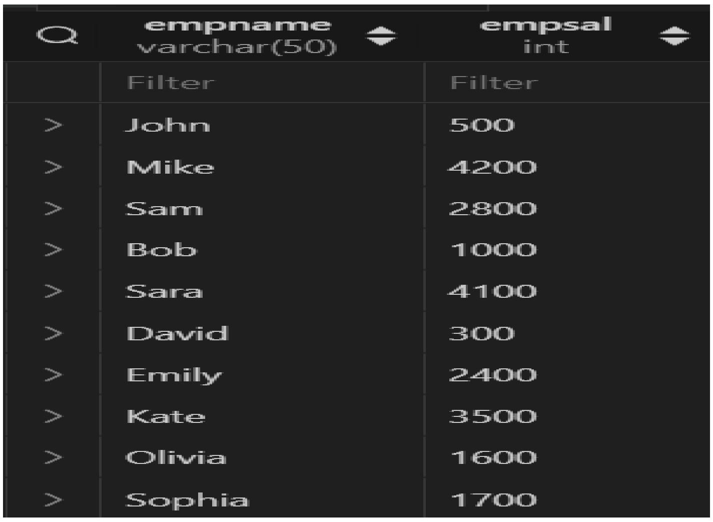
select * from Departmentpr3



	deptNo int	deptName varchar(50)	deptLoc varchar(50)
	Filter	Filter	Filter
>	10	sales	New York
>	20	analyst	Chicago
>	30	manager	Los Angeles
>	40	sales	New York
>	50	analyst	Chicago
>	60	manager	Los Angeles
>	70	sales	New York
>	80	analyst	Chicago
>	90	manager	Los Angeles
>	100	sales	New York

c) **List all employees names along with their salary from employee table.**

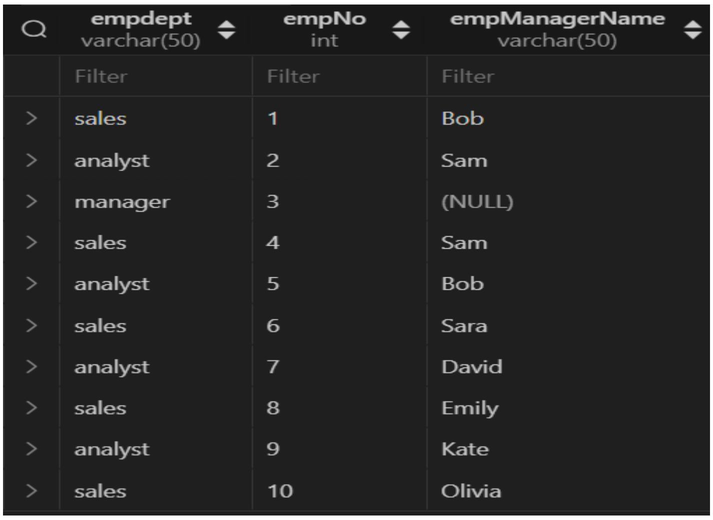
select empname, empsal from employeepr3



	empname varchar(50)	empsal int
	Filter	Filter
>	John	500
>	Mike	4200
>	Sam	2800
>	Bob	1000
>	Sara	4100
>	David	300
>	Emily	2400
>	Kate	3500
>	Olivia	1600
>	Sophia	1700

d) **List all department name, employee number, manager number from employee table**

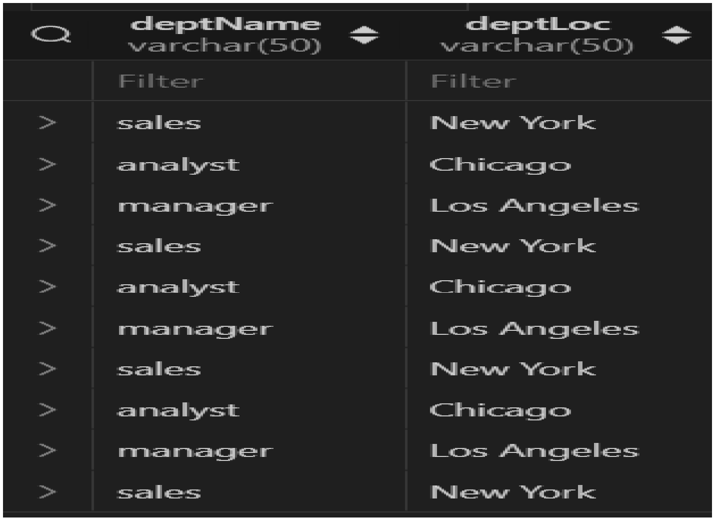
select empdept, empNo, empManagerName from employeepr3



	empdept varchar(50)	empNo int	empManagerName varchar(50)
	Filter	Filter	Filter
>	sales	1	Bob
>	analyst	2	Sam
>	manager	3	(NULL)
>	sales	4	Sam
>	analyst	5	Bob
>	sales	6	Sara
>	analyst	7	David
>	sales	8	Emily
>	analyst	9	Kate
>	sales	10	Olivia

e) List dept names and location from department table

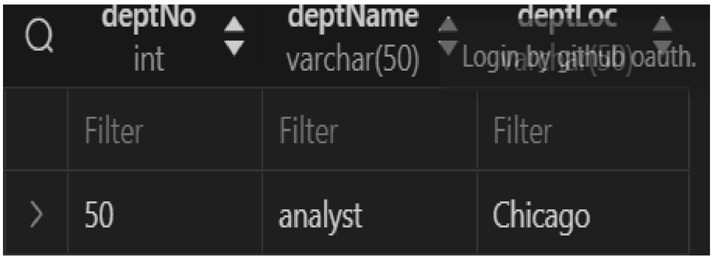
select deptName, deptLoc from Departmentpr3



deptName varchar(50)	deptLoc varchar(50)
Filter	Filter
sales	New York
analyst	Chicago
manager	Los Angeles
sales	New York
analyst	Chicago
manager	Los Angeles
sales	New York
analyst	Chicago
manager	Los Angeles
sales	New York

f) List the employees belong to department number=50

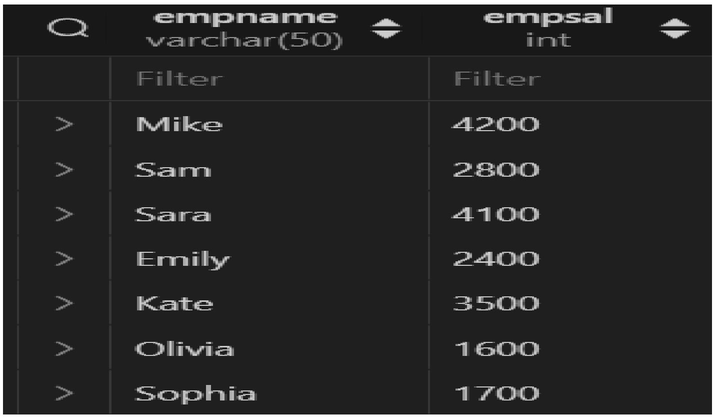
select * from departmentpr3 where `deptNo` = '50'



deptNo int	deptName varchar(50)	deptLoc varchar(50) Login by github oauth.
Filter	Filter	Filter
50	analyst	Chicago

g) List employee names and salary whose salary is greater than 1000

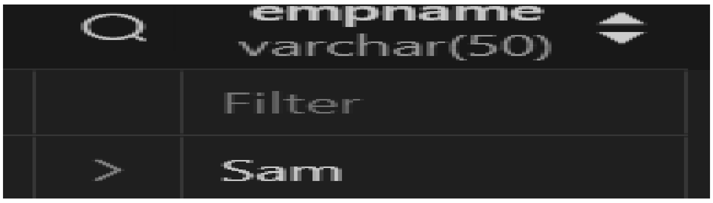
select empname, empsal from employeepr3 where empsal > 1000



empname varchar(50)	empsal int
Filter	Filter
Mike	4200
Sam	2800
Sara	4100
Emily	2400
Kate	3500
Olivia	1600
Sophia	1700

h) List names of clerk working in department 30

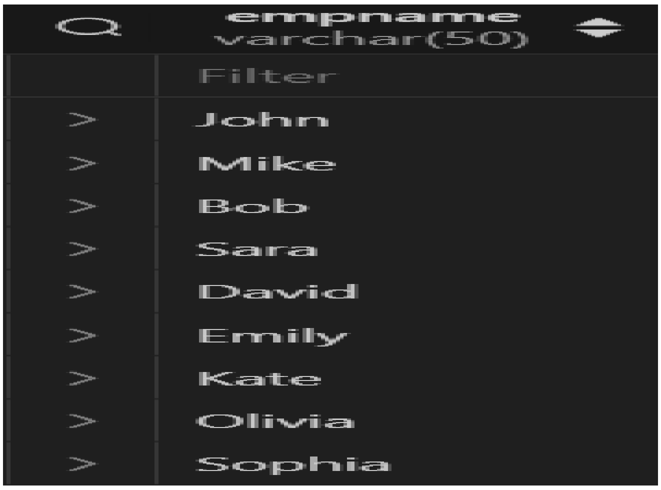
select empname from employeepr3 where deptNo = '30'



empname varchar(50)
Filter
Sam

i) List names of employees who are either analyst or salesman

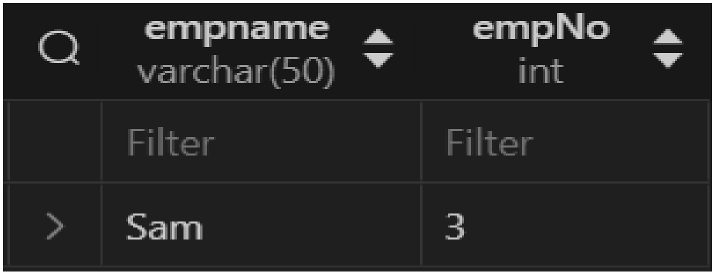
```
select empname
from employeepr3
where
    empJob = 'analyst'
        or empJob = 'salesman'
```



empname varchar(50)
Filter
John
Mike
Bob
Sara
David
Emily
Kate
Olivia
Sophia

j) List employee names and emp no who are manager

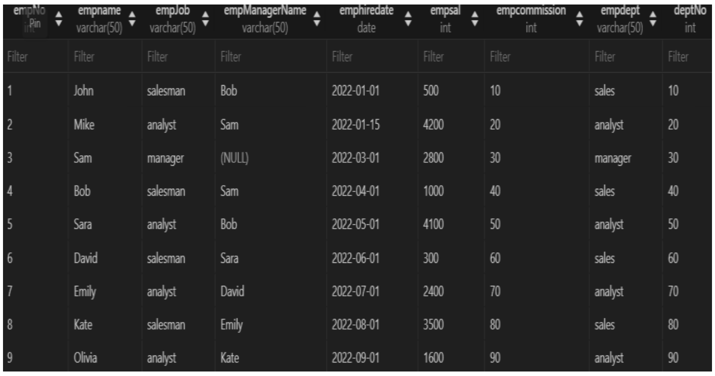
select empname, empNo from employeepr3 where empJob = 'manager'



	empname varchar(50)	empNo int
	Filter	Filter
>	Sam	3

k) List the details of employees who have joined before end of September'22

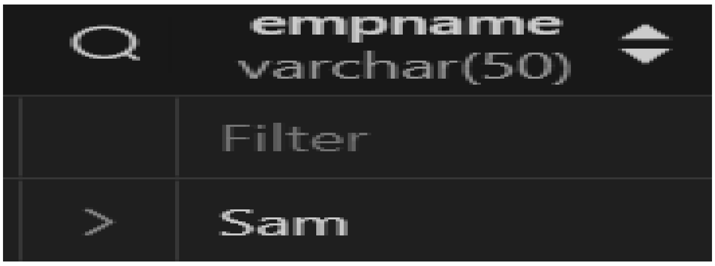
select * from employeepr3 where emphiredate < '2022-09-30'



empNo int	empname varchar(50)	empJob varchar(50)	empManagerName varchar(50)	emphiredate date	empsal int	empcommission int	empdept varchar(50)	deptNo int
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	John	salesman	Bob	2022-01-01	500	10	sales	10
2	Mike	analyst	Sam	2022-01-15	4200	20	analyst	20
3	Sam	manager	(NULL)	2022-03-01	2800	30	manager	30
4	Bob	salesman	Sam	2022-04-01	1000	40	sales	40
5	Sara	analyst	Bob	2022-05-01	4100	50	analyst	50
6	David	salesman	Sara	2022-06-01	300	60	sales	60
7	Emily	analyst	David	2022-07-01	2400	70	analyst	70
8	Kate	salesman	Emily	2022-08-01	3500	80	sales	80
9	Olivia	analyst	Kate	2022-09-01	1600	90	analyst	90

l) List names of employees who are not managers.

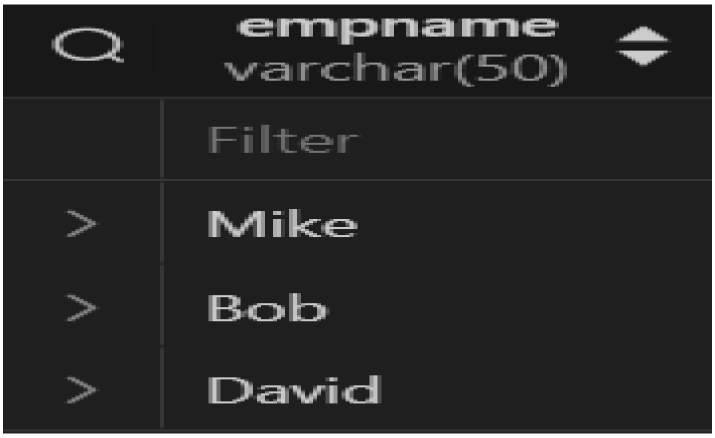
select empname from employeepr3 where empManagerName is null



	empname varchar(50)
	Filter
>	Sam

m) List the names of employees whose employee number is 20,40,60

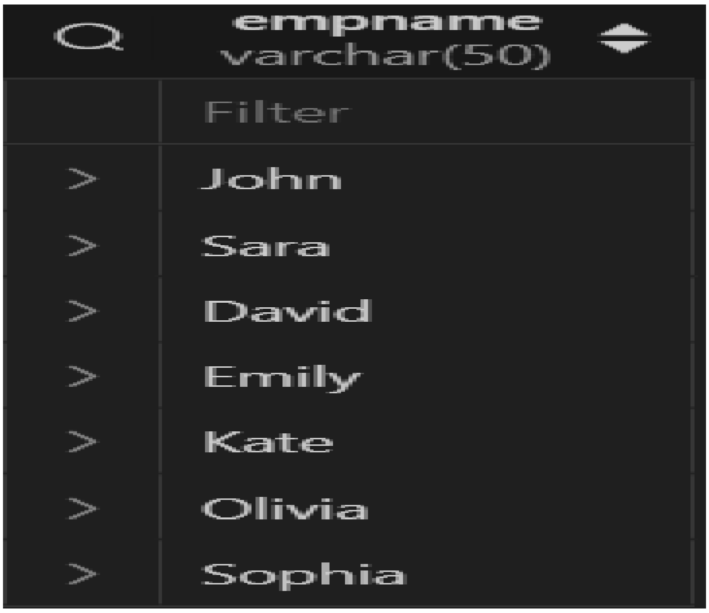
select empname from employeepr3 where deptNo in (20, 40, 60)



	empname varchar(50)
	Filter
>	Mike
>	Bob
>	David

n) List the names of employees not belonging to dept 20,30,40

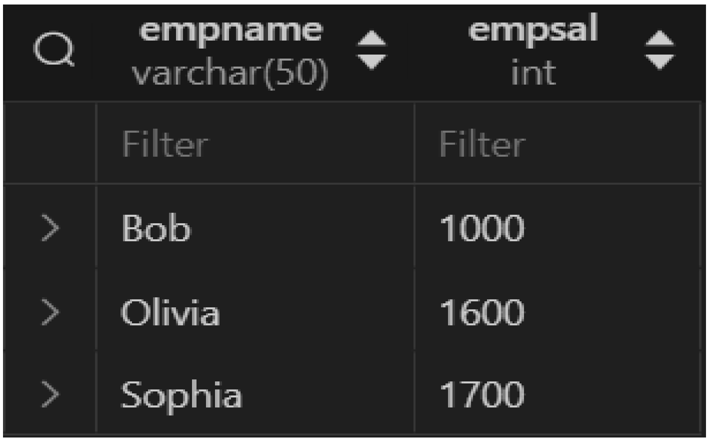
select empname from employeepr3 where deptNo not in(20, 30, 40)



	empname varchar(50)
	Filter
>	John
>	Sara
>	David
>	Emily
>	Kate
>	Olivia
>	Sophia

o) List the names and salary of employees whose salary is between 1000 and 2000

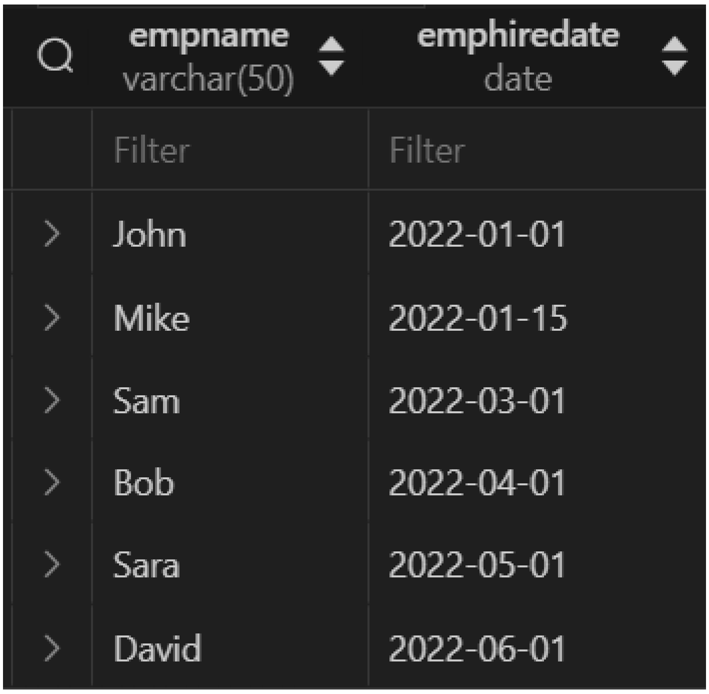
```
select empname, empsal
from employeepr3
where
   empsal between 1000 and 2000
```



empname varchar(50)	empsal int
Filter	Filter
Bob	1000
Olivia	1600
Sophia	1700

p) List the names of employees joined before 30 june'22 or after 31dec'22

```
select empname, emphiredate
from employeepr3
where
    emphiredate < '2022-06-30'
    or emphiredate > '2022-12-31'
```



empname varchar(50)	emphiredate date
Filter	Filter
John	2022-01-01
Mike	2022-01-15
Sam	2022-03-01
Bob	2022-04-01
Sara	2022-05-01
David	2022-06-01

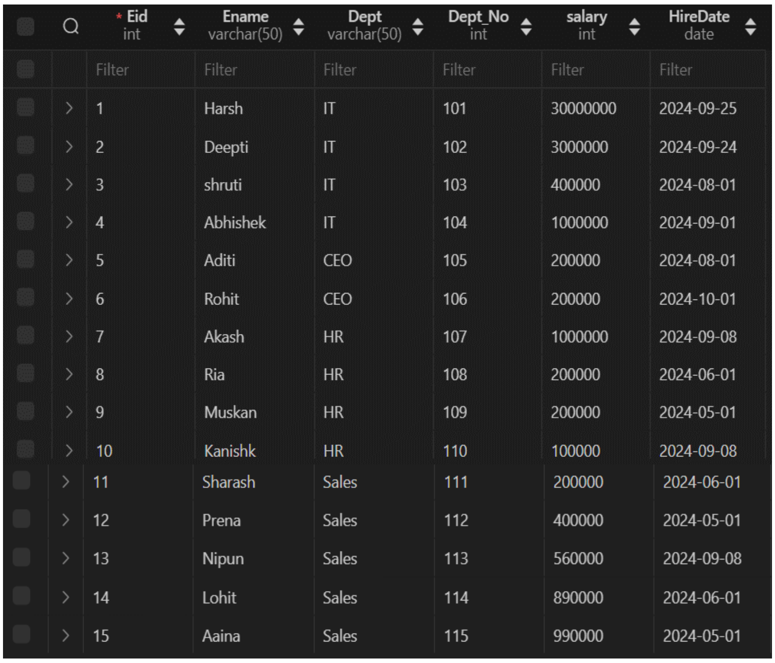
Date – 11/09/2024

Practical No : 4

AIM - Using Relational, Logical Operators

1) Display all records

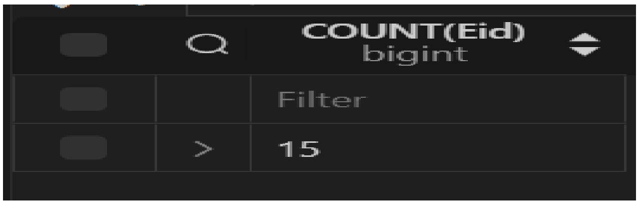
select * from employee;



* Eid int	Ename varchar(50)	Dept varchar(50)	Dept_No int	salary int	HireDate date
Filter	Filter	Filter	Filter	Filter	Filter
1	Harsh	IT	101	30000000	2024-09-25
2	Deepti	IT	102	3000000	2024-09-24
3	shruti	IT	103	400000	2024-08-01
4	Abhishek	IT	104	1000000	2024-09-01
5	Aditi	CEO	105	200000	2024-08-01
6	Rohit	CEO	106	200000	2024-10-01
7	Akash	HR	107	1000000	2024-09-08
8	Ria	HR	108	200000	2024-06-01
9	Muskan	HR	109	200000	2024-05-01
10	Kanishk	HR	110	100000	2024-09-08
11	Sharash	Sales	111	200000	2024-06-01
12	Prena	Sales	112	400000	2024-05-01
13	Nipun	Sales	113	560000	2024-09-08
14	Lohit	Sales	114	890000	2024-06-01
15	Aaina	Sales	115	990000	2024-05-01

2) Count total number of employees in the table.

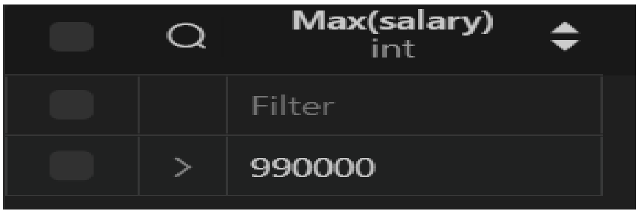
SELECT COUNT(Eid) FROM employee;



COUNT(Eid) bigint
Filter
15

3) List max salary of employee whose dept is ‘Sales’.

