# INTRODUCTION TO DATABASES

# WRITTEN TEST – June 24th, 2020

# Draft solution

ALGEBRA EXERCISES - POINTS: 4

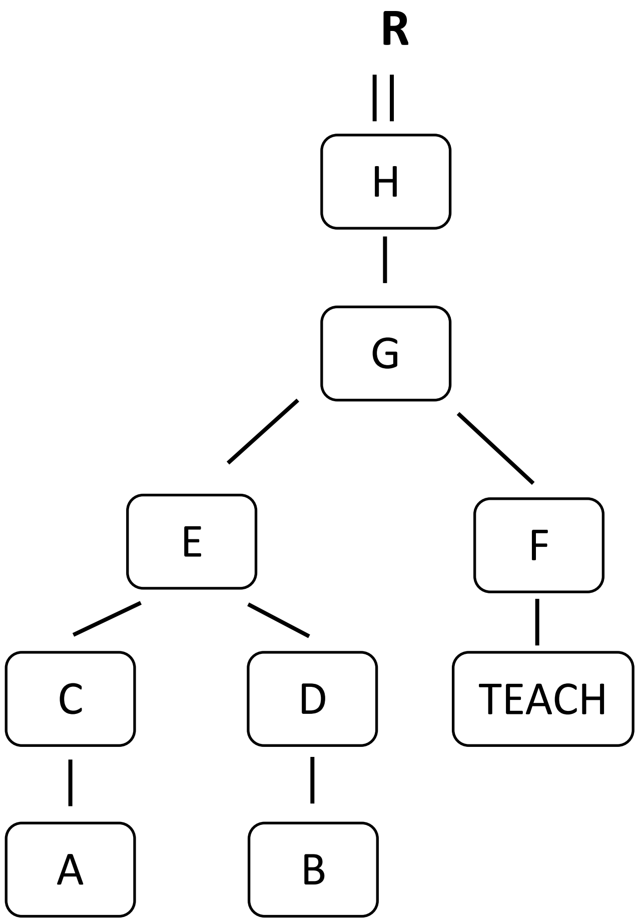
**Given the following relational tables:**

TEACHER (TeacherID, FirstName, LastName, Department)
COURSE (CourseID, CourseName, Year, Semester, TeacherID, Language)

Select the first and last names of the teachers of the Electronics department who teach at least two English-language courses in the first semester of the second year.

**Assignment for the exercise**:

The following query tree graphically represents the requested algebraic query. You are requested to indicate, for each box in the query tree (i.e., A, B, C, D, E, F, G, and H box), the relational table or the corresponding algebraic operator. Use the text box below to provide your solution. Note: each box in the query tree is associated with only one relational table or one algebraic operator.



Right solution

A: COURSE C1

B: COURSE C2

C: SELECTION C1.Year = 2 AND C1.Semester=1 AND Language = 'English'

D: SELECTION C2.Year=2 AND C2.Semester=1 AND Language='English'

E: THETA JOIN ON C1.TeacherId = C2.TeacherId AND C1.CourseId <>C2.CourseId

F: SELECTION T.Department = 'Electronics'

G: THETA JOIN ON E.teacherId = F.teacherId

H: PROJECTION G.FirstName, G.lastName

EXERCISES IN SQL

**EXERCISE No. 1 points 3**

**Given the following relational tables**

TEACHER (TeacherID, FirstName, LastName, Department)
COURSE (CourseID, CourseName, Year, Semester, MainTeacherID, Language)
STUDENT (StudentID, FirstName, LastName, BirthDate, Nationality)
STUDENT-COURSE-ENROLLMENT (CourseID, StudentID, EnrollmentYear)
STREAMING-VIDEOLECTURE (VideoLectureID, CourseID, Topic,
Date, DurationInMinutes, TeacherID)
STREAMING-VIDEOLECTURE-ATTENDANCE (StudentID, VideoLectureID, CourseID,
AttendanceInMinutes)

**Write the following query in SQL language:**

Select the student ID, first name, last name, and nationality of each student who has never attended a streaming video-lecture longer than 10% of the video-lecture duration.

