18/20

Yi Lin

CMPT220

Professor Arias

6 April 2018

Include in the document all the sections described, e.g. Abstract, Design, etc.

Project 2 Milestone

My database project utilizes three databases: student database, Marist database, and location database. Student database stores the academic, financial, and family information of the students. The Marist database stores the requirements for the students to continue college. The location database stores the students' home town economy information. The java program will pull data from those databases and use students' data as independent variables and location data and Marist data as dependent variables to calculate the possibility for student to continue college. I will use academic performance and financial ability as the measurement for student continuity.

My determination to work on database project was inspire from the project that my supervisor and colleague are working in their course. Their project focus student retention, but, unlike my database project, they consider various factors that allow student to continue college academic career. They consider not only academic performance and financial ability, but also psychological factors that cause student to dropout, travel distance that cause student to transfer, social issues that student faced in current college, and opportunities that the college offer. After listen to their discussion, I want to do similar things for my database project.

As the first step after proposing the project, I began with constructing the database starting with student database, Marist database and then location database. For accessibility and easiness, I decide to use Microsoft Access to construct the database. I fill student database with fictional student information. Then, I use data from Marist websites for Marist database and data from census.gov for location database. After completing those databases, I started to write java codes to construct the connection to the databases. I searched online and found sample java codes that construct the Microsoft connection using a JDBC driver called ucannaccess. I follow the sample code and create my first prototype, but it prompt

error when running. I went to my colleagues and professor for help and discover that one of the reference libraries that the driver require was outdate. After I updated the library, the code run fine. I began to pull data from student database and measure it against Marist database. I create three difference classes to construct the connections to those databases and pull data from those databases. During the process, syntax error on SQL queries in java codes and methods to pull the data have cause much delay. My next objective is to construction an algorithm that will use data pull from location database to calculate the student financial ability to continue class. Before completing that objective, I plan to create a new class called database as super class of those three database classes. I believe it will be a tough process since to determine the data to use from location database is very confusing.

In conclusion, when the project is finished, the program will read from the database, calculating students' possibility of retention, and present the possibility as console output along with student data. I wish by accomplish the project I will gain more experience and skills in write code in regarding java.

UML

<table>
<tr><td colspan="4">MaristDB</td></tr>
<tr><td colspan="4">-tuition: long -roomAndBoard:long -bookAndSupplies:long -personalMiscellaneous: long -minGPA: double</td></tr>
<tr><td>+ MaristDB ()
+ connect(): Connection</td><td>+ pullData(Connection conn): void
+ getPersonalMiscellaneous(): long</td><td>+ getTuition(): long
+ close(): void</td><td>+ getBookAndSupplies(): long
+ getMinGPA(): double
+ getRoomAndBoard(): long</td></tr>
<tr><td colspan="4">StudentDB</td></tr>
<tr><td>-ids: ArrayList<Integer>
-firstName: ArrayList<String>
-lastName: ArrayList<String>
-home: ArrayList<String></td><td>-gpa: ArrayList<Double>
-classPerform: ArrayList<Integer>
-studentStatus: ArrayList<String></td><td colspan="2">-awardSch: ArrayList<Integer>
-thirdSch: ArrayList<Integer>
-famCont: ArrayList<Integer></td></tr>
<tr><td>+ StudentDB()
+ connect(): Connection
+ close(): void
+ pullData(Connection conn): void
+ getIds():ArrayList<Integer></td><td colspan="2">+ getFirstName(): ArrayList<String>
+ getLastName(): ArrayList<String>
+ getGpa():ArrayList<Double>
+ getClassPerform(): ArrayList<Integer>
+ getAwardSch(): ArrayList<Integer></td><td>+ getThirdSch(): ArrayList<Integer>
+ getFamCont(): ArrayList<Integer>
+ getHome(): ArrayList<String>
+ getStudentStatus(): ArrayList<String></td></tr>
<tr><td colspan="4">LocationDB</td></tr>
</table>

