1.

CREATE DATABASE SQL\_ASSIGNMENT;

USE SQL\_ASSIGNMENT;

CREATE TABLE STUDENTS(

NAME VARCHAR(30),

ROLL INT PRIMARY KEY,

ADDRESS VARCHAR(50),

MAIN VARCHAR(20)

);

INSERT INTO STUDENTS(NAME,ROLL,ADDRESS,MAIN)

VALUES

('RAJKUMAR DAS',261,'BANKURA','COMPUTER SCIENCE'),

('SUKDEB MONDAL',340,'DHALLA','COMPUTER SCIENCE'),

('ARNAB MAITY',256,'BAGNAN','COMPUTER SCIENCE'),

('MANAB PATRA',257,'PURULIA','COMPUTER SCIENCE'),

('ARCHISMAN DE',316,'KOLKATA','COMPUTER SCIENCE');

CREATE TABLE OFFERING(

BRANCH VARCHAR(20),

COURSE VARCHAR(50) PRIMARY KEY

);

INSERT INTO OFFERING(BRANCH,COURSE)

VALUES

('COMPUTER SCIENCE','DATA STRUCTURE'),

('COMPUTER SCIENCE','ALGORITHMS'),

('COMPUTER SCIENCE','DATABASE MANAGEMENT'),

('COMPUTER SCIENCE','OPERATING SYSTEM'),

('COMPUTER SCIENCE','ARTIFICIAL INTELLIGENCE');

CREATE TABLE FACULTY(

COURSE VARCHAR(50),

FACULTY VARCHAR(30),

SEMESTER VARCHAR(20),

FOREIGN KEY (COURSE) REFERENCES OFFERING(COURSE)

);

INSERT INTO FACULTY(COURSE, FACULTY,SEMESTER)

VALUES

('DATA STRUCTURE','ANJAN G','SEM 1'),

('ALGORITHMS','SARBAJIT M','SEM 4'),

('DATABASE MANAGEMENT','SAMARESH M','SEM 4'),

('OPERATING SYSTEM','AVISHEK B','SEM 4'),

('ARTIFICIAL INTELLIGENCE','ARINDAM S','SEM 5');

CREATE TABLE ADMISSION(

ROLL INT,

COURSE VARCHAR(50),

SEMESTER VARCHAR(20),

FOREIGN KEY (ROLL) REFERENCES STUDENTS(ROLL),

FOREIGN KEY (COURSE) REFERENCES OFFERING(COURSE)

);

INSERT INTO ADMISSION(ROLL,COURSE,SEMESTER)

VALUES

(261,'DATA STRUCTURE','SEM 1'),

(261,'ARTIFICIAL INTELLIGENCE','SEM 5'),

(261,'ALGORITHMS','SEM 4'),

(257,'OPERATING SYSTEM','SEM 4'),

(257,'DATABASE MANAGEMENT','SEM 4'),

(257,'DATA STRUCTURE','SEM 1'),

(256,'ALGORITHMS','SEM 4'),

(256,'DATABASE MANAGEMENT','SEM 4'),

(340,'ARTIFICIAL INTELLIGENCE','SEM 5'),

(340,'OPERATING SYSTEM','SEM 4'),

(340,'ALGORITHMS','SEM 4'),

(316,'ALGORITHMS','SEM 4'),

(316,'ARTIFICIAL INTELLIGENCE','SEM 5');

-- 1. all courses taken by a given student.

SELECT S.NAME,A.ROLL,A.COURSE,A.SEMESTER FROM STUDENTS AS S

JOIN ADMISSION AS A

ON S.ROLL = A.ROLL

WHERE S.NAME = 'RAJKUMAR DAS'; -- USING STUDENT'S NAME

-- WHERE S.ROLL = 261; -- USING STUDENT'S ROLL

-- 2. names of the students admitted in a particular course in a given semester.

SELECT S.NAME,A.ROLL,A.COURSE,A.SEMESTER FROM STUDENTS AS S

JOIN ADMISSION AS A

ON S.ROLL = A.ROLL

WHERE A.SEMESTER = 'SEM 1' AND A.COURSE = 'DATA STRUCTURE';

-- 3. were two students (x and y ) ever admitted in the same course in the same semester .