**Assignment for the exercise**: use the text box below to provide your solution.

```
SELECT StudentID, FirstName, LastName, Nationality
FROM STUDENT S
WHERE StudentID NOT IN
        (SELECT StudentID
        FROM STREAMING-VIDEOLECTURE SV,
        STREAMING-VIDEOLECTURE-ATTENDANCE SVA
        WHERE VS.VideoLectureID = PVS.VideoLectureID AND
                VS.CourseID = PVS.CourseID
        AND AttendanceInMinutes > 10% * DurationInMinutes)
```

**EXERCISE 2 - POINTS 3**

**Given the following relational tables**

TEACHER (TeacherID, FirstName, LastName, Department)
COURSE (CourseID, CourseName, Year, Semester, TeacherID, Language)
STUDENT (StudentID, FirstName, LastName, BirthDate, Nationality)
STUDENT-COURSE-ENROLLMENT (CourseID, StudentID, EnrollmentYear)
STREAMING-VIDEOLECTURE (VideoLectureID, CourseID, Topic,
Date, DurationInMinutes, TeacherID)
STREAMING-VIDEOLECTURE-ATTENDANCE (StudentID, VideoLectureID, CourseID,
AttendanceInMinutes)

**Write the following query in SQL language:**

For each teacher of the Department of "Control and Computer Engineering" who teaches at least three courses, select the teacher’s first and last name, and the total number of the teacher’s streaming video-lectures.

**Assignment for the exercise**: use the text box below to provide your solution.

```
SELECT D.TeacherID, FirstName, LastName, COUNT(*)
FROM TEACHER T, STREAMING-VIDEOLECTURE SV
WHERE T.TeacherID = SV.TeacherID
AND D.TeacherID IN
                (SELECT TeacherID
                FROM COURSE C
                GROUP BY TeacherID
                HAVING COUNT(*)>=3)
AND Department = 'Control and Computer Engineering'
GROUP BY D.TeacherID, FirstName, LastName
```

EXERCISE 3 POINTS 5

**Given the following relational tables**

TEACHER (TeacherID, FirstName, LastName, Department)
COURSE (CourseID, CourseName, Year, Semester, TeacherID, Language)
STUDENT (StudentID, FirstName, LastName, BirthDate, Nationality)
STUDENT-COURSE-ENROLLMENT (CourseID, StudentID, EnrollmentYear)
STREAMING-VIDEOLECTURE (VideoLectureID, CourseID, Topic, Date, DurationInMinutes, TeacherID)
STREAMING-VIDEOLECTURE-ATTENDANCE (StudentID, VideoLectureID, CourseID, AttendanceInMinutes)

**Write the following query in SQL language:**

For each student enrolled in at least 3 courses in the 2019-2020 enrollment year, select the student ID and last name, and the ID of each course for which the student attended all streaming video-lectures of that course.

**Assignment for the exercise**: use the text box below to provide your solution.

```
SELECT PVS.StudentID, LastName, SVA.CourseID
FROM STUDENT S, STREAMING-VIDEOLECTURE-ATTENDANCE SVA
WHERE S.StudentID = SVA.StudentID
AND S.StudentID in
        (SELECT StudentID
        FROM STUDENT-COURSE-ENROLLMENT
        WHERE EnrollmentYear='2019-2020'
        GROUP BY StudentID
        HAVING COUNT(*) >= 3)
GROUP BY SVA.StudentID, LastName, SVA.CourseID
HAVING COUNT(*) =
                (SELECT COUNT(*)
                FROM STREAMING-VIDEOLECTURE SV
                WHERE SV.CourseID = SVA.CourseID)
```

**ER1 Immuni – POINTS 2**

**Describe the Entity-Relationship diagram corresponding to the following specifications.**

*You are requested to design the database for managing the users of a COVID-19 infection-tracking mobile app.*

Users of the app are characterized by their SSN (Social Security Number), first name, last name, date of birth, and place of residence. Users are divided into occasional users and registered users.

For registered users, their Local Health Agency and their registration date are known, whereas for occasional users, the brand and type of their mobile device are known. The Local Health Agency is identified by its reference city and a code (e.g. *Turin TO1*) and is characterized by an email address, a venue, and a list of phone numbers.

**Assignment for the exercise**: use the text box below to provide your ER diagram in textual form. Alternatively, you can use the drawing box to graphically represent the ER diagram.

