**SQL**

CREATE TABLE Courses(

CoursesId INTEGER PRIMARY KEY,

CourseName Text,

TeacherId INTEGER

)

INSERT INTO Courses(CoursesId, CourseName, TeacherId) VALUES(

1, "English", 100

)

INSERT INTO Courses(CoursesId, CourseName, TeacherId) VALUES(

2, "Geography", 101

)

INSERT INTO Courses(CoursesId, CourseName, TeacherId) VALUES(

3, "History", 102

)

INSERT INTO Courses(CoursesId, CourseName, TeacherId) VALUES(

4, "Physics", 103

)

INSERT INTO Courses(CoursesId, CourseName, TeacherId) VALUES(

5, "Chemistry", 104

)

SELECT \* FROM Courses

DROP TABLE Courses

#### Teachers

CREATE TABLE Teacher(

TeacherId INTEGER PRIMARY KEY,

TeacherName Text

)

CREATE TABLE Student(

StudentId INTEGER PRIMARY KEY,

StudentName Text

)

CREATE TABLE StudentCourses(

CourseId INTEGER PRIMARY KEY,

StudentId INTEGER

)

DROP TABLE StudentCourses

INSERT INTO Teacher (TeacherId, TeacherName) VALUES (100, "Vinee")

INSERT INTO Teacher (TeacherId, TeacherName) VALUES (101, "Yash")

INSERT INTO Teacher (TeacherId, TeacherName) VALUES (102, "Aman")

INSERT INTO Teacher (TeacherId, TeacherName) VALUES (103, "Honey")

INSERT INTO Teacher (TeacherId, TeacherName) VALUES (104, "Viman")

INSERT INTO Student (StudentId, StudentName) VALUES (11, "A")

INSERT INTO Student (StudentId, StudentName) VALUES (12, "B")

INSERT INTO Student (StudentId, StudentName) VALUES (13, "C")

INSERT INTO Student (StudentId, StudentName) VALUES (14, "D")

INSERT INTO Student (StudentId, StudentName) VALUES (15, "E")

INSERT INTO Student (StudentId, StudentName) VALUES (16, "F")

INSERT INTO StudentCourses (CourseId, StudentId) VALUES (1, 11)

INSERT INTO StudentCourses (CourseId, StudentId) VALUES (2, 11)

INSERT INTO StudentCourses (CourseId, StudentId) VALUES (3, 11)

INSERT INTO StudentCourses (CourseId, StudentId) VALUES (4, 14)

INSERT INTO StudentCourses (CourseId, StudentId) VALUES (5, 15)

DROP TABLE Student

DROP TABLE StudentCourses

SELECT \* FROM Teacher

SELECT \* FROM Student

SELECT \* FROM StudentCourses

SELECT Student.StudentName,COUNT(\*) FROM Student INNER JOIN StudentCourses ON

Student.StudentId = StudentCourses.StudentId

GROUP BY Student.studentid

SELECT Student.StudentName,COUNT(\*) FROM Student LEFT JOIN StudentCourses ON

Student.StudentId = StudentCourses.StudentId

GROUP BY Student.studentid

SELECT Student.StudentName,COUNT(\*) FROM Student JOIN StudentCourses ON

Student.StudentId = StudentCourses.StudentId

GROUP BY Student.studentid

SELECT Student.StudentName, Student.StudentId , Courses.coursesid, Courses.coursename FROM Student

SELECT Student.StudentName, Student.StudentId , StudentCourses.CourseId FROM Student LEFT JOIN StudentCourses

ON Student.StudentId = StudentCourses.StudentId GROUP BY Student.StudentId

-----------------------------------------------------------------------------------------------------

CREATE TABLE Customer(

CustomerID INTEGER PRIMARY KEY,

CityID INTEGER,

CustomerName TEXT

)

INSERT INTO Customer (CustomerID, CityID,CustomerName) VALUES (1, 1, "Bob Smith")

INSERT INTO Customer (CustomerID, CityID,CustomerName) VALUES (2, 1, "Sally Smith")

INSERT INTO Customer (CustomerID, CityID,CustomerName) VALUES (3, 2, "Tom Smith")

INSERT INTO Customer (CustomerID, CityID,CustomerName) VALUES (4, NULL, "Mary Smith")

SELECT \* FROM Customer

CREATE TABLE City(

CityID INTEGER,

CityName TEXT

)

INSERT INTO City (CityID, CityName) VALUES (1, "Kansad City")

INSERT INTO City (CityID, CityName) VALUES (2, "New York City")

INSERT INTO City (CityID, CityName) VALUES (3, "Houtson")

SELECT \* FROM Customer

SELECT \* FROM City

DROP TABLE City

#######Inner join

SELECT \* from Customer INNER JOIN City on Customer.CityID = City.CityID

Graphical user interface, application

Description automatically generated

SELECT \* from Customer LEFT JOIN City on Customer.CityID = City.CityID

Graphical user interface, application, Teams

Description automatically generated

SELECT \* from City LEFT JOIN Customer on Customer.CityID = City.CityID

Graphical user interface, application

Description automatically generated

CREATE TABLE emp (

empno decimal(4,0) NOT NULL,

ename varchar(10) default NULL,

job varchar(9) default NULL,

mgr decimal(4,0) default NULL,

hiredate date default NULL,

sal decimal(7,2) default NULL,

comm decimal(7,2) default NULL,

deptno decimal(2,0) default NULL

)