SELECT STUDENTS.NAME, ADMISSION.ROLL,ADMISSION.COURSE,ADMISSION.SEMESTER FROM STUDENTS

JOIN ADMISSION

ON ADMISSION.ROLL = STUDENTS.ROLL

WHERE ADMISSION.COURSE = "ALGORITHMS" AND ADMISSION.SEMESTER = 'SEM 4' AND ADMISSION.ROLL IN (261,256);

-- 4. students who have taken all courses offered by a given faculty.

SELECT S.NAME , A.COURSE FROM STUDENTS AS S

JOIN ADMISSION AS A

ON S.ROLL = A.ROLL

WHERE A.COURSE IN (

SELECT COURSE FROM FACULTY WHERE FACULTY = 'SARBAJIT M'

);

-- 5. find the name of the faculty who taught maximum courses.

SELECT FACULTY, COUNT(\*) AS COUNT FROM FACULTY

GROUP BY FACULTY

ORDER BY COUNT(COURSE) DESC;

2.

CREATE DATABASE SQL\_ASSIGNMENT2;

USE SQL\_ASSIGNMENT2;

CREATE TABLE STUDENT (

SID INT PRIMARY KEY,

SNAME VARCHAR(20),

SEX VARCHAR(10),

AGE INT,

YEAR varchar(5),

GPA float

);

INSERT INTO STUDENT(SID,SNAME,SEX,AGE,YEAR,GPA)

VALUES

(261,'RAJKUMAR DAS','M',21,'2ND',8.76), -- COMPUTER SCIENCE

(257,'MANAB PATRA','M',19,'2ND',9.45), -- COMPUTER SC

(256,'ARNAB MAITY','M',20,'2ND',9.31), -- COMPUTER SC

(340,'SUKDEB MONDAL','M',20,'2ND',8.3), -- COMPUTER SC

(196,'PALLABI GHOSH','F',19,'1ST',7.4), -- ZOOLOGY

(200,'SANGITA RAO','F',22,'3RD',8.9), -- CHEMISTRY

(123,'SHIRSHA BH','M',21,'2ND',9.2), -- ENGLISH

(312,'SOUMODEEP SAHO','M',21,'2ND',7.5), -- ZOOLOGY

(143,'ANISH CH','M',21,'2ND',9.2), -- ENGLISH

(225,'ANARGHA SARKAR','M',19,'1ST',8.5), -- MATH

(524,'ARKA BISWAS','M',21,'2ND',9.3), -- PHYSICS

(167,'ADITYA SEN','M',21,'2ND',7.3); -- MATH

CREATE TABLE DEPT(

DNAME VARCHAR(30) PRIMARY KEY,

NUMPHDS INT

);

INSERT INTO DEPT(DNAME,NUMPHDS)

VALUES

('COMPUTER SCIENCE',31),

('ZOOLOGY',16),

('ENGLISH',13),

('MATH',21),

('PHYSICS',19),

('CHEMISTRY',11);

CREATE TABLE PROF(

PNAME VARCHAR(30) PRIMARY KEY,

DNAME VARCHAR(30),

FOREIGN KEY (DNAME) REFERENCES DEPT(DNAME)

);

INSERT INTO PROF(PNAME,DNAME)

VALUES

('SAMARESH M','COMPUTER SCIENCE'),

('SARBAJIT M','COMPUTER SCIENCE'),

('ABHISEKH B','COMPUTER SCIENCE'),

('DEBANJAN M','ENGLISH'),

('SAPTARSHI K','ENGLISH'),

('SOUGATA M','ENGLISH'),

('SWAGATA D','ZOOLOGY'),

('JOYDIPM','MATH'),

('ARKIT D','MATH'),

('ARCHISMAN D','CHEMISTRY'),

('TAPASH G','CHEMISTRY'),

('JURAIN D','PHYSICS'),

('UTSAB M','PHYSICS');

CREATE TABLE COURSE(

CNO INT PRIMARY KEY,

CNAME VARCHAR(20),

DNAME VARCHAR(30),

FOREIGN KEY (DNAME) REFERENCES DEPT(DNAME)

);

