SQL Examples

CENG351

26.11.2021

Schema Specifications

Student (StudentId, Name, Address, Status, Gpa)

Professor (ProfessorId, Name, DeptID, Age)

Course (CourseId, DeptID, CourseName)

Transcript (StudentId, CourseId, Semester, Score)

Teaching (CourseId, Semester, ProfessorId)

StudentId	Name	Address	Status	Gpa	Age
1	Martin Prince	eskişehir yolu, ankara	Senior	2.5	22
2	Lisa Simpson	90. sk. eryaman , ankara	Junior	3.5	21
3	Milhouse Van Houten	ümitköy, ankara	Senior	2.7	21
4	Nelson Muntz	41. sk, eryaman, ankara	Senior	2	23
5	Ralph Wiggum	28. sk, eryaman, ankara	Freshman	2.5	19
6	Todd Flanders	yenimahalle, ankara	Sophomore	2.8	20
7	Shauna Chalmers	ümitköy, ankara	Sophomore	2.9	20

An instance of Student table

ProfessorId	Name	DeptId	Age
1	Waylon Smithers	CENG	35
2	Edna Krabappel	EE	42
3	Hans Moleman	CENG	63
4	Joe Quimby	CENG	45
5	Julius Hibbert	METE	50
6	Troy McClure	EE	35
7	Lenny Leonard	CENG	55
8	Kent Brockman	CENG	43

An instance of Professor table

CourseID	DeptId	CourseName
CENG100	CENG	Computer Engineering Orientation
CENG230	CENG	Introduction to C Programming
CENG223	CENG	Discrete Computational Structures
CENG491	CENG	Computer Engineering Design I
CENG492	CENG	Computer Engineering Design II
EE201	EE	Circuit Theory I
EE213	EE	Electrical Circuits Laboratory
METE201	METE	Materials Science I
METE202	METE	Materials Science II
ME202	ME	Manufacturing Technologies

An instance of Course table

StudentId	CourseId	Semester	Score
1	EE213	F2013	20
1	EE213	F2014	60
1	CENG230	S2014	65
2	CENG230	F2013	85
2	ME202	F2013	40
3	EE213	S2014	90
3	EE201	S2014	10
4	EE213	F2013	52
4	CENG230	F2015	55
5	ME202	F2013	49
5	CENG230	S2014	53
6	EE213	F2014	78
4	EE201	F2013	50
7	CENG100	F2013	0
7	CENG223	F2014	88

An instance of Transcript table

CourseId	Semester	ProfessorId
CENG100	F2013	1
CENG230	F2011	1
EE201	S2014	2
CENG230	F2015	2
CENG230	S2010	4
CENG100	F2010	4
CENG230	F2013	5
EE213	F2013	2
EE213	F2014	2
EE213	S2014	6
CENG100	F2012	7
CENG230	S2011	8
CENG100	F2014	8
CENG223	S2015	8

An instance of Teaching table

Q1. Delete all courses from Course table that have never been offered (that is, do not occur in the Transcript relation).

```
DELETE
FROM Course C
WHERE C.CourseID NOT IN (    SELECT T.CourseId
                             FROM Transcript T);
```

Q2. Modify the Transcript table by increasing the ”Score”s of each student whose status is ”Senior” by 10%.

```
UPDATE Transcript T
SET T.Score = T.Score * 1.1
WHERE T.StudentId IN (   SELECT S.StudentId
                         FROM Student
                         WHERE S.Status = ‘Senior’);
```

Q3. List the ProfessorId and the Age of the professors who taught all the courses that the professor named "Waylon Smithers" taught (i.e., on the same or different semesters). (List "ProfessorId" and "Age" ordered by Age)

```
SELECT P. ProfessorId, P. Age
FROM Professor P
WHERE NOT EXISTS (  SELECT T1.CourseId
                    FROM Teaching T1, Professor P1
                    WHERE T1.ProfessorId = P1.ProfessorId AND
                          P1.Name = 'Waylon Smithers')
                    EXCEPT
                  ( SELECT T2.CourseID
                    FROM Teaching T2
                    WHERE T2.ProfessorId = P. ProfessorId);
ORDER BY P.Age;
```

Q4. List the ProfessorId and the Age of the professors who taught all the courses that the professor named "Waylon Smithers" taught (i.e., on the same or different semesters). (List "ProfessorId" and "Age" ordered by Age)

Note: Do not include Waylon Smithers in the result

```
SELECT P. ProfessorId, P. Age
FROM Professor P
WHERE P.Name <> 'Waylon Smithers' AND
    NOT EXISTS ((SELECT T.CourseId
                FROM Teaching T, Professor P
                WHERE T.ProfessorId = P.ProfessorId AND P.Name = 'Waylon Smithers')
                    EXCEPT
                (SELECT T2.CourseId
                FROM Teaching T2
                WHERE T2.ProfessorId = P.ProfessorId))
```

```
ORDER BY P.Age;
```

Q5. List the average GPA of students who live in ”eryaman”(i.e., that includes ”eryaman” somewhere in the Address column).

```
SELECT AVG(S.GPA)
FROM Student S
WHERE S.Address LIKE ‘%eryaman%’;
```

Q6. List the average age of the students who enrolled to the course ME202 (enrolled in any semester).

```
SELECT AVG(S.Age)
FROM Student S, Transcript T
WHERE S.StudentId = T. StudentId AND T.CourseId = ‘ME202’;
```

Q7. List the age gap of the students who enrolled to the course ME202. (age gap: difference between ages of the oldest and the youngest student enrolled to that course in any year.)

```
SELECT AgeGap AS MAX(S.age) – MIN(S.Age)
FROM Student S, Transcript T
WHERE S.StudentId = T. StudentId AND T.CourseId = ‘ME202’;
```

Q8. List the StudentId and the name of the Students who enrolled to at least two courses.

```
SELECT S.StudentId, S.Name
FROM Student S, Transcript T1, Transcript T2
WHERE T1.StudentId = T2.StudentId AND T1.CourseId <> T2.CourseId AND
      S.StudentId = T1.StudentId;
```

Q9. List the StudentId and the name of the Students who enrolled to only a single course.

```
(  SELECT S.StudentId, S.Name
   FROM Student S, Transcript T
   WHERE S.StudentId = T.StudentId  )
   EXCEPT
(Solution of Q8);
```

Q10. List the ProfessorId and the name of the Professors who have never taught in Fall semester (i.e. Always taught in Spring semester)

```
SELECT P.ProfessorId, P.Name
FROM Professor P, Teaching T
WHERE P.ProfessorId = T.ProfessorId AND
      T.Semester = 'Spring' AND
      T.ProfessorId NOT IN (  SELECT T2.ProfessorId
                              FROM Teaching T2
                              WHERE T2.Semester = 'Fall');
```