Domanda 1

Punteggio max.: 1,00

1. Given the following relational schema (primary keys are underlined, optional attributes are indicated with *):

BOOK (BookCode, Title, Genre)
AUTHOR (AuthorCode, Name, Surname, BirthDate)
AUTHORS_BOOK (BookCode, AuthorCode)
PUBLISHER (PublisherCode, Name, City)
PUBLICATION (BookCode, PublisherCode, PublicationYear, NumberCopies)

Express the following query in SQL language:

- For each author who has published at least 3 books in the 'Science Fiction' genre, but who has never published any 'Horror' or 'Mystery' boos, display the author's name and surname and, for each published book, the title , the genre, and the total number of copies published of that book.

Domanda 2

Punteggio max.: 1,00

2. Given the following relational schema (primary keys are underlined, optional attributes are indicated with *):

EMPLOYEE (EmployeeNumber, Name, Surname, Department)
PROJECT (ProjectCode, Title, Description, ExpectedDeadline)
EVALUATION (EmployeeNumber, ProjectCode, DeliveryDate, AssessmentDate, AssessmentScore)

Express the following query in SQL language

- For each employee, view the employee's first and last name, and the title and description of each project for which the employee received an assessment score higher than the average score received by employees who worked on that project.

Domanda 3

Punteggio max.: 1,00

3. Given the following relational schema (primary keys are underlined, optional attributes are indicated with *):

CUSTOMER (CustomerID, FirstName, LastName, City)
STORE (StoreID, StoreName, Address, City)
ORDER (StoreID, CustomerID, Date, Amount)

Express the following query in SQL language

- For each store that received the highest number of orders among stores in the same city, display the store name, the city of the store, and for each order received, the first name and last name of the customer who placed the order, the order date, and the order amount.

Domanda 4

Punteggio max.: 1,00

4. Given the following relational schema (primary keys are underlined, optional attributes are indicated with *):

SERVICE_BOOKING (BookingID, LicensePlate, Model, Brand, RequestDate)
BOOKING_AVAILABILITY (Date, OpeningTime, ClosingTime, AvailableSlots)
REQUESTED_SERVICES (LicensePlate, ServiceType)
SERVICE_COSTS (ServiceType, Brand, Model, Cost)
SERVICE_NOTIFICATION (BookingID, LicensePlate, ServiceDate, OpeningTime, ClosingTime, TotalCost)

Write a trigger to manage the bookings for vehicle service at a specialized workshop.

A booking request for vehicle service is inserted (a record is inserted into the SERVICE_BOOKING table), specifying the type of vehicle (brand and model) and the date when the service is requested (attribute RequestDate). The trigger must perform the following operations:

Select the first date (on or after the requested date) where there are still available slots. The BOOKING_AVAILABILITY table indicates, for each date, the opening and closing times of the workshop, and the number of slots still available for vehicle service on that date.

If there are no available slots on any date, the booking request must be canceled. Otherwise, the following operations must be performed:

- Update the number of available slots on the selected date.
- Calculate the total cost of the service. The REQUESTED_SERVICES table stores the list of service types requested for the vehicle booked for service. The SERVICE_COSTS table stores the cost for each service type based on the brand and model of the vehicle.
- Notify the successful booking by inserting a new record into the SERVICE_NOTIFICATION table, providing details about the selected date and the opening and closing times on that date, and the total cost for the service.

Instructions for completing the task:

Write the trigger to manage hotel reviews according to the above specifications.

If necessary, use the *raise_application_error (...)* function to signal an error. There is no need to specify the parameters passed to the function.