INSERT INTO COURSE(CNO,CNAME,DNAME)

VALUES

(101,'MACHINE LEARNING','COMPUTER SCIENCE'),

(102,'AI','COMPUTER SCIENCE'),

(103,'GENERATIVE AI','COMPUTER SCIENCE'),

(201,'CALCULUS','MATH'),

(202,'REAL ANALYSIS','MATH'),

(301,'DNA FUNDAMENTALS','ZOOLOGY'),

(302,'PROTINE STRUCTURE','ZOOLOGY'),

(401,'MIDNIGHT EXPRESS','ENGLISH'),

(402,'THE EYES HAVE IT','ENGLISH'),

(501,'OPTICS','PHYSICS'),

(502,'THERMODYNAMICS','PHYSICS'),

(601,'ORGANIK CHEMISTRY','CHEMISTRY'),

(602,'PHYSICAL CHEMISTRY','CHEMISTRY');

CREATE TABLE MAJOR(

DNAME VARCHAR(30),

SID INT,

FOREIGN KEY (DNAME) REFERENCES DEPT(DNAME),

FOREIGN KEY (SID) REFERENCES STUDENT(SID)

);

INSERT INTO MAJOR(DNAME,SID)

VALUES

('COMPUTER SCIENCE',261),

('COMPUTER SCIENCE',257),

('COMPUTER SCIENCE',256),

('COMPUTER SCIENCE',340),

('ZOOLOGY',196),

('CHEMISTRY',200),

('ENGLISH',123),

('ZOOLOGY',312),

('ENGLISH',143),

('MATH',225),

('PHYSICS',524),

('MATH',167);

CREATE TABLE SECTION(

DNAME VARCHAR(30),

CNO INT,

SECTNO INT PRIMARY KEY,

PNAME VARCHAR(30),

FOREIGN KEY (DNAME) REFERENCES DEPT(DNAME),

FOREIGN KEY (CNO) REFERENCES COURSE(CNO),

FOREIGN KEY (PNAME) REFERENCES PROF(PNAME)

);

INSERT INTO SECTION(DNAME,CNO,SECTNO,PNAME)

VALUES

('COMPUTER SCIENCE',101,1,'SARBAJIT M'),

('COMPUTER SCIENCE',103,2,'ABHISEKH B'),

('MATH',201,3,'ARKIT D'),

('MATH',202,4,'JOYDIPM'),

('PHYSICS',502,5,'JURAIN D'),

('ZOOLOGY', 301, 6, 'SWAGATA D'),

('ZOOLOGY', 302, 7, 'SWAGATA D'),

('CHEMISTRY', 601, 8, 'ARCHISMAN D'),

('CHEMISTRY', 602, 9, 'TAPASH G'),

('ENGLISH', 401, 10, 'DEBANJAN M'),

('ENGLISH', 402, 11, 'SOUGATA M'),

('PHYSICS', 501, 12, 'UTSAB M');

CREATE TABLE ENROLL(

SID INT,

GRADE VARCHAR(4),

DNAME VARCHAR(30),

CNO INT ,

SECTNO INT,

FOREIGN KEY (SID) REFERENCES STUDENT(SID),

FOREIGN KEY (DNAME) REFERENCES DEPT(DNAME),

FOREIGN KEY (CNO) REFERENCES COURSE(CNO),

FOREIGN KEY (SECTNO) REFERENCES SECTION(SECTNO)

);

INSERT INTO ENROLL(SID, GRADE, DNAME, CNO, SECTNO)

VALUES

(261, 'A+', 'COMPUTER SCIENCE', 101, 1),

(257, 'A', 'COMPUTER SCIENCE', 103, 2),

(196, 'B+', 'COMPUTER SCIENCE', 101, 1),

(256, 'A-', 'COMPUTER SCIENCE', 103, 2),

(340, 'B', 'COMPUTER SCIENCE', 101, 1),

(196, 'A', 'ZOOLOGY', 301, 6),

(312, 'A-', 'ZOOLOGY', 302, 7),

(200, 'A', 'CHEMISTRY', 601, 8),

(200, 'B+', 'CHEMISTRY', 602, 9),

(123, 'A-', 'ENGLISH', 401, 10),

(143, 'B+', 'ENGLISH', 402, 11),

(261, 'A', 'MATH', 201, 3),

(225, 'B+', 'MATH', 202, 4),

(524, 'B', 'PHYSICS', 502, 5),

(167, 'A', 'PHYSICS', 501, 12);

