

LAB ASSIGNMENT DATABASE

TKM COLLEGE OF ENGINEERING

PROGRAM NO 1:

1. Consider the MOVIE DATABASE

Movies				Actors	
title	director	myear	rating	actor	ayear
Fargo	Coen	1996	8.2	Cage	1964
Raising Arizona	Coen	1987	7.6	Hanks	1956
Spiderman	Raimi	2002	7.4	Maguire	1975
Wonder Boys	Hanson	2000	7.6	McDormand	1957

Acts		Directors	
actor	title	director	dyear
Cage	Raising Arizona	Coen	1954
Maguire	Spiderman	Hanson	1945
Maguire	Wonder Boys	Raimi	1959
McDormand	Fargo		
McDormand	Raising Arizona		
McDormand	Wonder Boys		

Write following relational algebra queries for a given set of relations.

1. Find movies made after 1997
2. Find movies made by Hanson after 1997
3. Find all movies and their ratings
4. Find all actors and directors
5. Find Coen's movies with McDormand

SCRIPT:

```
CREATE DATABASE MOVIE;
```

```
USE MOVIE;
```

```
CREATE TABLE MOVIESS(TITLE VARCHAR(20),DIRECTORS VARCHAR(20),MYEAR INT,RATING  
FLOAT,PRIMARY KEY(TITLE));
```

```
INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('FARGO','COEN',1996,8.2);
```

```
INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('RAISING  
ARIZONA','COEN',1987,7.6);
```

```
INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('SPIDERMAN','RAIMI',2002,7.4);
```

```
INSERT INTO  
MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('WONDERBOYS','HANSON',2000,7.6);
```

```
SELECT * FROM MOVIESS WHERE MYEAR>1997;
```

```
SELECT TITLE FROM MOVIESS WHERE DIRECTORS='HANSON';
```

```
SELECT TITLE,RATING FROM MOVIESS;
```

```
USE MOVIE;
```

```
CREATE TABLE ACTORSS(ACTOR VARCHAR(20),AYEAR INT,PRIMARY KEY(ACTOR));
```

```
INSERT INTO ACTORSS(ACTOR,AYEAR)VALUES('CAGE',1964);
```

```
INSERT INTO ACTORSS(ACTOR,AYEAR)VALUES('HANKS',1956);
```

```
INSERT INTO ACTORSS(ACTOR,AYEAR)VALUES('MAGUIRE',1975);
```

```
INSERT INTO ACTORSS(ACTOR,AYEAR)VALUES('MCDORMANDS',1957);
```

```
SELECT * FROM ACTORSS;
```

```
USE MOVIE;
```

```

CREATE TABLE DIRECTORSS(DIRECTORS VARCHAR(20),DYEAR INT);

INSERT INTO DIRECTORSS(DIRECTORS,DYEAR)VALUES('COEN',1954);

INSERT INTO DIRECTORSS(DIRECTORS,DYEAR)VALUES('HANSON',1945);

INSERT INTO DIRECTORSS(DIRECTORS,DYEAR)VALUES('RAIMI',1959);

SELECT * FROM DIRECTORSS;

USE MOVIE;

```

```

CREATE TABLE ACT(ACTOR VARCHAR(20),TITLE VARCHAR(20),FOREIGN KEY(TITLE)REFERENCES
MOVIESS(TITLE));

INSERT INTO ACT(ACTOR,TITLE)VALUES('CAGE','RAISING ARIZONA');

INSERT INTO ACT(ACTOR,TITLE)VALUES('MAGUIRE','SPIDERMAN');

INSERT INTO ACT(ACTOR,TITLE)VALUES('MAGUIRE','WONDERBOYS');

INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','FARGO');

INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','RAISING ARIZONA');

INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','WONDERBOYS');

SELECT * FROM ACT;

```

```

SELECT MOVIESS.DIRECTORS,ACT.ACTOR FROM MOVIESS INNER JOIN ACT ON
MOVIESS.TITLE=ACT.TITLE;