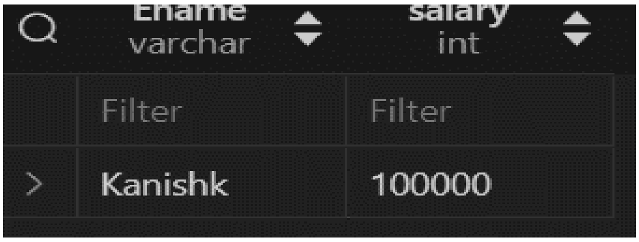
SELECT Max(salary) FROM employee WHERE dept = 'Sales';



Max(salary) int
Filter
990000

4) List min salary and name of employee in all of the department.

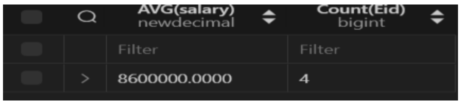
```
SELECT Ename, salary
FROM employee
WHERE
    salary = (
        SELECT MIN(salary)
        FROM employee
    );
```



Ename varchar	salary int
Filter	Filter
Kanishk	100000

5) List avg sal and number of employee working in dept no - 101

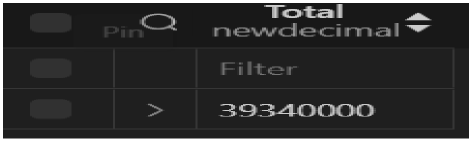
select AVG(salary), Count(Eid) FROM employee WHERE Dept_No = '101';



AVG(salary) newdecimal	Count(Eid) bigint
Filter	Filter
8600000.0000	4

6) Total salary of all employees

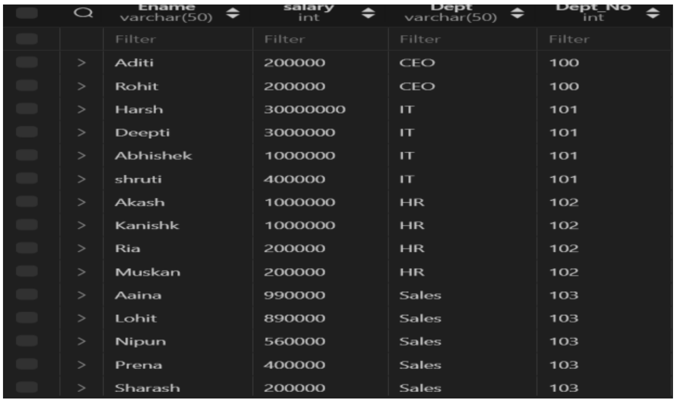
select sum(salary) as Total FROM employee;



Total newdecimal
39340000

7) List ename, salary ,dept and deptno in ascending order of dept_no & descending order of salary

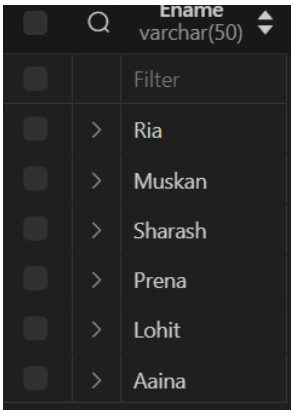
SELECT Ename, salary, Dept, Dept_No
FROM employee
ORDER BY Dept_No, salary DESC;



Ename varchar(50)	salary int	Dept varchar(50)	Dept_No int
Aditi	200000	CEO	100
Rohit	200000	CEO	100
Harsh	30000000	IT	101
Deepti	3000000	IT	101
Abhishek	1000000	IT	101
shruti	400000	IT	101
Akash	1000000	HR	102
Kanishk	1000000	HR	102
Ria	200000	HR	102
Muskan	200000	HR	102
Aaina	990000	Sales	103
Lohit	890000	Sales	103
Nipun	560000	Sales	103
Prena	400000	Sales	103
Sharash	200000	Sales	103

8) List the name of employees who are working in orgainzation for more than 2 months?

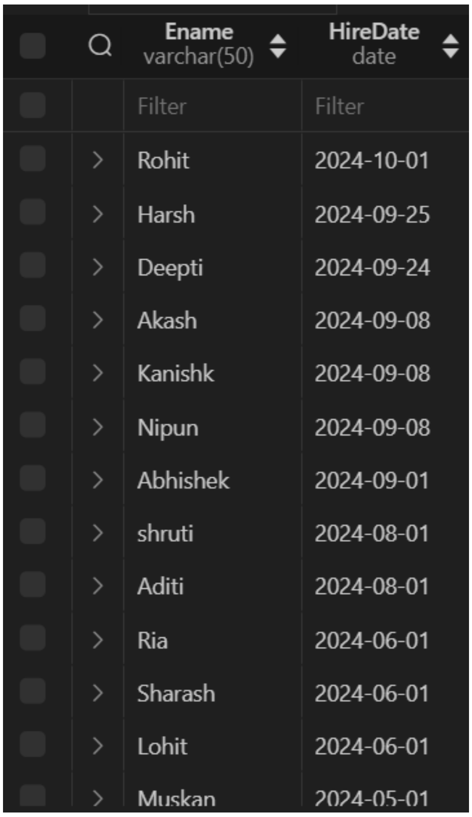
select Ename from employee where datediff(now(), HireDate) > 60



Ename varchar(50)
Ria
Muskan
Sharash
Prena
Lohit
Aaina

9) List ename & hire date in desc order of hire date

select Ename, HireDate from employee order by HireDate desc;



Ename varchar(50)	HireDate date
Rohit	2024-10-01
Harsh	2024-09-25
Deepti	2024-09-24
Akash	2024-09-08
Kanishk	2024-09-08
Nipun	2024-09-08
Abhishek	2024-09-01
shruti	2024-08-01
Aditi	2024-08-01
Ria	2024-06-01
Sharash	2024-06-01
Lohit	2024-06-01
Muskan	2024-05-01

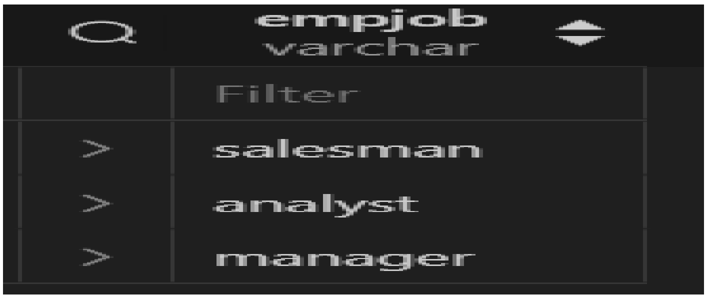
Date – 11/09/2024

Practical No : 5

AIM – Based on null distinct, null, like, derived attribute, order by clause.

1) **List the different jobs availabe in the emp table**

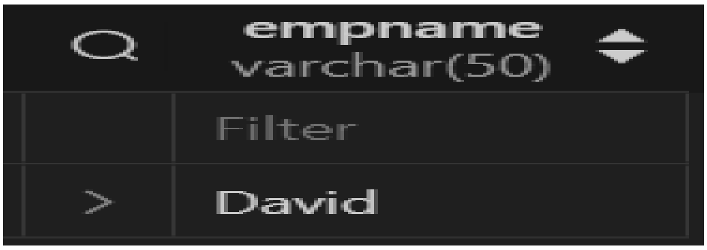
select distinct empjob from employeepr5;



	empjob varchar
	Filter
>	salesman
>	analyst
>	manager

2) **List the employees names that are not eligible for commission**

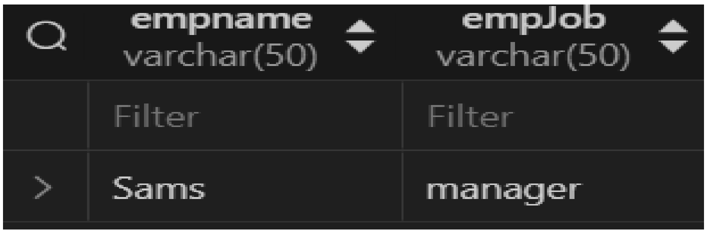
select empname from employeepr5 where empcommission is null;



	empname varchar(50)
	Filter
>	David

3) **List the name of emp and their designation who does not report to anyone**

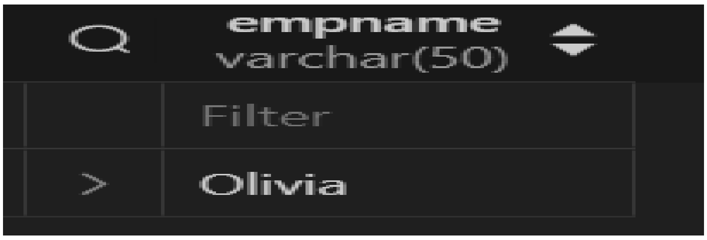
select empname, empJob from employeepr5 where empManagerName is null;



	empname varchar(50)	empJob varchar(50)
	Filter	Filter
>	Sams	manager

4) **List the emp who are not assign to any department**

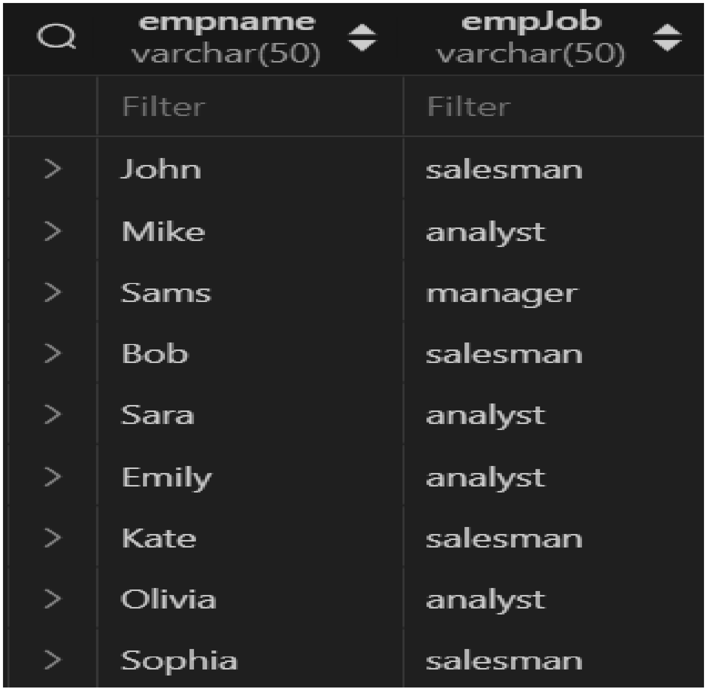
select empname from employeepr5 where empdept is null;



	empname varchar(50)
	Filter
>	Olivia

5) **List the emp name & their designation that is eligible for commission**

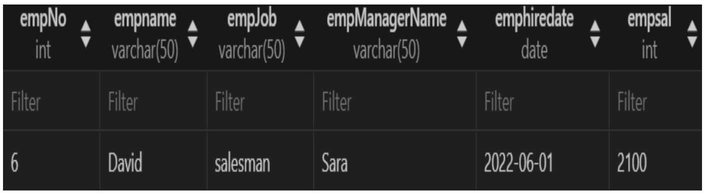
```
select empname, empJob
from employeepr5
where
    empcommission is not null;
```



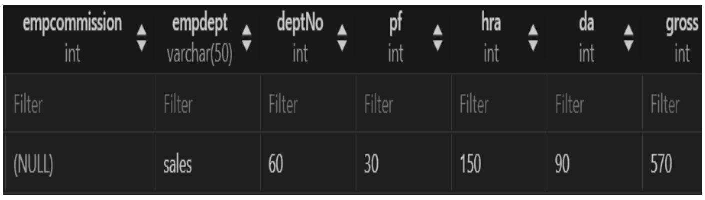
	empname varchar(50)	empJob varchar(50)
	Filter	Filter
>	John	salesman
>	Mike	analyst
>	Sams	manager
>	Bob	salesman
>	Sara	analyst
>	Emily	analyst
>	Kate	salesman
>	Olivia	analyst
>	Sophia	salesman

6) **List the details whose salary is greater than 2000 & commission is null**

```
select *
from employeepr5
where
    empsal > 2000
    and empcommission is null
```



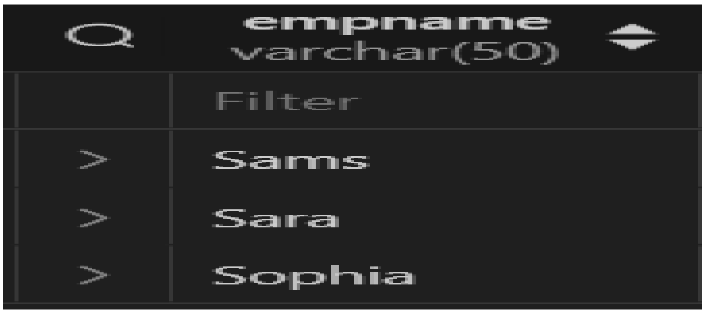
empNo int	empname varchar(50)	empJob varchar(50)	empManagerName varchar(50)	emphiredate date	empsal int
Filter	Filter	Filter	Filter	Filter	Filter
6	David	salesman	Sara	2022-06-01	2100



empcommission int	empdept varchar(50)	deptNo int	pf int	hra int	da int	gross int
Filter	Filter	Filter	Filter	Filter	Filter	Filter
(NULL)	sales	60	30	150	90	570

7) **list the name of employees whose name is starting with S**

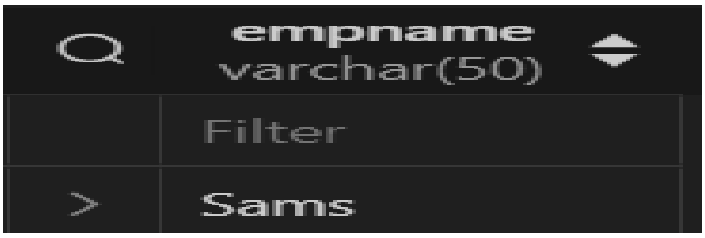
```
select empname from employeepr5 where
empname like 'S%';
```



	empname varchar(50)
	Filter
>	Sams
>	Sara
>	Sophia

8) **list the name of employees whose name is ending with S**

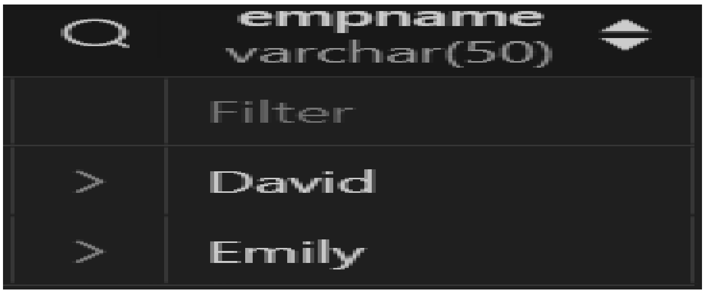
```
select empname from employeepr5 where
empname like '%S';
```



	empname varchar(50)
	Filter
>	Sams

9) **list the name of employees whose name has 5 characters**

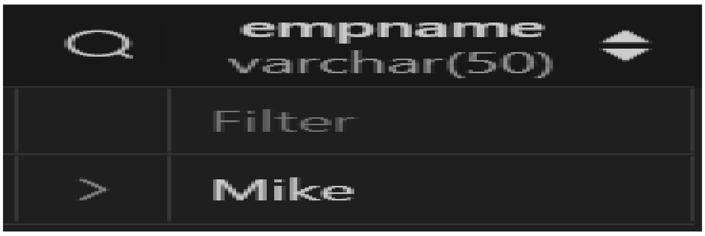
```
select empname from employeepr5 where
length(empname) = 5;
```



	empname varchar(50)
	Filter
>	David
>	Emily

10) list the name of employees whose is having I as second CHARACTER

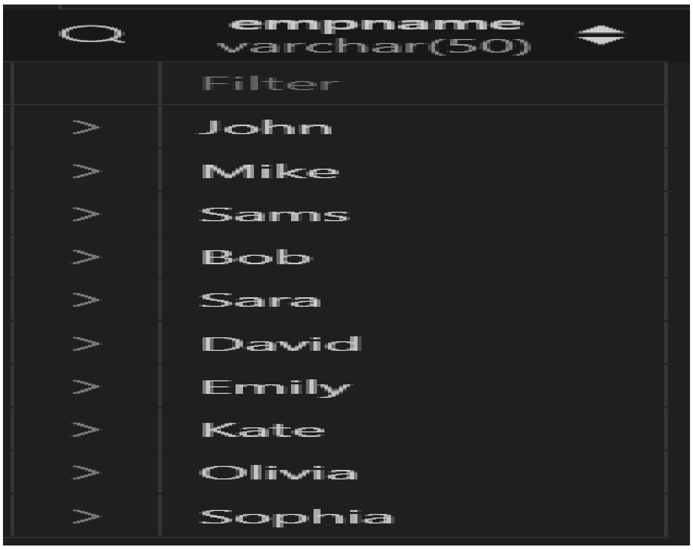
select empname from employeepr5 where empname like '_I%';



empname varchar(50)
Mike

11) List the name of employees who have been with the company for 2 years

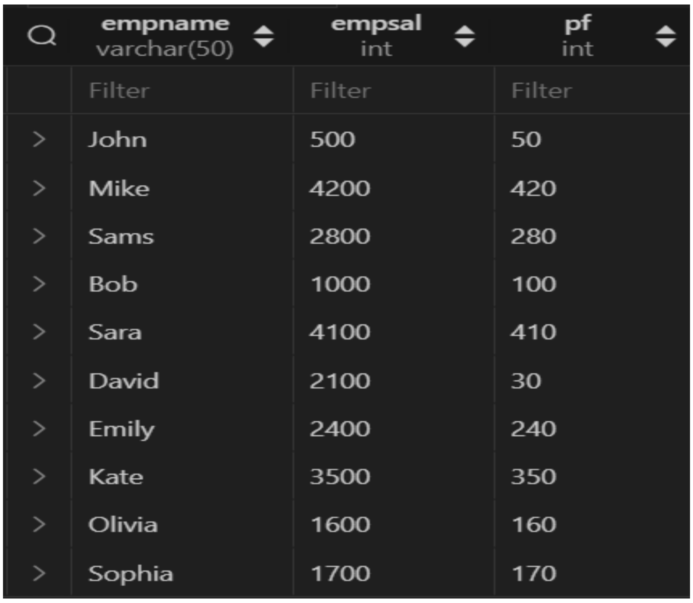
```
SELECT empname
FROM employeepr5
WHERE
   YEAR(CURDATE()) -
YEAR(emphiredate) = 2;
```



empname varchar(50)
John
Mike
Sams
Bob
Sara
David
Emily
Kate
Olivia
Sophia

12) List the name of emp, salary, and pf amount of employees

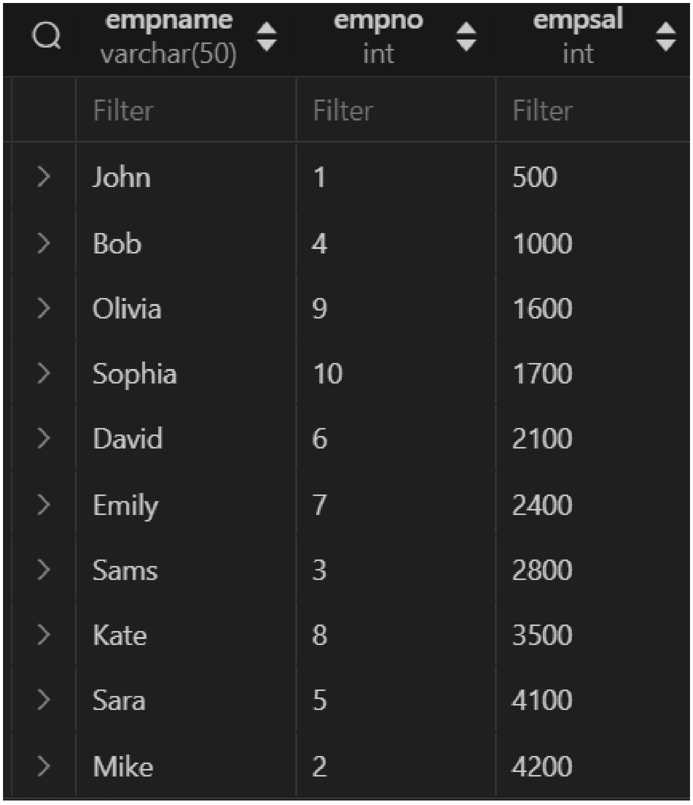
SELECT empname, empsal, pf FROM employeepr5;



empname varchar(50)	empsal int	pf int
John	500	50
Mike	4200	420
Sams	2800	280
Bob	1000	100
Sara	4100	410
David	2100	30
Emily	2400	240
Kate	3500	350
Olivia	1600	160
Sophia	1700	170

13) List the emp name,emp no, salary in ascending order of salary

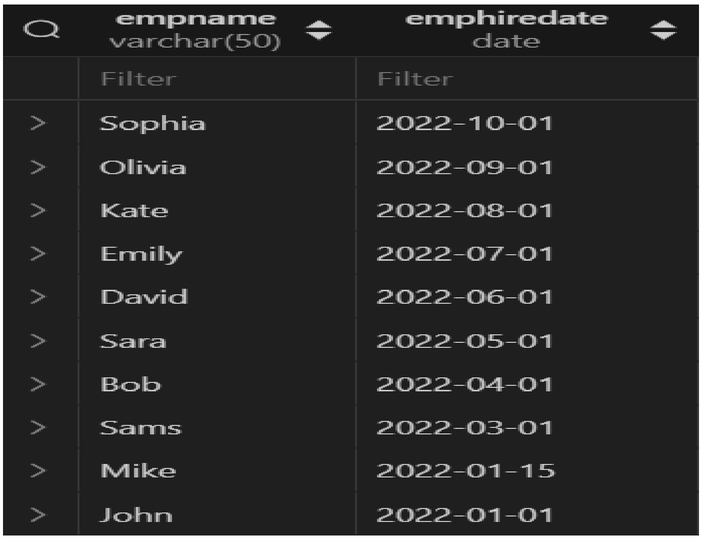
```
SELECT empname, empno, empsal
FROM employeepr5 ORDER BY empsal;
```



empname varchar(50)	empno int	empsal int
Filter	Filter	Filter
John	1	500
Bob	4	1000
Olivia	9	1600
Sophia	10	1700
David	6	2100
Emily	7	2400
Sams	3	2800
Kate	8	3500
Sara	5	4100
Mike	2	4200

