SQL 3 PROJECT

سميره عابد الحصيني – فريق الدرعية

اسم المشرف: حاتم السلمان

مشروع ٣

– وصف المشروع

باستخدام ما تعلمته خلال هذه الدورة قم بتطبيق الآتي, بحيث أن هذا المشروع مكمّل لمشروع SQL المستوى الثاني.

المتطلبات:

- انشاء علاقة بين جدول المعلمين والطلاب (بحيث أن المعلم يدرّس اكثر من طالب، والطالب يقوم بتدريسه أكثر من معلم)

```
/*
create a relationship between the table of teachers and students
(the teacher teaches more than one student,
and the student is taught by more than one teacher) */
CREATE TABLE stuRelTech (
student_id int not null,
teacher_id int not null,
 FOREIGN KEY (student_id) REFERENCES Students (Student_id),
 FOREIGN KEY (teacher_id) REFERENCES Teachers (Teacher_id),
 PRIMARY KEY (student_id,teacher_id)
  );



 /* create a relationship between the table of subjects and teachers
 (so that the teacher teaches only one subject,
 and the subject is taught by more than one teacher) */
```

Output

Action Output

#	Time	Action	Message
10	12:17:30	SELECT * FROM Students LIMIT 0, 1000	31 row(s) returned
11	12:17:50	UPDATE Students SET Student_GPA = Student_GPA +5 WHERE Student_GPA < 60	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
12	12:18:17	SELECT * FROM students LIMIT 0, 1000	31 row(s) returned
13	17:31:50	CREATE TABLE stuRelTech (student_id int not null, teacher_id int not null, FOREIGN KEY (student_id) REFERENCE...	0 row(s) affected

- انشاء علاقة بين جدول المواد والمعلمين (بحيث أن المعلم يقوم بتدريس مادة واحدة فقط، والمادة يقوم بتدريسها أكثر من معلم).

```
 /* create a relationship between the table of subjects and teachers
 (so that the teacher teaches only one subject,
 and the subject is taught by more than one teacher) */

 ALTER TABLE Teachers
 ADD COLUMN subject_id int,
 ADD FOREIGN KEY (subject_id) REFERENCES Courses(Subject_id);


/*
create a relationship between the table of subjects and students
(So that the student studies more than one subject,
 and the subject is studied by more than one student).
```

Output

Action Output

#	Time	Action	Message
8	12:16:41	SELECT * FROM Students LIMIT 0, 1000	31 row(s) returned
9	12:17:30	UPDATE Students SET Student_gender = 'MALE' WHERE Student_gender ='M'	19 row(s) affected Rows matched: 19 Changed: 19 Warnings: 0
10	12:17:30	SELECT * FROM Students LIMIT 0, 1000	31 row(s) returned
11	12:17:50	UPDATE Students SET Student_GPA = Student_GPA +5 WHERE Student_GPA < 60	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0
12	12:18:17	SELECT * FROM students LIMIT 0, 1000	31 row(s) returned
13	17:31:50	CREATE TABLE stuRelTech (student_id int not null, teacher_id int not null, FOREIGN KEY (student_id) REFERENCE...	0 row(s) affected
14	17:32:59	ALTER TABLE Teachers ADD COLUMN subject_id int, ADD FOREIGN KEY (subject_id) REFERENCES Courses(Subj...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0

- انشاء علاقة بين جدول المواد والطلاب (بحيث أن الطالب يدرس أكثر مادة، والمادة يدرسها أكثر من طالب).

```
/*
create a relationship between the table of subjects and students
(So that the student studies more than one subject,
 and the subject is studied by more than one student).
*/
CREATE TABLE subjRelSTU (
subject_id int not null,
student_id int not null,
 FOREIGN KEY (subject_id) REFERENCES Courses (Subject_id),
 FOREIGN KEY (student_id) REFERENCES Students (Student_id),
 PRIMARY KEY (subject_id,student_id)
  );


/* Create the procedure called student_info that displays the names of the students and the subjects
contain all the data shared between the subjects and students table */
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
10	12:17:30	SELECT * FROM Students LIMIT 0, 1000	31 row(s) returned	0.000 sec / 0.000 sec
11	12:17:50	UPDATE Students SET Student_GPA = Student_GPA +5 WHERE Student_GPA < 60	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.016 sec
12	12:18:17	SELECT * FROM students LIMIT 0, 1000	31 row(s) returned	0.000 sec / 0.000 sec
13	17:31:50	CREATE TABLE stuRelTech (student_id int not null, teacher_id int not null, FOREIGN KEY (student_id) REFERENCE...	0 row(s) affected	0.094 sec
14	17:32:59	ALTER TABLE Teachers ADD COLUMN subject_id int, ADD FOREIGN KEY (subject_id) REFERENCES Courses(Subj...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.157 sec
15	17:34:14	CREATE TABLE subjRelSTU (subject_id int not null, student_id int not null, FOREIGN KEY (subject_id) REFERENCES...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server versi...	0.000 sec
16	17:38:06	CREATE TABLE subjRelSTU (subject_id int not null, student_id int not null, FOREIGN KEY (subject_id) REFERENCES...	0 row(s) affected	0.063 sec

- قم بانشاء Procedureباسم student_info يعرض اسماء الطلاب و المواد يحتوي على جميع البيانات المشتركه بين جدول المواد و الطلاب