CREATE TABLE dept (

deptno decimal(2,0) default NULL,

dname varchar(14) default NULL,

loc varchar(13) default NULL

)

INSERT INTO emp VALUES ('7369','SMITH','CLERK','7902','1980-12-17','800.00',NULL,'20')

INSERT INTO emp VALUES ('7499','ALLEN','SALESMAN','7698','1981-02-20','1600.00','300.00','30');

INSERT INTO emp VALUES ('7521','WARD','SALESMAN','7698','1981-02-22','1250.00','500.00','30');

INSERT INTO emp VALUES ('7566','JONES','MANAGER','7839','1981-04-02','2975.00',NULL,'20');

INSERT INTO emp VALUES ('7654','MARTIN','SALESMAN','7698','1981-09-28','1250.00','1400.00','30');

INSERT INTO emp VALUES ('7698','BLAKE','MANAGER','7839','1981-05-01','2850.00',NULL,'30');

INSERT INTO emp VALUES ('7782','CLARK','MANAGER','7839','1981-06-09','2450.00',NULL,'10');

INSERT INTO emp VALUES ('7788','SCOTT','ANALYST','7566','1982-12-09','3000.00',NULL,'20');

INSERT INTO emp VALUES ('7839','KING','PRESIDENT',NULL,'1981-11-17','5000.00',NULL,'10');

INSERT INTO emp VALUES ('7844','TURNER','SALESMAN','7698','1981-09-08','1500.00','0.00','30');

INSERT INTO emp VALUES ('7876','ADAMS','CLERK','7788','1983-01-12','1100.00',NULL,'20');

INSERT INTO emp VALUES ('7900','JAMES','CLERK','7698','1981-12-03','950.00',NULL,'30');

INSERT INTO emp VALUES ('7902','FORD','ANALYST','7566','1981-12-03','3000.00',NULL,'20');

INSERT INTO emp VALUES ('7934','MILLER','CLERK','7782','1982-01-23','1300.00',NULL,'10');

INSERT INTO dept VALUES ('10','ACCOUNTING','NEW YORK');

INSERT INTO dept VALUES ('20','RESEARCH','DALLAS');

INSERT INTO dept VALUES ('30','SALES','CHICAGO');

INSERT INTO dept VALUES ('40','OPERATIONS','BOSTON');

SELECT \* from emp WHERE deptno = 10

SELECT \* from emp WHERE deptno >= 20

SELECT \* from emp WHERE deptno != 20

SELECT \* from emp WHERE deptno <> 10

1.4

SELECT \* from emp WHERE (deptno = 10

or comm != NULL

or sal <= 2000) and deptno = 20

1.5

select sal as salary , comm as commission from emp

select sal as salary , comm as commission from emp

select sal as salary , comm as commission from emp WHERE salary > 800

1.6

select \* from

(select sal as salary ,

comm as commission

from emp) as innertable

WHERE salary > 1000

1.7

SELECT ename || ' WORKS AS A ' || job as msg

from emp where deptno = 10

1.8

SELECT ename, sal,

case

when sal <= 2000 THEN 'UNDERPAID'

when sal >= 4000 then 'OVERPAID'

else 'OK' END

as status

from emp

1.9

SELECT \* from emp FETCH FIRST 5 ROWS ONLY

1.10

SELECT \* from emp LIMIT 2

1.11

SELECT ename, job from emp LIMIT 5

SELECT ename, job from emp ORDER by random() LIMIT 5

1.12

SELECT \* from emp where comm is NULL

1.13

select ename, job, deptno

from emp

where deptno in (10,20)

select ename, job, deptno

from emp

where deptno in (10,20)

and (ename like '%I%' or job LIKE 'CL%')

Graphical user interface, application

Description automatically generated

2nd Chapter

2.1

SELECT ename, job, sal

from emp

where deptno = 10

order by sal ASC

Graphical user interface, application

Description automatically generated

SELECT ename, job, sal

from emp

where deptno = 10

order by sal DESC

Graphical user interface, application, Teams

Description automatically generated

SELECT ename, job, sal, deptno

from emp

order by 4 ASC

LIMIT 5

A picture containing graphical user interface

Description automatically generated

SELECT job, SUBSTRING( job, LENGTH(job)-1) as lengthOfString FROM emp;

Background pattern

Description automatically generated

SELECT ename, job from emp ORDER by SUBSTRING(job, LENGTH(job)-1)

Background pattern

Description automatically generated

To find if the ID is a primary key or not :

A picture containing funnel chart

Description automatically generated

SELECT CoursesId,CourseName,TeacherId,ROW\_NUMBER() OVER(ORDER BY CoursesId) RowNumber

FROM Courses