14) List the name and hiredate in decsending order of hiredate

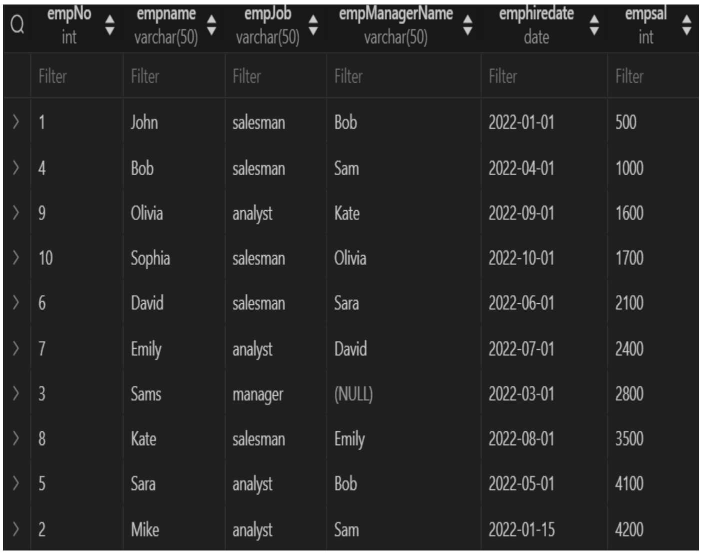
```
SELECT empname, emphiredate
FROM employeepr5
ORDER BY emphiredate DESC;
```



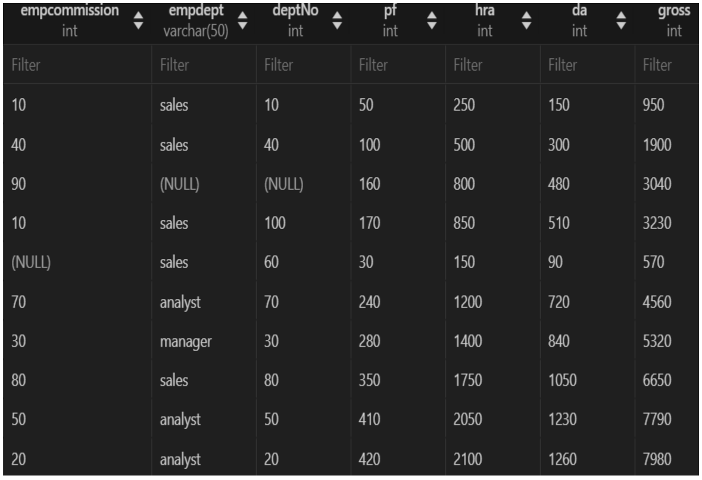
empname varchar(50)	emphiredate date
Filter	Filter
Sophia	2022-10-01
Olivia	2022-09-01
Kate	2022-08-01
Emily	2022-07-01
David	2022-06-01
Sara	2022-05-01
Bob	2022-04-01
Sams	2022-03-01
Mike	2022-01-15
John	2022-01-01

15) List the emp details in ascending order of salary

```
SELECT * FROM employeepr5 ORDER
BY empsal;
```



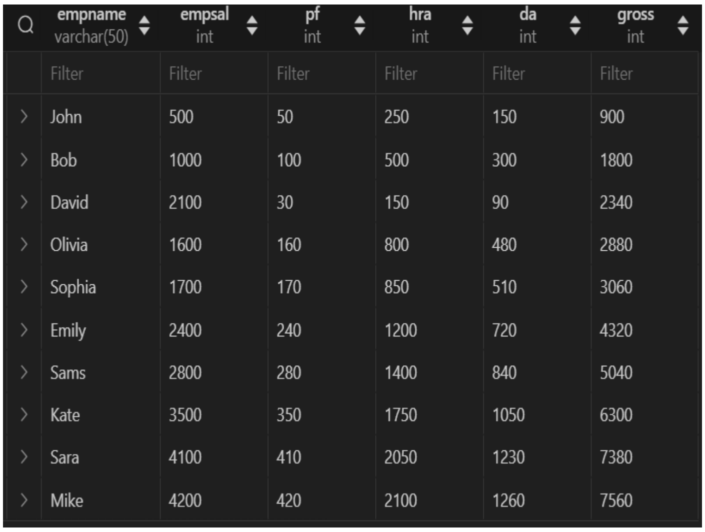
empNo int	empname varchar(50)	empJob varchar(50)	empManagerName varchar(50)	emphiredate date	empsal int
Filter	Filter	Filter	Filter	Filter	Filter
1	John	salesman	Bob	2022-01-01	500
4	Bob	salesman	Sam	2022-04-01	1000
9	Olivia	analyst	Kate	2022-09-01	1600
10	Sophia	salesman	Olivia	2022-10-01	1700
6	David	salesman	Sara	2022-06-01	2100
7	Emily	analyst	David	2022-07-01	2400
3	Sams	manager	(NULL)	2022-03-01	2800
8	Kate	salesman	Emily	2022-08-01	3500
5	Sara	analyst	Bob	2022-05-01	4100
2	Mike	analyst	Sam	2022-01-15	4200



empcommission int	empdept varchar(50)	deptNo int	pf int	hra int	da int	gross int
Filter	Filter	Filter	Filter	Filter	Filter	Filter
10	sales	10	50	250	150	950
40	sales	40	100	500	300	1900
90	(NULL)	(NULL)	160	800	480	3040
10	sales	100	170	850	510	3230
(NULL)	sales	60	30	150	90	570
70	analyst	70	240	1200	720	4560
30	manager	30	280	1400	840	5320
80	sales	80	350	1750	1050	6650
50	analyst	50	410	2050	1230	7790
20	analyst	20	420	2100	1260	7980

16) List the emp name, salary, pf, hra, da and gross order on the basis of gross

```
SELECT
    empname,
    empsal,
    pf,
    hra,
    da,
    (empsal + hra + da) AS gross
FROM employeepr5
ORDER BY gross;
```



empname varchar(50)	empsal int	pf int	hra int	da int	gross int
Filter	Filter	Filter	Filter	Filter	Filter
John	500	50	250	150	900
Bob	1000	100	500	300	1800
David	2100	30	150	90	2340
Olivia	1600	160	800	480	2880
Sophia	1700	170	850	510	3060
Emily	2400	240	1200	720	4320
Sams	2800	280	1400	840	5040
Kate	3500	350	1750	1050	6300
Sara	4100	410	2050	1230	7380
Mike	4200	420	2100	1260	7560

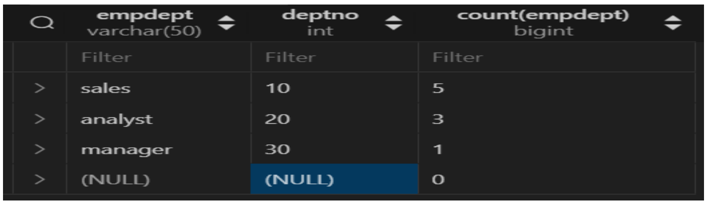
Date – 11/09/2024

Practical No : 6

AIM – Based on null distinct, null, like, derived attribute, order by clause.

1) List the department number & no of employees in each department

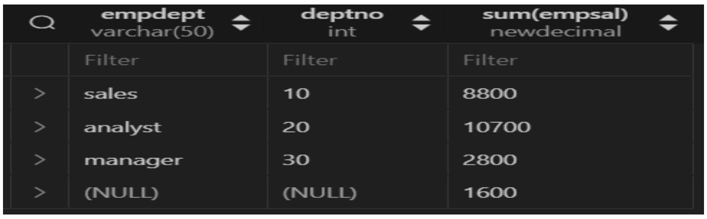
```
select empdept, deptno, count(empdept)
from employeepr6
group by deptno, empdept
```



empdept varchar(50)	deptno int	count(empdept) bigint
Filter	Filter	Filter
sales	10	5
analyst	20	3
manager	30	1
(NULL)	(NULL)	0

2) List the department number, total salary payable to each department

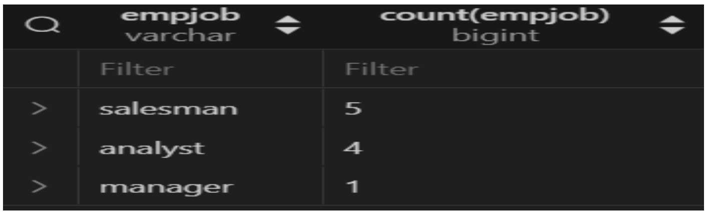
```
select empdept, deptno, sum(empsal)
from employeepr6
group by deptno, empdept
```



empdept varchar(50)	deptno int	sum(empsal) newdecimal
Filter	Filter	Filter
sales	10	8800
analyst	20	10700
manager	30	2800
(NULL)	(NULL)	1600

3) List the jobs & no of emp in each job result must be in descending order of no of employees

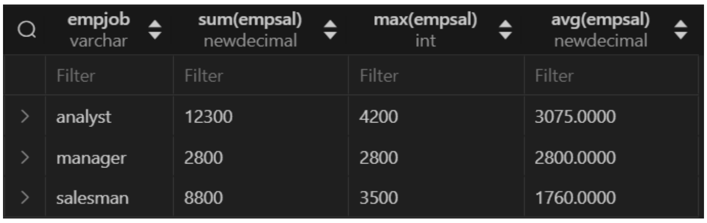
```
select empjob, count(empjob)
from employeepr6
group by empjob
order by count(empjob) desc
```



empjob varchar	count(empjob) bigint
Filter	Filter
salesman	5
analyst	4
manager	1

4) List the sum of all salary, maximum salary & average salary of employees job wise

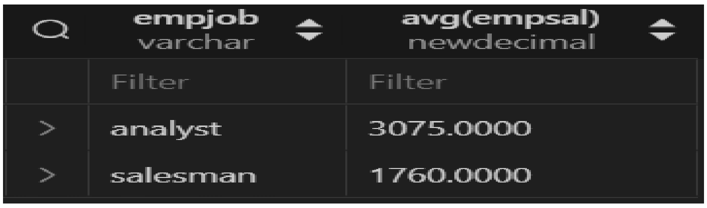
```
select empjob, sum(empsal),
max(empsal), avg(empsal)
from employeepr6
group by empjob
order by empjob
```



empjob varchar	sum(empsal) newdecimal	max(empsal) int	avg(empsal) newdecimal
Filter	Filter	Filter	Filter
analyst	12300	4200	3075.0000
manager	2800	2800	2800.0000
salesman	8800	3500	1760.0000

5) List the average salary from each job excluding manager

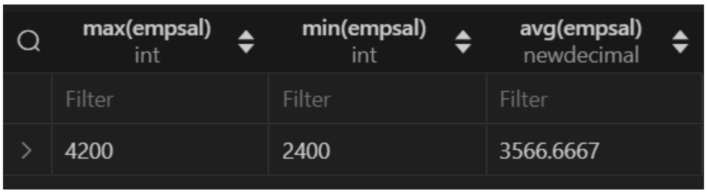
```
select empjob, avg(empsal)
from employeepr6
where empjob != 'Manager'
group by empjob
order by empjob
```



empjob varchar	avg(empsal) newdecimal
Filter	Filter
analyst	3075.0000
salesman	1760.0000

6) List the total salary max & min salary, avg sal of employees jobwise for department number 20

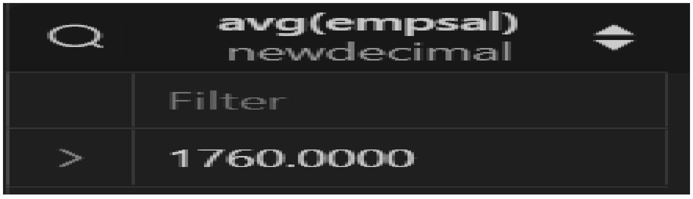
```
select max(empsal), min(empsal),
avg(empsal)
from employeepr6
where deptno = 20
order by empjob
```



max(empsal) int	min(empsal) int	avg(empsal) newdecimal
Filter	Filter	Filter
4200	2400	3566.6667

7) **List the avgrage salary of all the departments employing more than 4 people**

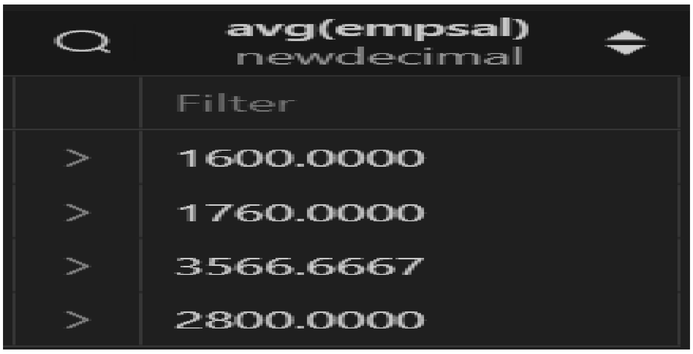
```
select avg(empsal)
from employeepr6
WHERE
   deptno IN (
      SELECT deptno
      FROM employeepr6
      GROUP BY
         deptno
      HAVING
         count(deptno) > 4
   )
```



```
avg(empsal)
newdecimal
Filter
>   1760.0000
```

8) **List avgrage monthly salary for each job type within department number outer group by department number in a group by job**

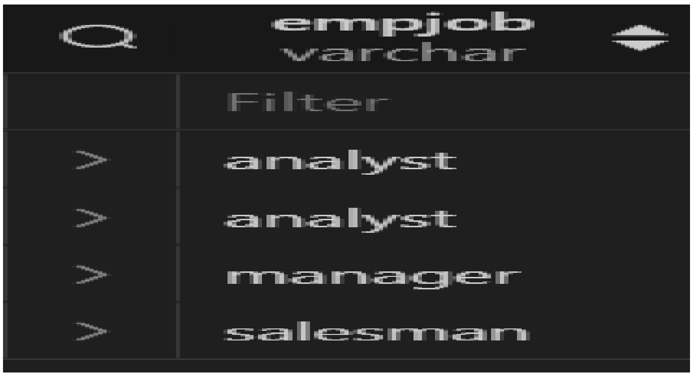
```
select avg(empsal)
from employeepr6
group by deptno, empjob
order by deptno, empjob
```



```
avg(empsal)
newdecimal
Filter
>   1600.0000
>   1760.0000
>   3566.6667
>   2800.0000
```

9) **List job of employees where salary >=2500**

```
select empjob
from employeepr6
where empsal >= 2500
order by empjob
```



```
empjob
varchar
Filter
>   analyst
>   analyst
>   manager
>   salesman
```

Date – 11/09/2024

Practical No : 7

AIM –

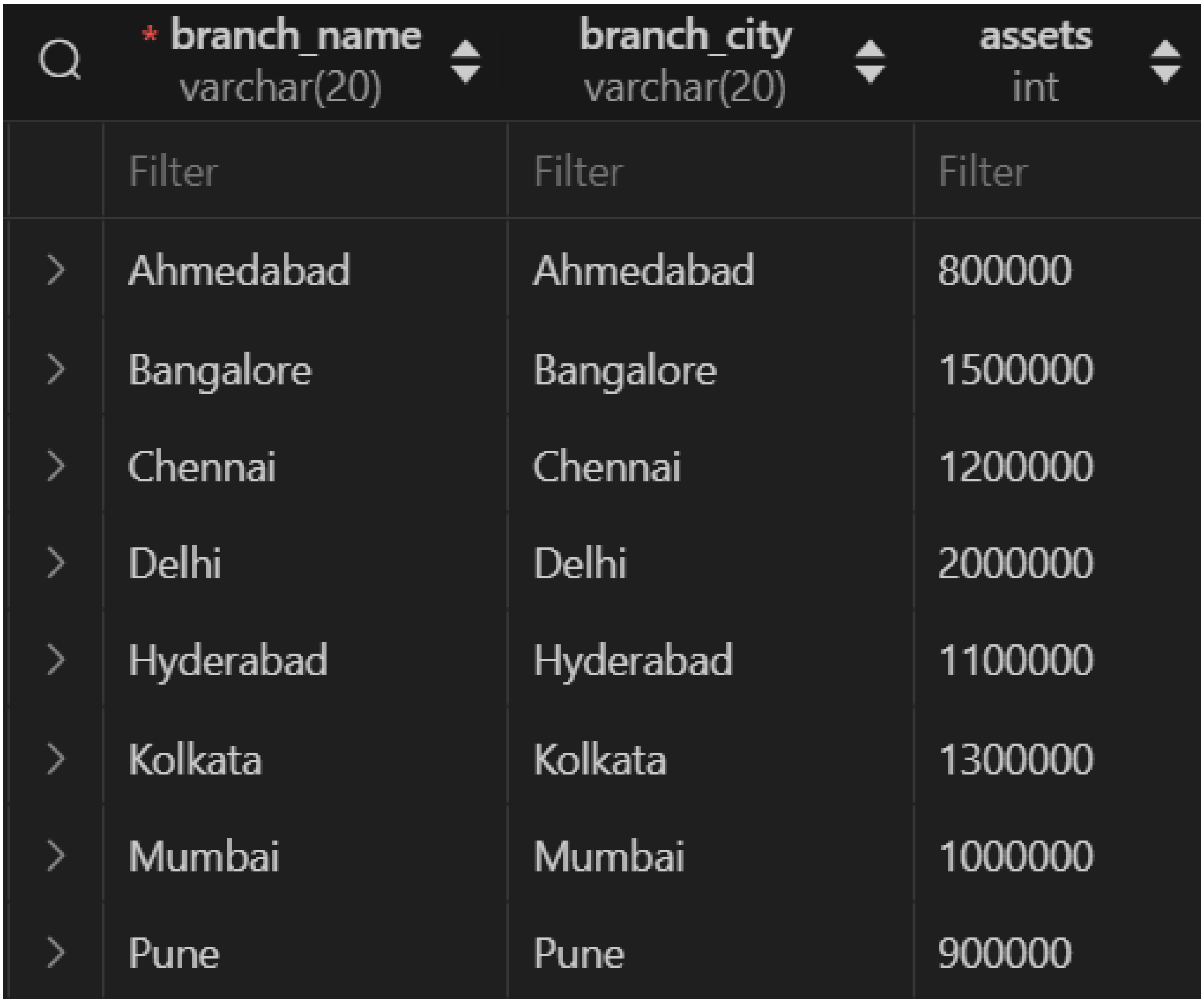
1) Create table Branch and insert values into it.

i)

```
create Table branchpr7 (
  branch_name varchar(20) primary key,
  branch_city varchar(20),
  assets int
);
```

ii)

```
insert into
 branchpr7
values ('Mumbai', 'Mumbai', 1000000),
 ('Delhi', 'Delhi', 2000000),
 (
    'Bangalore',
    'Bangalore',
    1500000
 ),
 ('Chennai', 'Chennai', 1200000),
 ('Kolkata', 'Kolkata', 1300000),
 (
    'Hyderabad',
    'Hyderabad',
    1100000
 ),
 ('Pune', 'Pune', 900000),
(
    'Ahmedabad',
    'Ahmedabad',
    800000
);
```



* branch_name varchar(20)	branch_city varchar(20)	assets int
Filter	Filter	Filter
Ahmedabad	Ahmedabad	800000
Bangalore	Bangalore	1500000
Chennai	Chennai	1200000
Delhi	Delhi	2000000
Hyderabad	Hyderabad	1100000
Kolkata	Kolkata	1300000
Mumbai	Mumbai	1000000
Pune	Pune	900000

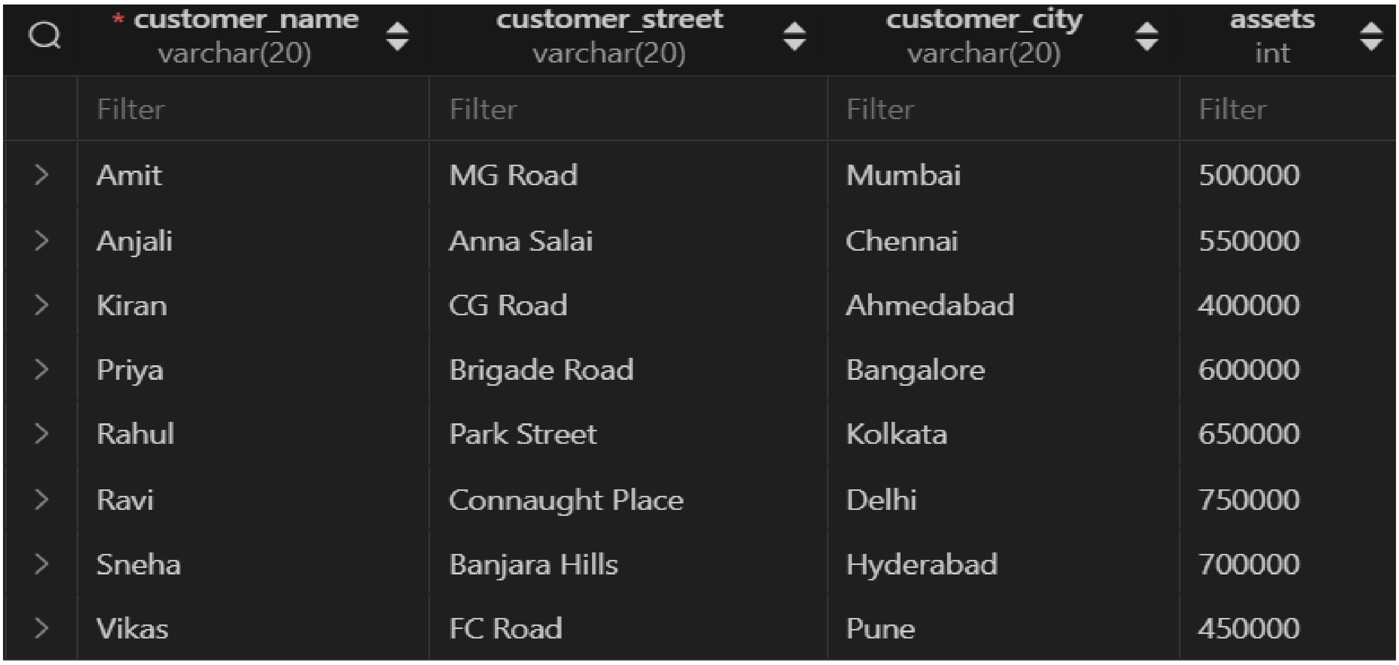
2) Create table Customer and insert values into it.

