Gabriela Serrano Echenagucia

CSC 423 – Project 3: Implementation

GitHub: https://github.com/gmserranoe/CSC423.git

Translate the logical data model for the Oracle Enterprise DBMS. (12/09/21)

- a. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps.
- b. Create at least 5 tuples for each relation in your database.
- c. Develop 5 SQL queries using embedded SQL (see Python tutorial).
- d. Upload all the code and documentation to GitHub.

A. Develop SQL code to create the entire database schema, reflecting the constraints identified in previous steps

```
Department
```

```
CREATE TABLE Department(
  deptNo VARCHAR(5) NOT NULL CHECK (deptNo LIKE 'D%'),
  deptName VARCHAR(100) UNIQUE CHECK (deptName LIKE
'Department%'),
  chairName VARCHAR(100),
  faculty_count INT,
  PRIMARY KEY(deptNo));
Student
CREATE TABLE Student(
  studentID VARCHAR(5) NOT NULL CHECK (studentID LIKE 'R%'),
  studentName VARCHAR(100),
  student_initials VARCHAR(4) CHECK (student_initials LIKE '__%'),
  PRIMARY KEY(studentID) );
Major
CREATE TABLE Major(
  majorCode CHAR(3) NOT NULL CHECK (majorCode LIKE '____'),
  majorName VARCHAR(100),
  deptNo VARCHAR(5),
  PRIMARY KEY(majorCode),
  FOREIGN KEY(deptNo) REFERENCES Department(deptNo)
  ON DELETE CASCADE );
```

```
CREATE TABLE Event(
     eventNo VARCHAR(5) NOT NULL CHECK (eventNo LIKE 'EV%'),
     eventName VARCHAR(100),
     start_date date NOT NULL CHECK (start_date >= 2022-12-02),
     end_date date NOT NULL,
     CONSTRAINT chk_date CHECK (end_date > start_date),
     PRIMARY KEY(eventNo));
Department Event
CREATE TABLE DeptEvent(
     deptNo VARCHAR(5) NOT NULL CHECK (deptNo LIKE 'D%'),
     eventNo VARCHAR(5) NOT NULL CHECK (eventNo LIKE 'E%'),
     PRIMARY KEY(deptNo, eventNo),
     FOREIGN KEY(deptNo) REFERENCES Department(deptNo)
           ON DELETE CASCADE,
     FOREIGN KEY(eventNo) REFERENCES Event(eventNo)
           ON DELETE CASCADE );
Student Attendance
CREATE TABLE StudentAttendance(
  studentID VARCHAR(5) NOT NULL CHECK (studentID LIKE 'R%'),
  eventNo VARCHAR(5) NOT NULL CHECK (eventNo LIKE 'E%'),
  PRIMARY KEY(studentID, eventNo),
  FOREIGN KEY(studentID) REFERENCES Student(studentID)
     ON DELETE CASCADE,
```

FOREIGN KEY(eventNo) REFERENCES Event(eventNo) ON DELETE CASCADE);

Student Major

CREATE TABLE StudentMajor(

studentID VARCHAR(5) NOT NULL CHECK (studentID LIKE 'R%'),

majorCode CHAR(3) NOT NULL CHECK (majorCode LIKE '____'),

PRIMARY KEY(studentID, majorCode),

FOREIGN KEY(studentID) REFERENCES Student(studentID)

ON DELETE CASCADE,

FOREIGN KEY(majorCode) REFERENCES Major(majorCode)

ON DELETE CASCADE);

(Results in next page)

(Using DB Browser for SQLite to display results)

	Туре
Tables (7)	
Department	
<page-header> deptNo</page-header>	VARCHAR(5)
deptName	VARCHAR(100)
chairName	VARCHAR(100)
faculty_count	INT
DeptEvent	
눩 deptNo	VARCHAR(5)
eventNo	VARCHAR(5)
Event	
eventNo	VARCHAR(5)
eventName	VARCHAR(100)
start_date	date
end_date	date
Major	
majorCode	CHAR(3)
majorName	VARCHAR(100)
🔙 deptNo	VARCHAR(5)
Student	
studentID	VARCHAR(5)
studentName	VARCHAR(100)
student_initials	VARCHAR(4)
StudentAttendance	
studentID	VARCHAR(5)
eventNo	VARCHAR(5)
StudentMajor	
studentID	VARCHAR(5)
majorCode	CHAR(3)
	Department deptNo deptName chairName faculty_count DeptEvent deptNo eventNo eventNo eventName start_date end_date Major majorCode majorName deptNo Student StudentID studentAttendance studentID eventNo eventNo StudentID studentID studentID studentAttendance studentID studentID studentID studentAttendance studentID studentID studentID studentID studentAttendance studentID studentNo StudentID studentNo studentID

B. Create at least 5 tuples for each relation in your database.

Department

INSERT INTO Department

VALUES ('DEP01', 'Department of Business', 'James Morrison', 162);

INSERT INTO Department

VALUES ('DEP02', 'Department of Science', 'Lily Watson', 134);

