Database Design Project - Draft

**Content**

[Introduction 2](#_Toc435978113)

[Objective 2](#_Toc435978114)

[Study Design 2](#_Toc435978115)

[Overview 2](#_Toc435978116)

[Methods 2](#_Toc435978117)

[Results 3](#_Toc435978118)

[List of Entities and ERDs 3](#_Toc435978119)

[Table 1. Student 4](#_Toc435978120)

[Table 2. Enrollment 4](#_Toc435978121)

[Table 3. Offering 4](#_Toc435978122)

[Table 4. Course 5](#_Toc435978123)

[Table 5. Assignment 5](#_Toc435978124)

[Table 6. Discussion 5](#_Toc435978125)

[Table 7. Quiz 6](#_Toc435978126)

[Table 8. Demographic1: Personal 6](#_Toc435978127)

[Table 9. Demographic2: School Related 6](#_Toc435978128)

[Transformation of Entities 6](#_Toc435978129)

[Splitting Compound Attributes 6](#_Toc435978130)

[Expanding an Entity Type 7](#_Toc435978131)

[M-Way Relationship 7](#_Toc435978132)

[Finalizing the ERDs 7](#_Toc435978133)

[Documenting the ERDs 7](#_Toc435978134)

[Checking for Design Errors 8](#_Toc435978135)

[Converting the ERDs into Relational Tables 8](#_Toc435978136)

[Inserting Data into the Tables 8](#_Toc435978137)

[Normalization of Relational Tables 16](#_Toc435978138)

[Presentation of the Learning Analytics Model 17](#_Toc435978139)

[Queries for the Learning Analytics Table 18](#_Toc435978140)

# Introduction

The University of Florida (UF) is starting its Analytics Program. Since Business and Learning Analytics have been growing in the United States and no program has been officially launched at UF, the Provost and Academic Technology started to show concern. Particularly, we felt very behind to the current progress and evolution of the field after participating in the Learning Analytics and Knowledge Conference carried out at Marist College, Poughkeepsie, New York, in March 2015. We are determined to start the program and a committee was established to guide this process. I am a member of this committee as a researcher with expertise in statistical analysis. I will not be involved in the database design, but I would like to use this opportunity to see deeply the development process and outcomes in order to make suggestions of improvement by the time I receive the database for analysis. If this proposal is acceptable, I would like to design a “prototype” database for a general Learning Analytics Program. This will be the first program to start at UF because the Warrington College of Business will carry out the Business Analytics Program.

# Objective

To design a general database that could be used to predict students at risk to successfully complete their courses.

# Study Design

## Overview

In general, there is always a small percentage of students who fail to successfully complete their courses. If these students could be predicted at the beginning of the semester, an intervention program might help them to pass their courses. This will significantly increase students’ performance, graduation rate, and eventually recruitment rate. The prediction stage uses predictive models that rely on accurate and reliable databases. Therefore, the first phase of this alert system process is to create the database needed for the program. The Registrar Office and the Learning Management System are the main sources with the required information for the databases. Database designers have the role to create the database specifically needed to perform specific probabilistic models. The database will be created according to the variables (columns) that will be used by the probabilistic models to predict students at risk.

## Methods

The procedures used to design the database were the following:

1. List of the tables (student’s performance and demographic) needed for the database.
   * Creating entities:
     + Registrar Office: student, enrollment, offering, course, and demographics.
     + Learning Management System: assignments, discussions, and quizzes.
   * Defining the PKs and FKs.
   * Identifying relationships 1-1 and 1-M.
2. Create the ERDs with SQL Developer Data Modeler for the student’s performance and demographic tables.
   * Drawing entities with their respective 1-1 and 1-M relationship.
   * Transforming entities by using:
     + Splitting compound attributes
     + Expanding entity types
     + M-Way relationship
   * Finalizing the ERD:
     + Documenting the ERD
     + Detecting design errors
3. Convert ERDs into relational tables:
   * Creating the SQL script from the ERDs by using SQL Developer.
   * Following basic conversion rules
4. Build a fake sample data – inserting data into the tables
5. Work with functional dependencies to remove redundancies in the database:
   * Removing potential functional dependencies with sample data criterion.
   * Deleting redundancies by using the simple synthesis procedure.
6. Presentation of the model.
7. Queries to obtain the required database:
   * I used the SELECT statement in SQL to make the necessary queries to join a set of tables and obtain the necessary database for this research.
   * The variables (columns) needed in the database will be the attributes of the tables that I plan to create for this project.
   * I created a view for the final table.