```
/* Create the procedure called student_info that displays the names of the students and the subjects
contain all the data shared between the subjects and students table */
delimiter //
CREATE PROCEDURE student_info()
BEGIN
SELECT Student_name, Subject_name
FROM subjRelSTU
JOIN Students ON Students.Student_id = student_id
JOIN Courses ON Courses.Subject_id = subject_id;
END ;
```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
11	12:17:50	UPDATE Students SET Student_GPA = Student_GPA +5 WHERE Student_GPA < 60	0 row(s) affected Rows matched: 0 Changed: 0 Warnings: 0	0.016 sec
12	12:18:17	SELECT * FROM students LIMIT 0, 1000	31 row(s) returned	0.000 sec / 0.000 sec
13	17:31:50	CREATE TABLE stuRelTech (student_id int not null, teacher_id int not null, FOREIGN KEY (student_id) REFERENCE...	0 row(s) affected	0.094 sec
14	17:32:59	ALTER TABLE Teachers ADD COLUMN subject_id int, ADD FOREIGN KEY (subject_id) REFERENCES Courses(Subj...	10 row(s) affected Records: 10 Duplicates: 0 Warnings: 0	0.157 sec
15	17:34:14	CREATE TABLE subjRelSTU (subject_id int not null, student_id int not null, FOREIGN KEY (subject_id) REFERENCES...	Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server versi...	0.000 sec
16	17:38:06	CREATE TABLE subjRelSTU (subject_id int not null, student_id int not null, FOREIGN KEY (subject_id) REFERENCES...	0 row(s) affected	0.063 sec
17	17:39:03	CREATE PROCEDURE student_info() BEGIN SELECT Student_name, Subject_name FROM subjRelSTU JOIN Stude...	0 row(s) affected	0.031 sec

- قم باستدعائها.

```
267
268
269
270
271     -- call this procedure
272     CALL student_info();
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Student_name	Subject_name

Result 4

- قم بانشاء viewباسم teacher_infoيحتوي على اسم المعلم و رقم المكتب و اسم المادة التي يتم تدريسها.

```
/* Create a view with the name teacher_info that contains the teacher's name,
 the office number, and the name of the subject being taught. */
CREATE VIEW teacher_info AS
SELECT Teacher_name, OfficeNum, Subject_name
FROM Teachers
join Courses on Courses.Subject_id= Teachers.subject_id
```

  - قم بعرضview

```
283     -- show the view
284     SELECT * FROM teacher_info;
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Teacher_name	OfficeNum	Subject_name

  - قم بحذفview

```
285     -- drop view
286     DROP VIEW teacher_info;
```

- قم بإنشاء indexللبحث باستخدام اسماء الطلاب ابجدياً.

```
-- Create an index to search using student names alphabetically
CREATE INDEX StudentIndex
ON Students (Student_name)
```

- قم بعرض index

```
289  -- show index
290  SHOW INDEX FROM Students
```

Result Grid | Filter Rows: | Export: | Wrap Cell Content:

Table	Non_unique	Key_name	Seq_in_index	Column_name	Collation	Cardinality	Sub_part	Packed	Null	Index_type	Comment	Index_comment	Visible	Expression
students	0	PRIMARY	1	Student_id	A	30	NULL	NULL		BTREE			YES	NULL
students	1	StudentIndex	1	Student_name	A	NULL	NULL	NULL	YES	BTREE			YES	NULL

- قم بحذف index

```
291  -- drop index
292  DROP INDEX StudentIndex  on Students
293
```

ملاحظة: علاقة many to many ينتج عنها جدول جديد.

Source Code

```
-- SQL3

/*
create a relationship between the table of teachers and students
(the teacher teaches more than one student,
and the student is taught by more than one teacher) */
CREATE TABLE stuRelTech (
student_id int not null,
teacher_id int not null,
 FOREIGN KEY (student_id) REFERENCES Students (Student_id),
 FOREIGN KEY (teacher_id) REFERENCES Teachers (Teacher_id),
 PRIMARY KEY (student_id,teacher_id)
  );

  /* create a relationship between the table of subjects and teachers
 (so that the teacher teaches only one subject,
 and the subject is taught by more than one teacher) */

 ALTER TABLE Teachers
 ADD COLUMN subject_id int,
```

```
 ADD FOREIGN KEY (subject_id) REFERENCES Courses(Subject_id);


 /*
create a relationship between the table of subjects and students
(So that the student studies more than one subject,
 and the subject is studied by more than one student).
*/
CREATE TABLE subjRelSTU (
subject_id int not null,
student_id int not null,
 FOREIGN KEY (subject_id) REFERENCES Courses (Subject_id),
 FOREIGN KEY (student_id) REFERENCES Students (Student_id),
 PRIMARY KEY (subject_id,student_id)
  );

  /* Create the procedure called student_info that displays the names o
f the students and the subjects
contain all the data shared between the subjects and students table */
delimiter //
CREATE PROCEDURE student_info()
BEGIN
SELECT Student_name, Subject_name
FROM subjRelSTU
JOIN Students ON Students.Student_id = subjRelSTU.student_id
JOIN Courses ON Courses.Subject_id = subjRelSTU.subject_id;
END
-- call this procedure
CALL student_info();

/* Create a view with the name teacher_info that contains the teacher's
 name,
 the office number, and the name of the subject being taught. */
 CREATE VIEW teacher_info AS
 SELECT Teacher_name, OfficeNum, Subject_name
 FROM Teachers
 join Courses on Courses.Subject_id= Teachers.subject_id

 -- show the view
 SELECT * FROM teacher_info;
 -- drop view
 DROP VIEW teacher_info;
 -- Create an index to search using student names alphabetically
 CREATE INDEX StudentIndex
 ON Students (Student_name)
 -- show index
 SHOW INDEX FROM Students
 -- drop index
 DROP INDEX StudentIndex  on Students
```