

CP363: Databases I

Assignment #1

CP363, WLU, 2022

Instructor: Syed Nasir Daniel

Due: Tuesday February 1st, 2022

Nishant Tewari

190684430

tewa4430@mylaurier.ca

Q1:

Student Table

User Category	Application Type	Interface Type
→ System Administrator Ex. University Admissions Office	This application would be used to insert data of undergraduate students who are attending the university	Form-based Interface → Users will be easily able to insert new data and fill out specific entries from the students.

Course Table

User Category	Application Type	Interface Type
→ Casual/Temporary Users Ex. Students	This application is used by students to primarily view the courses they enrolled in for the semester. This could also be used as a part of enrollment system for students to enroll into specific classes and have it entered into the database.	Menu-Based Interface → Presented with a list of options such as menus to implement simple commands to enroll for classes

Section Table

User Category	Application Type	Interface Type
→ Database Administrator Ex. University Registration Office	This application would be used by the registration office to assign students to various sections of classes, check on their status within the course. This table can also be implemented by adding entries for grades and another section to view students that have dropped or added other courses to their semester.	Menu-Based Interface → Presented with a list of options such as menus to implement simple commands, assign students their sections, semester they are taking the course, year and instructor who will be teaching.

Grade_Report Table

User Category	Application Type	Interface Type
→ Database Administrator Ex. University Transcript Office	This application would be handled by the transcript office in charge of uploading the students grades into a database. With that database being created, they can further create a sheet of grades (Transcript) for students to access and view their semester grades.	Menu-Based Interface, Parametric User Interface → Presented with a list of options such as menus to implement simple commands to assign certain grade levels to students.

Prerequisite Table

User Category	Type of Applications	Type of Interface
→ Naive/Parametric Users Ex. Faculty Department, Professors teaching a specific class	This application can be used by professors to notify students of what prerequisites are required for each course. Professors will have an understanding of the class they teach and advise students the appropriate pre-requisites	Menu-Based Interface Parametric User Interface → Presented with a list of options such as menus to implement simple commands to assign certain pre-requisite courses to specific classes that professors are teaching.

Q2:

Student Table

- Student_number column is unique because the student numbers should be different regardless if more than one student has the exact same name

Course Table

- Course_number column is unique because there can't be two course numbers which are the same. Even if a course were to have a part 2 to it for example [Database I (CP363) & Database II (CP463)] the course number is different.

Prerequisite Table

- There are no unique values because a single course in the upper year may have more than one pre-requisite. For example in the picture below you can see that CP-363 requires two prerequisites (CS3320 & MATH2410).

PREREQUISITE	
Course_number	Prerequisite_number
CP-363	CS3320
CP-363	MATH2410
CS3320	CS1310

Section Table

- The Section_Identifier column is unique because each course will have a specific section it is assigned to. For example, if there is a large class that requires multiple sections, they will be addressed accordingly as the following; CP-363-A (Section A) & CP-363-B (Section B)

Grade_Report Table

- There are no unique columns because the grade report displays all grades that a student has completed. If a student takes more than one class, their student number will show up more than once. The same can be said for Grade column where multiple similar grades will appear on this table

Q3:

If I were designing a web-based system to make airline reservations and sell airline tickets, I would go with a Multi-User Enterprise DBMS Architecture. The reason behind this is because there are thousands of customers every day who are flying across the globe and it would be necessary to have more than one database across several computers in the airport. Multi - User Enterprise database supports a large number of users. With many customers making airline reservations at once and by chance one database failed, the customers would still be able to access other files.

The reason I didn't choose a centralized DBMS Architecture is because there is a single database at only one location. Yes there are advantages of having a centralized over a distributed database such as, it is much easier to synchronize and update data. However, for this specific problem of having customers make airline reservations, there would be thousands of users accessing the same database which could lead to slower results. With that being said, if the database were to crash/fail due to the amount of users, customers will lose access and not be able to make airline reservations.

There are many types of Databases and they all have their advantages and disadvantages however it is dependent on which database fits best given the scenario at hand.