# Results

## List of Entities and ERDs

Tables 1-9 present the list of entities used for creating the relational model for the Learning Analytics Program. This model is not build from a narrative. Instead, the model is created from this list of tables that should be taken from the Registrar Office and the Learning Management System. They are as follows: Student, Enrollment, Offering, Course, Assignment, Discussion, Quiz, Demographic1, and Demographic2. The relationships are described below and Figure 1 illustrates the entity relationship diagrams (ERDs).

Enrollment is the center of the ERDs Database Model. It has two primary-foreign (PF) keys, Offering\_OfferNo establishes a 1-M relationship Offering-Enrollment and Student\_StudentNo creates a 1-M relationship Student-Enrollment.

Offering has OfferNo as a primary (P) key and Course\_CourseNo as a foreign (F) key. This F Key makes the 1-M relationship Course-Offering. Furthermore, Assignment, Discussion, and Quiz have their respectives P ID keys and Course\_CourseNo as the F Key. This F Key builds the 1-M relationship Course-Assignment, Course-Discussion, and Course-Quiz.

The Student entity has a 1-1 relationship with Demographic1 and Demographic2 through the PF key Student\_StudentNo, which it is the PF for Demographic1 and 2. The relationships are Student-Demographic1 and Student-Demographic2.

All 1-M relationships are mandatory, and they have 1-1 and 1-M cardinalities. The 1-1 Student-Demographics 1 and 2 relationships are mandatory, and the cardinalities are 1-1 and 1-1. All primary keys are stable and single purposes.

Figure 1 illustrates the simple model just described and it is the path to follow in order to obtain the required columns that the probabilistic models will use to predict students at risk.

### Table 1. Student

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **StuNo** | Char (11) | Student Number (Primary Key) |
| StuFirstName | VarChar2 (20) | Student First Name |
| StuLastName | VarChar2 (30) | Student Last Name |
| City | VarChar2 (30) | Student City |
| State | Char (2) | Student State |
| Zip | Char (10) | Student Zip Code |
| Major | VarChar2 ( 20) | Student Major |
| GPA | Numeric (1, 2) | Student GPA |

### Table 2. Enrollment

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ***OfferNo*** | Integer | Foreign key to Offering Number |
| ***StuNo*** | Char (11) | Foreign key to Student Number |
| Grade | Numeric (1, 2) | Grade for the enrollment |

### Table 3. Offering

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **OfferNo** | Integer | Offering Number (Primary Key) |
| ***CourseNo*** | Char (6) | Foreign key to course number |
| Location | VarChar2 (10) | Building and room number |
| Days | Char (6) | Days of the week course was offered |
| Term | Char (6) | Semester course was offered |
| Year | Integer | Year course was offered |

### Table 4. Course

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **CourseNo** | Char (6) | Course Number (Primary Key) |
| CourseName | VarChar2 (50) | Course Name |
| CourseDes | VarChar2 (90) | Course Description |
| Units | Char (10) | Course Units |
| Creation | Date | Date course was created |
| Start | Date | Date course started |
| End | Date | Date Course ended |
|  |  |  |
|  |  |  |

### Table 5. Assignment

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **AssigID** | Integer | Assignment ID (Primary Key) |
| ***CourseNo*** | Char (6) | Foreign key to course number |
| AssigName | VarChar2 (50) | Assignment name |
| AssigDesc | VarChar2 (150) | Assignment description |
| AssigMaxPoints | Numeric (2, 2) | Assignment maximum number of points |
| AssigScore | Numeric (2, 2) | Assignment score |
| AssigAvaDate | Date | Assignment available date |
| AssigSubDate | Date | Assignment submission date |
| AssigUntDate | Date | Assignment until date |
|  |  |  |