INSERT INTO Department

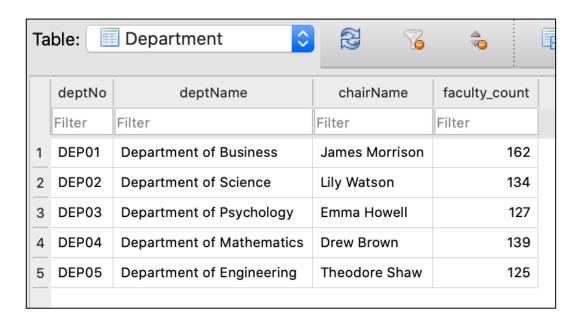
VALUES ('DEP03', 'Department of Psychology', 'Emma Howell', 127);

INSERT INTO Department

VALUES ('DEP04', 'Department of Mathematics', 'Drew Brown', 139);

INSERT INTO Department

VALUES ('DEP05', 'Department of Engineering', 'Theodore Shaw', 125);



INSERT INTO Student

VALUES ('R0001', 'Gabriela Serrano Echenagucia', 'GSE');

INSERT INTO Student

VALUES ('R0002', 'Andrea Smith', 'AS');

INSERT INTO Student

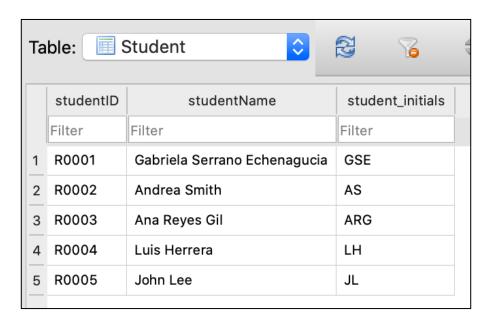
VALUES ('R0003', 'Ana Reyes Gil', 'ARG');

INSERT INTO Student

VALUES ('R0004', 'Luis Herrera', 'LH');

INSERT INTO Student

VALUES ('R0005', 'John Lee', 'JL');



INSERT INTO Major

VALUES ('BUS', 'Business', 'DEP01');

INSERT INTO Major

VALUES ('BIO', 'Biology', 'DEP02');

INSERT INTO Major

VALUES ('PSY', 'Psychology', 'DEP03');

INSERT INTO Major

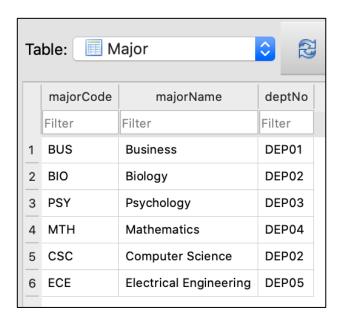
VALUES ('MTH', 'Mathematics', 'DEP04');

INSERT INTO Major

VALUES ('CSC', 'Computer Science', 'DEP02');

INSERT INTO Major

VALUES ('ECE', 'Electrical Engineering', 'DEP05');



INSERT INTO Event

VALUES ('EV000', 'Public Speaking 101', '2022-01-31', '2022-02-01');

INSERT INTO Event

VALUES ('EV001', 'Networking in Business', '2022-03-16', '2022-03-18');

INSERT INTO Event

VALUES ('EV002', 'Green Day', '2022-03-02', '2022-03-03');

INSERT INTO Event

VALUES ('EV003', 'Logic and Games', '2022-01-14', '2022-01-15');

INSERT INTO Event

VALUES ('EV004', 'Spring Hackathon', '2022-02-21', '2022-02-26');

Та	ble:	Event		6 ₽
	eventNo	eventName	start_date	end_date
	Filter	Filter	Filter	Filter
1	EV000	Public Speaking 101	2022-01-31	2022-02-01
2	EV001	Networking in Business	2022-03-16	2022-03-18
3	EV002	Green Day	2022-03-02	2022-03-03
4	EV003	Logic and Games	2022-01-14	2022-01-15
5	EV004	Spring Hackathon	2022-02-21	2022-02-26

INSERT INTO DeptEvent

VALUES ('DEP01', 'EV000');

INSERT INTO DeptEvent

VALUES ('DEP01', 'EV001');

INSERT INTO DeptEvent

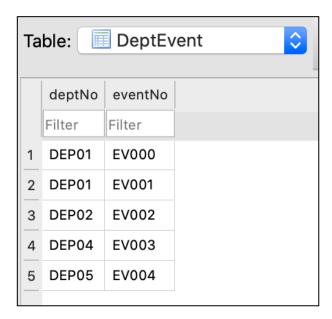
VALUES ('DEP02', 'EV002');

INSERT INTO DeptEvent

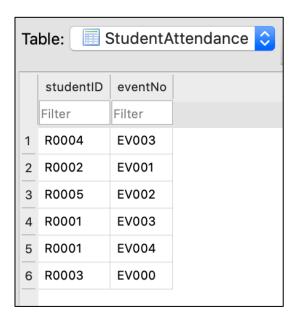
VALUES ('DEP04', 'EV003');

INSERT INTO DeptEvent

VALUES ('DEP05', 'EV004');