```

```

SELECT MOVIESS.TITLE FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=ACT.TITLE WHERE
DIRECTORS='COEN' AND ACTOR='MCDORMAND';

```

- 1) SELECT * FROM MOVIESS WHERE MYEAR>1997;
- 2) SELECT TITLE FROM MOVIESS WHERE DIRECTORS='HANSON';
- 3) SELECT TITLE,RATING FROM MOVIESS;
- 4) SELECT MOVIESS.DIRECTORS,ACT.ACTOR FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=ACT.TITLE;
- 5) SELECT MOVIESS.TITLE FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=ACT.TITLE WHERE DIRECTORS='COEN' AND ACTOR='MCDORMAND';

Output:

1)

Result Grid

Filter Rows:

Edit: Export/Import: Wrap Cell Contents:

	TITLE	DIRECTORS	MYEAR	RATING
▶	SPIDERMAN	RAIMI	2002	7.4
	WONDERBOYS	HANSON	2000	7.6
•	MOVIES	MOVIES	MOVIES	MOVIES

MOVIESS 1

Apply Revert Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
✓	4 10:42:41	CREATE TABLE MOVIESS(TITLE VARCHAR(20),DIRECTORS VARCHAR(20),MYEAR INT,RATING FLOAT,...	0 row(s) affected	0.109 sec
✓	5 10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(FARGO','COEN',1996,8.2)	1 row(s) affected	0.016 sec
✓	6 10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(RAISING ARIZONA','COEN',1987,...	1 row(s) affected	0.000 sec
✓	7 10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(SPIDERMAN','RAIMI',2002,7.4)	1 row(s) affected	0.015 sec
✓	8 10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(WONDERBOYS','HANSON',2000,...	1 row(s) affected	0.016 sec
✓	9 10:42:47	SELECT * FROM MOVIESS WHERE MYEAR>1997 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

2)

Result Grid					Filter Rows:					Edit:					Export/Import:					Wrap Cell Content:				
TITLE	DIRECTORS	MYEAR	RATING																					
WONDERBOYS	HANSON	2000	7.6																					
...																					

MOVIESS 2

ApplyRevertContext HelpSnippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
5	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(FARGO','COEN',1996,8.2)	1 row(s) affected	0.016 sec
6	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(RAISING ARIZONA','COEN',1987,...	1 row(s) affected	0.000 sec
7	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(SPIDERMAN','RAIMI',2002,7.4)	1 row(s) affected	0.015 sec
8	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)/VALUES(WONDERBOYS','HANSON',2000,...	1 row(s) affected	0.016 sec
9	10:42:47	SELECT * FROM MOVIESS WHERE MYEAR>1997 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
10	10:43:58	SELECT TITLE FROM MOVIESS WHERE DIRECTORS='HANSON' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

3)

Result Grid		Filter Rows:	Edit:	Export/Import:	Wrap Cell Content:
TITLE	RATING				
FARGO	8.2				
RAISING ARIZONA	7.6				
SPIDERMAN	7.4				
WONDERBOYS	7.6				

MOVIESS 3		Apply	Revert
Output			
Action Output			
#	Time	Action	Message
6	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('RAISING ARIZONA','COEN',1987,...	1 row(s) affected
7	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('SPIDERMAN','RAIMI',2002,7.4)	1 row(s) affected
8	10:42:41	INSERT INTO MOVIESS(TITLE,DIRECTORS,MYEAR,RATING)VALUES('WONDERBOYS','HANSON',2000,...	1 row(s) affected
9	10:42:47	SELECT * FROM MOVIESS WHERE MYEAR>1997 LIMIT 0, 1000	2 row(s) returned
10	10:43:58	SELECT TITLE FROM MOVIESS WHERE DIRECTORS='HANSON' LIMIT 0, 1000	1 row(s) returned
11	10:44:51	SELECT TITLE,RATING FROM MOVIESS LIMIT 0, 1000	4 row(s) returned