-city: String	-less15kHouseholdPer: double	-less75kMarriedCouplePer: double
-state: String	-less15kFamiliesPer: double	-less75kNonfamilyPer: double
-totalHousehold: int	-less15kMarriedCouplePer: double	-less100kHouseholdPer: double
-families: int	-less15kNonfamilyPer: double	-less100kFamiliesPer: double
-marriedCouple: int	-less25kHouseholdPer: double	-less100kMarriedCouplePer: double
-nonfamily: int	-less25kFamiliesPer: double	-less100kNonfamilyPer: double
-totalHouseholdMedianIncome: int	-less25kMarriedCouplePer: double	-less150kHouseholdPer: double
-familiesMedianIncome: int	-less25kNonfamilyPer: double	-less150kFamiliesPer: double
-marriedCoupleMedianIncome: int	-less35kHouseholdPer: double	-less150kMarriedCouplePer: double
-nonfamilyMedianIncome: int	-less35kFamiliesPer: double	-less150kNonfamilyPer: double
-totalHouseholdMeanIncome: int	-less35kMarriedCouplePer: double	-less200kHouseholdPer: double
-familiesMeanIncome: int	-less35kNonfamilyPer: double	-less200kFamiliesPer: double
-marriedCoupleMeanIncome: int	-less50kHouseholdPer: double	-less200kMarriedCouplePer: double
-nonfamilyMeanIncome: int	-less50kFamiliesPer: double	-less200kNonfamilyPer: double
-less10kHouseholdPer: double	-less50kMarriedCouplePer: double	-more200kHouseholdPer: double
-less10kFamiliesPer: double	-less50kNonfamilyPer: double	-more200kFamiliesPer: double
-less10kMarriedCouplePer: double	-less75kHouseholdPer: double	-more200kMarriedCouplePer: double
-less10kNonfamilyPer: double	-less75kFamiliesPer: double	-more200kNonfamilyPer: double
+LocationDB ()	+getNonfamilyMeanIncome: int	+getLess75kHouseholdPer: double
+connect(): Connection	+getLess10kFamiliesPer: double	+getLess75kFamiliesPer: double
+close(): void	+getLess10kHouseholdPer: double	+getLess75kMarriedCouplePer: double
+pullData(Connection conn): void	+getLess10kMarriedCouplePer: double	+getLess75kNonfamilyPer: double
+getCity(): String	+getLess10kNonfamilyPer: double	+getLess100kHouseholdPer: double
+getState: String	+getLess15kHouseholdPer: double	+getLess100kFamiliesPer: double
+getTotalHousehold: int	+getLess15kFamiliesPer: double	+getLess100kMarriedCouplePer: double
+getFamilies: int	+getLess15kMarriedCouplePer: double	+getLess100kNonfamilyPer: double
+getMarriedCouple: int	+ getLess15kNonfamilyPer: double	+getLess150kHouseholdPer: double
+getNonfamily: int	+getLess25kHouseholdPer: double	+getLess150kFamiliesPer: double
+ getTotalHouseholdMedianIncome: int	+getLess25kFamiliesPer: double	+getLess150kMarriedCouplePer: double
	+getLess25kMarriedCouplePer: double	+getLess150kNonfamilyPer: double
+getFamiliesMedianIncome: int	+getLess25kNonfamilyPer: double	+getLess200kHouseholdPer: double
+getMarriedCoupleMedianIncome: int	+getLess35kHouseholdPer: double	+getLess200kFamiliesPer: double
+getNonfamilyMedianIncome: int	+getLess35kFamiliesPer: double	+getLess200kMarriedCouplePer: double
+getTotalHouseholdMeanIncome: int	+getLess35kMarriedCouplePer: double	+getLess200kNonfamilyPer: double
+getFamiliesMeanIncome: int	+getLess35kNonfamilyPer: double	+getMore200kHouseholdPer: double
+getMarriedCoupleMeanIncome: int	+getLess50kHouseholdPer: double	+getMore200kFamiliesPer: double
	+getLess50kFamiliesPer: double	+getMore200kMarriedCouplePer: double
	+getLess50kMarriedCouplePer: double	+getMore200kNonfamilyPer: double
	+getLess50kNonfamilyPer: double	

Reference

SJ. “Java JDBC: An Example to Connect MS Access Database.” BenchResources.Net, 24 Apr. 2017, www.benchresources.net/jdbc-msaccess-database-connection-steps/.

Saleem, Aamir, director. Java Program to Access the MS Access Database with Ucanaccess Library with Java 8 or Above. Youtube, 2 Dec. 2016, www.youtube.com/watch?v=Zb1AWUroicM.