LaSalle College
Montréal

Project 1: 15%

Course Identification

Name of program – Code:	
Course title:	**DATABASE II**
Course number:	420-BD2-AS
Group:	07438
Teacher’s name:	Jean-François Parent
Duration:	Extended
Semester:	Fall 2023

Student Identification

Name: _____ Student number: _____

Date: _____ Result: _____

☐ I declare that this is an original work, and that I credited all content sources of which I am not the author (online and printed, images, graphics, films, etc.), in the required quotation and citation style for this work.

Standard of the Evaluated Competency
Statement of the evaluated competency – Code
Use a database management system – 00Q7
Evaluated elements of the competency 00Q7
1. Create the database. *2. Formulate read requests, insertion requests, modification requests and deletion requests.* *3. Ensure data confidentiality and consistency.* *4. Program automated data processing operations.* *5. Save and restore the database.*

Instructions

- Create a database (schema) according to the requirements
- It is the teacher's responsibility to identify language errors. If such errors are found, teachers may apply a penalty of up to 20% of the grade (IPEL – Article 5.7).
- Plagiarism, attempts at plagiarism or complicity in plagiarism during a summative evaluation results in a mark of zero (0). In the case of recidivism, in the same course or in another course, the student will be given a grade of '0' for the course in question. (IPEL – Article 5.16).
- Submit dates on Omnivox must be respected.

- Please consult the correction grid at the end of document for more details.

TOTAL: 100 POINTS

Because this work must be done at home, if we think that the answers are not yours, the department reserves the right to complete your evaluation with a virtual meeting to verify that you attained the required competencies.

Project 1

Directives

Note: This is **NOT** a team work

You must complete this project by implementing all the good working methods shown in class:

1. For objects names, only use the recommended **characters** (A-Z (objects names must begin by a letter), 0-9 and the underscore (_).

2. Each field must have the correct **data type**. The size must be big enough to store all required data, but not too big to use too much space on the disk.

3. Unless specified otherwise, each field must contain data and must **not** accept **NULL** values and/or **empty strings** (").

4. Specify a **default value** where you can do it.

5. Specify **constraints** where you can do it to prevent users to enter unwanted data.

6. Each table must have a **primary key** which identifies each row with a unique value (GUID).

7. You must create **indexes** for each field used in a search, for each foreign key, and for each column which must be unique.

8. Each SQL query must use good **indentation** (for example, no query on a single line).

9. Write comments to describe what the SQL code does.

10. For each table and object, you must write the **revision history**. It must contains the **date**, your **name**, your **student number** and the **modifications** made. For tables, enter this information in the comments of the primary key. For the other objects (stored procedures, functions, etc.) you must inscribe these informations in the comments. For example:

Table:

```
2023-10-29    Bill Torvalds (1244556)    Created the customers table.
2023-10-31    Bill Torvalds (1244556)    Added the column firstname.
```

Stored procedure:

```
--Revision history:
--2023-11-01    Bill Torvalds (1244556)    Created the procedure customers_select.
--2023-11-03    Bill Torvalds (1244556)    Fixed the ORDER BY bug.
```

Project description

For this project, you must create a database to manage a school. Create a database (schema) named **SCHOOL_LASTNAME_STUDENTNUMBER**. Replace "LASTNAME" with your own last name and "STUDENTNUMBER" with your own student number, for example, SCHOOL_TORVALDS_1244556.

(00Q7.1 / 00Q7.4 / 00Q7.4) (41 pt)
Create tables to manage the following elements, and enter data (at least 3 rows) into each of these tables:

- **Teachers**
  This table must contain the name, firstname, email and salary (hourly rate).

- **Courses**
  This table must contain the name of the course (maximum 25 characters, for example "geography"), the code of the course (must be unique, maximum 10 characters, for example "420-BD2-AS"). There must also be a column to enter the course description (maximum 5000 characters).

- **Students**
  This table must contain the name, firstname, address, city and postal code for each student. It must also contain the student number (maximum 9 characters). In this table, create a row that contains YOUR name, firstname and student number (it is not necessary to enter your real address).

- **Reports**
  This table must use 2 foreign keys to identify the student and the teacher who gave the course. It must also contain a column result (between 0 and 100) and a comments column which is optional (maximum 250 characters).

(00Q7.2 / 00Q7.3 / 00Q7.4) (47 pt)
To acces the data, create stored procedures and/or functions to perform the following operations on the students and teachers tables:

- INSERT
- UPDATE
- SELECT (all)
- SELECT (only one row with the primary key)
- DELETE (only one row)

(00Q7.3) (00Q7.4) (7 pt)
In each of the 4 tables, create 2 fields to know the date and time for the row creation and modification. Perform all the required steps to make sure these fields are handled automatically by the system.

(00Q7.5) (5pt)
When your database is ready, export it into a text file (.sql). The filename must contain at least your lastname and the date when the backup was made.

This exported file only creates the objects and not the database. You must thus add manually the SQL code to create and use a database (schema) which has the same name than your database. Use the password “123”. The script must create all the objects (database/schema, tables, procedures, etc.) so test it properly.

Your script must NOT delete any existing database/schema if it already contains object(s).

Your script must be error-free.

Submit this unique file on Omnivox.

CORRECTION GRID FOR REQUIREMENTS

Competency : Use a database management system – 00Q7	
Elements of the competency: Create the database. (00Q7.1)	
Performance criteria	**weight**
Accurate analysis of the data model (00Q7.1.1)	/12.5
Accurate analysis of the specifications of the database management system. (00Q7.1.2)	/11.5
Appropriate coding of the instructions for creating the database. (00Q7.1.3)	/10.5
Elements of the competency: Formulate read requests, insertion requests, modification requests and deletion requests. (00Q7.2)	
Performance criteria	**weight**
Accurate identification of the types of requests to be formulated. (00Q7.2.1)	/6
Appropriate use of clauses, operators, commands and parameters. (00Q7.2.2)	/6
Proper performance of requests. (00Q7.2.4)	/6
Elements of the competency: Ensure data confidentiality and consistency. (00Q7.3)	
Performance criteria	**weight**
Accurate identification of the techniques to be used. (00Q7.3.1)	/13.5
Appropriate use of referential integrity constraints, triggers and transactions. (00Q7.3.4)	/6
Elements of the competency: Program automated data processing operations. (00Q7.4)	
Performance criteria	**weight**
Accurate identification of data processing operations to be automated. (00Q7.4.1)	/6
Appropriate creation of stored procedures and scripts. (00Q7.4.2)	/10
Clear record of programming support documentation. (00Q7.4.3)	/7
Elements of the competency: Save and restore the database. (00Q7.5)	
Performance criteria	**weight**
Astute choice of techniques to be used for saving and restoring. (00Q7.5.1)	/2
Appropriate use of techniques for saving and restoring the database. (00Q7.5.2)	/2
Compliance with the procedure and frequency for saving the database. (00Q7.5.3)	/1