4)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

DIRECTORS	ACTOR
COEN	MCDORMAND
COEN	CAGE
COEN	MCDORMAND
RAIMI	MAGUIRE
HANSON	MAGUIRE

Result 7

Read Only

Output:

Action Output

#	Time	Action	Message
29	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MAGUIRE','WONDERBOYS')	1 row(s) affected
30	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','FARGO')	1 row(s) affected
31	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','RAISING ARIZONA')	1 row(s) affected
32	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','WONDERBOYS')	1 row(s) affected
33	10:45:54	SELECT * FROM ACT LIMIT 0, 1000	6 row(s) returned
34	10:46:04	SELECT MOVIESS.DIRECTORS,ACT.ACTOR FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=AC...	6 row(s) returned

5)

Result Grid | Filter Rows: | Export: | Wrap Cell Content: |

TITLE
FARGO
RAISING ARIZONA

Result 8 x Read Only

Output

Action Output

#	Time	Action	Message
✓ 30	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','FARGO')	1 row(s) affected
✓ 31	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','RAISING ARIZONA')	1 row(s) affected
✓ 32	10:45:54	INSERT INTO ACT(ACTOR,TITLE)VALUES('MCDORMAND','WONDERBOYS')	1 row(s) affected
✓ 33	10:45:54	SELECT * FROM ACT LIMIT 0, 1000	6 row(s) returned
✓ 34	10:46:04	SELECT MOVIESS.DIRECTORS,ACT.ACTOR FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=AC...	6 row(s) returned
✓ 35	10:46:58	SELECT MOVIESS.TITLE FROM MOVIESS INNER JOIN ACT ON MOVIESS.TITLE=ACT.TITLE WHERE DI...	2 row(s) returned

RESULT: Output obtained successfully.

PROGRAM NO 2:

Consider Dept table

<u>DEPTNO</u>	DNAME	LOC
---------------	-------	-----

Perform the following:

1. Rename the table dept as department
2. Add a new column PINCODE with not null constraints to the existing table DEPT
3. All constraints and views that reference the column are dropped automatically, along with the column.
4. Rename the column DNAME to DEPT_NAME in dept table
5. Change the data type of column loc as CHAR with size 10
6. Delete table

SCRIPT:

```
CREATE DATABASE DEPARTMENTS;
```

```
USE DEPARTMENTS;
```

```
CREATE TABLE DEPT(DEPTNO VARCHAR(10) NOT NULL,
```

```
DNAME VARCHAR(20) NOT NULL,
```

```
LOC VARCHAR(20) NOT NULL);
```

```
SELECT * FROM DEPT;
```

```
SELECT * FROM DEPARTMENT;
```

```
RENAME TABLE DEPT TO DEPARTMENT;
```

```
ALTER TABLE DEPT ADD COLUMN PINCODE VARCHAR(2) NOT NULL;
```

```
ALTER TABLE DEPARTMENT DROP COLUMN PINCODE;
```

```
SELECT * FROM DEPARTMENT;
```

```
ALTER TABLE DEPARTMENT CHANGE DNAME DEPT_NAME varchar(30);
```

```
ALTER TABLE DEPARTMENT MODIFY COLUMN LOC char(10);
```

```
SELECT * FROM DEPARTMENT;
```

```
USE DATABASENAME;
```

```
SHOW TABLES;
```

```
DROP TABLE department;
```

```
USE DATABASENAME;
```

```
SHOW TABLES;
```

- 1) RENAME TABLE DEPT TO DEPARTMENT
- 2) ALTER TABLE DEPT ADD COLUMN PINCODE VARCHAR(2) NOT NULL;
- 3) ALTER TABLE DEPT ADD COLUMN PINCODE VARCHAR(2) NOT NULL;
- 4) ALTER TABLE DEPARTMENT CHANGE DNAME DEPT_NAME varchar(30);
- 5) ALTER TABLE DEPARTMENT MODIFY COLUMN LOC char(10);
- 6) DROP TABLE department;