### Table 6. Discussion

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **DiscID** | Integer | Discussion ID (Primary Key) |
| ***CourseNo*** | Char (6) | Foreign key to course number |
| DiscName | VarChar2 (50) | Discussion Name |
| DiscDesc | VarChar2 (150 | Discussion description |
| DiscMaxPoints | Numeric (2, 2) | Discussion maximum number of points |
| DiscScore | Numeric (2, 2) | Discussion score |
| DiscGroup | Char (3) | Group discussion |
| DiscAvaDate | Date | Discussion available date |
| DiscUntDate | Date | Discussion until date |

### Table 7. Quiz

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| **QuizID** | Integer | Quiz ID (Primary Key) |
| ***CourseNo*** | Char (6) | Foreign key to course number |
| QuizName | VarChar2 (50) | Quiz name |
| QuizMaxPoints | Numeric (2, 2) | Quiz maximum number of points |
| QuizScore | Numeric (2, 2) | Quiz score |
| QuizTimeLimit | Char (2) | Quiz time limit in minutes |
| QuizResponses | Char (3) | Quiz view responses |
| QuizAnswers | Char (3) | Quiz show correct answers |
| QuizAvaDate | Date | Quiz available date |
| QuizSubDate | Date | Quiz submission date |
|  |  |  |

### Table 8. Demographic1: Personal

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ***StuNo*** | Char (11) | Foreign key to student number |
| Age | Char (2) | Student age |
| Gender | VarChar2 (15) | Student gender |
| Race | VarChar2 (15) | Student race |
| Ethnicity | VarChar2 (20) | Student ethnicity |
| MaritalStatus | VarChar2 (15) | Student marital status |
| FamilySize | Char (2) | Student family size |
| EmpStatus | VarChar2 (15) | Student employment status |
| IncomeLevel | Char (7) | Student income level |

### Table 9. Demographic2: School Related

| **Name** | **Type** | **Description** |
| --- | --- | --- |
| ***StuNo*** | Char (11) | Foreign key to student number |
| SatVerbal | Numeric (2, 2) | Student Sat verbal score |
| SatMath | Numeric (2, 2) | Student Sat math score |
| Class | VarChar2 (10) | Student class |
| Major | VarChar2 (20) | Student major |
| NoCreditHours | Char (2) | Number of credit hour taken per semester |

## Transformation of Entities

### Splitting Compound Attributes

Student had address as a compound attribute that I split into City, State, and Zip. Date in Assignment, Discussion, and Quiz was split into Available Date, Submission Date, and Until Date. These transformations increase the amount of information regarding a specific category in an attribute and facilitate search.

### Expanding an Entity Type

Expanding Demographic into Demographic 1 and Demographic 2 is an example of an entity expansion without a relationship. The list of attribute in Demographic could be divided into two categories: personal and school related. Even though the relationship is not established between them, there is no doubt this expansion makes sense to facilitate search of additional demographic data.

### M-Way Relationship

I do not see the need of creating an M-Way relationship between Course and Assignment, Discussion, and Quiz. This will make Couse a weak entity, which this model does not need it. I would rather to keep the ERDs simple.

## Finalizing the ERDs

### Documenting the ERDs

The description of the attributes are displayed from Table 1 to Table 9. The idea of the detail descriptions is to clearly communicate the attribute name and to avoid misunderstanding in terminology. All attributes have names that indicate their purposes.

The dataset chosen for the Learning Analytics Program must be a complete dataset, no missing values, and with a minimum cardinality of 1. Since the dataset will be used to predict students at risk to complete their courses, the dataset must be as much complete and accurate as possible to avoid wrong predictions. Therefore, the dataset cannot be at random and the relationships must have a minimum cardinality of 1. Selecting the dataset helps to obtain completeness. Otherwise, there is always a possibility of obtaining offering without enrollments, students without enrollments, courses without offerings, and missing values. This is the reason why all 1-M relationships in the ERDs model are mandatory and have cardinalities 1-1 and 1-M. The exceptions are Student-Demographics 1 or 2 relationships, which they are 1-1, mandatory, and have 1-1 and 1-1 cardinalities. In addition, the following checkup was carried out to guarantee completeness:

1. Naming rules: all entities, relationships, and attributes were named.
2. Primary key rule: all entities had a primary key.
3. Cardinality rule: cardinalities were specified for both entities types in a relationship.
4. Entity participation rule: all entities had at least one relationship.

