***Information Systems and Data Management File***

*COE-317*

*Manish Kumar Gupta*

*260/CO/08*

*COE-I (VI Semester)*

***Assignment 1***

Create Table student

(S No integer PRIMARY KEY, Roll No integer UNIQUE NOT NULL, Name text NOT NULL, Age integer,

DOB date, Department text, Marks integer);

Insert into student values

(1,12,’Amit Verma’,19,’ 1991-02-28’,’COE’,87),

(2,24,’ Yogesh Sharma’,20,’ 1990-02-14’,’MPA’,80),

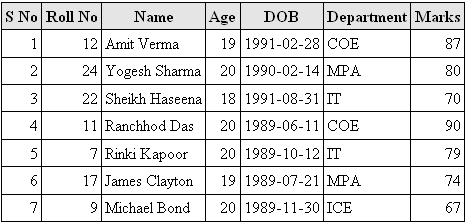
(3,22,’ Sheikh Haseena’,18,’ 1991-08-31’,’IT’,70),

(4,11,’ Ranchhod Das’,20, ‘1989-06-11’,’COE’,90),

(5,7,’ Rinki Kapoor’,20,’ 1989-10-12’,’IT’,79),

(6,17,’ James Clayton’,19,’ 1989-07-21’,’MPA’,74),

(7,9,’ Michael Bond’,20,’ 1989-11-30’,’ICE’,67);



**Queries**

*1. Display names of students by roll no.*

SELECT `Name` FROM `student` WHERE `Roll No` =7

Name: Rinki Kapoor

SELECT `Name` FROM `student` WHERE `Roll No` =24

Name: Yogesh Sharma

SELECT `Name` FROM `student` WHERE `Roll No` =20

MySQL returned an empty result set (i.e. zero rows)

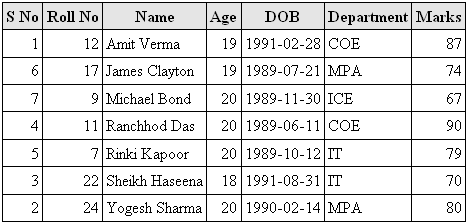
*2. Names of students having age <=20 years.*

SELECT `Name` FROM `student` WHERE `Age`<=20



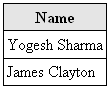
*3. Display table in ascending order.*

SELECT \* FROM `student` ORDER BY `Name` ASC



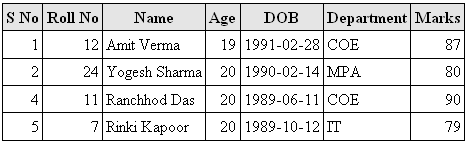
*4. Display names of students having department MPA and marks >=70*

SELECT `Name` FROM `student` WHERE `Department`="MPA" AND `Marks`>=70



*5. Display students with marks greater than average marks.*

SELECT \* FROM student WHERE marks>(SELECT AVG(marks) FROM student)



***Assignment 2***

Create Table student

(SID char(4 ) PRIMARY KEY , Name text, Major text, Credits integer, CHECK (Credits >0));

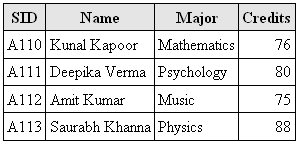
Insert into student values

(‘A110’,’Kunal Kapoor’,’Mathematics’,76),

(‘A111’,’Deepika Verma’,’Psychology’,80),

(‘A112’,’Amit Kumar’,’Music’,75),

(‘A113’,’Saurabh Khanna’,’Physics’,88);



Create Table Faculty

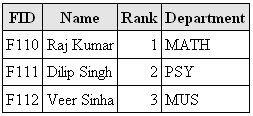
(FID char (4) PRIMARY KEY, Name text NOT NULL, Rank integer(2), Department char(4));

Insert into Faculty values

('F110', 'Raj Kumar', '1', 'MATH'),

('F111', 'Dilip Singh', '2', 'PSY'),

('F112', 'Veer Sinha', '3', 'MUS');



Create Table class

(CourseNo char(7), Schedule text, FID char(4), Room char(4));

Alter Table class Add Foreign Key (FID) References Faculty(FID);

