|  |
| --- |
|  |
|  |  |
|  |  |
|  | 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) |
|  | 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 |
|  | 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) |
|  | 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) |
|  | ); |
|  |  |
|  | 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); |