Output:

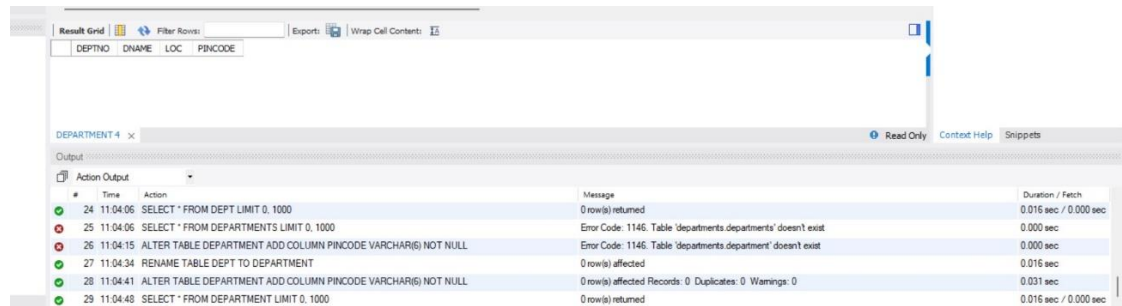
1)



The screenshot shows the 'Output' window of a SQL IDE. It contains a table with columns: #, Time, Action, Message, and Duration / Fetch. The table lists the execution of several SQL statements. The first statement, 'CREATE DATABASE DEPARTMENTS', is successful. The second, 'USE DEPARTMENTS', is also successful. The third, 'CREATE TABLE DEPT(DEPTNO VARCHAR(10) NOT NULL, DNAME VARCHAR(20) NOT NULL, LOC VARCHAR(10) NOT NULL)', is successful. The fourth, 'SELECT * FROM DEPT LIMIT 0, 1000', is successful. The fifth, 'SELECT * FROM DEPARTMENTS LIMIT 0, 1000', results in an error: 'Error Code: 1146. Table 'departments.departments' doesn't exist'. The sixth, 'RENAME TABLE DEPT TO DEPARTMENT', is successful.

#	Time	Action	Message	Duration / Fetch
13	11:02:17	CREATE DATABASE DEPARTMENTS	1 row(s) affected	0.015 sec
14	11:02:17	USE DEPARTMENTS	0 row(s) affected	0.000 sec
15	11:02:25	CREATE TABLE DEPT(DEPTNO VARCHAR(10) NOT NULL, DNAME VARCHAR(20) NOT NULL, LOC VAR...	0 row(s) affected	0.062 sec
16	11:02:25	SELECT * FROM DEPT LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
17	11:02:25	SELECT * FROM DEPARTMENTS LIMIT 0, 1000	Error Code: 1146. Table 'departments.departments' doesn't exist	0.000 sec
18	11:02:32	RENAME TABLE DEPT TO DEPARTMENT	0 row(s) affected	0.016 sec

2)



The screenshot shows the 'Output' window of a SQL IDE. It contains a table with columns: #, Time, Action, Message, and Duration / Fetch. The table lists the execution of several SQL statements. The first, 'SELECT * FROM DEPT LIMIT 0, 1000', is successful. The second, 'SELECT * FROM DEPARTMENTS LIMIT 0, 1000', results in an error: 'Error Code: 1146. Table 'departments.departments' doesn't exist'. The third, 'ALTER TABLE DEPARTMENT ADD COLUMN PINCODE VARCHAR(5) NOT NULL', results in an error: 'Error Code: 1146. Table 'departments.department' doesn't exist'. The fourth, 'RENAME TABLE DEPT TO DEPARTMENT', is successful. The fifth, 'ALTER TABLE DEPARTMENT ADD COLUMN PINCODE VARCHAR(5) NOT NULL', is successful. The sixth, 'SELECT * FROM DEPARTMENT LIMIT 0, 1000', is successful.