Insert into class values

('MPA304', 'Monday', 'F111', 'R311'),

('PS234', 'Tuesday', 'F110', 'R214'),

(MAT123', 'Friday', 'F112', 'R112');



Create Table enroll

(Course No char(7), SID char(4), Grade char(4));

Alter Table Enroll Add Foreign Key (SID) References student(SID)

Alter Table Enroll Add Foreign Key (CourseNo) References class(CourseNo)

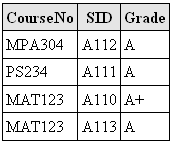
Insert into enroll values

('MPA304', 'A112', 'A'),

('PS234', 'A111', 'A'),

('MAT123', 'A110', 'A+')

('MAT123', 'A113', 'A');



**Queries**

*1. Get name, id and no of credits of Mathematics major.*

SELECT Name,SID,Credits FROM `student` WHERE 'major'="Mathematics"



*2. Get id and name of all students taking course MPA304.*

SELECT student.SID, name FROM student, enroll WHERE enroll.CourseNo=MPA304 AND student.SID = enroll.SID



*3. Find course no, names and major of all students enrolled under faculty F110.*

SELECT student.name, major, class.courseno FROM student, class, enroll WHERE class.FID = "F110" AND class.courseno = enroll.courseno AND student.sid = enroll.sid



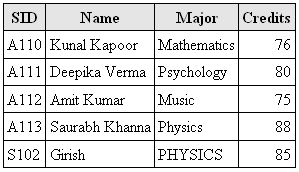
*4. Find student with largest no of credits.*

SELECT \* FROM student WHERE Credits=(SELECT MAX(Credits)FROM student)



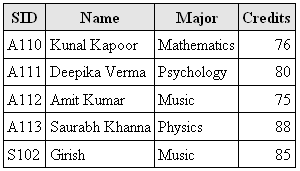
*5. Insert a new student record.*

Insert into student values(‘S102', 'Girish', 'PHYSICS', 85);



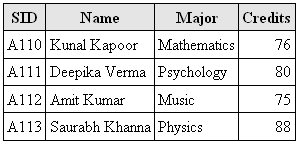
*6. Change Major of student with id S102 to Music*

UPDATE student SET Major="Music" WHERE SID="S102"



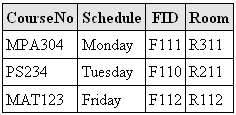
*7. Delete the new student record.*

DELETE FROM student WHERE SID="S102"



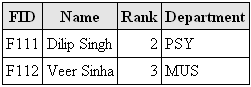
*8. Change room no to R211of all courses taught by Raj Kumar.*

UPDATE class,faculty SET room = 'R211' WHERE faculty.name = "Raj Kumar" AND faculty.fid = class.fid



*9. Delete a faculty record.*

DELETE FROM faculty WHERE faculty.FID = "F110"



***Assignment 3***

Create Table student

(sname char(20) PRIMARY KEY, classno integer, theorymarks integer, drivingmarks integer);

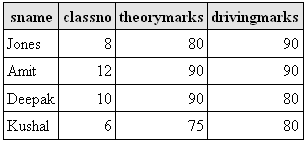
Insert into student values

('Jones',8,80,90),

('Amit',12,90,90),

('Deepak',10,90,80),

('Kushal',6,75,80);



Create Table studentdrivingteacher

(drivingteachername char(20), sname char(20) , FOREIGN KEY (sname) REFERENCES student(sname));

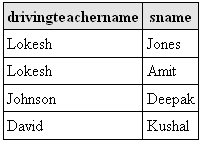
Insert into studentdrivingteacher values

('Lokesh','Jones'),

('Lokesh','Amit'),

('Johnson','Deepak'),

('David','Kushal');



Create Table terachervehicle

(drivingteachername char(20), vehicle char(10), theory char(3), practical char(3),

FOREIGN KEY (drivingteachername) REFERENCES studentdrivingteacher(drivingteachername),

PRIMARY KEY (drivingteachername,vehicle));

Insert into terachervehicle values

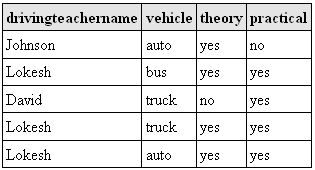
('Johnson','auto','yes','no'),

('Lokesh','bus','yes','yes'),

('David','truck','no','yes'),

('Lokesh','truck','yes','yes'),

('Lokesh','auto','yes','yes');