-- 1.List the names of professors who work in department that have fewer than 20 PhD.

select p.pname from prof as p

join dept as d

on p.dname = d.dname

where d.numphds < 20;

-- 2.List the names of students with the lowest GPA.

select sname ,gpa from student

where gpa = (select min(gpa) from student);

-- 3.List the name and sid of student enrolled in maximum classes.

SELECT s.sid, s.sname

FROM student s

JOIN enroll e ON s.sid = e.sid

GROUP BY s.sid, s.sname

HAVING COUNT(\*) = (

SELECT MAX(cnt)

FROM (

SELECT COUNT(\*) AS cnt

FROM enroll

GROUP BY sid

) AS sub

);

-- 4.List the name of student and department name whose sid=’261’

SELECT s.sname, m.dname

FROM student s

JOIN major m ON s.sid = m.sid

WHERE s.sid = 261;

-- 5.List the names of students who are taking both a Computer Science and Mathematics course.

SELECT s.sname

FROM student s

WHERE s.sid IN (

SELECT sid FROM enroll WHERE dname = 'COMPUTER SCIENCE'

)

AND s.sid IN (

SELECT sid FROM enroll WHERE dname = 'MATH'

);

3.

CREATE DATABASE SQL\_ASSIGNMENT3;

USE SQL\_ASSIGNMENT3;

CREATE TABLE DEPARTMENT (

DEPT\_ID INT PRIMARY KEY,

DEPT\_NAME VARCHAR(50),

MGR\_ID INT,

FLOOR\_NO INT

);

INSERT INTO DEPARTMENT (DEPT\_ID, DEPT\_NAME, MGR\_ID, FLOOR\_NO)

VALUES

(1, 'COMPUTER SCIENCE', 201, 10),

(2, 'PHYSICS', 202, 9),

(3, 'ELECTORNICS', 203, 10),

(4, 'ZOOLOGY', 204, 8),

(5, 'MATH', 205, 9),

(6, 'CHEMISTRY', 206, 10),

(7, 'MATH', 206, 7),

(6, 'CHEMISTRY', 207, 10);

CREATE TABLE EMPLOYEE (

EMP\_ID INT PRIMARY KEY,

EMP\_NAME VARCHAR(50),

SALARY INT,

DEPT\_ID INT,

FOREIGN KEY (DEPT\_ID) REFERENCES DEPARTMENT(DEPT\_ID)

);

INSERT INTO EMPLOYEE (EMP\_ID, EMP\_NAME, SALARY, DEPT\_ID)

VALUES

(101, 'RJKUMAR DAS', 32000, 1),

(102, 'SUKDEB MONDAL', 28000, 2),

(103, 'ARCHISMAN DE', 9500, 3),

(104, 'MANAB PATRA', 45000, 4),

(105, 'JURAIN DE', 48000, 5),

(106, 'AKASH MONDAL', 52000, 6),

(107, 'AMIT SAHO', 8500, 3),

(108, 'ASHIS DEY', 7200, 6),

(109, 'PRIYAM DAS', 56000, 1),

(110, 'JOYDIP GHOSH', 67000, 5);

-- 1.Display the names of all employees who work on 10th floor and earn less than Rs. 10,000.

select e.emp\_name,d.floor\_no,e.salary from employee as e

join department as d

on e.dept\_id = d.dept\_id

where d.floor\_no = 10 and e.salary < 10000;

-- 2.List names of all managers who manage two or more departments on the same floor.

SELECT dept\_name, floor\_no

FROM department

GROUP BY dept\_name, floor\_no

HAVING COUNT(\*) = 2;

-- 3.Give 10% hike to the salary to every employee who works in computer science department and display the appropriate record.