INSERT INTO StudentAttendance
VALUES ('R0004', 'EV003');
INSERT INTO StudentAttendance
VALUES ('R0002', 'EV001');
INSERT INTO StudentAttendance
VALUES ('R0005', 'EV002');
INSERT INTO StudentAttendance
VALUES ('R0001', 'EV003');
INSERT INTO StudentAttendance
VALUES ('R0001', 'EV004');
INSERT INTO StudentAttendance
VALUES ('R0001', 'EV0004');
INSERT INTO StudentAttendance
VALUES ('R0003', 'EV0000');



```
INSERT INTO StudentMajor

VALUES ('R0001', 'CSC');

INSERT INTO StudentMajor

VALUES ('R0002', 'BUS');

INSERT INTO StudentMajor

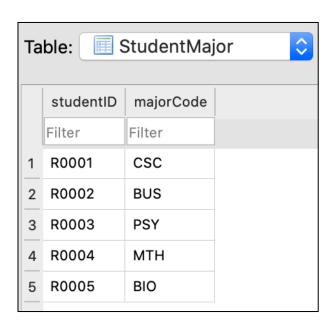
VALUES ('R0003', 'PSY');

INSERT INTO StudentMajor

VALUES ('R0004', 'MTH');

INSERT INTO StudentMajor

VALUES ('R0005', 'BIO');
```



C. Develop 5 SQL queries using embedded SQL (see Python tutorial).

List all events that take place in February

SELECT *

FROM Event

WHERE (start_date LIKE '____-02-__') OR (end_date LIKE '____-02-__');

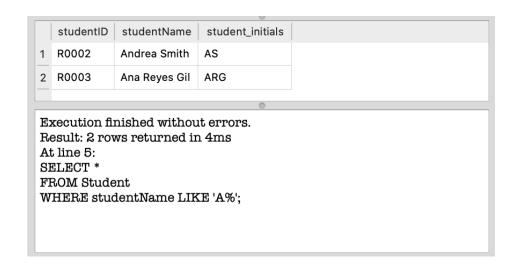
	0							
	eventNo	eventName	start_date	end_date				
1	EV000	Public Speaking 101	2022-01-31	2022-02-01				
2	EV004	Spring Hackathon	2022-02-21	2022-02-26				
R A	Execution finished without errors. Result: 2 rows returned in 14ms At line 1: SELECT * FROM Event WHERE start_date LIKE '02' OR end_date LIKE '02';							

List all students whose name starts with 'A'

SELECT *

FROM Student

WHERE studentName LIKE 'A%';



List all students who plan to attend the Spring Hackathon

SELECT s.studentID, s.studentName, e.eventNo, e.eventName, e.start_date, e.end_date

FROM StudentAttendance a, Student s, Event e

WHERE (a.studentID = s.studentID) AND (a.eventNo = e.eventNo)

AND (e.eventName = 'Spring Hackathon');

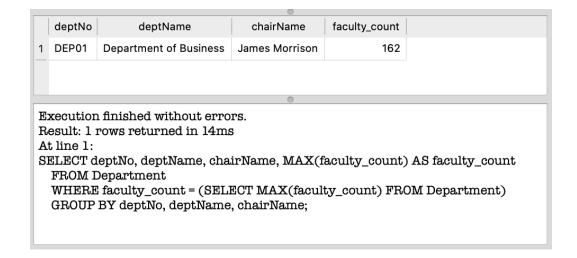
	studentID	studentName	eventNo	eventName	start_date	end_date
1	R0001	Gabriela Serrano Echenagucia	EV004	Spring Hackathon	2022-02-21	2022-02-26
A S F	t line 9: ELECT s.st ROM Stude	ws returned in 39ms sudentID, s.studentName, e. entAttendance a, Student s, studentID = s.studentID) AN kathon');	Event e	,	_ ,	_

List all details of the Department that has the most faculty

SELECT deptNo, deptName, chairName, MAX(faculty_count) AS faculty_count FROM Department

WHERE faculty_count = (SELECT MAX(faculty_count) FROM Department)
GROUP BY deptNo, deptName, chairName;

(Result in next page)



List all students whose major belongs to the Department of Science

SELECT s.studentID, s.studentName, m.majorCode, d.deptName

FROM Student s, Department d, Major m, StudentMajor sm

WHERE (s.studentID = sm.studentID) AND (m.majorCode = sm.majorCode)

AND (d.deptNo = m.deptNo) AND (d.deptName = 'Department of Science');

	studentID	studentName	majorCode	deptName	
1	R0001	Gabriela Serrano Echenagucia	CSC	Department of Science	
2	R0005	John Lee	BIO	Department of Science	
SI FI	ROM Stude HERE (s.s	udentID, s.studentName, m ent s, Department d, Major tudentID = sm.studentID) A m.deptNo) AND (d.deptNar	m, Studenti AND (m.maj	Major sm jorCode = sm.majorCod	de) AND

D. Upload all the code and documentation to GitHub.

GitHub: https://github.com/gmserranoe/CSC423.git