#	Time	Action	Message	Duration / Fetch
24	11:04:06	SELECT * FROM DEPT LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec
25	11:04:06	SELECT * FROM DEPARTMENTS LIMIT 0, 1000	Error Code: 1146. Table 'departments.departments' doesn't exist	0.000 sec
26	11:04:15	ALTER TABLE DEPARTMENT ADD COLUMN PINCODE VARCHAR(5) NOT NULL	Error Code: 1146. Table 'departments.department' doesn't exist	0.000 sec
27	11:04:34	RENAME TABLE DEPT TO DEPARTMENT	0 row(s) affected	0.016 sec
28	11:04:41	ALTER TABLE DEPARTMENT ADD COLUMN PINCODE VARCHAR(5) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
29	11:04:48	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned	0.016 sec / 0.000 sec

3)

Result Grid			
DEPTNO	DNAME	LOC	

DEPARTMENT 5			
Output			
Action Output			
#	Time	Action	Message
27	11:04:34	RENAME TABLE DEPT TO DEPARTMENT	0 row(s) affected
28	11:04:41	ALTER TABLE DEPARTMENT ADD COLUMN PINCODE VARCHAR(6) NOT NULL	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
29	11:04:48	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned
30	11:05:52	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE1	Error Code: 1091. Can't DROP 'PINCODE1'; check that column/key exists
31	11:06:06	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
32	11:06:12	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned

4)

Result Grid			
DEPTNO	DNAME	LOC	

DEPARTMENT 6			
Output			
Action Output			
#	Time	Action	Message
30	11:05:52	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE1	Error Code: 1091. Can't DROP 'PINCODE1'; check that column/key exists
31	11:06:06	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
32	11:06:12	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned
33	11:06:53	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE	Error Code: 1091. Can't DROP 'PINCODE'; check that column/key exists
34	11:08:32	ALTER TABLE DEPARTMENT MODIFY COLUMN LOC char(10)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
35	11:08:32	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned

5)

Result Grid			
Tables_in_departments			

Result 7			
Output			
Action Output			
#	Time	Action	Message
33	11:06:53	ALTER TABLE DEPARTMENT DROP COLUMN PINCODE	Error Code: 1091. Can't DROP 'PINCODE'; check that column/key exists
34	11:08:32	ALTER TABLE DEPARTMENT MODIFY COLUMN LOC char(10)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
35	11:08:32	SELECT * FROM DEPARTMENT LIMIT 0, 1000	0 row(s) returned
36	11:09:14	DROP TABLE department	0 row(s) affected
37	11:09:14	USE DATABASENAME	Error Code: 1049. Unknown database 'databaseName'
38	11:09:14	SHOW TABLES	0 row(s) returned

RESULT: Output obtained successfully.

PROGRAM NO 3:

Consider Employee table

EMPNO	EMP_NAME	DEPT	SALARY	DOJ	BRANCH
E101	Amit	Production	45000	12-Mar-00	Bangalore
E102	Amit	HR	70000	03-Jul-02	Bangalore
E103	sunita	Management	120000	11-Jan-01	mysore
E105	sunita	IT	67000	01-Aug-01	mysore
E106	mahesh	Civil	145000	20-Sep-03	Mumbai

Perform the following

1. Display all the fields of employee table
2. Retrieve employee number and their salary
3. Retrieve average salary of all employee
4. Retrieve number of employee
5. Retrieve distinct number of employee
6. Retrieve total salary of employee group by employee name and count similar names
7. Retrieve total salary of employee which is greater than >120000
8. Display name of employee in descending order
9. Display details of employee whose name is AMIT and salary greater than 50000