All relationships were checked out for inconsistency. Oracle sometimes added unnecessary F Keys, and they were removed. Furthermore, the 1-M mandatory relationships among Student, Enrollment, Offering, Course, Assignment, Discussion, and Quiz ensure all relationships to be consistence. For instance, a student might have one or several enrollments, each enrollment has only one offering, and one offering is associated with a single course. Finally, each course might have one or more assignments, discussion, and quizzes. Furthermore, the 1-1 relationship also ensure consistency. Each student will have a 1-1 relationship with Demographic 1 and 2. The following checkup was performed to guarantee consistency:

1. Entity name rule: each entity type had a unique name.
2. Attribute name rule: each attribute name was unique within the entity type and the relationship.
3. Relationship/entity type connection rule: all relationships connected two entity types.
4. Relationship/relationship connection rule: relationships were not connected to other relationships.
5. Redundant foreign key rule: redundant foreign keys were not used.

### Checking for Design Errors

Reviewing the procedures carried out for building the ERDs is a very important step to verify the accuracy of the model. Furthermore, the ERDs should consider all possible queries that the database should support. The following steps were performed for error checking:

1. Diagram rules to make sure that that no conspicuous errors were in the ERDs on Figure 1.
2. Entity connections to find out if they were using the correct relationship. I took into consideration the type of relationship (optional or mandatory) and the cardinalities to ensure completeness and consistency. The direction of the relationships were also checked. There are no missing, misplaced, or redundant relationship.
3. The ERDs diagram model is simple and does not use specialized data modeling construct such as generalization hierarchy.

## Converting the ERDs into Relational Tables

I converted the ERD model into a SQL script by using SQL Developer, and the script is presented in Figure 2. I followed the basic conversion rules such as:

1. Entity Type Rule: an entity type with its attributes and primary key becomes a table with its columns and primary key, respectively, after conversion.
2. 1-M Relationship Rule: each 1-M relationship becomes a foreign key in the child table. If the minimum cardinality is one on the parent side of the relationship, the foreign key cannot accept null values.

## Inserting Data into the Tables

I inserted fake data into the tables by using the “Insert into” command in Oracle (Figure 2). I inserted 10 rows per table with the exception of courses, assignments, discussions, and quizzes, in which I inserted eight rows corresponding to eight courses. I used the opportunity to double check that all data type and constraints for P, F, and PF keys were correct.

**Figure 2. SQL script for the ERDs converted into tables.**

CREATE TABLE Assignment

(

AssigID CHAR (7) NOT NULL ,

Course\_CourseNo CHAR (7) NOT NULL ,

AssigName VARCHAR2 (50) ,

AssigDesc VARCHAR2 (150) ,

AssigMaxPoints NUMBER (5,2) ,

AssigScore NUMBER (5,2) ,

AssigAvaiDate DATE ,

AssigSubDate DATE ,

AssigUntDate DATE

) ;

ALTER TABLE Assignment ADD CONSTRAINT Assignment\_PK PRIMARY KEY ( AssigID ) ;

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('703521', 'BUS4120','Career Field Analysis', 'In deepth analysis of your career', 50, 47, TO\_DATE ('29/January/16','DD/MM/YY'), TO\_DATE ('5/February/16','DD/MM/YY'), TO\_DATE ('05/February/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('723727', 'BUS4330','Persuasive Presentation', 'How to influence our customers', 50, 45, TO\_DATE ('29/January/16','DD/MM/YY'), TO\_DATE ('5/February/16','DD/MM/YY'), TO\_DATE ('05/February/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('728361', 'WIS3934','Predatory Behaviour', 'Description of general predatory behaviors', 30, 30, TO\_DATE ('11/January/16','DD/MM/YY'), TO\_DATE ('17/January/16','DD/MM/YY'), TO\_DATE ('19/January/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('702947', 'FOR4120','The Pine Forest', 'Ecolgy of pine forest', 70, 65, TO\_DATE ('4/January/16','DD/MM/YY'), TO\_DATE ('11/January/16','DD/MM/YY'), TO\_DATE ('13/January/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('710983', 'ARC2180','Cube', 'Interior design of the cube', 50, 47, TO\_DATE ('15/February/16','DD/MM/YY'), TO\_DATE ('20/February/16','DD/MM/YY'), TO\_DATE ('22/February/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('719362', 'GEO2220','Gulfs of America', 'Geology of gulfs', 90, 79, TO\_DATE ('29/January/16','DD/MM/YY'), TO\_DATE ('5/February/16','DD/MM/YY'), TO\_DATE ('05/February/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('729472', 'STA2023','Exploratory Data Analysis', 'How to do the normality test', 50, 41, TO\_DATE ('4/January/16','DD/MM/YY'), TO\_DATE ('8/January/16','DD/MM/YY'), TO\_DATE ('10/January/16','DD/MM/YY'));