i)

```
create Table customerpr7 (
    customer_name varchar(20) primary key,
    customer_street varchar(20),
    customer_city varchar(20),
    assets int
);
```

ii)

```
insert into
    customerpr7
values (
        'Amit',
        'MG Road',
        'Mumbai',
        500000
    ),(
        'Ravi',
        'Connaught Place',
        'Delhi',
        750000
    ), (
        'Priya',
        'Brigade Road',
        'Bangalore',
        600000
    ), (
        'Anjali',
        'Anna Salai',
        'Chennai',
    550000
), (
    'Rahul',
    'Park Street',
    'Kolkata',
    650000
), (
    'Sneha',
    'Banjara Hills',
    'Hyderabad',
    700000
), (
    'Vikas',
    'FC Road',
    'Pune',
    450000
), (
    'Kiran',
    'CG Road',
    'Ahmedabad',
    400000
);
```



	* customer_name varchar(20)	customer_street varchar(20)	customer_city varchar(20)	assets int
	Filter	Filter	Filter	Filter
>	Amit	MG Road	Mumbai	500000
>	Anjali	Anna Salai	Chennai	550000
>	Kiran	CG Road	Ahmedabad	400000
>	Priya	Brigade Road	Bangalore	600000
>	Rahul	Park Street	Kolkata	650000
>	Ravi	Connaught Place	Delhi	750000
>	Sneha	Banjara Hills	Hyderabad	700000
>	Vikas	FC Road	Pune	450000

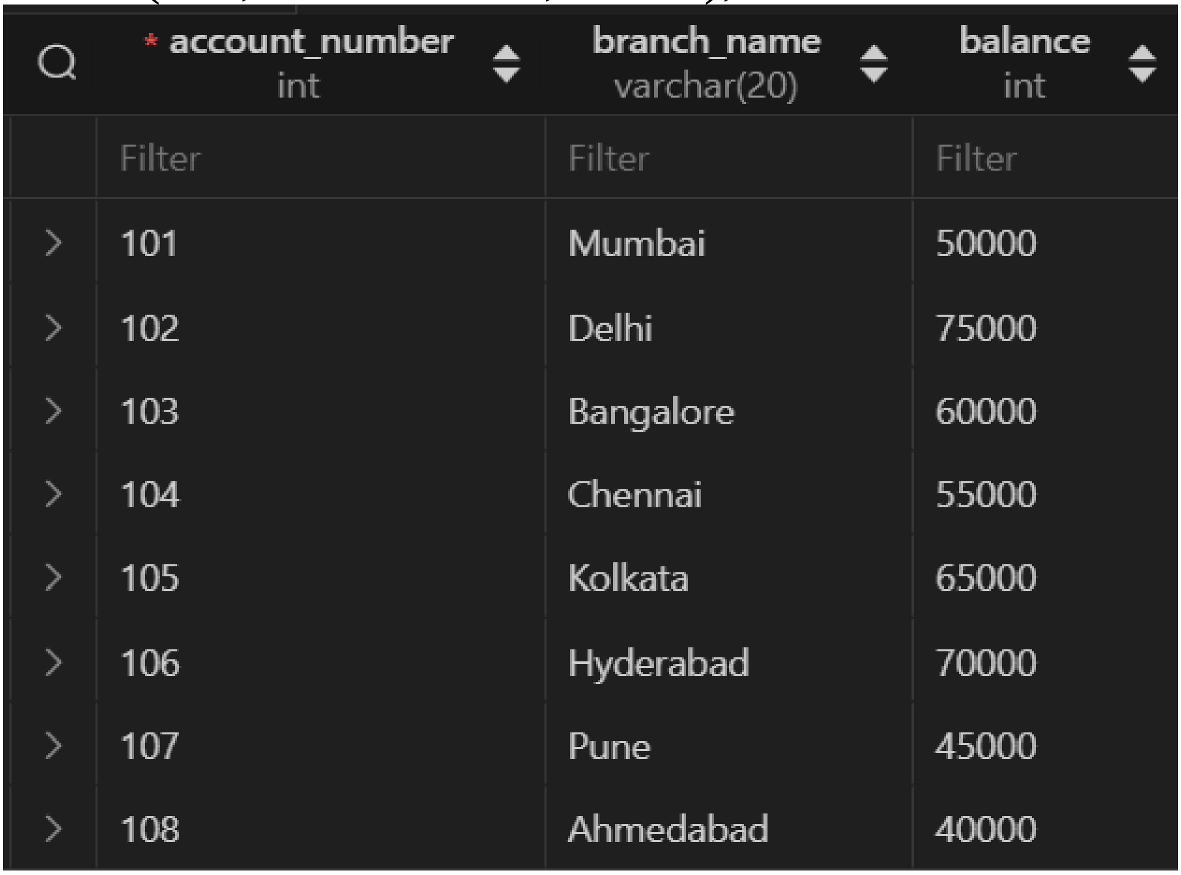
3) **Create table Account and insert values into it.**

i)

```
create Table accountpr7 (
    account_number int primary key,
    branch_name varchar(20),
    balance int
);
```

ii)

```
insert into
    accountpr7
values (101, 'Mumbai', 50000),
    (102, 'Delhi', 75000),
    (103, 'Bangalore', 60000),
    (104, 'Chennai', 55000),
    (105, 'Kolkata', 65000),
    (106, 'Hyderabad', 70000),
    (107, 'Pune', 45000),
    (108, 'Ahmedabad', 40000);
```



	* account_number int	branch_name varchar(20)	balance int
	Filter	Filter	Filter
>	101	Mumbai	50000
>	102	Delhi	75000
>	103	Bangalore	60000
>	104	Chennai	55000
>	105	Kolkata	65000
>	106	Hyderabad	70000
>	107	Pune	45000
>	108	Ahmedabad	40000

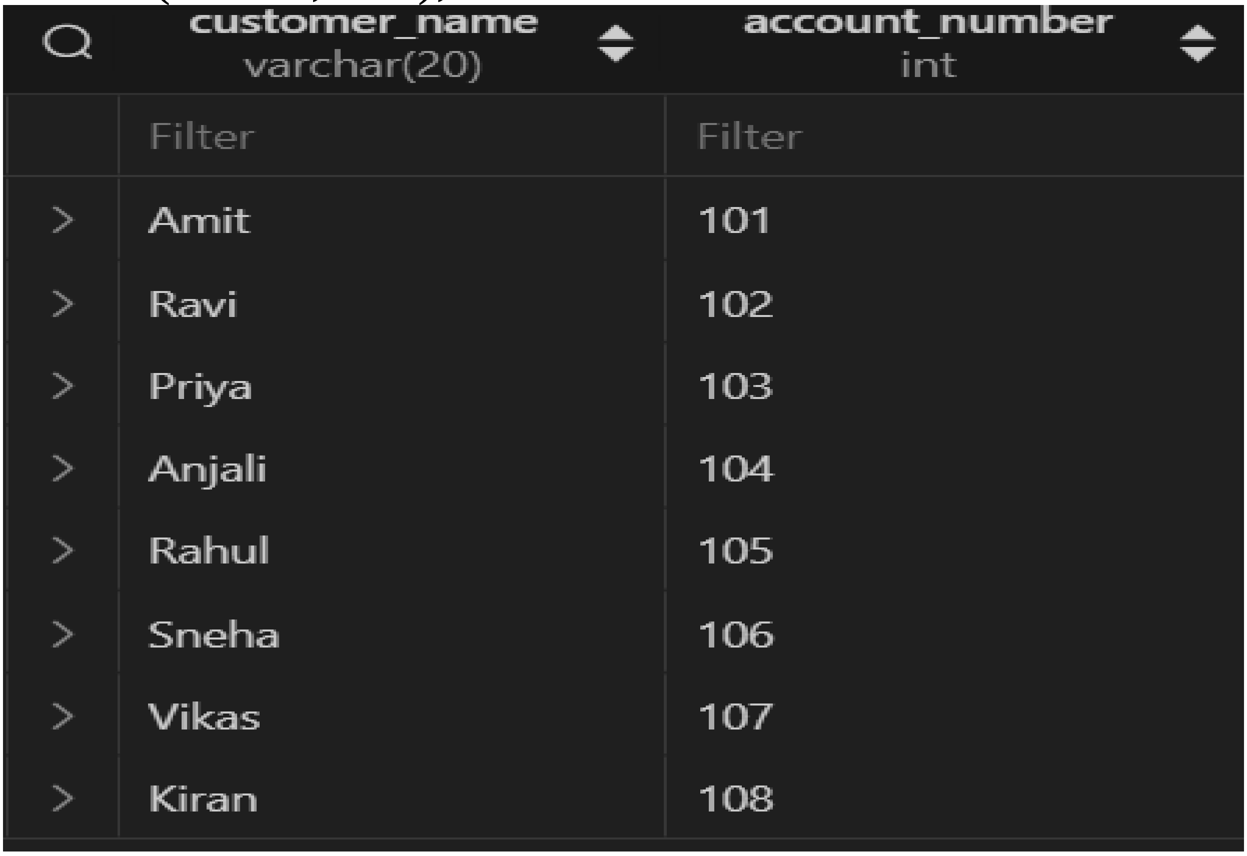
4) **Create table Depositor and insert values into it.**

i)

```
create Table Depositorpr7 (
    customer_name varchar(20),
    account_number int
);
```

ii)

```
insert into
    Depositorpr7
values ('Amit', 101),
    ('Ravi', 102),
    ('Priya', 103),
    ('Anjali', 104),
    ('Rahul', 105),
    ('Sneha', 106),
    ('Vikas', 107),
    ('Kiran', 108);
```



customer_name varchar(20)	account_number int
Amit	101
Ravi	102
Priya	103
Anjali	104
Rahul	105
Sneha	106
Vikas	107
Kiran	108

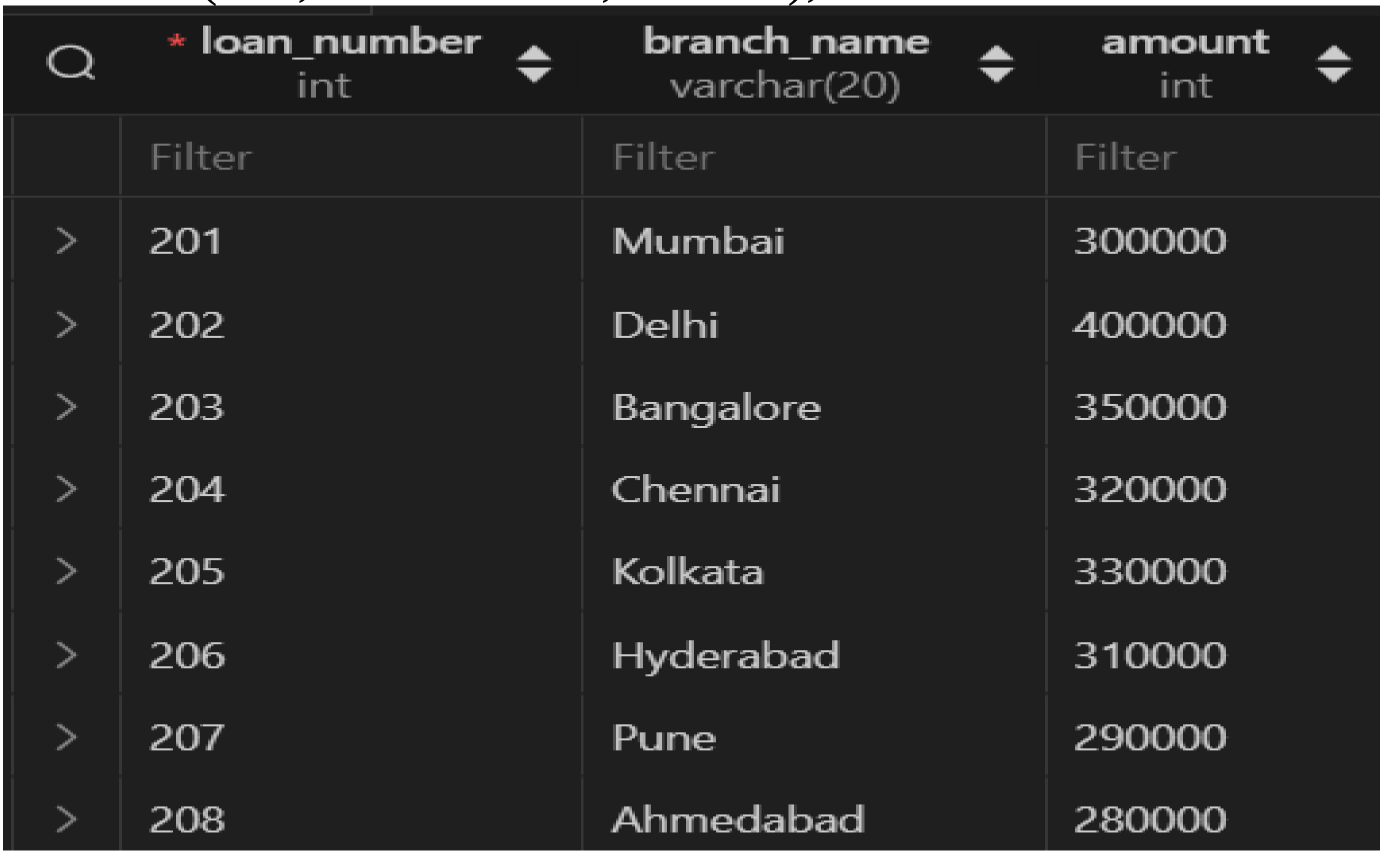
5) **Create table Loan and insert values into it.**

i)

```
create Table loanpr7 (
    loan_number int primary key,
    branch_name varchar(20),
    amount int
);
```

ii)

```
insert into
    loanpr7
values (201, 'Mumbai', 300000),
    (202, 'Delhi', 400000),
    (203, 'Bangalore', 350000),
    (204, 'Chennai', 320000),
    (205, 'Kolkata', 330000),
    (206, 'Hyderabad', 310000),
    (207, 'Pune', 290000),
    (208, 'Ahmedabad', 280000);
```



* loan_number int	branch_name varchar(20)	amount int
Filter	Filter	Filter
201	Mumbai	300000
202	Delhi	400000
203	Bangalore	350000
204	Chennai	320000
205	Kolkata	330000
206	Hyderabad	310000
207	Pune	290000
208	Ahmedabad	280000

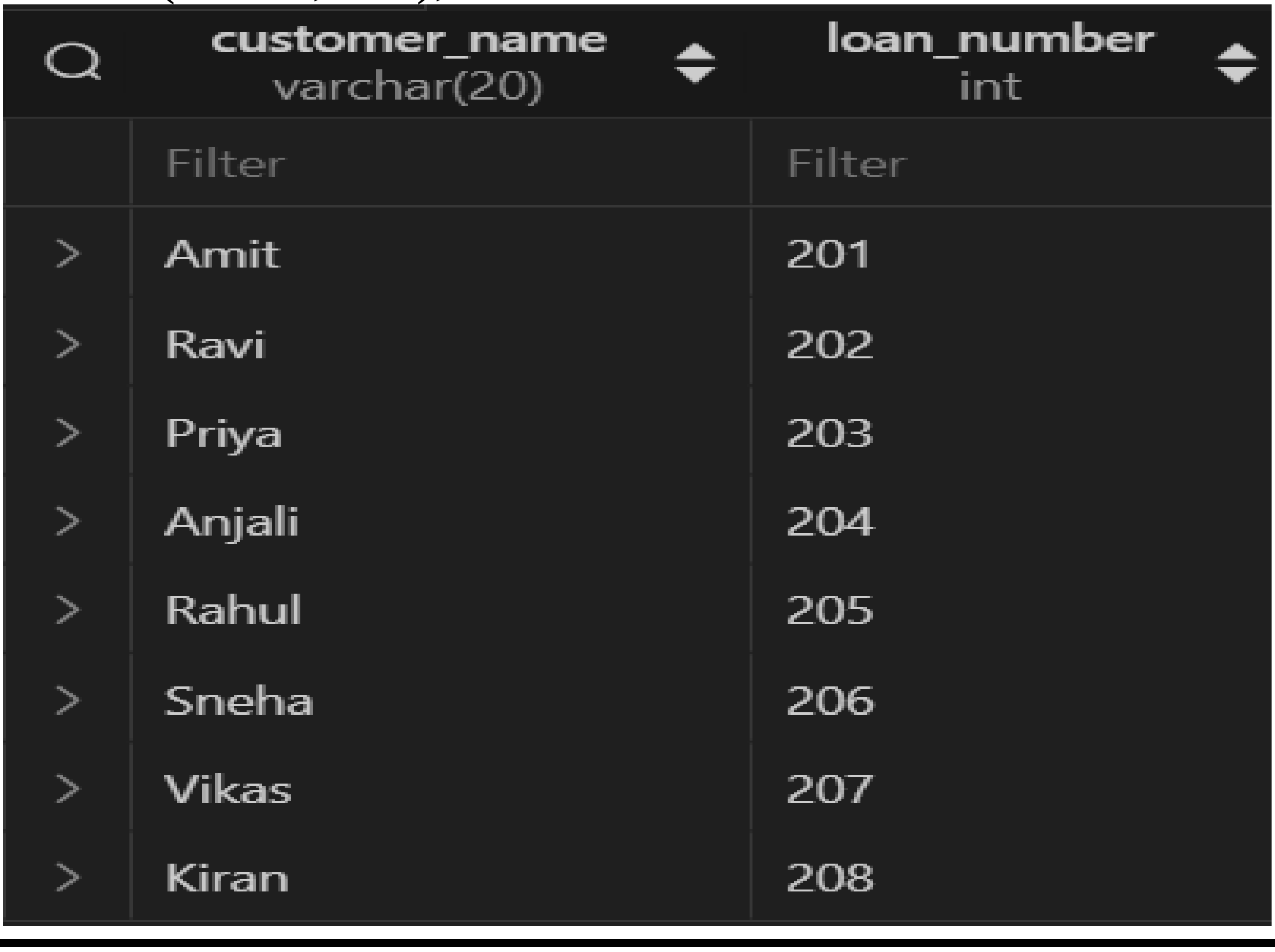
6) Create table Borrower and insert values into it.

i)

```
create Table borrowerpr7 (
    customer_name varchar(20),
    loan_number int
);
```

ii)

```
insert into
    borrowerpr7
values ('Amit', 201),
    ('Ravi', 202),
    ('Priya', 203),
    ('Anjali', 204),
    ('Rahul', 205),
    ('Sneha', 206),
    ('Vikas', 207),
    ('Kiran', 208);
```



customer_name varchar(20)	loan_number int
Filter	Filter
Amit	201
Ravi	202
Priya	203
Anjali	204
Rahul	205
Sneha	206
Vikas	207
Kiran	208

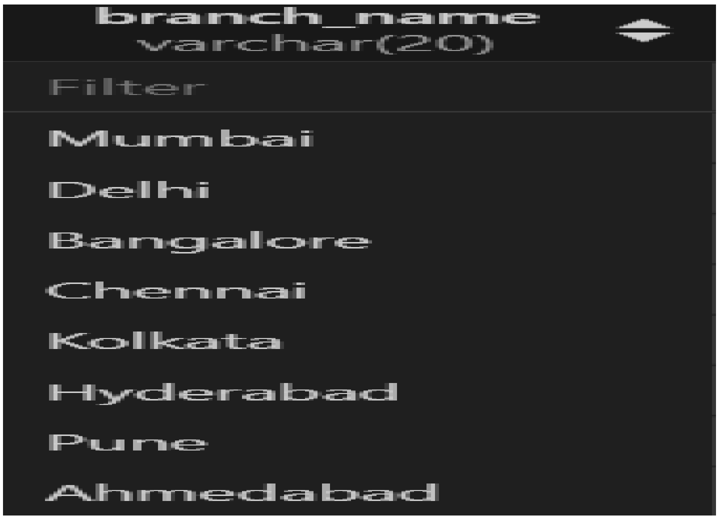
Date – 11/09/2024

Practical No : 8

AIM –

1) **Find the names of all branches in the loan relation**

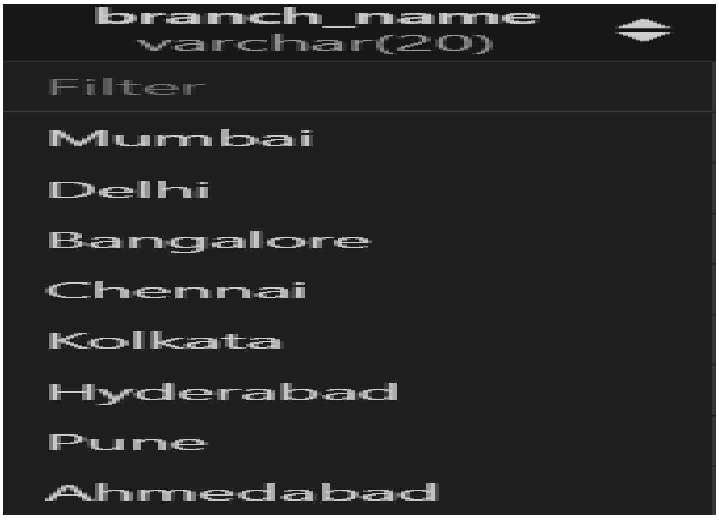
select branch_name from loanpr7;



```
branch_name
varchar(20)
Filter
Mumbai
Delhi
Bangalore
Chennai
Kolkata
Hyderabad
Pune
Ahmedabad
```

2) **Find the names of all branches in the loan relation and remove duplicates**

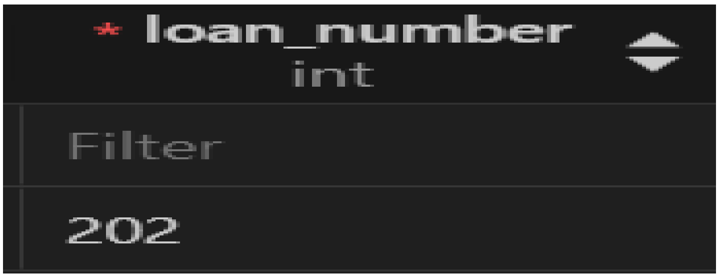
select distinct branch_name from loanpr7;



```
branch_name
varchar(20)
Filter
Mumbai
Delhi
Bangalore
Chennai
Kolkata
Hyderabad
Pune
Ahmedabad
```

3) **Find the loan number for loans made by delhi branch with loan amount >320000**