SCRIPT:

```
CREATE DATABASE EMPLOYEES;  
USE EMPLOYEES;
```

```
CREATE table EMPLOYEE(  
Empno varchar(5) not null,  
Emp_name varchar(20) not null,  
dept varchar(30) not null,  
salary int not null,  
dob date not null,  
branch varchar(30) not null  
);
```

```

insert into
EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E101","Amit","production",45000,"2000-
03-12","Banglore");
insert into
EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E102","Amit","HR",70000,"2002-07-
03","Banglore");
insert into
EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E103","Sunita","Manager",120000,"2001
-01-11","Mysore");
insert into
EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E105","Sunita","IT",67000,"2001-08-
01","Mysore");
insert into
EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E106","Mahesh","Civil",145000,"2003-
09-20","Mumbai");
SELECT * FROM EMPLOYEE;
SELECT EMPNO,SALARY FROM EMPLOYEE;
SELECT AVG(SALARY) FROM EMPLOYEE;
SELECT COUNT(*)FROM EMPLOYEE;
select COUNT( DISTINCT Emp_name) from employee;
SELECT SUM(SALARY), EMP_NAME,count(*) AS occurence from employee group by Emp_name;
SELECT SALARY FROM EMPLOYEE WHERE SALARY>120000 ;
SELECT DISTINCT EMP_NAME FROM EMPLOYEE ORDER BY EMP_NAME DESC;
select * from employee where Emp_name="Amit" and salary>50000;

```

- 1) SELECT * FROM EMPLOYEE;
- 2) SELECT EMPNO,SALARY FROM EMPLOYEE;
- 3) SELECT AVG(SALARY) FROM EMPLOYEE;
- 4) SELECT COUNT(*)FROM EMPLOYEE;
- 5) select COUNT(DISTINCT Emp_name) from employee;
- 6) SELECT SUM(SALARY), EMP_NAME,count(*) AS occurence from employee group by Emp_name;
- 7) SELECT SALARY FROM EMPLOYEE WHERE SALARY>120000 ;
- 8) SELECT DISTINCT EMP_NAME FROM EMPLOYEE ORDER BY EMP_NAME DESC;
- 9) select * from employee where Emp_name="Amit" and salary>50000;

Output:

1)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	Empno	Emp_name	dept	salary	dob	branch
▶	E101	Amit	production	45000	2000-03-12	Banglore
	E102	Amit	HR	70000	2002-07-03	Banglore
	E103	Sunita	Manager	120000	2001-01-11	Mysore
	E105	Sunita	IT	67000	2001-08-01	Mysore
	E106	Maresh	Civil	145000	2003-09-20	Mumbai

EMPLOYEE 2

Read Only

Output

Action Output

#	Time	Action	Message
✓ 54	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E101","Amit","production",45000,"...	1 row(s) affected
✓ 55	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E102","Amit","HR",70000,"2002-0...	1 row(s) affected
✓ 56	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E103","Sunita","Manager",120000,...	1 row(s) affected
✓ 57	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E105","Sunita","IT",67000,"2001-0...	1 row(s) affected
✓ 58	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E106","Maresh","Civil",145000,"2...	1 row(s) affected
✓ 59	11:47:08	SELECT * FROM EMPLOYEE LIMIT 0, 1000	5 row(s) returned

2)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

AVG(SALARY)
89400.0000

Result 3

Read Only

Output

Action Output

#	Time	Action	Message
55	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E102","Amit","HR",70000,"2002-0...	1 row(s) affected
56	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E103","Sunita","Manager",120000,...	1 row(s) affected
57	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E105","Sunita","IT",67000,"2001-0...	1 row(s) affected
58	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E106","Maresh","Civil",145000,"2...	1 row(s) affected
59	11:47:08	SELECT * FROM EMPLOYEE LIMIT 0, 1000	5 row(s) returned
60	11:48:21	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned

3)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

AVG(SALARY)

89400.0000

Result 4

Read Only

Output

Action Output

#	Time	Action	Message
56	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E103","Sunita","Manager",120000,...	1 row(s) affected
57	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E105","Sunita","IT",67000,"2001-0...	1 row(s) affected
58	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E106","Maresh","Civil",145000,"2...	1 row(s) affected
59	11:47:08	SELECT * FROM EMPLOYEE LIMIT 0, 1000	5 row(s) returned
60	11:48:21	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
61	11:49:11	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned

