SQL PROJECT:

**Students Databases Management of School**

-- I am going to do the project of SQL.

-- step 1: create the database and tables.

-- create database schoolDB

-- use schoolDB

-- (table for storing student info)

-- create table students(

-- student\_id int primary key,

-- first\_name varchar(50),

-- last\_name varchar(50),

-- date\_of\_birth date,

-- gender char(1)

-- )

-- select \* from students

-- (table for storing course info)

-- create table course(

-- course\_id int primary key,

-- course\_name varchar(100),

-- credits int

-- )

-- (table for storing enrollment info)

-- create table enrollments(

-- enrollment\_id int primary key,

-- student\_id int,

-- course\_id int,

-- enrollment\_date date,

-- grade char(1),

-- foreign key (student\_id) references students(student\_id),

-- foreign key (course\_id) references course(course\_id)

-- )

-- step 2: insert sample data 1. \*insert data into student table\*

-- insert into students(student\_id, first\_name, last\_name, date\_of\_birth, gender)values

-- (1,'john','doe','2000-01-15','M'),

-- (2,'jane','smith','1999-03-22','F'),

-- (3,'mike','johnson','2001-07-30','M'),

-- (4,'emily','davis','2002-10-05','F')

-- (2.insert data into course table)

-- insert into course(course\_id, course\_name, credits) values

-- (1,'mathematics',3),

-- (2,'english',4),

-- (3,'computer science',3),

-- (4,'history',3)

-- (3.insert data into enrollement table)

-- insert into enrollments(enrollment\_id, student\_id, course\_id, enrollment\_date, grade)values

-- (1,1,1,'2023-01-15','A'),

-- (2,2,2,'2023-01-16','B'),

-- (3,1,3,'2023-01-17','A'),

-- (4,3,1,'2023-01-18','C'),

-- (5,4,4,'2023-01-19','B')

-- step 3: write SQL queries. 1. retrive all students

-- select \* from students

-- (2.retrive all courses)

-- select \* from course

-- (3.retrive all enrollments)

-- select \* from enrollments

-- (4.get the names of students enrolled in a specific courses)

-- select students.first\_name, students.last\_name from students

-- join enrollments on students.student\_id = enrollments.student\_id

-- where enrollments.course\_id = 1

-- (5.get the list of courses a specific student is enrolled in)

-- select course.course\_name from course join

-- enrollments on course.course\_id = enrollments.course\_id where

-- enrollments.student\_id = 1

-- -- (6.get the average grade of a specific course)

-- select course.course\_name, avg(case

-- when grade = 'A' then 4.0

-- when grade = 'B' then 3.0

-- when grade = 'C' then 2.0

-- when grade = 'D' then 1.0

-- else 0.0

-- end) as average\_grade from course

-- join enrollments on course.course\_id = enrollments.course\_id

-- where course.course\_id = 1 group by

-- course.course\_name

-- (7. count the number of students in each course)

-- select course.course\_name, count(enrollments.student\_id) as student\_count from course

-- join enrollments on course.course\_id = enrollments.course\_id group by

-- course.course\_name

-- (8. list all students with their enrolled course)

-- select students.first\_name, students.last\_name, course.course\_name from students

-- join enrollments on students.student\_id = enrollments.student\_id join course on

-- course.course\_id = enrollments.course\_id order by students.last\_name, students.first\_name

-- step 4: advanced sql queries \*find students with no enrollments\*

-- select students.first\_name, students.last\_name

-- from students

-- left join enrollments on students.student\_id = enrollments.student\_id where enrollments.student\_id is null

-- (10. get the total number of enrollments per students)

-- select students.first\_name, students.last\_name, count(enrollments.enrollment\_id)

-- as total\_enrollments from students

-- left join enrollments on students.student\_id = enrollments.student\_id group by students.student\_id,

-- students.first\_name, students.last\_name

-- (11.find the highest grade for each course)

-- select course.course\_name, max(enrollments.grade) as highest\_grade

-- from course

-- join enrollments on course.course\_id = enrollments.course\_id

-- group by course.course\_name

-- (12.calculate the average grade per students)

-- select students.first\_name, students.last\_name,

-- avg(case

-- when grade = 'A' then 4.0

-- when grade = 'B' then 3.0

-- when grade = 'C' then 2.0

-- when grade = 'D' then 1.0

-- else 0.0

-- end) as average\_grade from students

-- join enrollments on students.student\_id = enrollments.student\_id

-- group by students.first\_name, students.last\_name

-- (13. identify courses with more than 2 students enrolled)

-- select course.course\_name, count(enrollments.student\_id) as student\_count

-- from course

-- join enrollments on course.course\_id = enrollments.course\_id

-- group by course.course\_name

-- having count(enrollments.student\_id) > 2

-- (14. list students and their grades for a specific course)

-- select students.first\_name, students.last\_name, enrollments.grade

-- from students

-- join enrollments on students.student\_id = enrollments.student\_id

-- where enrollments.course\_id = 1 (replcae with the course\_id you are intrested in)

-- (15. get enrollments details for students born after a specific date)

-- select students.first\_name, students.last\_name, enrollments.enrollment\_date, course.course\_name

-- from students

-- join enrollments on students.student\_id = enrollments.student\_id

-- join course on enrollments.course\_id = course.course\_id

-- where students.date\_of\_birth > '2000-01-01' (replace with rhw date you are intrested in)

-- step 5: adding new features (16.\* addatable for teachers \*)

-- create table teachers(

-- teacher\_id int primary key,

-- first\_name varchar(50),

-- last\_name varchar(50),

-- hire\_date date)

-- (17. link courses to teachers)

-- alter table course

-- add teacher\_id int,

-- add foreign key (teacher\_id) references teachers(teacher\_id)

-- 18. insert sample data into teachers table

-- insert into teachers(teacher\_id, first\_name, last\_name, hire\_date) values

-- (1, 'alice', 'brown', '2015-08-01'),

-- (2, 'bob', 'green', '2018-01-15')

-- update course set teacher\_id = 1 where course\_id in(1,2) (assigned mathematics,english)

-- update course set teacher\_id = 2 where course\_id in(3,4) (assigned computer science, history)

-- step 6: (more complex queries)

-- (19. find courses taught by each teacher)

-- select teachers.first\_name as teacher\_first\_name,

-- teachers.last\_name as teacher\_last\_name, course.course\_name

-- from teachers

-- join course on teachers.teacher\_id = course.teacher\_id

-- (20. calculate average grade given by each teacher)

-- select teachers.first\_name, teachers.last\_name,

-- avg(case

-- when grade = 'A' then 4.0

-- when grade = 'B' then 3.0

-- when grade = 'C' then 2.0

-- when grade = 'D' then 1.0

-- else 0.0

-- end) as average\_grade from teachers

-- join course on teachers.teacher\_id = course.teacher\_id

-- join enrollments on course.course\_id = enrollments.course\_id

-- group by teachers.first\_name, teachers.last\_name

-- (21. list teachers and the number of courses they teach)

-- select teachers.first\_name, teachers.last\_name, count(course.course\_id) as course\_count

-- from teachers

-- left join course on teachers.teacher\_id = course.teacher\_id

-- group by teachers.first\_name, teachers.last\_name