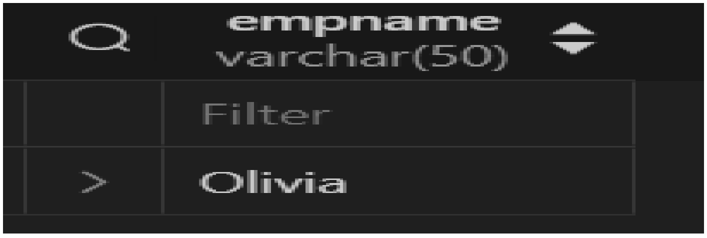
```
select loan_number
from loanpr7
where
    branch_name = 'Delhi'
    and amount > 320000;
```



```
* loan_number
int
Filter
202
```

4) List the emp who are not assign to any department

select empname from employeepr5 where empdept is null;

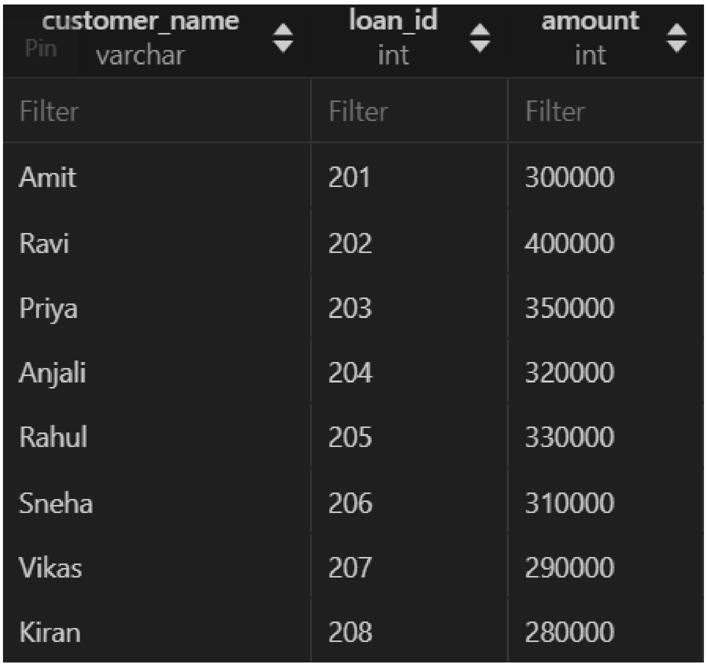


empname varchar(50)
Filter
Olivia

5)

6) Find the name, loan number, loan amount of all customers & rename the columns name as loan_number as loan_id

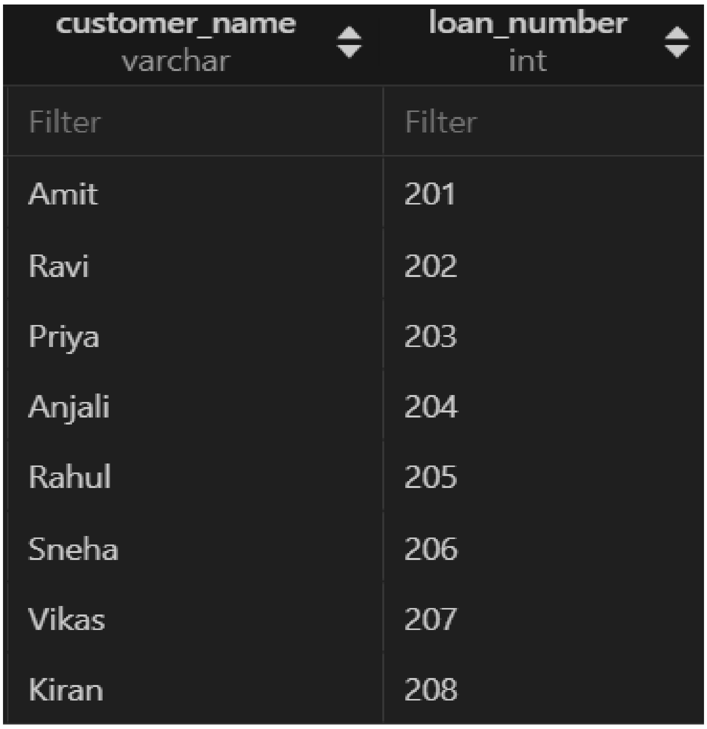
```
select
    customer_name,
    loan_number as loan_id,
    amount
from borrowerpr7
    natural join loanpr7;
```



customer_name varchar	loan_id int	amount int
Filter	Filter	Filter
Amit	201	300000
Ravi	202	400000
Priya	203	350000
Anjali	204	320000
Rahul	205	330000
Sneha	206	310000
Vikas	207	290000
Kiran	208	280000

7) Find the customer names & their loan numbers for all the customers having loan at some branch

```
select customer_name,
loan_number
from borrowerpr7
    natural join loanpr7;
```



customer_name varchar	loan_number int
Filter	Filter
Amit	201
Ravi	202
Priya	203
Anjali	204
Rahul	205
Sneha	206
Vikas	207
Kiran	208

8) Find the names of all branches that have greater assests then some branch located in Jaipur

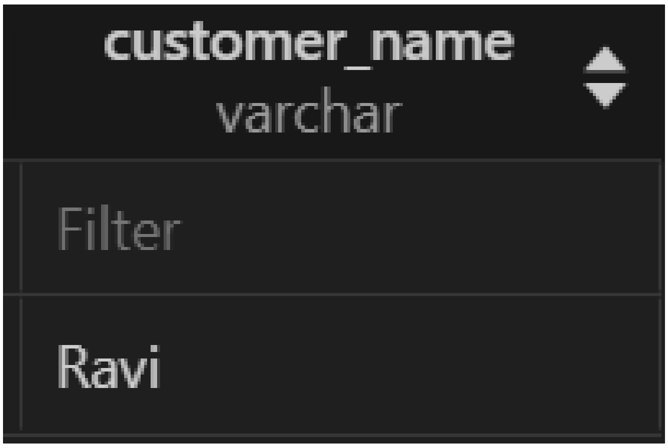
```
select branch_name
from loanpr7
where
    amount > any (
        select amount
        from loanpr7
        where
            branch_name = 'Jaipur'
    );
```

9) Find the names of all customes whose string incluedes the string "Main"

```
select customer_name
from borrowerpr7
where
    customer_name like
'%Main%';
```

10) List in aplhabetical order the names of all the customers having loan in delhi branch

```
select customer_name
from borrowerpr7
    natural join loanpr7
where
    branch_name = 'Delhi'
order by customer_name;
```



```
customer_name
varchar
Ravi
```

11) Find the all customers who have loan & account or both

```
select customer_name
from borrowerpr7
    natural join accountpr7
    natural join loanpr7;
```

12) Find the customers who have both loan & account

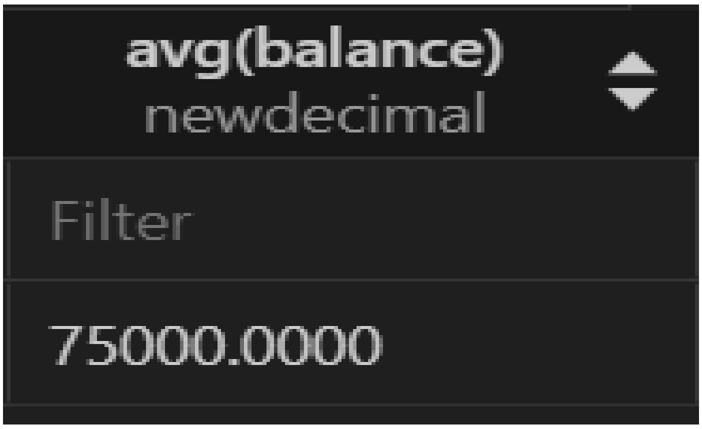
```
select customer_name
from borrowerpr7
    natural join accountpr7
    natural join loanpr7;
```

13) Find the customers who have either loan or account or both

```
select customer_name
from
    borrowerpr7 natural full
    join accountpr7 natural full
    join loanpr7;
```

14) Find the average account balance at delhi branch

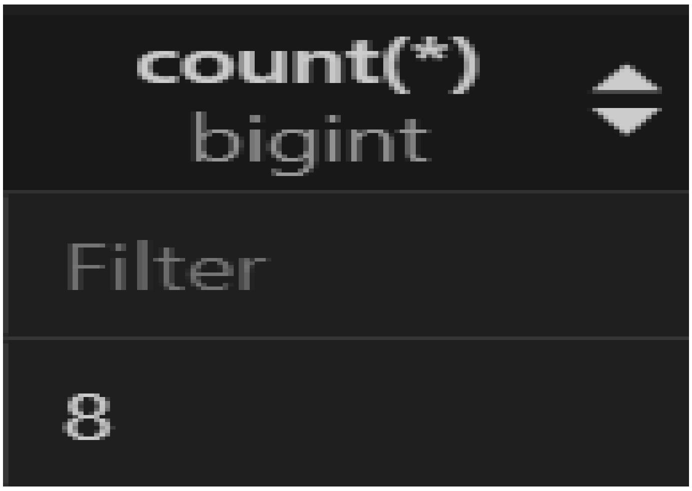
```
select avg(balance) from
accountpr7 where
branch_name = 'Delhi';
```



```
avg(balance)
newdecimal
Filter
75000.0000
```

15) Find the number of tuples in the customer relation

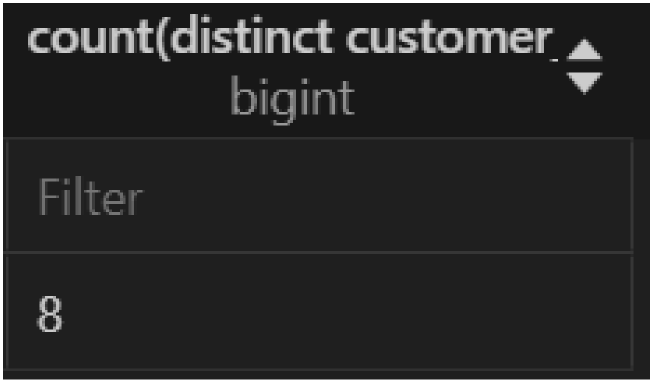
```
select count(*) from
customerpr7;
```



```
count(*)
bigint
Filter
8
```

16) Find the number of depositors on the bank

```
select count(distinct
customer_name) from
depositorpr7;
```



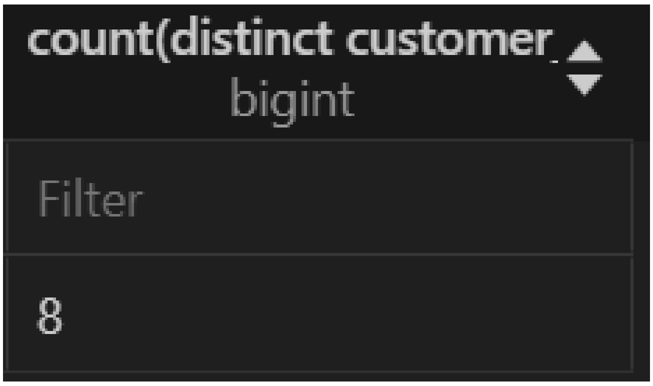
count(distinct customer_
bigint

Filter

8

17) Find the number of depositors in each branch

```
select count(distinct
customer_name) from
depositorpr7;
```



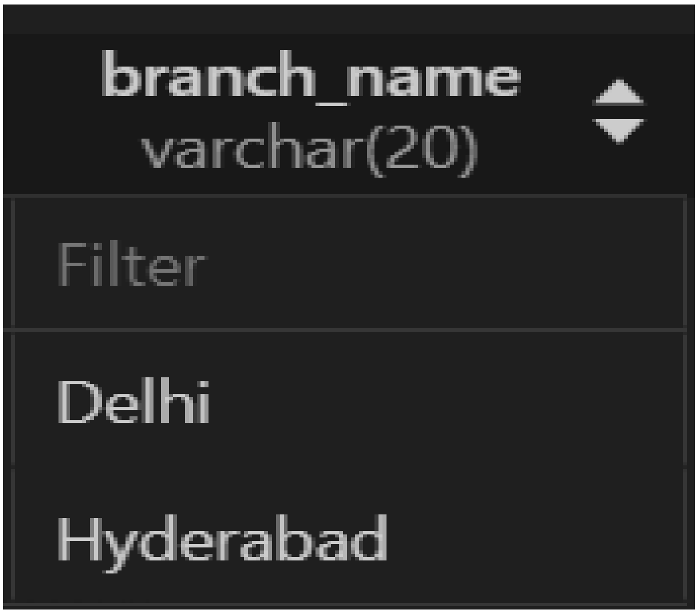
count(distinct customer_
bigint

Filter

8

18) Find the names of all branches where the average account balance is greater than 65000

```
select branch_name
from accountpr7
group by
    branch_name
having
    avg(balance) > 65000;
```



branch_name
varchar(20)

Filter

Delhi

Hyderabad

Oracle Express Edition

1. Introduction to Oracle Express Edition (XE)

- **Overview**: Describe Oracle XE as a free-to-use, entry-level version of Oracle's RDBMS. Mention its suitability for beginners, developers, and small applications.
- **Purpose and Target Audience**: Explain that Oracle XE is intended for students, developers, and small businesses needing database management without the cost.
- **Key Advantages**: Highlight the benefits of using XE, like a full-featured database experience, lower resource requirements, and easy setup.

2. Features of Oracle XE

- **Free to Use**: Emphasize that XE is a no-cost, fully functional version of Oracle's database.
- **Lightweight and Limited Resource Usage**: Suitable for smaller applications due to limitations (e.g., 2GB memory limit, up to 12GB data).
- **Comprehensive Database Features**: Includes SQL, PL/SQL, RESTful services, JSON, and support for Oracle Application Express (APEX).
- **Platform Compatibility**: Works on various platforms, including Windows and Linux.
- **Community and Learning Support**: Access to extensive Oracle documentation and community resources.

3. Limitations and Comparisons to Other Oracle Editions

- **Usage Limits**: Explain the storage (12GB), CPU (up to 2 CPU threads), and RAM (2GB) restrictions.
- **Feature Limitations**: Some advanced features (e.g., clustering, certain performance enhancements) are not available in XE.
- **Comparison with Other Editions**: Briefly compare with Standard, Enterprise, and Cloud Editions to contextualize where XE fits.

4. Technical Features and Capabilities

- **Storage and Performance Management**: Describe its capability for handling moderate-sized datasets.
- **Data Security**: Explain features such as encryption, data masking, and user access control.
- **Data Access and APIs**: Support for RESTful services, allowing external applications to interact with data via web services.
- **Development Tools and Compatibility with APEX**: Oracle XE supports APEX, enabling web-based application development.
- **Backup and Recovery Options**: Describe the options for backup and recovery to protect data.

5. Installing Oracle XE

- **System Requirements**: Detail requirements such as OS compatibility, minimum storage, RAM, and CPU specifications.
- **Downloading Oracle XE**: Guide users to Oracle's official site for downloading.
- **Installation Steps**:
  - **Windows Installation**:
    1. **Download the Installer**: Download the XE installer for Windows.
    2. **Run the Installer**: Walk through installation steps, including accepting license agreements.
    3. **Set up Default Passwords**: Briefly explain setting up default SYS and SYSTEM user passwords.
  - **Linux Installation**:
    1. **Download and Unpack the RPM File**: Step-by-step guide for installing RPM packages on Linux.
    2. **Setup and Configuration**: Walkthrough of setup scripts and database start commands.
- **Testing the Installation**: Demonstrate logging into the XE database and running a sample query.

6. Basic Usage of Oracle XE

- **Connecting to the Database**: Instructions for logging in using SQL*Plus or Oracle SQL Developer.
- **Creating and Managing Tables**: Basic SQL commands for creating tables, inserting data, and querying.
- **Database Management**: Simple tasks such as creating users, assigning roles, and basic security settings.
- **Using APEX for Application Development**: Brief introduction to setting up APEX and creating a sample application.

7. Use Cases for Oracle XE

- **Education and Learning**: Ideal for students and educators learning SQL and database management.
- **Application Prototyping**: Useful for developers to test and prototype applications.
- **Small Business Applications**: Discuss use cases like inventory management, CRM, etc.
- **Personal Projects and Experiments**: Mention how hobbyists and individual developers benefit from XE.

8. Conclusion

- **Summary of Key Points**: Recap the benefits, features, and installation of Oracle XE.
- **Recommendation for Use**: Summarize who should use Oracle XE and its value for specific users and organizations.
- **Future Prospects**: Mention how users can upgrade to higher Oracle editions if needed.

Installing the Oracle SQL Developer Client

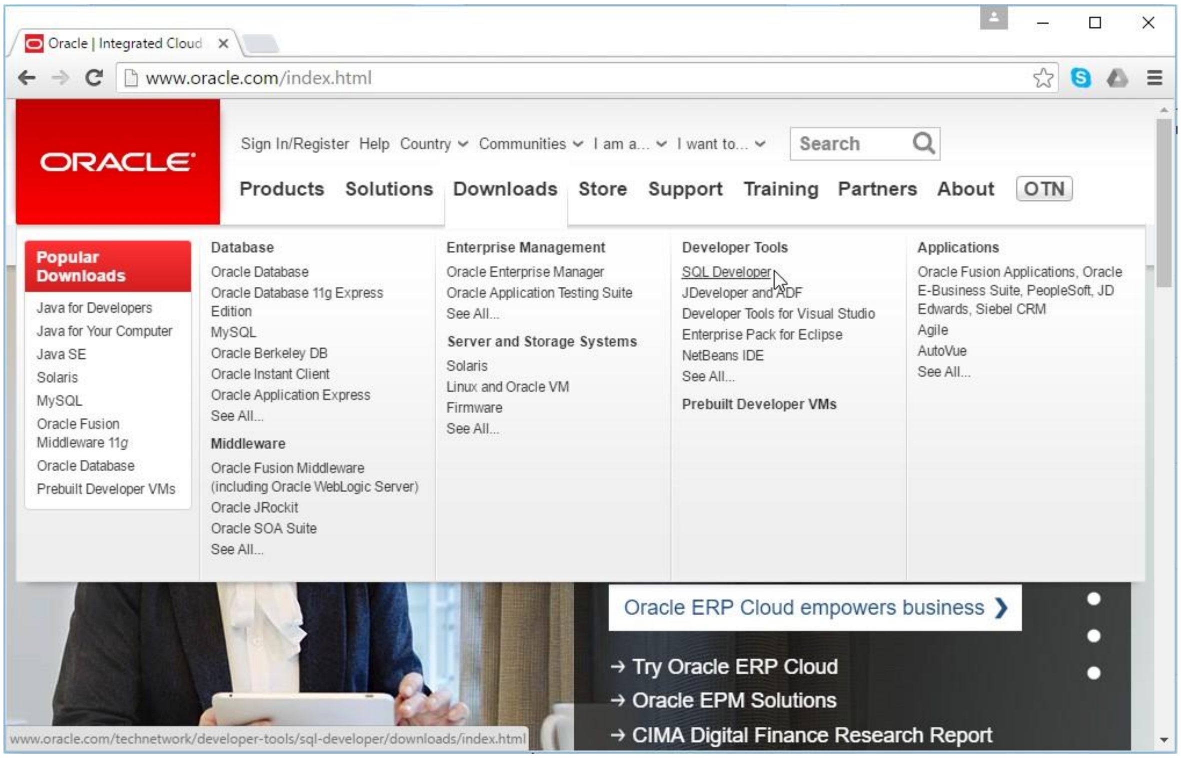
Step 1: Downloading Oracle SQL Developer

Introduction

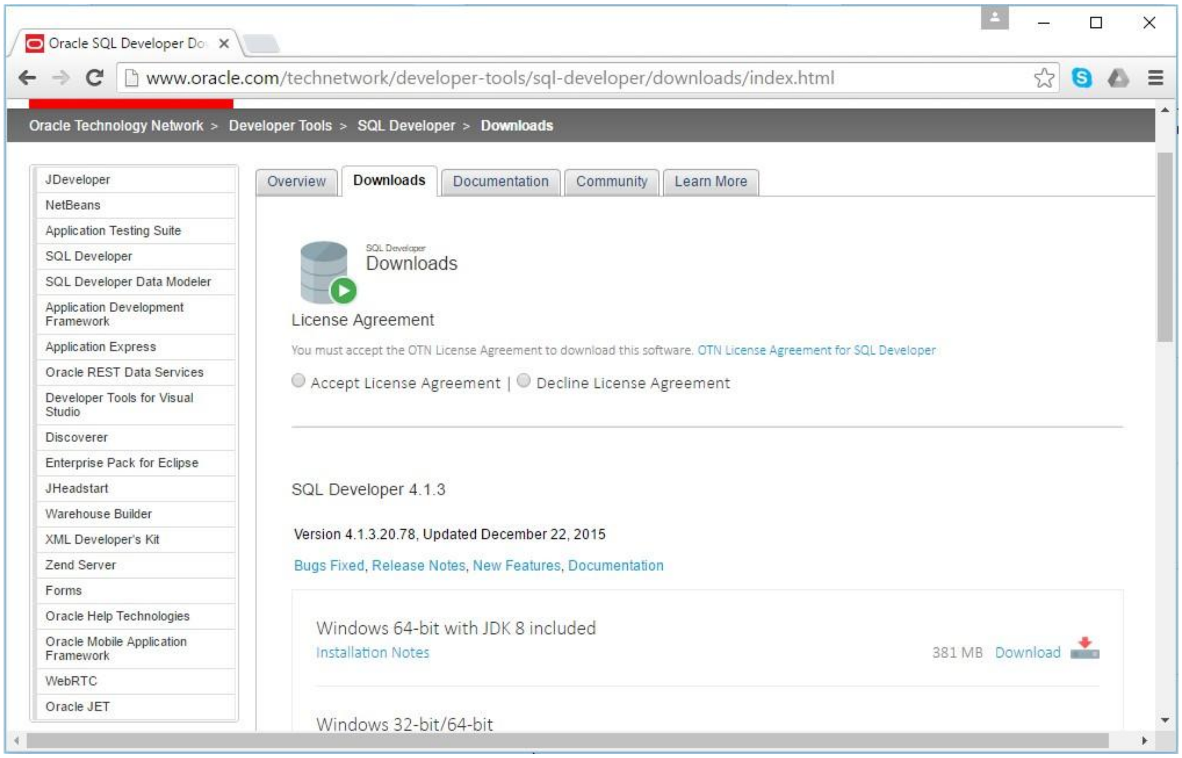
With the database installed, you now need to download and use a client that is capable of connecting to the database. Modern clients are graphical and have many features to enhance productivity. If you already have a working copy of a SQL client you prefer, you do not need to complete this section. Simply setup a connection to your installed database and continue with the next section. Otherwise, it is recommended that you install the free Oracle SQL Developer client. This section contains step-by-step instructions for downloading and using Oracle SQL Developer.