4)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

COUNT(DISTINCT Emp_name)
3

Result 6

Read Only

Output

Action Output

#	Time	Action	Message
58	11:47:08	insert into EMPLOYEE(Empno,Emp_name,dept,salary,dob,branch)values("E106","Mahesh","Civil",145000,"2...	1 row(s) affected
59	11:47:08	SELECT * FROM EMPLOYEE LIMIT 0, 1000	5 row(s) returned
60	11:48:21	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
61	11:49:11	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
62	11:49:57	SELECT COUNT(*)FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
63	11:50:45	select COUNT(DISTINCT Emp_name)from employee LIMIT 0, 1000	1 row(s) returned

5)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

	SUM(SALARY)	EMP_NAME	occurrence
▶	115000	Amit	2
	187000	Sunita	2
	145000	Mahesh	1

Result 7

Read Only

Output

Action Output

#	Time	Action	Message
✓ 59	11:47:08	SELECT * FROM EMPLOYEE LIMIT 0, 1000	5 row(s) returned
✓ 60	11:48:21	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
✓ 61	11:49:11	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
✓ 62	11:49:57	SELECT COUNT(*)FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
✓ 63	11:50:45	select COUNT(DISTINCT Emp_name)from employee LIMIT 0, 1000	1 row(s) returned
✓ 64	11:51:37	SELECT SUM(SALARY), EMP_NAME,count(*) AS occurrence from employee group by Emp_name LIMIT 0, 1...	3 row(s) returned

6)

Result Grid

Filter Rows:

Export:

Wrap Cell Content:

SALARY

145000

EMPLOYEE 8

Read Only

Output

Action Output

#	Time	Action	Message
60	11:48:21	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
61	11:49:11	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
62	11:49:57	SELECT COUNT(*)FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
63	11:50:45	select COUNT(DISTINCT Emp_name)from employee LIMIT 0, 1000	1 row(s) returned
64	11:51:37	SELECT SUM(SALARY), EMP_NAME,count(*) AS occurrence from employee group by Emp_name LIMIT 0, 1...	3 row(s) returned
65	11:52:23	SELECT SALARY FROM EMPLOYEE WHERE SALARY>120000 LIMIT 0, 1000	1 row(s) returned

7)

Result Grid				Filter Rows:	Exports:	Wrap Cell Content:
EMP_NAME						
▶ Sunita						
▶ Mahesh						
▶ Amit						

EMPLOYEE 9	Read Only
------------	-----------

#	Time	Action	Message
61	11:49:11	SELECT AVG(SALARY) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
62	11:49:57	SELECT COUNT(*) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
63	11:50:45	select COUNT(DISTINCT Emp_name) from employee LIMIT 0, 1000	1 row(s) returned
64	11:51:37	SELECT SUM(SALARY), EMP_NAME,count(*) AS occurence from employee group by Emp_name LIMIT 0, 1...	3 row(s) returned
65	11:52:23	SELECT SALARY FROM EMPLOYEE WHERE SALARY>120000 LIMIT 0, 1000	1 row(s) returned
66	11:53:12	SELECT DISTINCT EMP_NAME FROM EMPLOYEE ORDER BY EMP_NAME DESC LIMIT 0, 1000	3 row(s) returned

8)

