SQL Language

Queries in SQL

Tania Cerquitelli

```
AUTHOR(AuthorCode, Name, Surname, Department, University)
ARTICLE(ArticleCode, Title, Topic)
AUTHORS_OF_ARTICLE(ArticleCode, AuthorCode)
EDITIONS_OF_CONFERENCE(Conference, Edition, EditionName, StartDate, EndDate, Editor)
AUTHOR_PRESENTS_ARTICLE(AuthorCode, Date, StartTime, EndTime, Room,
                                    ArticleCode, Conference, Edition)
```

Write the following query in SQL

(a) For the authors who have *exclusively* presented articles with topic 'Data Mining', show the code of the author, the surname of the author, her/his university, and the total number of articles presented by the author in each edition of every conference.

```
AUTHOR(AuthorCode, Name, Surname, Department, University)
ARTICLE(ArticleCode, Title, Topic)
AUTHORS_OF_ARTICLE(ArticleCode, AuthorCode)
EDITIONS_OF_CONFERENCE(Conference, Edition, EditionName, StartDate, EndDate, Editor)
AUTHOR_PRESENTS_ARTICLE(AuthorCode, Date, StartTime, EndTime, Room,
                                     ArticleCode, Conference, Edition)
```

Write the following queries in SQL

(a) Considering the conferences with at least 10 editions, for each edition of the conference show the name of the edition and the code of the author who presented the highest number of articles in the edition.

The following relations are given (primary keys are underlined):

```
STUDENT(StudentID, Name, Surname, DegreeProgramme)
ASSIGNMENT_TO_BE_DELIVERED(ACode, Title, Topic, ScheduledExpirationDate)
TEACHER(TeacherID, Name, Surname, Department)
EVALUATION_OF_DELIVERED_ASSIGNMENT(StudentID, ACode, TeacherID,
                                    DeliveryDate, EvaluationDate, Score)
```

Write the following query in SQL

(a) For each student who has delivered at least 3 assignments with score greater than 4, show the surname of the student, the total number of assignments delivered by the student, the average score of all delivered assignments, and the number of different teachers who evaluated their delivered assignments.

```
STUDENT(StudentID, Name, Surname, DegreeProgramme)
ASSIGNMENT_TO_BE_DELIVERED(ACode, Title, Topic, ScheduledExpirationDate)
TEACHER(TeacherID, Name, Surname, Department)
EVALUATION_OF_DELIVERED_ASSIGNMENT(StudentID, ACode, TeacherID,
                                   DeliveryDate, EvaluationDate, Score)
```

Write the following query in SQL

(a) Show the identifier, surname and degree programme of the students who have *never* delivered an assignment after the scheduled expiration date, and who have delivered *all* the assignments due always getting the highest score.

```
SEMINAR(SCode, STitle, Topic, Duration)
SPEAKER(S-SSN, SName, BirthDate)
SEMINAR-CALENDAR(SCode, Date, StartTime, S-SSN, Room)
EXPERTISE(S-SSN, Topic)
```

Write the following query in SQL

(a) Show the code of the seminars for which at least one scheduled presentation was held by the speaker with the highest number of topics of expertise.

The following relations are given (primary keys are underlined):

```
TEACHER(TCode, TName, TSurname, Department, ResearchGroupName, ResearchArea)
COURSE(CCode, CName, EnrollingStudent#, TCode, Topic)
CLASSROOM(RoomID, Floor#, Video_Kit, Seat#)
LECTURE(RoomID, Date, StartHour, EndHour, CCode, AttendingStudent#)
Video_Kit = {yes, no}
```

Write the following query in SQL

(a) For each teacher who has taught *exclusively* courses whose topic is databases, select the code of the teacher and, among her courses, the code of the course for which the average number of students attending the course lectures is the highest.