SELECT e.emp\_name, e.salary, e.salary \* 1.10 AS hiked\_salary

FROM employee e

JOIN department d ON e.dept\_id = d.dept\_id

WHERE d.dept\_name = 'COMPUTER SCIENCE';

UPDATE employee

SET salary = salary \* 1.10

WHERE dept\_id IN (

SELECT dept\_id FROM department WHERE dept\_name = 'COMPUTER SCIENCE'

);

SELECT e.emp\_name, e.salary,d.dept\_name FROM employee as e

join department as d

on e.dept\_id = d.dept\_id

where d.dept\_id = 1;

-- 4.Display the names of employee who earn more than every employee of IT department.

select emp\_name from employee

where salary = (select max(salary) from employee);

-- 5.Print the names of employees who earn more than Rs. 50,000 and work in either sales or marketing departments.

SELECT e.emp\_name,e.salary,d.dept\_name

FROM employee e

JOIN department d ON e.dept\_id = d.dept\_id

WHERE e.salary > 40000 AND d.dept\_name IN ('ZOOLOGY', 'MATH');

4.

CREATE DATABASE SQL\_ASSIGNMENT4;

USE SQL\_ASSIGNMENT4;

CREATE TABLE CUSTOMER (

CUST\_NAME VARCHAR(50) PRIMARY KEY,

STREET VARCHAR(50),

CITY VARCHAR(50)

);

INSERT INTO CUSTOMER (CUST\_NAME, STREET, CITY)

VALUES

('Rajkumar Das', 'MG Road', 'Howrah'),

('Sukdeb Mondal', 'College Street', 'Kolkata'),

('Archisman De', 'Ballygunge', 'Kolkata'),

('Manab Patra', 'Shibpur', 'Howrah'),

('Amit Kumar', 'New Town', 'Delhi'),

('Suvash sen', 'Park Street', 'Mumbai');

CREATE TABLE BRANCH (

BR\_NAME VARCHAR(50) PRIMARY KEY,

ASSETS INT,

CITY VARCHAR(50)

);

INSERT INTO BRANCH (BR\_NAME, ASSETS, CITY) VALUES

('SBI, Belur Math', 5000000, 'Howrah'),

('SBI, Kolkata', 7000000, 'Kolkata'),

('HDFC, Park Street', 6000000, 'Kolkata');

CREATE TABLE DEPOSIT (

BR\_NAME VARCHAR(50),

ACC\_NO INT PRIMARY KEY,

CUST\_NAME VARCHAR(50),

BALANCE INT,

FOREIGN KEY (CUST\_NAME) REFERENCES CUSTOMER(CUST\_NAME),

FOREIGN KEY (BR\_NAME) REFERENCES BRANCH(BR\_NAME)

);

INSERT INTO DEPOSIT (BR\_NAME, ACC\_NO, CUST\_NAME, BALANCE) VALUES

('SBI, Belur Math', 1001, 'Rajkumar Das', 85000),

('SBI, Kolkata', 1002, 'Sukdeb Mondal', 92000),

('HDFC, Park Street', 1003, 'Suvash sen', 75000),

('SBI, Belur Math', 1004, 'Manab Patra', 65000);

CREATE TABLE BORROW (

BR\_NAME VARCHAR(50),

LOAN\_NO INT PRIMARY KEY,

CUST\_NAME VARCHAR(50),

AMOUNT INT,

FOREIGN KEY (CUST\_NAME) REFERENCES CUSTOMER(CUST\_NAME),

FOREIGN KEY (BR\_NAME) REFERENCES BRANCH(BR\_NAME)

);

INSERT INTO BORROW (BR\_NAME, LOAN\_NO, CUST\_NAME, AMOUNT) VALUES

('SBI, Kolkata', 2001, 'Rajkumar Das', 120000),

('SBI, Belur Math', 2002, 'Archisman De', 80000),

('HDFC, Park Street', 2003, 'Suvash sen', 30000),

('SBI, Kolkata', 2004, 'Manab Patra', 55000),

('SBI, Belur Math', 2005, 'Amit Kumar', 60000);

-- 1.Find all customers who have either account or loan or both in SBI, Belur Math Branch

SELECT DISTINCT cust\_name

FROM (

SELECT cust\_name FROM deposit WHERE br\_name = 'SBI, Belur Math'

UNION

SELECT cust\_name FROM borrow WHERE br\_name = 'SBI, Belur Math'

) AS result;

-- 2.Find customers with names and cities in which they live who have taken loan from SBI, Kolkata Branch.