INSERT INTO Assignment (AssigID, Course\_CourseNo, AssigName, AssigDesc, AssigMaxPoints, AssigScore, AssigAvaiDate, AssigSubDate, AssigUntDate) VALUES ('772648', 'STA4905','Model Assumptions', 'Testing model assumptions', 20, 17, TO\_DATE ('11/January/16','DD/MM/YY'), TO\_DATE ('15/January/16','DD/MM/YY'), TO\_DATE ('15/January/16','DD/MM/YY'));

CREATE TABLE Course

(

CourseNo CHAR (7) NOT NULL ,

CourseName VARCHAR2 (50) ,

CourseDes VARCHAR2 (90) ,

Units INTEGER ,

Creation DATE ,

CouStart DATE ,

CouEnd DATE

) ;

ALTER TABLE Course ADD CONSTRAINT Course\_PK PRIMARY KEY ( CourseNo ) ;

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('BUS4120', 'Professional Business', 'Introduction to Business', 4, TO\_DATE ('5/November/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('BUS4330', 'Professional Communication', 'Communication Strategies', 3, TO\_DATE ('7/November/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('WIS3934', 'Mammalogy', 'Ecology of Mammals', 4, TO\_DATE ('25/November/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('FOR4120', 'Forest Ecology', 'Ecology of Forests', 4, TO\_DATE ('5/December/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('ARC2180', 'Architecture Design', 'Design Principles', 4, TO\_DATE ('9/October/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('GEO2220', 'Geography of America', 'Geography of USA', 4, TO\_DATE ('31/October/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('STA2023', 'Introduction to Statistics', 'Basic Statistics Analysis', 4, TO\_DATE ('23/November/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

INSERT INTO Course (CourseNo, CourseName, CourseDes, Units, Creation, CouStart, CouEnd) VALUES ('STA4905', 'Regression Analysis', 'Statistics of Linear Regression', 4, TO\_DATE ('17/October/15','DD/MM/YY'), TO\_DATE ('5/January/16','DD/MM/YY'), TO\_DATE ('20/May/16','DD/MM/YY'));

CREATE TABLE Demographic1

(

Student\_StdNo CHAR (10) NOT NULL ,

Age CHAR (2) ,

Gender VARCHAR2 (6) ,

Race VARCHAR2 (15) ,

Ethnicity VARCHAR2 (20) ,

MaritalStatus VARCHAR2 (15) ,

FamilySize CHAR (2) ,

EmpStatus VARCHAR2 (15) ,

AnnualIncome NUMBER (12,0)

) ;

ALTER TABLE Demographic1 ADD CONSTRAINT "Demographic I\_PK" PRIMARY KEY ( Student\_StdNo ) ;

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0655', '25', 'Male', 'White', 'American', 'Married', '3', 'Employee', 48000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0656', '35', 'Male', 'White', 'American', 'Single', '1', 'Employee', 38000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0657', '27', 'Male', 'Black', 'American', 'Living Together', '2', 'Employee', 35000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0658', '30', 'Male', 'Black', 'American', 'Married', '4', 'Employee', 48000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0659', '28', 'Male', 'White', 'Latino', 'Married', '3', 'Employee', 48000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0660', '29', 'Male', 'White', 'Latino', 'Single', '1', 'Unemployee', 25000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0661', '31', 'Female', 'White', 'American', 'Married', '2', 'Employee', 47000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0662', '33', 'Female', 'White', 'Latino', 'Single', '1', 'Unemployee', 25000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0663', '23', 'Female', 'White', 'Latino', 'Single', '1', 'Unemployee', 27000);