**USER ENTITY**
Id: SSN
Attributes: first name, last name, date_of_birth, place of residence

GENERALIZATION (t,e) ON USER [FATHER ENTITY]
children entities: REGISTERED, OCCASIONAL

REGISTERED ENTITY
Attributes: registration date (It can be also defined as a property of the Binary Relationship Local Health Agency – Registered Users)

OCCASIONAL ENTITY
Attributes: mobile phone brand, mobile phone type

Local Health Agency ENTITY
Id: ref. city, code
Attributes: email, venue, phone numbers (1-N)

BINARY **Relationship:** Local Health Agency – Registered
REGISTERED (1,1)
Local Health Agency (0,N)

**ER2 Immuni - POINTS 2**

**Describe the Entity-Relationship diagram corresponding to the following specifications.**

*You are requested to design the database for managing alert notifications of a COVID-19 infection-tracking mobile app.*

App users are identified by a code and are characterized by their first and last name. If the app incurs an issue or the Bluetooth protocol is turned off, the app notifies an alert to the user. Each alert notification is identified by a unique code and is characterized by a predefined message text (e.g. "*The Bluetooth connection is turned off. Restore it as soon as possible.*").

For each user, record the date and time of receipt of the alert notifications. Assume that the same alert notification can be sent multiple times to the same user (i.e., several times on the same date with different times or on different dates) or to different users (on the same date and time, or on different dates and/or times).

**Assignment for the exercise**: use the text box below to provide your ER diagram in textual form. Alternatively, you can use the drawing box to graphically represent the ER diagram.

**USER ENTITY**
Id: UserID
Attributes: first name, last name

Alert ENTITY
ID: AlertID
Attributes: Message text

TIME ENTITY
ID: date, time

Alert notification TERNARY relationship
ALERT (0,N)
USER (0,N)
TIME (1,N)

ER3 Immuni - POINTS 3

**Describe the Entity-Relationship diagram corresponding to the following specifications.**

*You are requested to design the database for managing alert notifications of a COVID-19 infection-tracking mobile app and the doctor visits booked by users.*

The users of the app are identified by a code and are characterized by their first and last name. Users receive notifications of potential exposure to infection. Each notification is identified by a progressive number, unique for each user, and is characterized by a description and a specific risk level.

Users can book medical visits to doctors in the area. Each doctor, identified by the VAT registration number, has a first name, last name, and number of years of experience. For each booking of a medical examination, you are requested to record the date, the requesting user, the doctor in charge of performing the examination, and the place where the examination will be held. Assume that a user can book multiple visits on different dates with the same doctor or with different doctors, but on the same date a user can book at most one examination. Assume also that a doctor can have multiple examinations on the same date.

**Assignment for the exercise**: use the text box below to provide your ER diagram in textual form. Alternatively, you can use the drawing box to graphically represent the ER diagram.

**USER ENTITY**
Id: UserID
Attributes: first name, last name

**NOTIFICATION ENTITY**
Internal Id: Number
External Id: USER entity id
Attributes: description, risk level

**BINARY RELATIONSHIP: USER-NOTIFICATION**
USER (0,N)
NOTIFICATION (1,1)

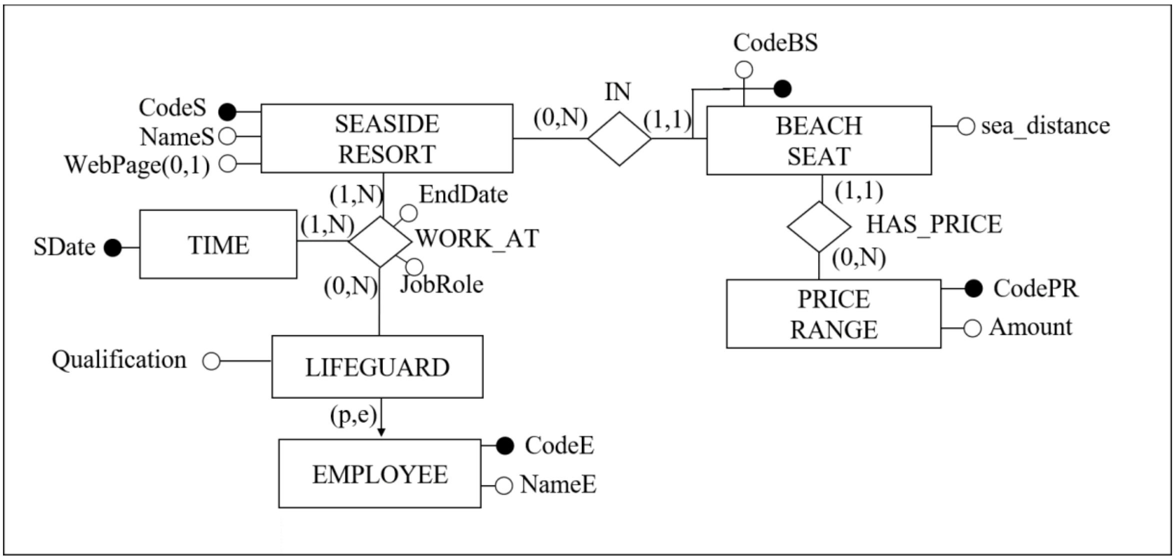
**DOCTOR ENTITY**
Id: VATnumber
Attributes: First name, last name, ActivityStartDate

