

SQL ASSIGNMENTS

1)

```
SELECT COUNT(*) - COUNT(DISTINCT CITY)
FROM STATION;
```

2)

```
CREATE TABLE Teams (
    team_id INTEGER,
    team_name VARCHAR(255),
    team_city VARCHAR(255),
    coach VARCHAR(255),
    captain_id INTEGER,
    PRIMARY KEY (team_id)
);

CREATE TABLE Players (
    player_id INTEGER,
    team_id INTEGER,
    player_name VARCHAR(255),
    position VARCHAR(255),
    skill_level INTEGER,
    PRIMARY KEY (player_id),
    FOREIGN KEY (team_id) REFERENCES Teams(team_id)
);

CREATE TABLE Injury_Records (
    player_id INTEGER,
    injury VARCHAR(255),
    severity INTEGER,
    FOREIGN KEY (player_id) REFERENCES Players(player_id)
);

CREATE TABLE Games (
    game_id INTEGER,
    host_team_id INTEGER,
    guest_team_id INTEGER,
    date DATE,
    score VARCHAR(255),
    PRIMARY KEY (game_id),
    FOREIGN KEY (host_team_id) REFERENCES Teams(team_id),
    FOREIGN KEY (guest_team_id) REFERENCES Teams(team_id)
);

INSERT INTO Teams (team_id, team_name, team_city, coach, captain_id)
VALUES (1, 'Toronto Maple Leafs', 'Toronto', 'John Smith', 7);
INSERT INTO Players (player_id, team_id, player_name, position, skill_level)
VALUES (7, 1, 'John Doe', 'Right Wing', 9);
INSERT INTO Injury_Records (player_id, injury, severity)
VALUES (7, 'Sprained Ankle', 3);
INSERT INTO Games (game_id, host_team_id, guest_team_id, date, score)
```

SQL ASSIGNMENTS

```
VALUES (1, 1, 2, 'May 11th, 1999', '4 to 2');
3)
create table courses (
  course_id int primary key,
  title varchar(50),
  credits int,
  syllabus varchar(1000),
  prerequisites varchar(50)
);
create table course_offerings (
  course_offering_id int primary key,
  course_id int,
  year int,
  semester int,
  section_number int,
  instructor_id int,
  timings varchar(50),
  classroom varchar(50),
  foreign key (course_id) references courses (course_id),
  foreign key (instructor_id) references instructors (instructor_id)
);
create table students (
  student_id int primary key,
  name varchar(50),
  program varchar(50)
);
create table instructors (
  instructor_id int primary key,
  name varchar(50),
  department varchar(50),
  title varchar(50)
);
create table course_enrollment (
  course_offering_id int,
  student_id int,
  grade varchar(50),
  foreign key (course_offering_id) references course_offerings (course_offering_id),
  foreign key (student_id) references students (student_id)
);
SELECT course.title, offering.year, offering.semester, offering.section_number,
offering.instructor_id
FROM courses AS course
INNER JOIN course_offerings AS offering
ON course.course_id = offering.course_id;
SELECT student.name, offering.course_id, enrollment.grade
FROM students AS student
```

SQL ASSIGNMENTS

```
INNER JOIN course_enrollment AS enrollment
ON student.student_id = enrollment.student_id
INNER JOIN course_offerings AS offering
ON enrollment.course_offering_id = offering.course_offering_id;
SELECT instructor.name, offering.course_id, offering.section_number
FROM instructors AS instructor
INNER JOIN course_offerings AS offering
ON instructor.instructor_id = offering.instructor_id;
```

4)

```
CREATE TABLE team_stats
(
    match_id INT NOT NULL AUTO_INCREMENT,
    team VARCHAR(100) NOT NULL,
    opponent VARCHAR(100) NOT NULL,
    match_date DATE NOT NULL,
    score INT NOT NULL,
    PRIMARY KEY (match_id)
);
CREATE TABLE players
(
    player_id INT NOT NULL AUTO_INCREMENT,
    player_name VARCHAR(100) NOT NULL,
    PRIMARY KEY (player_id)
);
CREATE TABLE match_players
(
    match_id INT NOT NULL,
    player_id INT NOT NULL,
    PRIMARY KEY (match_id, player_id),
    FOREIGN KEY (match_id) REFERENCES team_stats (match_id),
    FOREIGN KEY (player_id) REFERENCES players (player_id)
);
CREATE TABLE player_stats
(
    match_id INT NOT NULL,
    player_id INT NOT NULL,
    goals INT NOT NULL,
    assists INT NOT NULL,
    PRIMARY KEY (match_id, player_id),
    FOREIGN KEY (match_id) REFERENCES team_stats (match_id),
    FOREIGN KEY (player_id) REFERENCES players (player_id)
);
INSERT INTO team_stats (team, opponent, match_date, score)
VALUES ('India', 'Australia', '2020-01-02', 5);
INSERT INTO players (player_name)
```

SQL ASSIGNMENTS

```
VALUES ('Virat Kohli'), ('Rohit Sharma'), ('MS Dhoni');
INSERT INTO match_players (match_id, player_id)
VALUES (1, 1), (1, 2), (1, 3);
INSERT INTO player_stats (match_id, player_id, goals, assists)
VALUES (1, 1, 2, 0), (1, 2, 3, 2), (1, 3, 0, 0);
SELECT t.team, t.opponent, t.match_date, t.score, p.player_name, s.goals, s.assists
FROM team_stats t
INNER JOIN match_players m ON t.match_id = m.match_id
INNER JOIN players p ON m.player_id = p.player_id
INNER JOIN player_stats s ON t.match_id = s.match_id
WHERE t.team = 'India';
```

5)

```
create table EXAM
(
    Course_Name VARCHAR(20),
    Section_Number INTEGER,
    Room_Number INTEGER,
    Time TIME,
    PRIMARY KEY(Course_Name, Section_Number)
);
create table COURSE
(
    Name VARCHAR(20),
    Department VARCHAR(20),
    C_Number INTEGER,
    PRIMARY KEY(Name, Department, C_Number)
);
create table SECTION
(
    S_Number INTEGER,
    Enrollment INTEGER,
    Course_Name VARCHAR(20),
    Department VARCHAR(20),
    C_Number INTEGER,
    PRIMARY KEY (S_Number),
    FOREIGN KEY (Course_Name, Department, C_Number) REFERENCES COURSE(Name,
Department, C_Number)
);
create table ROOM
(
    R_Number INTEGER,
    Capacity INTEGER,
    Building VARCHAR(20),
    PRIMARY KEY(R_Number)
);
```

SQL ASSIGNMENTS

```
create table DEPENDENT
(
  Course_Name VARCHAR(20),
  Department VARCHAR(20),
  C_Number INTEGER,
  S_Number INTEGER,
  PRIMARY KEY (Course_Name, Department, C_Number, S_Number),
  FOREIGN KEY (Course_Name, Department, C_Number) REFERENCES COURSE(Name,
Department, C_Number),
  FOREIGN KEY (S_Number) REFERENCES SECTION(S_Number)
);
ALTER TABLE EXAM
ADD FOREIGN KEY (Room_Number) REFERENCES ROOM (R_Number);
ALTER TABLE EXAM
ADD FOREIGN KEY (Course_Name, Department, C_Number) REFERENCES DEPENDENT
(Course_Name, Department, C_Number);
```