Create Table vehicle

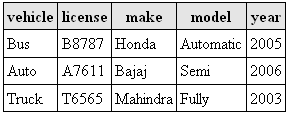
(vehicle char(10), license char(10) PRIMARY KEY, make char(10), model char(10), year integer,  
 FOREIGN KEY (vehicle) REFERENCES teachervehicle(vehicle));

Insert into vehicle values

('Bus','B8787','Honda','Automatic',2005),

('Auto','A7611','Bajaj','Semi',2006),

('Truck','T6565','Mahindra','Fully',2003);



**Queries**

*1. Find teachers that teach theory and give driving lessons on all vehicles.*

SELECT drivingteachername FROM terachervehicle GROUP BY drivingteachername HAVING count(drivingteachername)= (SELECT count(DISTINCT vehicle) FROM vehicle);



*2. Find pair of students having:-*

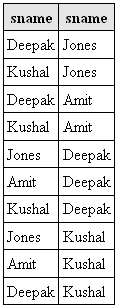
*a) same marks*

SELECT s1.sname FROM student AS s1, student AS s2 WHERE (s1.theorymarks + s1.drivingmarks)=(s2.theorymarks + s2.drivingmarks) AND s1.sname <> s2.sname



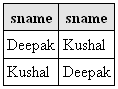
*b) different theory teachers*

SELECT s1.sname,s2.sname FROM studentdrivingteacher AS s1, studentdrivingteacher AS s2  
WHERE s1.drivingteachername <> s2.drivingteachername



*c) same driving marks but different driving teachers*

SELECT s1.sname,student.sname FROM student AS s1, student AS s2, studentdrivingteacher AS s3, studentdrivingteacher AS s4 WHERE s1.drivingmarks=s2.drivingmarks AND s1.sname<>s2.sname AND s3.drivingteachername<>s4.drivingteachername AND s3.sname<>s4.sname AND s2.sname=s4.sname AND s1.sname=s3.sname



*3. List of students taught neither theory nor driving lesson from Johnson.*

SELECT sname FROM studentdrivingteacher WHERE drivingteachername <> “Johnson”



*4. Find list of students having more marks than Jones in theory as well as driving.*

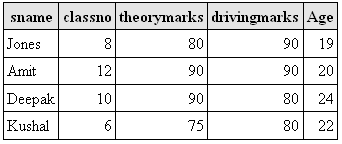
SELECT sname FROM student WHERE sname<>"Jones" AND theorymarks>=(SELECT theorymarks FROM student WHERE sname="Jones") AND drivingmarks>=(SELECT drivingmarks FROM student WHERE sname="Jones")



*5. Add field ‘Age ‘ in student table.*

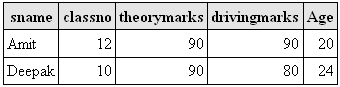
ALTER TABLE student ADD Age integer CHECK (Age>0);

UPDATE student SET Age = 19 WHERE sname = “Jones”;(Similarly add age value of others)



*6. Find records of students having marks > average theory marks of class 8.*

SELECT \* FROM student WHERE theorymarks > (SELECT AVG(theorymarks) FROM student WHERE classno=8);



***Assignment 4***

CREATE TABLE `emp`

( `empno` int(11) PRIMARY KEY, `ename` varchar(20), `job` varchar(9), `mgr` int(11) DEFAULT NULL, `hiredate` date, `sal` float, `comm` float, `deptno` int(11))

INSERT INTO `emp` VALUES

(1, 'Michael Bond', 'clerk', 20, '2008-04-16', 10000, 1000, 5),

(2, 'James Clayton', 'clerk', 22, '2002-06-17', 15000, 2000, 4),

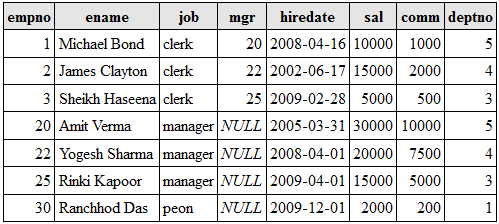
(3, 'Sheikh Haseena', 'clerk', 25, '2009-02-28', 5000, 500, 3),