Download Oracle SQL Developer

On Oracle.com, click on the Downloads/SQL Developer link:


Oracle | Integrated Cloud
www.oracle.com/index.html
ORACLE
Sign In/Register Help Country Communities I am a... I want to...
Search
Products Solutions Downloads Store Support Training Partners About OTN
Popular Downloads
Java for Developers
Java for Your Computer
Java SE
Solaris
MySQL
Oracle Fusion Middleware 11g
Oracle Database
Prebuilt Developer VMs
Database
Oracle Database
Oracle Database 11g Express Edition
MySQL
Oracle Berkeley DB
Oracle Instant Client
Oracle Application Express
See All...
Middleware
Oracle Fusion Middleware (including Oracle WebLogic Server)
Oracle JRockit
Oracle SOA Suite
See All...
Enterprise Management
Oracle Enterprise Manager
Oracle Application Testing Suite
See All...
Server and Storage Systems
Solaris
Linux and Oracle VM
Firmware
See All...
Developer Tools
SQL Developer
JDeveloper and ADF
Developer Tools for Visual Studio
Enterprise Pack for Eclipse
NetBeans IDE
See All...
Prebuilt Developer VMs
Applications
Oracle Fusion Applications, Oracle E-Business Suite, PeopleSoft, JD Edwards, Siebel CRM
Agile
AutoVue
See All...
Oracle ERP Cloud empowers business ❯
→ Try Oracle ERP Cloud
→ Oracle EPM Solutions
→ CIMA Digital Finance Research Report
www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html

This will take you to a page of Oracle SQL Developer downloads for different platforms and different Java configurations.


Oracle Technology Network > Developer Tools > SQL Developer > Downloads

Overview | Downloads | Documentation | Community | Learn More

SQL Developer Downloads

License Agreement

You must accept the OTN License Agreement to download this software. OTN License Agreement for SQL Developer

○ Accept License Agreement | ○ Decline License Agreement

SQL Developer 4.1.3

Version 4.1.3.20.78, Updated December 22, 2015

Bugs Fixed, Release Notes, New Features, Documentation

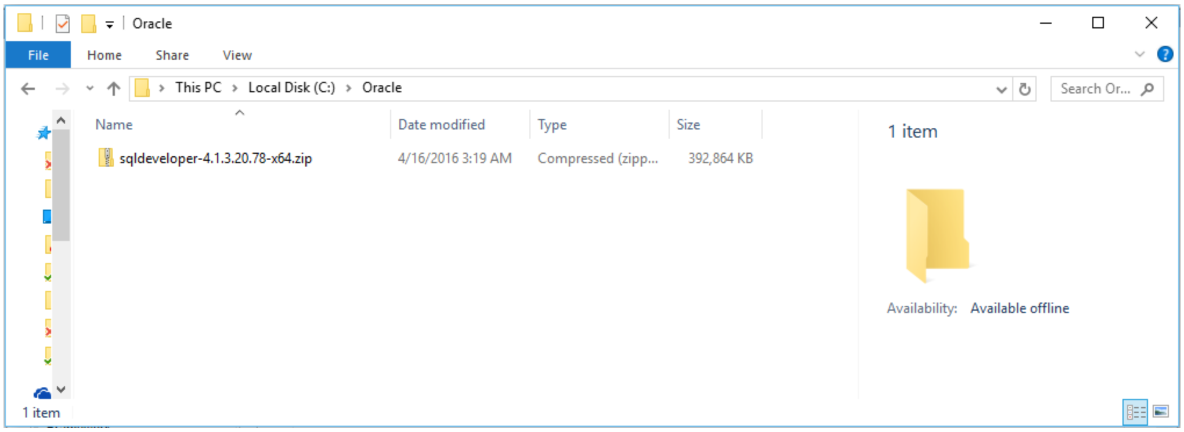
Windows 64-bit with JDK 8 included
Installation Notes — 381 MB Download

Windows 32-bit/64-bit

Oracle SQL Developer Do
www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html
JDeveloper
NetBeans
Application Testing Suite
SQL Developer
SQL Developer Data Modeler
Application Development Framework
Application Express
Oracle REST Data Services
Developer Tools for Visual Studio
Discoverer
Enterprise Pack for Eclipse
JHeadstart
Warehouse Builder
XML Developer's Kit
Zend Server
Forms
Oracle Help Technologies
Oracle Mobile Application Framework
WebRTC
Oracle JET

Before downloading, you must first accept the license agreement. Then click on "Download" link for the "Windows 64-bit with JDK 8 included" option. This option includesJava and allows you to run the client by simply downloading and unzipping it, without a more complex installation process. However, if you have Windows 32-bit, you will need touse the alternative "Windows 32-bit/64-bit" option, and ensure that you also have Java 8 installed.

Since you have already logged into the OTN to download Oracle, clicking on the link here should immediately begin the file download. The file name will start with "sqldeveloper",and then be followed by the version that is being downloaded. Save the file into the C:\Oracle directory you used to download the database, or into another directory of yourchoice that is easy for you to remember.


This PC > Local Disk (C:) > Oracle

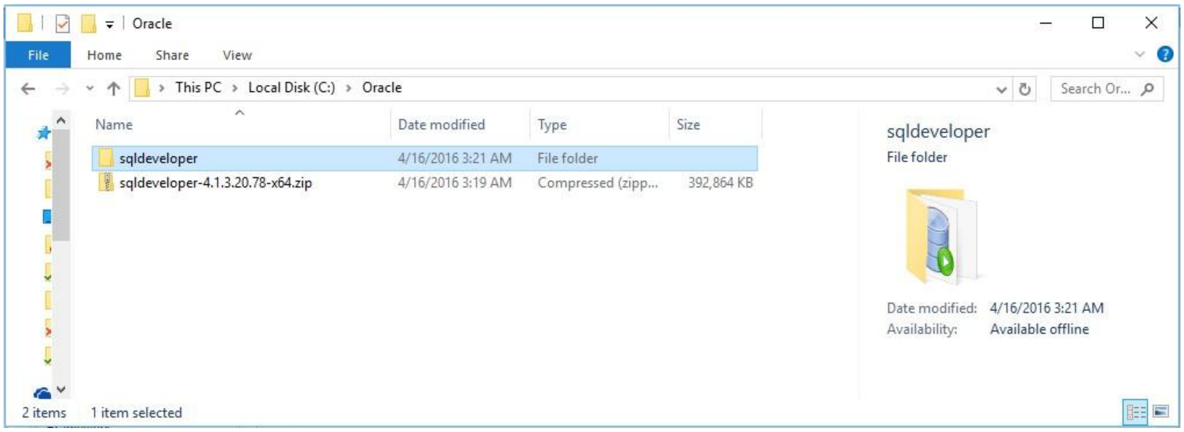
Name	Date modified	Type	Size
sqldeveloper-4.1.3.20.78-x64.zip	4/16/2016 3:19 AM	Compressed (zipp...	392,864 KB

1 item

Availability: Available offline

Unzip SQL Developer

Use the zipping program of your choice to unzip the file. There are many such tools, but ifyou do not have one already, you can download 7-zip at http://www.7-zip.org/download.html. Once the file is unzipped, you will see a directory named "sqldeveloper". It will look something like this:



Name	Date modified	Type	Size
sqldeveloper	4/16/2016 3:21 AM	File folder	
sqldeveloper-4.1.3.20.78-x64.zip	4/16/2016 3:19 AM	Compressed (zipp...	392,864 KB

sqldeveloper
File folder

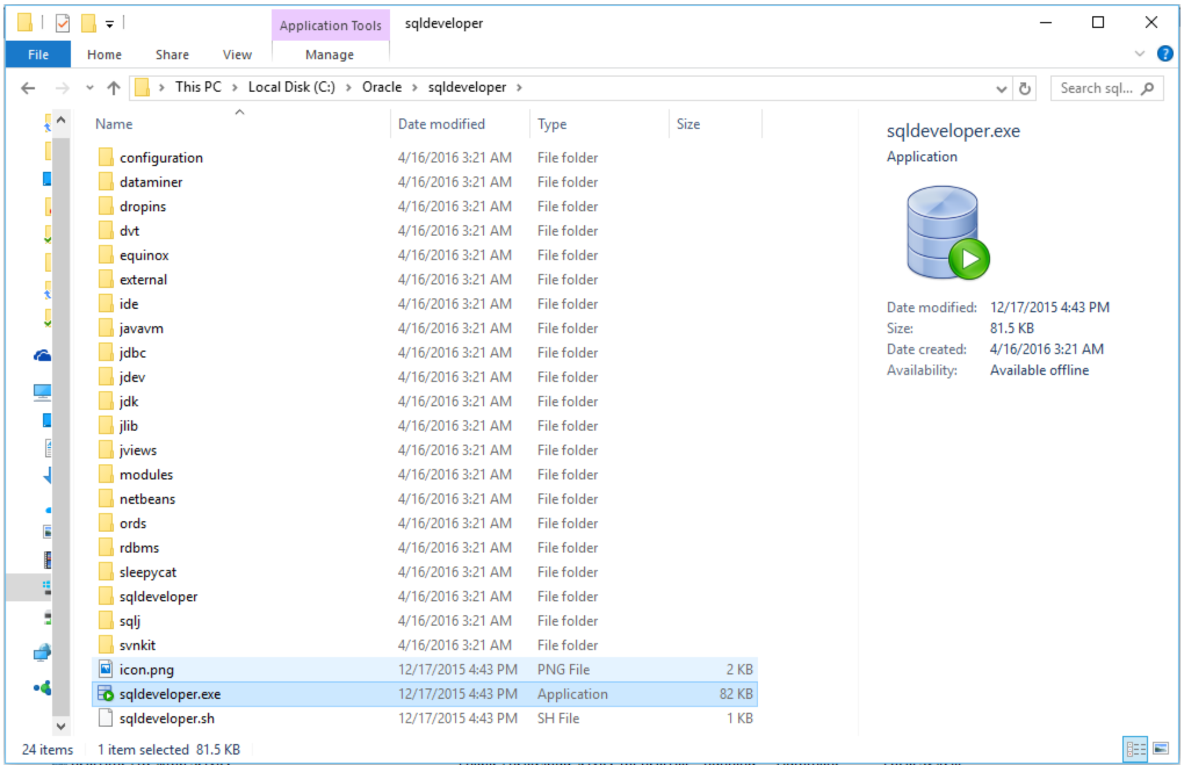
Date modified: 4/16/2016 3:21 AM
Availability: Available offline

2 items 1 item selected

Step 2: Launching Oracle SQL Developer

Launch Oracle SQL Developer

To launch Oracle SQL Developer, simply navigate into the newly created "sqldeveloper"directory, and execute "sqldeveloper". This standalone client does not require an installation before being used.



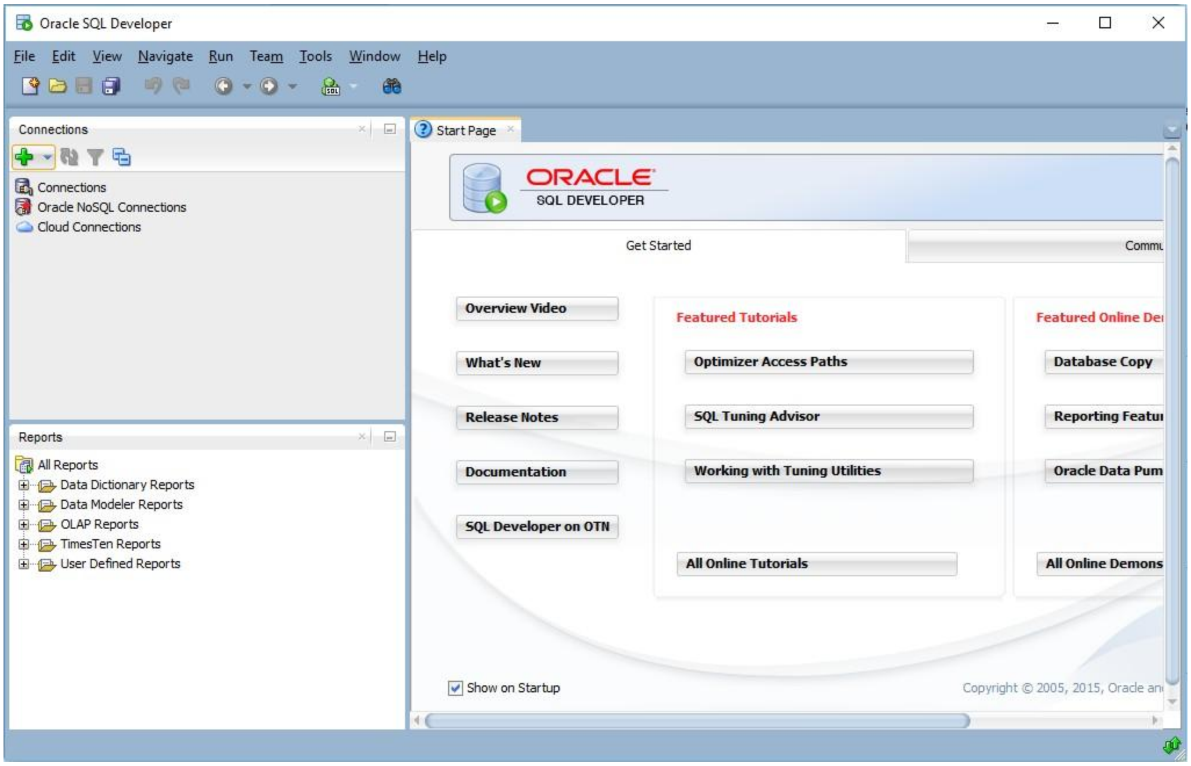
Name	Date modified	Type	Size
configuration	4/16/2016 3:21 AM	File folder	
dataminer	4/16/2016 3:21 AM	File folder	
dropins	4/16/2016 3:21 AM	File folder	
dvt	4/16/2016 3:21 AM	File folder	
equinox	4/16/2016 3:21 AM	File folder	
external	4/16/2016 3:21 AM	File folder	
ide	4/16/2016 3:21 AM	File folder	
javavm	4/16/2016 3:21 AM	File folder	
jdbc	4/16/2016 3:21 AM	File folder	
jdev	4/16/2016 3:21 AM	File folder	
jdk	4/16/2016 3:21 AM	File folder	
jlib	4/16/2016 3:21 AM	File folder	
jviews	4/16/2016 3:21 AM	File folder	
modules	4/16/2016 3:21 AM	File folder	
netbeans	4/16/2016 3:21 AM	File folder	
ords	4/16/2016 3:21 AM	File folder	
rdbms	4/16/2016 3:21 AM	File folder	
sleepycat	4/16/2016 3:21 AM	File folder	
sqldeveloper	4/16/2016 3:21 AM	File folder	
sqlj	4/16/2016 3:21 AM	File folder	
svnkit	4/16/2016 3:21 AM	File folder	
icon.png	12/17/2015 4:43 PM	PNG File	2 KB
sqldeveloper.exe	12/17/2015 4:43 PM	Application	82 KB
sqldeveloper.sh	12/17/2015 4:43 PM	SH File	1 KB

sqldeveloper.exe
Application

Date modified: 12/17/2015 4:43 PM
Size: 81.5 KB
Date created: 4/16/2016 3:21 AM
Availability: Available offline

24 items 1 item selected 81.5 KB

Simply double-click “sqldeveloper” to launch the program. You can return to this directoryeach time you wish to launch Oracle SQL Developer, or you can create a shortcut and place it on your desktop.



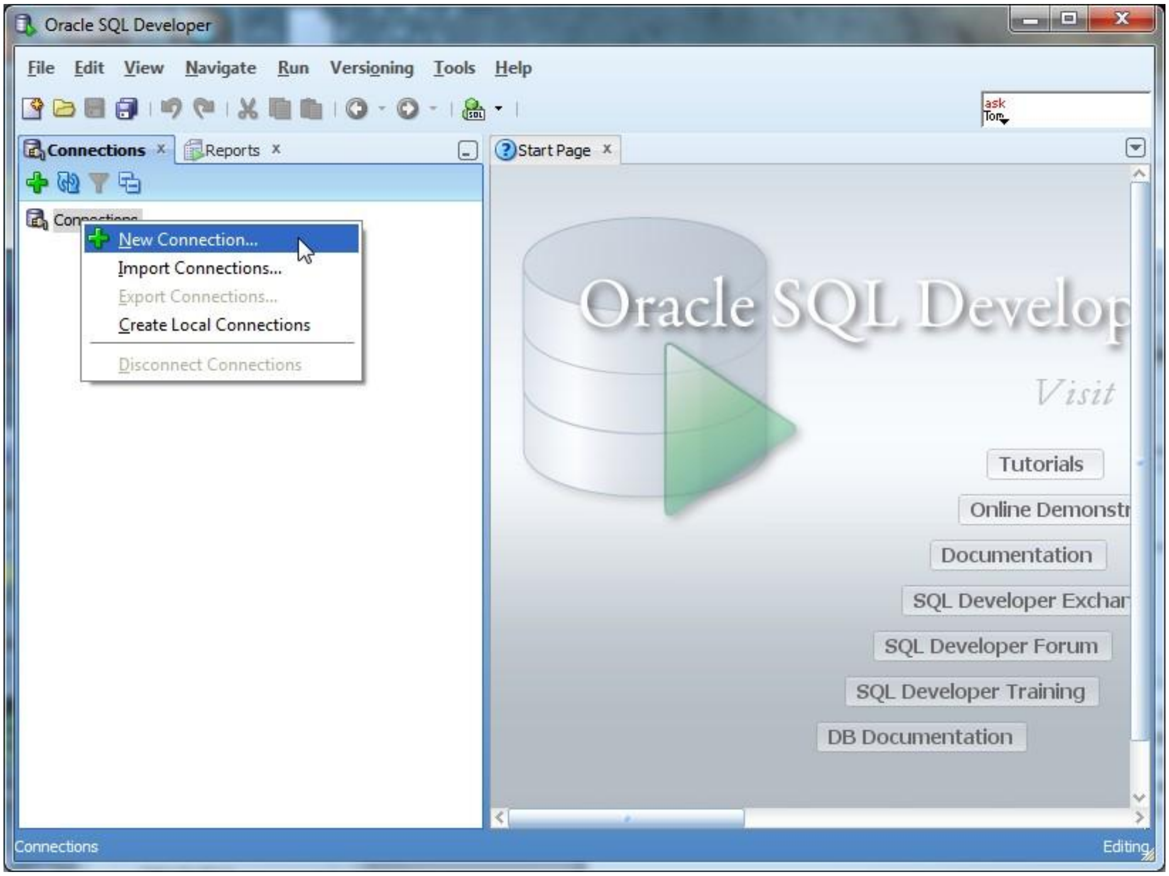
Step 3: Connecting to Your Database

Introduction

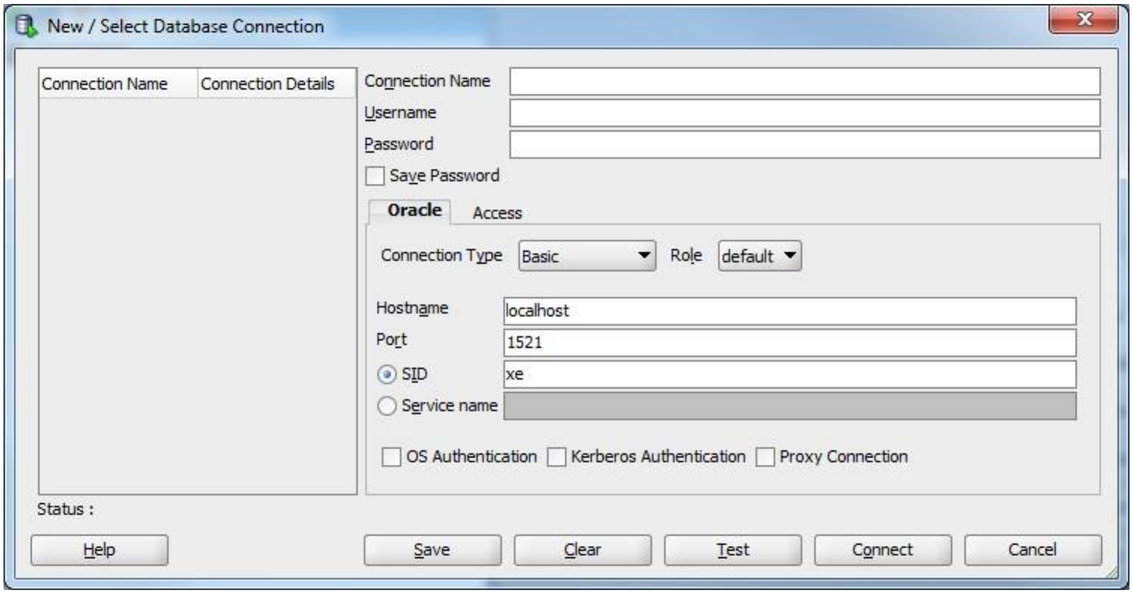
In any SQL client, we must first establish a connection to the database before typing our SQL commands. A connection opens a communication pathway between the client and the database. In Oracle SQL Developer, the term “connection” is overloaded to indicate both inactive and active communication pathways, as well as all of the configuration information necessary to connect again and again to the same database.

Connecting as the SystemUser

We first need to connect to the database as the "system" user, so that we can create ourown user to complete the assignments. To start, right click on the Connections category, then select "New Connection…" from the context menu.



Oracle SQL Developer will launch the New Connection dialog, which requests configuration information about this new connection.