SELECT c.cust\_name, c.city

FROM customer c

JOIN borrow b ON c.cust\_name = b.cust\_name

WHERE b.br\_name = 'SBI, Kolkata';

-- 3.Find customers name and amount including name of the branch for loan over Rs.50,000.

SELECT b.cust\_name, b.br\_name, b.amount

FROM borrow b

WHERE b.amount > 50000;

-- 4.Find customer having highest balance.

SELECT cust\_name, balance

FROM deposit

WHERE balance = (SELECT MAX(balance) FROM deposit);

-- 5.Find branch having highest loan.

SELECT br\_name, SUM(amount) AS total\_loan

FROM borrow

GROUP BY br\_name

ORDER BY total\_loan DESC

LIMIT 1;

5.

CREATE DATABASE SQL\_ASSIGNMENT5;

USE SQL\_ASSIGNMENT5;

CREATE TABLE STUDENT (

SNUM INT PRIMARY KEY,

SNAME VARCHAR(50),

MAJOR VARCHAR(50),

LEVEL VARCHAR(10),

AGE INT

);

INSERT INTO STUDENT (SNUM, SNAME, MAJOR, LEVEL, AGE) VALUES

(1, 'Rajkumar Das', 'Computer Science', 'SR', 22),

(2, 'Sukdeb Mondal', 'Physics', 'JR', 20),

(3, 'Archisman De', 'Mathematics', 'SR', 23),

(4, 'Manab Patra', 'Electronics', 'SO', 21),

(5, 'Amit Kumar', 'Zoology', 'FR', 19),

(6, 'Suvash Kumar', 'Mathematics', 'SR', 24);

CREATE TABLE FACULTY (

FID INT PRIMARY KEY,

FNAME VARCHAR(50),

DEPTID INT

);

INSERT INTO FACULTY (FID, FNAME, DEPTID) VALUES

(101, 'Ravi Kumar', 1),

(102, 'Priya Sharma', 2),

(103, 'Arjun Mehta', 3);

CREATE TABLE CLASS (

NAME VARCHAR(50) PRIMARY KEY,

MEETS\_AT VARCHAR(20),

ROOM VARCHAR(10),

FID INT,

FOREIGN KEY (FID) REFERENCES FACULTY(FID)

);

INSERT INTO CLASS (NAME, MEETS\_AT, ROOM, FID) VALUES

('Database Systems', '10 AM', 'R128', 101),

('Quantum Physics', '2 PM', 'R128', 102),

('Linear Algebra', '1 PM', 'R200', 103),

('Machine Learning', '4 PM', 'R300', 101);

CREATE TABLE ENROLLED (

SNUM INT,

CNAME VARCHAR(50),

FOREIGN KEY (SNUM) REFERENCES STUDENT(SNUM),

FOREIGN KEY (CNAME) REFERENCES CLASS(NAME)

);

INSERT INTO ENROLLED (SNUM, CNAME) VALUES

(1, 'Database Systems'),

(2, 'Quantum Physics'),

(3, 'Linear Algebra'),

(4, 'Database Systems'),

(5, 'Quantum Physics'),

(6, 'Machine Learning'),

(1, 'Linear Algebra'),

(3, 'Database Systems'),

(4, 'Quantum Physics'),

(5, 'Machine Learning');

-- 1.Find the names of faculty members for whom the combined enrollment of the courses that they teach is less than five.

SELECT f.fname

FROM faculty f

JOIN class c ON f.fid = c.fid

LEFT JOIN enrolled e ON c.name = e.cname

GROUP BY f.fid, f.fname

HAVING COUNT(e.snum) < 5;

-- 2.For each level, print the level and the average age of students for that level.

SELECT level, AVG(age) AS avg\_age

FROM student

GROUP BY level;

-- 3.For all levels except JR, print the level and the average age of students for that level.

SELECT level, AVG(age) AS avg\_age

FROM student

WHERE level <> 'JR'

GROUP BY level;

-- 4.For each faculty member that has taught classes only in room R128, print the faculty member's name and the total number of classes she or he has taught.