(20, 'Amit Verma', 'manager', NULL, '2005-03-31', 30000, 10000, 5),

(22, 'Yogesh Sharma', 'manager', NULL, '2008-04-01', 20000, 7500, 4),

(25, 'Rinki Kapoor', 'manager', NULL, '2009-04-01', 15000, 5000, 3),

(30, 'Ranchhod Das', 'peon', NULL, '2009-12-01', 2000, 200, 1);



CREATE TABLE `dept`

( `deptno` int(11) PRIMARY KEY, `dname` varchar(14) DEFAULT NULL, `loc` varchar(13))

INSERT INTO `dept` VALUES

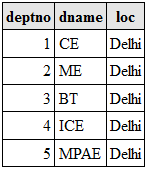
(1, 'CE', 'Delhi'),

(2, 'ME', 'Delhi'),

(3, 'BT', 'Delhi'),

(4, 'ICE', 'Delhi'),

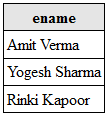
(5, 'MPAE', 'Delhi');



**Queries**

*1. List all employees who have at least one person reporting to them.*

SELECT distinct(ename) FROM `emp` WHERE empno in (select mgr from emp where mgr is not null);



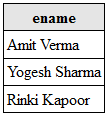
*2. List all employees iff more than 1 employee is present in department 5.*

SELECT distinct(ename) FROM `emp` WHERE `deptno`=5;



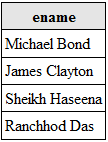
*3. List names of employee with immediate authority.*

SELECT distinct(ename) FROM `emp` WHERE mgr is null and empno in (select mgr from emp where mgr is not null);



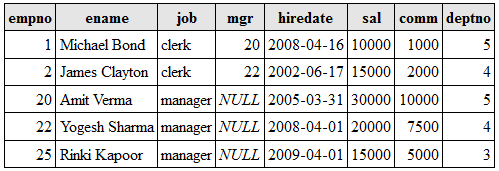
*4. List all employees who do not manage anyone.*

SELECT ename FROM `emp` WHERE empno not in (select mgr from emp where mgr is not null);



*5. List employee details whose salary is > lowest salary of employee belonging to dept. no 3.*

SELECT \* FROM `emp` WHERE `sal`>(select min(sal) from emp where `deptno`=3);



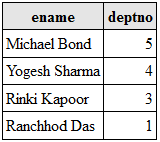
*6. List details of employee earning more than highest paid manager.*

SELECT \* FROM `emp` WHERE `sal`>(select max(sal) from emp where mgr is null);

*MySQL returned an empty result set (i.e. zero rows)*

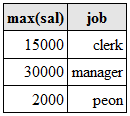
*7. Find the most recently employed employee in each dept.*

select ename,deptno from emp where hiredate in (select max(`hiredate`) from emp group by deptno);



*8. List highest salary paid for each job.*

SELECT max( sal ) , job FROM emp GROUP BY job;



*9. In which year did most people join company? Display year & no. of employee.*

create view v as(SELECT count(\*) as no,year(`hiredate`)as year FROM `emp` group by year(`hiredate`));

select no,year from v where no in (select max(no) from v);



*10. Which department has highest remuneration bill.*

create view v as (select sum(sal) as sal,deptno from emp group by deptno);

select deptno from v where sal in(select max(sal) from v);

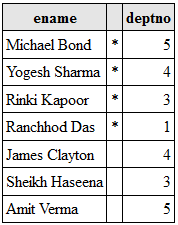


11. Write query to display ‘\*’ against row of most recently hired employee.

select ename,'\*' as '',deptno from emp where hiredate in (select max(`hiredate`) from emp group by deptno)

**union**

select ename ,'',deptno from emp where ename not in(select ename from emp where hiredate in (select max(`hiredate`) from emp group by deptno));



*12. Display employees who earn more than avg salary of their dept.*

create view v as(select avg(sal)as sal, deptno from emp group by deptno);

select ename,emp.sal,emp.deptno from emp,v where (emp.sal>v.sal) and (emp.deptno=v.deptno);