**EXAMINATION BOOKING ENTITY**
External Id: USER id
Internal Id: Date
Attributes: Place

**BINARY RELATIONSHIP: USER-EXAMINATION BOOKING**
USER (0,N)
EXAMINATION (1,1)

**BINARY RELATIONSHIP: DOCTOR-EXAMINATION BOOKING**
EXAMINATION (1,1)
DOCTOR (0,N)

Given the following Entity-Relationship diagram



You are required to:

1) Provide a normalized relational logical schema for the same database (N.B. It is not mandatory to report possible changes to the E-R diagram)

2) Define referential integrity constraints for the 2 relationships of the conceptual schema.

Assignment: Use the text box below to provide your solution.

Normalized relational logical schema
EMPLOYEE( CodeE, NameE, Qualification*, Type)
SEASIDE_RESORT(CodeS, NameS, WebPage*)
TIME (SDate)
WORK_AT (CodeE, CodeS, SDate, EndDate, JobRole)
BEACH_SEAT(CodeS, CodeBS, sea_distance, CodePR)
PRICE_RANGE(CodPR, Amount)

Referential integrity constraints:
In the CREATE TABLE BEACH_SEAT(CodeS) REFERENCES SEASIDE_RESORT(CodeS)

In the CREATE TABLE WORK_AT
(CodeE) REFERENCES EMPLOYEE(CodeE)
(CodeS) REFERENCES SEASIDE_RESORT(CodeS)

# TRIGGER 1: POINTS 2

**The following relational schema is given (primary keys are underlined):**
STUDENT (StudentID, Name, Surname, MasterDegree)
COURSE (CourseID, Name, NumOfCredits)
EXAM (StudentID, CourseID, Date, Mark)
STUDENT_CAREER_INFO_NOTIFICATION (StudentID, RequestDate, NumOfPassedExams)
STUDENT_CAREER_REQUEST (RequestID, StudentID, RequestDate)

A student requests her/his career information (i.e., insertion of a record in the table STUDENT_CAREER_REQUEST). Write the trigger to address the following activities.

The number of exams passed by the student must be calculated. The EXAM table contains the list of exams taken by students. Consider that an exam is passed if the mark is greater than or equal to 18.

If the student has not passed any exam, the trigger ends with an error. Otherwise, a new record must be inserted in the STUDENT_CAREER_INFO_NOTIFICATION table with the computed information.

**Indications for carrying out the exercise**:

*Given the following incomplete solution of the trigger, you are asked to complete **Part A** in bold by specifying the body of the trigger. Use the text box below to provide your solution.*

*Use the raise_application_error (....) function to raise an error. It is not required to specify the parameters passed to the function.*

```
create or replace trigger StudentCareerInfo
after insert on STUDENT_CAREER_REQUEST
for each row
```

**Part A**

---------------------------------- ------------------------------------------------ ----------------------------------------

**Solution -** Block A:

```
declare
X number ;
begin
 --- calculate the number of exams passed by the student
select count (*) INTO X
from EXAMS
where StudentID =: NEW.StudentID AND Mark >= 18;

if (X = 0) then
    raise_application_error (....);
else
---- insert the calculated values
INSERT INTO STUDENT_CAREER_INFO_NOTIFICATION (StudentID, RequestDate,
NumOfPassedExams)
values (: New.StudentID,: NEW.RequestDate, X);
END;
```