Result Grid				Filter Rows:	Exports:	Wrap Cell Content:
Empno	Emp_name	dept	salary	dob	branch	
▶ E102	Amit	HR	70000	2002-07-03	Banglore	

employee 10	Read Only
-------------	-----------

#	Time	Action	Message
62	11:49:57	SELECT COUNT(*) FROM EMPLOYEE LIMIT 0, 1000	1 row(s) returned
63	11:50:45	select COUNT(DISTINCT Emp_name) from employee LIMIT 0, 1000	1 row(s) returned
64	11:51:37	SELECT SUM(SALARY), EMP_NAME,count(*) AS occurence from employee group by Emp_name LIMIT 0, 1...	3 row(s) returned
65	11:52:23	SELECT SALARY FROM EMPLOYEE WHERE SALARY>120000 LIMIT 0, 1000	1 row(s) returned
66	11:53:12	SELECT DISTINCT EMP_NAME FROM EMPLOYEE ORDER BY EMP_NAME DESC LIMIT 0, 1000	3 row(s) returned
67	11:53:54	select * from employee where Emp_name="Amit" and salary>50000 LIMIT 0, 1000	1 row(s) returned

9)

Empno	Emp_name	dept	salary	dob	branch
▶ E102	Amit	HR	70000	2002-07-03	Banglore

RESULT: Output obtained successfully.

TCL & DCL commands

TCL CODE:

Script:

```
create database tcl;
```

```
create table dept(deptno varchar(20) not null,dname varchar(20) not null,loc varchar(20) not null,primary key(deptno));
```

```
insert into dept values("d001","finance","kollam");
```

```
insert into dept values("d002","it","ernakulam");
```

```
insert into dept values("d003","management","thrissur");
```

```
set autocommit=0;
```

```
insert into dept values("d004","it","kozhikode");
```

```
savepoint b;
```

```
rollback;
```

```
select *from dept;
```

```
insert into dept values("d005","finance","kozhikode");
```

```
savepoint c;
```

```
insert into dept values("d006","finance","malappuram");
```

```
savepoint d;
```

```
rollback to c;
```

```
commit;
```

OUTPUT:

	deptno	dname	loc
▶	d001	finance	kollam
	d002	it	ernakulam
	d003	management	thrissur
	d005	finance	kozhikode
*	NULL	NULL	NULL

dent 3 v

DCL CODE:

```
CREATE DATABASE employee;

USE employee;

START transaction;

CREATE table employee(empno VARCHAR(20) NOT NULL PRIMARY
KEY,emp_name VARCHAR(20) NOT NULL,dept VARCHAR(20) NOT
NULL,salary INT NOT NULL,dob DATE NOT NULL,branch VARCHAR(20)
NOT NULL);

DESCRIBE employee;

INSERT INTO employee (empno,emp_name,dept,salary,dob,branch)
VALUES ('E101','Amit','Production',45000,'2000-03-12','Bangalore');

INSERT INTO employee (empno,emp_name,dept,salary,dob,branch)
VALUES ('E102','Amit','HR',70000,'2002-07-03','Bangalore');

INSERT INTO employee (empno,emp_name,dept,salary,dob,branch)
VALUES ('E103','sunita','Manager',120000,'2001-01-11','Mysore');

INSERT INTO employee (empno,emp_name,dept,salary,dob,branch)
VALUES ('E104','sunita','IT',67000,'2001-08-01','Mysore');

INSERT INTO employee (empno,emp_name,dept,salary,dob,branch)
VALUES ('E105','mahesh','Civil',145000,'2003-09-20','Mumbai');

SELECT * FROM employee;

delete from employee where empno="E101";
```

```
use
employee;
```

```
GRANT DELETE ON employee TO 'heylo'@'localhost';

REVOKE DELETE ON employee FROM 'heylo'@'localhost';
```


REVOKE DELETE ON *.* FROM 'heylo'@'localhost';

SHOW GRANTS FOR 'heylo'@'localhost';

OUTPUT:

	empno	emp_name	dept	salary	dob	branch
►	E102	Amit	HR	70000	2002-07-03	Bangalore
	E103	sunita	Manager	120000	2001-01-11	Mysore
	E104	sunita	IT	67000	2001-08-01	Mysore
	E105	mahesh	Civil	145000	2003-09-20	Mumbai
*	NULL	NULL	NULL	NULL	NULL	NULL

	Grants for heylo@localhost
►	GRANT USAGE ON *.* TO 'heylo'@'localhost'