SELECT f.fname, COUNT(\*) AS total\_classes

FROM faculty f

JOIN class c ON f.fid = c.fid

GROUP BY f.fid, f.fname

HAVING COUNT(DISTINCT c.room) = 1 AND MAX(c.room) = 'R128';

-- 5.Find the names of students enrolled in the maximum number of classes.

SELECT s.sname

FROM student s

JOIN enrolled e ON s.snum = e.snum

GROUP BY s.snum, s.sname

HAVING COUNT(e.cname) = (

SELECT MAX(class\_count)

FROM (

SELECT COUNT(\*) AS class\_count

FROM enrolled

GROUP BY snum

) AS sub

);

6.

-- in the railway reservation system, it contains the following table

-- t\_info(t\_no,t\_name,source,destination)

-- t\_schedule(date,t\_id,deperture,duration)

-- reservation(t\_id,p\_id,pnr,date,seat\_no,source,destination)

-- passenger(p\_id,name,city,mob)

-- if any train is cancelled on schedule date and all the reservation for

-- that train on same date but at different time maintaining all the reservation

-- status then write down the PL/SQL above incident to perform the necessary

-- changes in the above mention table.

CREATE DATABASE SQL\_ASSIGNMENT6;

USE SQL\_ASSIGNMENT6;

CREATE TABLE t\_info (

t\_no INT PRIMARY KEY,

t\_name VARCHAR(50),

source VARCHAR(50),

destination VARCHAR(50)

);

INSERT INTO t\_info VALUES

(101, 'Howrah Express', 'Howrah', 'Sealdah'),

(102, 'Arambagh Local', 'Arambagh', 'Bardhaman'),

(103, 'Bandel Shuttle', 'Bandel', 'Howrah'),

(104, 'Kolkata Duronto', 'Kolkata', 'New Jalpaiguri'),

(105, 'Siliguri Express', 'Siliguri', 'Malda'),

(106, 'Sealdah Mail', 'Sealdah', 'Asansol'),

(107, 'Kharagpur Fast', 'Howrah', 'Kharagpur'),

(108, 'Shantiniketan Express', 'Howrah', 'Bolpur'),

(109, 'Rampurhat Local', 'Rampurhat', 'Katwa'),

(110, 'Burdwan Memu', 'Burdwan', 'Howrah');

CREATE TABLE t\_schedule (

date DATE,

t\_id INT,

departure TIME,

duration VARCHAR(10), -- New format: 'HH:MM'

PRIMARY KEY (date, t\_id),

FOREIGN KEY (t\_id) REFERENCES t\_info(t\_no)

);

INSERT INTO t\_schedule VALUES

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 101, '08:00:00', '02:00'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 102, '10:30:00', '01:00'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 103, '11:00:00', '01:45'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 104, '06:15:00', '07:30'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 105, '13:00:00', '05:00'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 106, '15:10:00', '03:20'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 107, '09:45:00', '02:15'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 108, '07:20:00', '02:30'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 109, '16:00:00', '01:20'),

(TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 110, '12:30:00', '01:50');

CREATE TABLE passenger (

p\_id INT PRIMARY KEY,

name VARCHAR(50),

city VARCHAR(50),

mob VARCHAR(15)

);

INSERT INTO passenger VALUES

(1, 'Rajkumar Das', 'Howrah', '9876543210'),

(2, 'Sukdeb Mondal', 'Arambagh', '9876500000'),

(3, 'Archisman De', 'Bandel', '9876512345'),

(4, 'Manab Patra', 'Sealdah', '9876523456'),

(5, 'Amit Kumar', 'Siliguri', '9876534567');

CREATE TABLE reservation (

t\_id INT,

p\_id INT,

pnr INT PRIMARY KEY,

date DATE,

seat\_no VARCHAR(10),

source VARCHAR(50),

destination VARCHAR(50),

status VARCHAR(20),

FOREIGN KEY (t\_id) REFERENCES t\_info(t\_no),

FOREIGN KEY (p\_id) REFERENCES passenger(p\_id)

);

