
Homework 3: SQL queries

1. The following relations are given (primary keys are underlined; optional attributes are denoted with *):

AUTHOR (CodAuthor, Name, Surname, Department, University)

ARTICLE (CodArticle, Title, Topic)

ARTICLE_AUTHORS (CodArticle, CodAuthor)

CONFERENCE_EDITIONS (Conference, Edition, EditionName, StartDate, EndDate, Editor)

AUTHOR_PRESENTS_ARTICLE (CodAuthor, Date, StartHour, EndHour, Room, CodArticle, Conference, Edition)

Write the following query in SQL language:

- For each author who has presented at least 4 articles on the topic 'Data Mining', but who has never submitted articles on the topic 'Deep Learning' or 'CyberSecurity', show the name, surname, university of the author and the total number of articles presented by the author in each edition of each conference.

2. The following relations are given (primary keys are underlined; optional attributes are denoted with *):

STUDENT (StudentID, Name, Surname, Degree Course)

HOMEWORK_TO_SUBMIT (HWCode, Title, Topic, ExpectedDueDate)

PROFESSOR (PCode, Name, Surname, Department)

SUBMITTED_HOMEWORK_EVALUATION (StudentID, HWCode, PCode, SubmissionDate, EvaluationDate, Evaluation)

Write the following query in SQL language:

- For each student, show the student's first and last name, and the title and subject of each homework delivered by the student, for which the student has obtained a higher rating than the average rating achieved on that homework by all students.

3. The following relations are given (primary keys are underlined; optional attributes are denoted with *):

STUDENT (StudentID, Name, Surname, Degree Course)

HOMEWORK_TO_SUBMIT (HWCode, Title, Topic, ExpectedDueDate)

PROFESSOR (PCode, Name, Surname, Department)

SUBMITTED_HOMEWORK_EVALUATION (StudentID, HWCode, PCode, SubmissionDate, EvaluationDate, Evaluation)

Write the following query in SQL language:

- For each student who has delivered *all* the homeworks of topic 'databases', and always *before* the expected due date ($\text{SubmissionDate} < \text{ExpectedDueDate}$), view the student's surname and, with regard to the submitted homeworks of the topic 'databases', the average evaluation received, the total number of different teachers who carried out the evaluations, and the average number of days in which the student submitted the homeworks in advance of the expected due date ($\text{ExpectedDueDate} - \text{SubmissionDate}$).

4. The following relations are given (primary keys are underlined; optional attributes are denoted with *):

CLIENT (TaxCode, NameC, Surname, BirthDate, City)
 RESTAURANT (IdS, NameR, Address, City, Cuisine)
 ORDER (CodO, TaxCode, Date, Time, Price, IdS)
 EMPLOYEE (EID, NameE, Surname, Qualification)
 WORKS_IN (EID, Date, IdS)

Write the following query in SQL language:

- For each restaurant that received the most orders from restaurants in its city, view the name of the restaurant and the total revenue earned by the restaurant on each date.

5. Trigger

The following relations are given (primary keys are underlined; optional attributes are denoted with *):

HOTEL (CodH, Name, Category, Address, City)
 HOTEL_REVIEW (CodR, ReviewDate, CodH, Score, Comment)
 REVIEWS_SUMMARY (CodH, ReviewsNumber, TotalScore)
 REVIEWS_OUTCOME_NOTIFICATION (CodH, ReviewDate, HotelTotalScore, CategoryAvgScore)

Write the trigger to manage the submitted hotel reviews collected through a web portal.

The HOTEL table contains the list of hotels for which you can submit a review. The table REVIEWS_SUMMARY contains, for each hotel, the *total number* of reviews received, and the *overall score* assigned to each hotel through the reviews. Consider that a hotel is listed in the REVIEWS_SUMMARY table only if at least one review has been entered for that hotel.

A new review for a hotel is submitted through the portal (insertion of a record in the table HOTEL_REVIEW). You must update the table REVIEWS_SUMMARY considering the review just submitted. Also consider the case that this is the first review entered for the hotel.

You must then enter a new record in the table `REVIEWS_OUTCOME_NOTIFICATION` with information on the *overall score assigned to the hotel* (`HotelTotalScore` attribute). The *average score assigned per category* (`CategoryAvgScore` attribute) calculated by considering the overall score assigned to all hotels in the same category of the hotel that received the review must also be notified.

Indications for the performance of the exercise:

You are asked to write the trigger to manage hotel reviews in the manner described above.

If necessary, use the function `raise_application_error (...)` to report an error. You are not required to specify parameters passed to the function.