New / Select Database Connection

Connection Name	Connection Details

Connection Name:

Username:

Password:

□ Save Password

Oracle | Access

Connection Type: Basic — Role: default

Hostname: localhost

Port: 1521

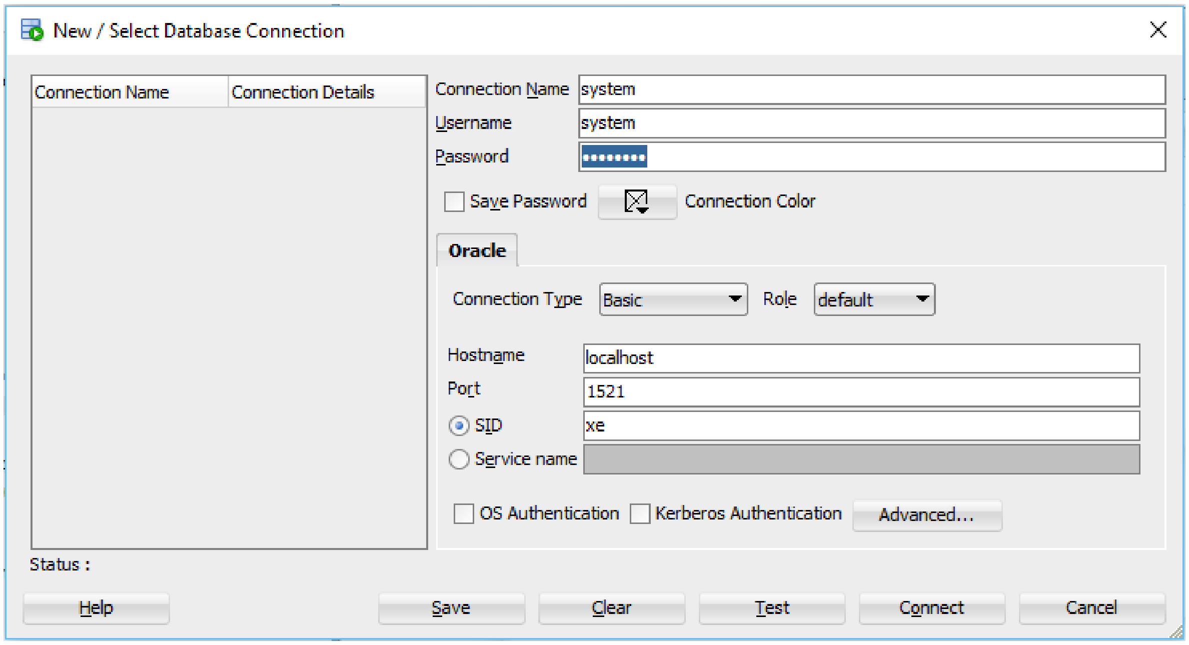
● SID: xe

○ Service name:

□ OS Authentication □ Kerberos Authentication □ Proxy Connection

Status :

You will need to modify the following fields – Connection Name, Username, and Password. The Connection Name is just an identifier that you are giving your connection, so that you may identify it in the future. You can type anything you want that is memorable. We will use the name "system" because this connection is for the system user. The username is "system". The password is whatever you chose your password to be during the Oracle installation process.



New / Select Database Connection

Connection Name	Connection Details

Connection Name: system

Username: system

Password: ••••••••

□ Save Password — Connection Color

Oracle

Connection Type: Basic — Role: default

Hostname: localhost

Port: 1521

● SID: xe

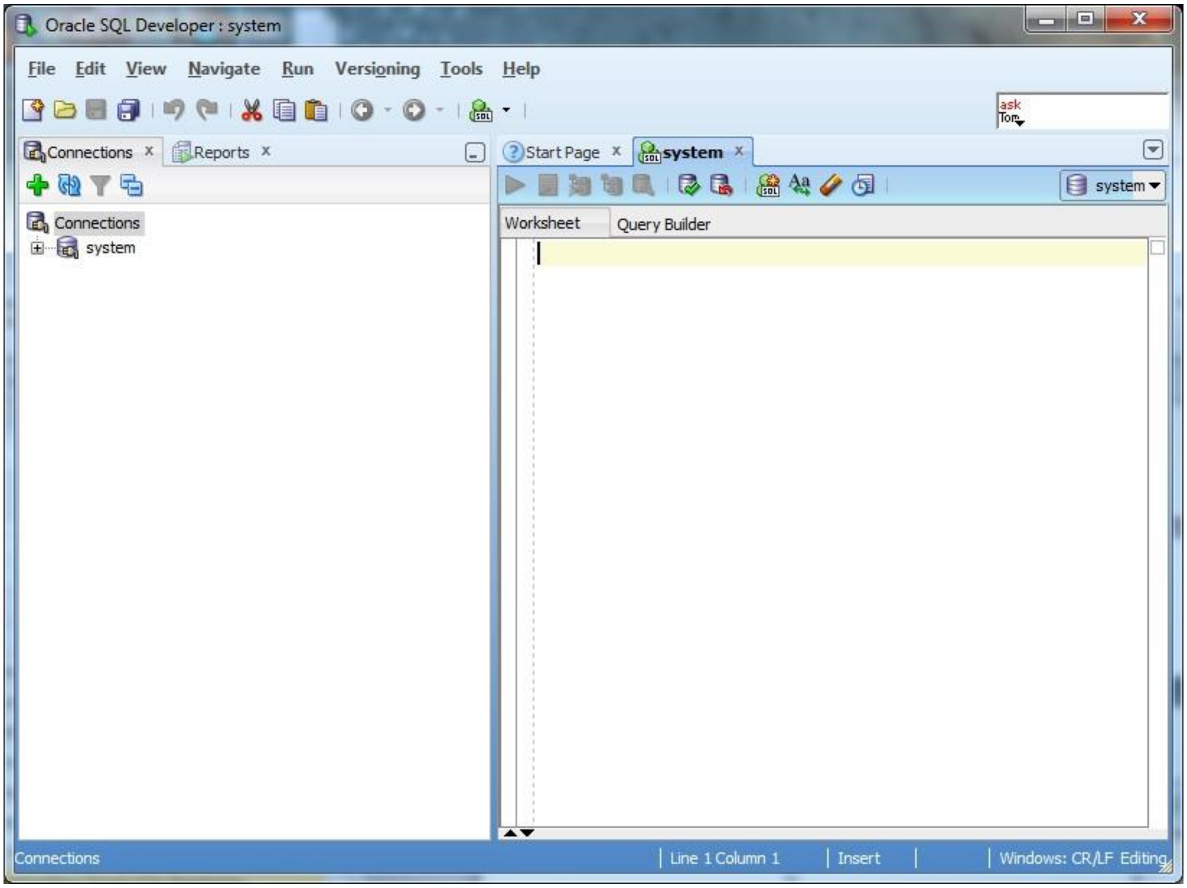
○ Service name:

□ OS Authentication □ Kerberos Authentication

Status :

Click the Test button to ensure that you are able to connect. If you are able to connect, it will state “Status: Success” in the lower left-hand part of the dialog. If there is an issue, it will display the issue in the place of the success status. You will need to fix any issue before proceeding to connect.

Now simply click the Connect button to establish the connection. When you do so,Oracle SQL Developer will open up a blank SQL worksheet.



In this SQL worksheet, you can type commands, execute them, and see the results.

Congratulations! You have now successfully launched Oracle SQL Developer andestablished a connection to your database.

Oracle Express Edition

1. Introduction to Oracle Express Edition (XE)

- **Overview**: Describe Oracle XE as a free-to-use, entry-level version of Oracle's RDBMS. Mention its suitability for beginners, developers, and small applications.
- **Purpose and Target Audience**: Explain that Oracle XE is intended for students, developers, and small businesses needing database management without the cost.
- **Key Advantages**: Highlight the benefits of using XE, like a full-featured database experience, lower resource requirements, and easy setup.

2. Features of Oracle XE

- **Free to Use**: Emphasize that XE is a no-cost, fully functional version of Oracle's database.
- **Lightweight and Limited Resource Usage**: Suitable for smaller applications due to limitations (e.g., 2GB memory limit, up to 12GB data).
- **Comprehensive Database Features**: Includes SQL, PL/SQL, RESTful services, JSON, and support for Oracle Application Express (APEX).
- **Platform Compatibility**: Works on various platforms, including Windows and Linux.
- **Community and Learning Support**: Access to extensive Oracle documentation and community resources.

3. Limitations and Comparisons to Other Oracle Editions

- **Usage Limits**: Explain the storage (12GB), CPU (up to 2 CPU threads), and RAM (2GB) restrictions.
- **Feature Limitations**: Some advanced features (e.g., clustering, certain performance enhancements) are not available in XE.
- **Comparison with Other Editions**: Briefly compare with Standard, Enterprise, and Cloud Editions to contextualize where XE fits.

4. Technical Features and Capabilities

- **Storage and Performance Management**: Describe its capability for handling moderate-sized datasets.
- **Data Security**: Explain features such as encryption, data masking, and user access control.
- **Data Access and APIs**: Support for RESTful services, allowing external applications to interact with data via web services.
- **Development Tools and Compatibility with APEX**: Oracle XE supports APEX, enabling web-based application development.
- **Backup and Recovery Options**: Describe the options for backup and recovery to protect data.

5. Installing Oracle XE

- **System Requirements**: Detail requirements such as OS compatibility, minimum storage, RAM, and CPU specifications.
- **Downloading Oracle XE**: Guide users to Oracle's official site for downloading.
- **Installation Steps**:
  - **Windows Installation**:
    1. **Download the Installer**: Download the XE installer for Windows.
    2. **Run the Installer**: Walk through installation steps, including accepting license agreements.
    3. **Set up Default Passwords**: Briefly explain setting up default SYS and SYSTEM user passwords.
  - **Linux Installation**:
    1. **Download and Unpack the RPM File**: Step-by-step guide for installing RPM packages on Linux.
    2. **Setup and Configuration**: Walkthrough of setup scripts and database start commands.
- **Testing the Installation**: Demonstrate logging into the XE database and running a sample query.

6. Basic Usage of Oracle XE

- **Connecting to the Database**: Instructions for logging in using SQL*Plus or Oracle SQL Developer.
- **Creating and Managing Tables**: Basic SQL commands for creating tables, inserting data, and querying.
- **Database Management**: Simple tasks such as creating users, assigning roles, and basic security settings.
- **Using APEX for Application Development**: Brief introduction to setting up APEX and creating a sample application.

7. Use Cases for Oracle XE

- **Education and Learning**: Ideal for students and educators learning SQL and database management.
- **Application Prototyping**: Useful for developers to test and prototype applications.
- **Small Business Applications**: Discuss use cases like inventory management, CRM, etc.
- **Personal Projects and Experiments**: Mention how hobbyists and individual developers benefit from XE.

8. Conclusion

- **Summary of Key Points**: Recap the benefits, features, and installation of Oracle XE.
- **Recommendation for Use**: Summarize who should use Oracle XE and its value for specific users and organizations.
- **Future Prospects**: Mention how users can upgrade to higher Oracle editions if needed.

Installing the Oracle SQL Developer Client

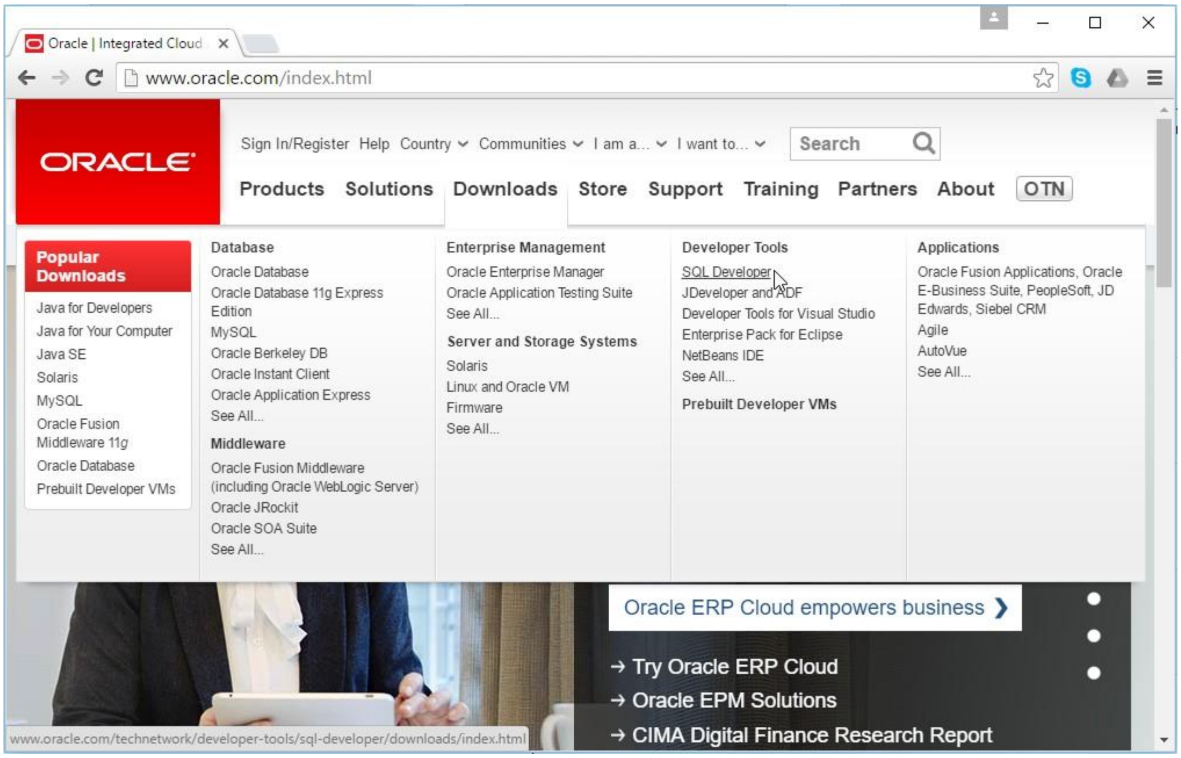
Step 1: Downloading Oracle SQL Developer

Introduction

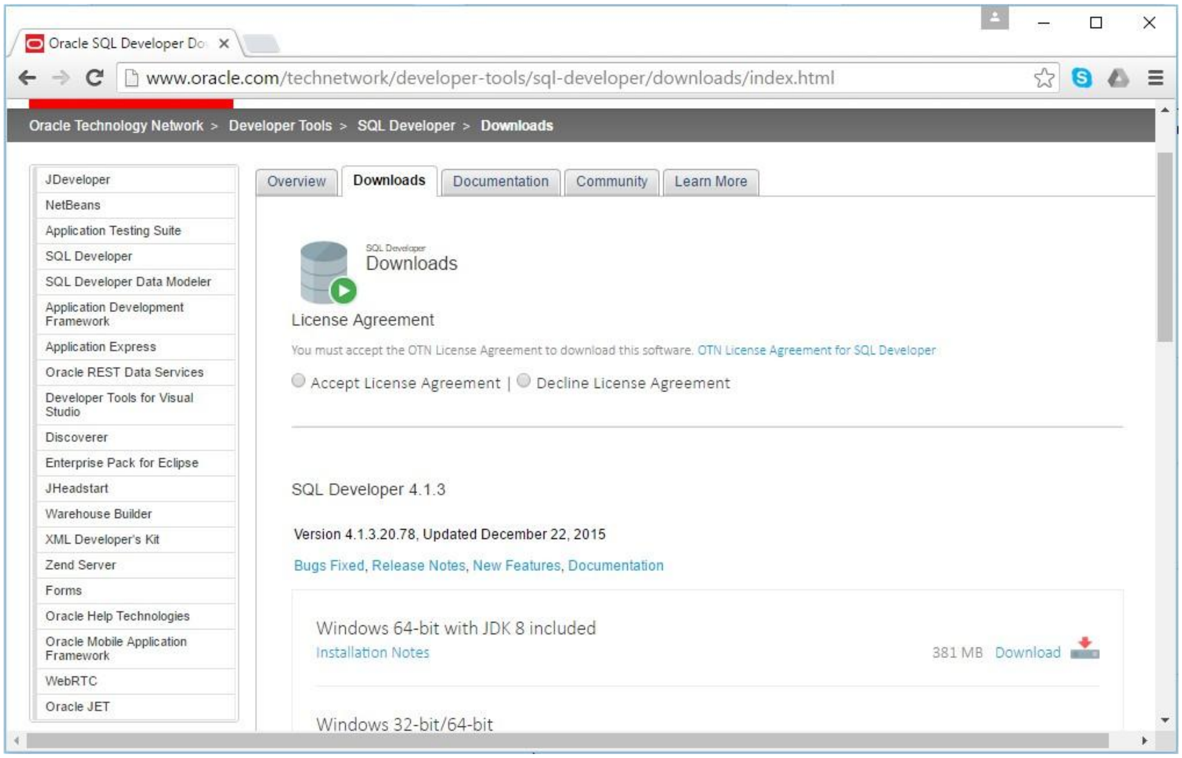
With the database installed, you now need to download and use a client that is capable of connecting to the database. Modern clients are graphical and have many features to enhance productivity. If you already have a working copy of a SQL client you prefer, you do not need to complete this section. Simply setup a connection to your installed database and continue with the next section. Otherwise, it is recommended that you install the free Oracle SQL Developer client. This section contains step-by-step instructions for downloading and using Oracle SQL Developer.

Download Oracle SQL Developer

On Oracle.com, click on the Downloads/SQL Developer link:


Oracle | Integrated Cloud
www.oracle.com/index.html
ORACLE
Sign In/Register Help Country Communities I am a... I want to...
Search
Products Solutions Downloads Store Support Training Partners About OTN
Popular Downloads
Java for Developers
Java for Your Computer
Java SE
Solaris
MySQL
Oracle Fusion Middleware 11g
Oracle Database
Prebuilt Developer VMs
Database
Oracle Database
Oracle Database 11g Express Edition
MySQL
Oracle Berkeley DB
Oracle Instant Client
Oracle Application Express
See All...
Middleware
Oracle Fusion Middleware (including Oracle WebLogic Server)
Oracle JRockit
Oracle SOA Suite
See All...
Enterprise Management
Oracle Enterprise Manager
Oracle Application Testing Suite
See All...
Server and Storage Systems
Solaris
Linux and Oracle VM
Firmware
See All...
Developer Tools
SQL Developer
JDeveloper and ADF
Developer Tools for Visual Studio
Enterprise Pack for Eclipse
NetBeans IDE
See All...
Prebuilt Developer VMs
Applications
Oracle Fusion Applications, Oracle E-Business Suite, PeopleSoft, JD Edwards, Siebel CRM
Agile
AutoVue
See All...
Oracle ERP Cloud empowers business ❯
→ Try Oracle ERP Cloud
→ Oracle EPM Solutions
→ CIMA Digital Finance Research Report
www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html

This will take you to a page of Oracle SQL Developer downloads for different platforms and different Java configurations.


Oracle SQL Developer Do
www.oracle.com/technetwork/developer-tools/sql-developer/downloads/index.html
Oracle Technology Network > Developer Tools > SQL Developer > Downloads
JDeveloper
NetBeans
Application Testing Suite
SQL Developer
SQL Developer Data Modeler
Application Development Framework
Application Express
Oracle REST Data Services
Developer Tools for Visual Studio
Discoverer
Enterprise Pack for Eclipse
JHeadstart
Warehouse Builder
XML Developer's Kit
Zend Server
Forms
Oracle Help Technologies
Oracle Mobile Application Framework
WebRTC
Oracle JET
Overview
Downloads
Documentation
Community
Learn More
SQL Developer
Downloads

License Agreement

You must accept the OTN License Agreement to download this software. OTN License Agreement for SQL Developer

○ Accept License Agreement | ○ Decline License Agreement

SQL Developer 4.1.3

Version 4.1.3.20.78, Updated December 22, 2015

Bugs Fixed, Release Notes, New Features, Documentation

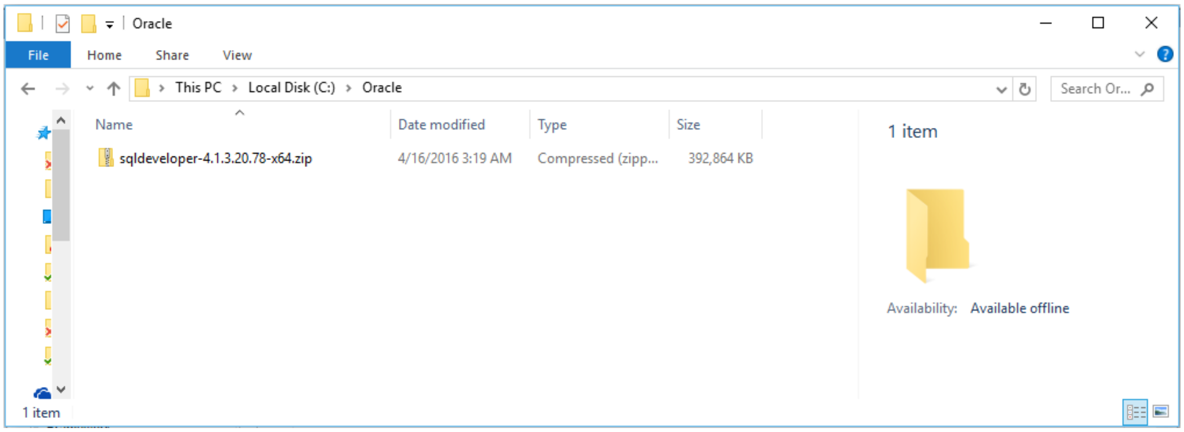
Windows 64-bit with JDK 8 included

Installation Notes 381 MB Download

Windows 32-bit/64-bit

Before downloading, you must first accept the license agreement. Then click on "Download" link for the "Windows 64-bit with JDK 8 included" option. This option includesJava and allows you to run the client by simply downloading and unzipping it, without a more complex installation process. However, if you have Windows 32-bit, you will need touse the alternative "Windows 32-bit/64-bit" option, and ensure that you also have Java 8 installed.

Since you have already logged into the OTN to download Oracle, clicking on the link here should immediately begin the file download. The file name will start with "sqldeveloper",and then be followed by the version that is being downloaded. Save the file into the C:\Oracle directory you used to download the database, or into another directory of yourchoice that is easy for you to remember.