INSERT INTO reservation VALUES

(101, 1, 1001, TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 'A1-05', 'Howrah', 'Sealdah', 'Confirmed'),

(102, 2, 1002, TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 'S2-01', 'Arambagh', 'Bardhaman', 'Confirmed'),

(103, 3, 1003, TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 'C1-10', 'Bandel', 'Howrah', 'Confirmed'),

(104, 4, 1004, TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 'B2-02', 'Kolkata', 'New Jalpaiguri', 'Confirmed'),

(105, 5, 1005, TO\_DATE('2025-04-07', 'YYYY-MM-DD'), 'B3-15', 'Siliguri', 'Malda', 'Confirmed');

7.

-- supplier(s\_id,sname, address)

-- parts(pid,pname,colour)

-- catalogue( sid,pid,cost)

-- the catalogue relation list the price charged for part by supplier

-- write down the following queries in SQL

-- 1. find the sname who supplies only red parts

-- 2. find the sname who supplies the different parts in minimum rate show the parts with their rate.

-- 3. find the suppliers who charged less for some part than the average cost of that part

-- 4. find the pnames of blue parts for which there is some suppliers.

CREATE DATABASE SQL\_ASSIGNMENT7;

USE SQL\_ASSIGNMENT7;

CREATE TABLE supplier (

s\_id INT PRIMARY KEY,

sname VARCHAR(50),

address VARCHAR(100)

);

INSERT INTO supplier (s\_id, sname, address) VALUES

(1, 'Raj Suppliers', 'Kolkata'),

(2, 'Das Distributors', 'Howrah'),

(3, 'Mondal Traders', 'Delhi'),

(4, 'Archi Parts Co.', 'Mumbai'),

(5, 'Manab Supplies', 'Chennai'),

(6, 'Kumar Hardware', 'Bangalore');

CREATE TABLE parts (

p\_id INT PRIMARY KEY,

pname VARCHAR(50),

colour VARCHAR(20)

);

INSERT INTO parts (p\_id, pname, colour) VALUES

(101, 'Bolt', 'red'),

(102, 'Nut', 'blue'),

(103, 'Screw', 'green'),

(104, 'Washer', 'red'),

(105, 'Pin', 'blue'),

(106, 'Cap', 'red');

CREATE TABLE catalogue (

s\_id INT,

p\_id INT,

cost INT,

FOREIGN KEY (s\_id) REFERENCES supplier(s\_id),

FOREIGN KEY (p\_id) REFERENCES parts(p\_id)

);

INSERT INTO catalogue (s\_id, p\_id, cost) VALUES

(1, 101, 15),

(1, 104, 20),

(2, 101, 18),

(2, 102, 25),

(3, 103, 30),

(4, 105, 22),

(5, 106, 17),

(6, 102, 19),

(6, 105, 21);

-- 1. find the sname who supplies only red parts.

SELECT s.sname

FROM supplier s

WHERE s.s\_id IN (

SELECT c.s\_id

FROM catalogue c

JOIN parts p ON c.p\_id = p.p\_id

WHERE p.colour = 'red'

);

-- 2. find the sname who supplies the different parts in minimum rate show the parts with their rate.

SELECT s.sname, p.pname, c.cost

FROM catalogue c

JOIN (

SELECT p\_id, MIN(cost) AS min\_cost

FROM catalogue

GROUP BY p\_id

) min\_costs ON c.p\_id = min\_costs.p\_id AND c.cost = min\_costs.min\_cost

JOIN supplier s ON c.s\_id = s.s\_id

JOIN parts p ON c.p\_id = p.p\_id;

-- 3. find the suppliers who charged less for some part than the average cost of that part

SELECT DISTINCT s.sname

FROM catalogue c

JOIN parts p ON c.p\_id = p.p\_id

JOIN supplier s ON c.s\_id = s.s\_id

WHERE c.cost < (

SELECT AVG(c2.cost)

FROM catalogue c2

WHERE c2.p\_id = c.p\_id

);

-- 4. find the pnames of blue parts for which there is some suppliers.

SELECT DISTINCT p.pname

FROM parts p

JOIN catalogue c ON p.p\_id = c.p\_id

WHERE p.colour = 'blue';