Name	Date modified	Type	Size
sqldeveloper-4.1.3.20.78-x64.zip	4/16/2016 3:19 AM	Compressed (zipp...	392,864 KB

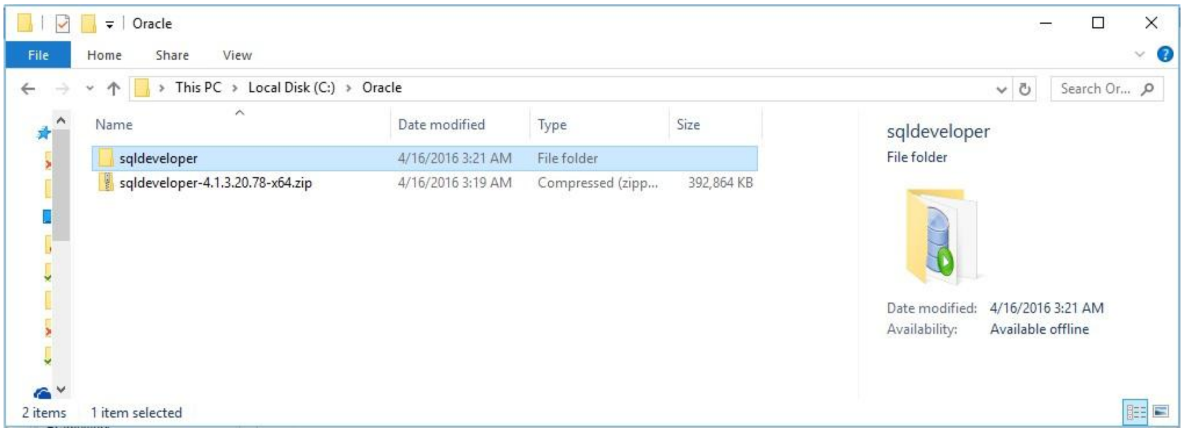
1 item

Availability: Available offline

1 item

Unzip SQL Developer

Use the zipping program of your choice to unzip the file. There are many such tools, but ifyou do not have one already, you can download 7-zip at http://www.7-zip.org/download.html. Once the file is unzipped, you will see a directory named “sqldeveloper”. It will look something like this:



Name	Date modified	Type	Size
sqldeveloper	4/16/2016 3:21 AM	File folder	
sqldeveloper-4.1.3.20.78-x64.zip	4/16/2016 3:19 AM	Compressed (zipp...	392,864 KB

sqldeveloper
File folder

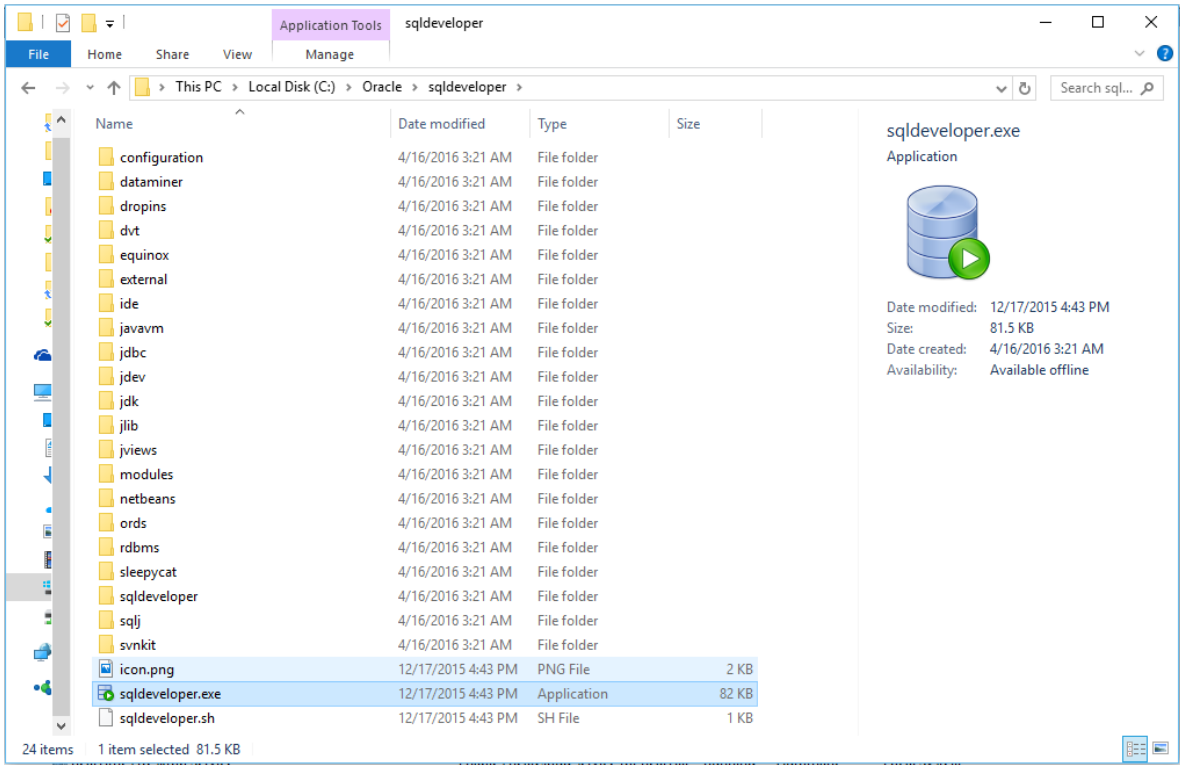
Date modified: 4/16/2016 3:21 AM
Availability: Available offline

2 items 1 item selected

Step 2: Launching Oracle SQL Developer

Launch Oracle SQL Developer

To launch Oracle SQL Developer, simply navigate into the newly created “sqldeveloper”directory, and execute “sqldeveloper”. This standalone client does not require an installation before being used.



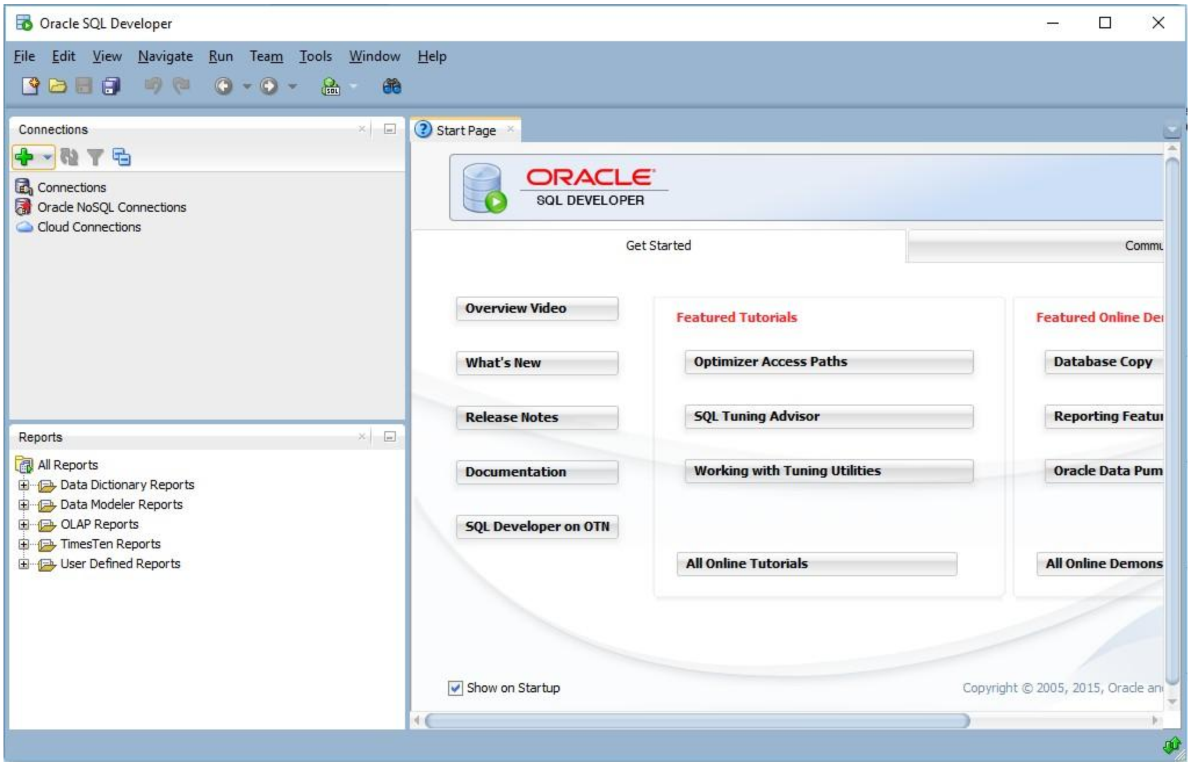
Name	Date modified	Type	Size
configuration	4/16/2016 3:21 AM	File folder	
dataminer	4/16/2016 3:21 AM	File folder	
dropins	4/16/2016 3:21 AM	File folder	
dvt	4/16/2016 3:21 AM	File folder	
equinox	4/16/2016 3:21 AM	File folder	
external	4/16/2016 3:21 AM	File folder	
ide	4/16/2016 3:21 AM	File folder	
javavm	4/16/2016 3:21 AM	File folder	
jdbc	4/16/2016 3:21 AM	File folder	
jdev	4/16/2016 3:21 AM	File folder	
jdk	4/16/2016 3:21 AM	File folder	
jlib	4/16/2016 3:21 AM	File folder	
jviews	4/16/2016 3:21 AM	File folder	
modules	4/16/2016 3:21 AM	File folder	
netbeans	4/16/2016 3:21 AM	File folder	
ords	4/16/2016 3:21 AM	File folder	
rdbms	4/16/2016 3:21 AM	File folder	
sleepycat	4/16/2016 3:21 AM	File folder	
sqldeveloper	4/16/2016 3:21 AM	File folder	
sqlj	4/16/2016 3:21 AM	File folder	
svnkit	4/16/2016 3:21 AM	File folder	
icon.png	12/17/2015 4:43 PM	PNG File	2 KB
sqldeveloper.exe	12/17/2015 4:43 PM	Application	82 KB
sqldeveloper.sh	12/17/2015 4:43 PM	SH File	1 KB

sqldeveloper.exe
Application

Date modified: 12/17/2015 4:43 PM
Size: 81.5 KB
Date created: 4/16/2016 3:21 AM
Availability: Available offline

24 items 1 item selected 81.5 KB

Simply double-click “sqldeveloper” to launch the program. You can return to this directoryeach time you wish to launch Oracle SQL Developer, or you can create a shortcut and place it on your desktop.



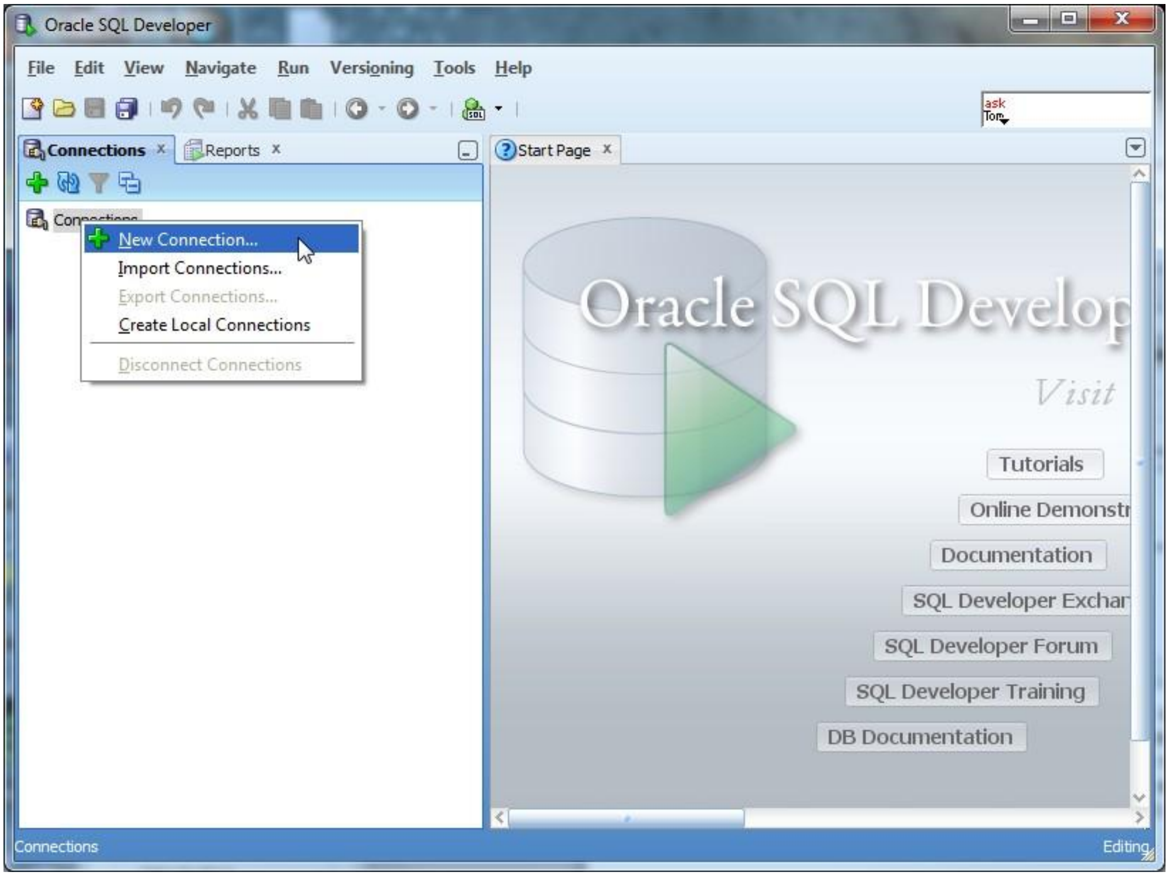
Step 3: Connecting to Your Database

Introduction

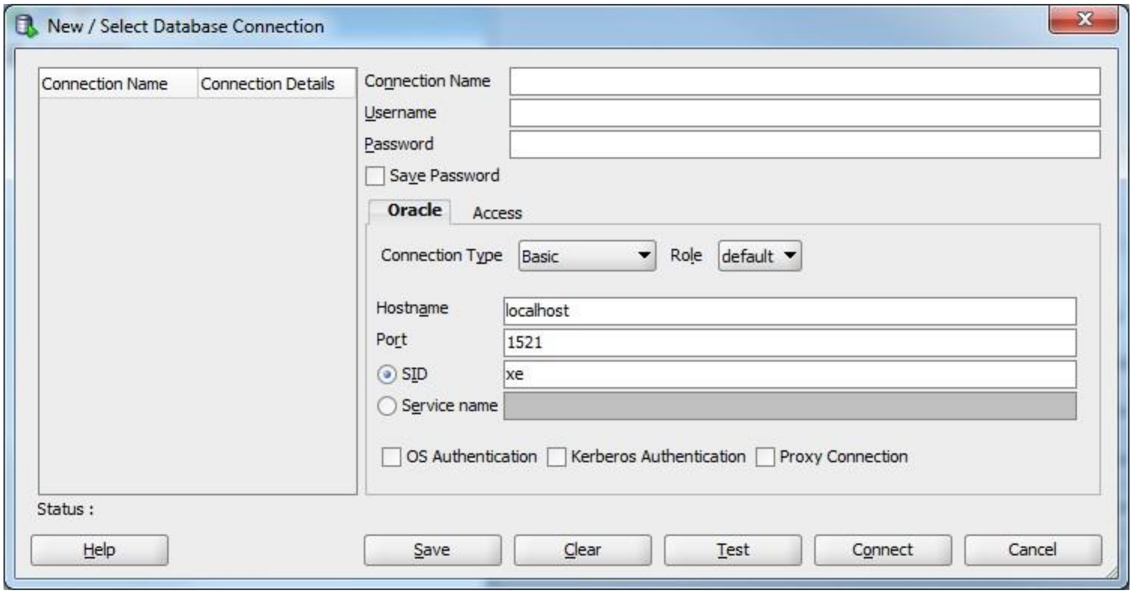
In any SQL client, we must first establish a connection to the database before typing our SQL commands. A connection opens a communication pathway between the client and the database. In Oracle SQL Developer, the term “connection” is overloaded to indicate both inactive and active communication pathways, as well as all of the configuration information necessary to connect again and again to the same database.

Connecting as the SystemUser

We first need to connect to the database as the “system” user, so that we can create ourown user to complete the assignments. To start, right click on the Connections category, then select “New Connection…” from the context menu.



Oracle SQL Developer will launch the New Connection dialog, which requests configuration information about this new connection.


New / Select Database Connection

Connection Name

Username

Password

□ Save Password

Connection Type: Basic

Role: default

Hostname: localhost

Port: 1521

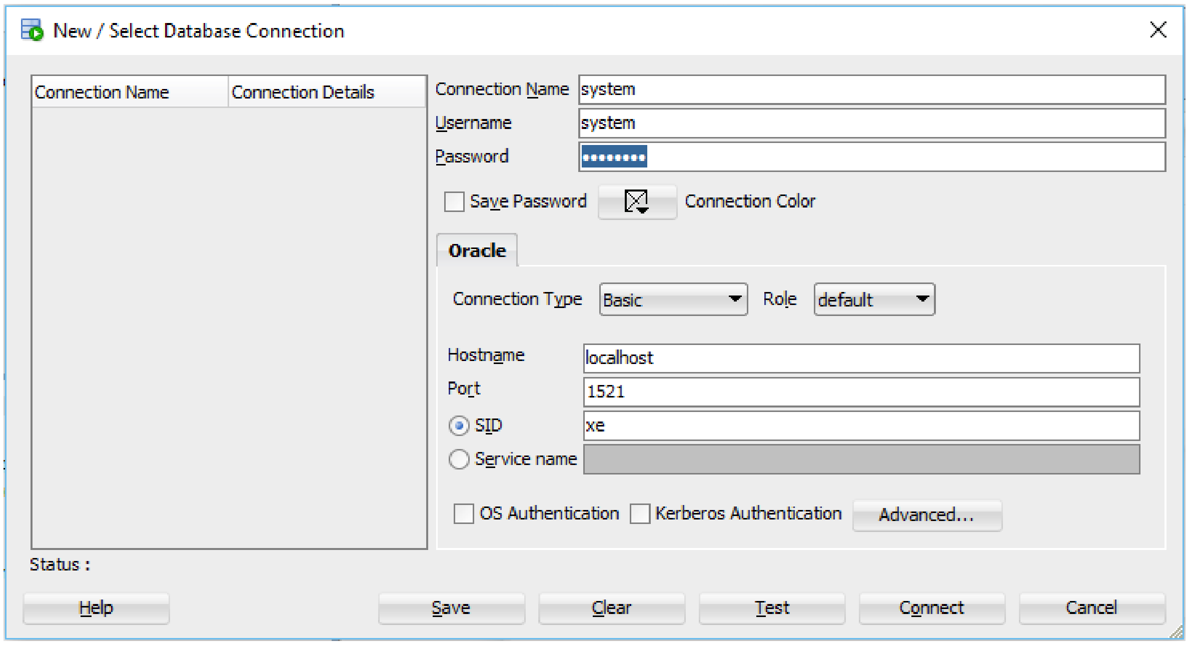
● SID: xe

○ Service name

□ OS Authentication □ Kerberos Authentication □ Proxy Connection

Status :

You will need to modify the following fields – Connection Name, Username, and Password. The Connection Name is just an identifier that you are giving your connection, so that you may identify it in the future. You can type anything you want that is memorable. We will use the name “system” because this connection is for the system user. The username is “system”. The password is whatever you chose your password to be during the Oracle installation process.


New / Select Database Connection

Connection Name: system

Username: system

Password: ••••••••

□ Save Password

Connection Type: Basic

Role: default

Hostname: localhost

Port: 1521

● SID: xe

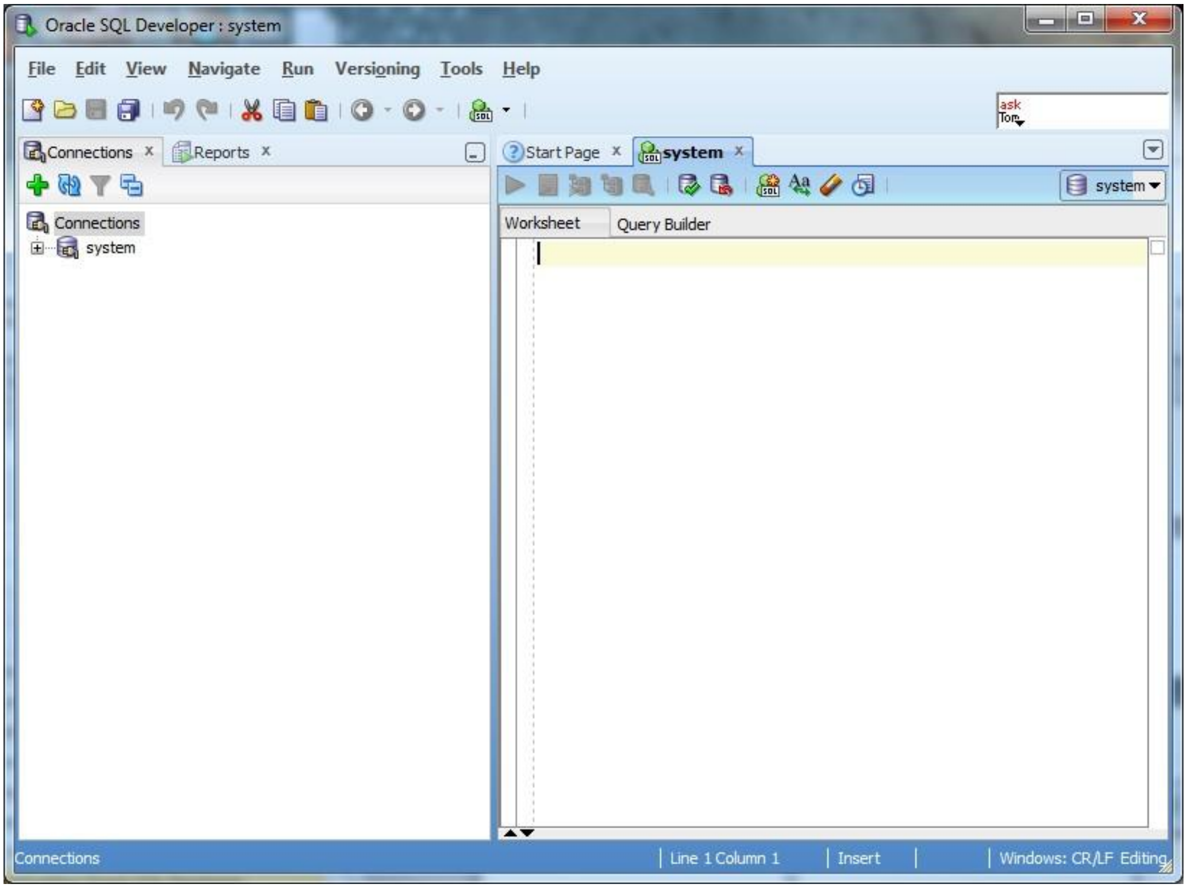
○ Service name

□ OS Authentication □ Kerberos Authentication

Status :

Click the Test button to ensure that you are able to connect. If you are able to connect, it will state “Status: Success” in the lower left-hand part of the dialog. If there is an issue, it will display the issue in the place of the success status. You will need to fix any issue before proceeding to connect.

Now simply click the Connect button to establish the connection. When you do so,Oracle SQL Developer will open up a blank SQL worksheet.



In this SQL worksheet, you can type commands, execute them, and see the results.

Congratulations! You have now successfully launched Oracle SQL Developer andestablished a connection to your database.