INSERT INTO Demographic1 (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome) VALUES ('S2210-0664', '34', 'Female', 'Black', 'American', 'Living Together', '2', 'Employee', 32000);

CREATE TABLE Demographic2

(

Student\_StdNo CHAR (10) NOT NULL ,

SatVerbal NUMBER (12,2) ,

SatMath NUMBER (12,2) ,

Class CHAR (2) ,

Major VARCHAR2 (30) ,

NoCreditHours CHAR (2)

) ;

ALTER TABLE Demographic2 ADD CONSTRAINT Demographic2\_PK PRIMARY KEY ( Student\_StdNo ) ;

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0655', 660, 770,'SO', 'Business', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0656', 590, 680,'FR', 'Wildlife', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0657', 560, 670,'JR', 'Forestry', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0658', 530, 580,'SR', 'Architecture', '9');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0659', 565, 770,'FR', 'Geography', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0660', 537, 577,'SO', 'Statistics', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0661', 710, 680,'JR', 'Business', '9');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0662', 650, 760,'SR', 'Forestry', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0663', 620, 710,'FR', 'Statistics', '12');

INSERT INTO Demographic2 (Student\_StdNo, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0664', 592, 540,'SO', 'Wildlife', '12');

CREATE TABLE Discussion

(

DiscID CHAR (7) NOT NULL ,

Course\_CourseNo CHAR (7) NOT NULL ,

DiscName VARCHAR2 (50) ,

DiscDesc VARCHAR2 (150) ,

DiscMaxPoints NUMBER (5,2) ,

DiscScore NUMBER (5,2) ,

DiscGroup CHAR (3) ,

DiscAvaiDate DATE ,

DiscUntDate DATE

) ;

ALTER TABLE Discussion ADD CONSTRAINT Discussion\_PK PRIMARY KEY ( DiscID ) ;

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('324578','BUS4120','What is a Good Deal?', 'Criteria to evaluate business', 20, 15, 'Yes', TO\_DATE ('7/March/16','DD/MM/YY'), TO\_DATE ('11/March/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('315437','BUS4330','Seven Habits of Highly Effective People', 'Habits to be successful in business', 10, 10, 'Yes', TO\_DATE ('7/March/16','DD/MM/YY'), TO\_DATE ('18/March/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('393721','WIS3934','Do Lions Think?', 'Lions hunting strategy', 20, 17, 'Yes', TO\_DATE ('8/February/16','DD/MM/YY'), TO\_DATE ('12/February/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('384736','FOR4120','What is the equilibrium?', 'Forest maintenance through time', 10, 10, 'Yes', TO\_DATE ('21/March/16','DD/MM/YY'), TO\_DATE ('25/March/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('318473','ARC2180','How Layers Work for Designers', 'The best use of designing by layers', 20, 19, 'Yes', TO\_DATE ('7/March/16','DD/MM/YY'), TO\_DATE ('11/March/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('383726','GEO2220','How does erosion change the landscape?', 'The role of wind and water on landscapes', 10, 10, 'Yes', TO\_DATE ('14/March/16','DD/MM/YY'), TO\_DATE ('18/March/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('395837','STA2023','Should we transform variables?', 'The advantages of transformations in Statistics', 10, 7, 'Yes', TO\_DATE ('28/March/16','DD/MM/YY'), TO\_DATE ('1/April/16','DD/MM/YY'));

INSERT INTO Discussion (DiscID, Course\_CourseNo, DiscName, DiscDesc, DiscMaxPoints, DiscScore, DiscGroup, DiscAvaiDate, DiscUntDate) VALUES ('319482','STA4905','Should we use the Durbin-Watson test?', 'The importance of autocorrelation of residuals', 20, 18, 'Yes', TO\_DATE ('7/March/16','DD/MM/YY'), TO\_DATE ('18/March/16','DD/MM/YY'));

CREATE TABLE Enrollment

(

Offering\_OfferNo INTEGER NOT NULL ,

Student\_StdNo CHAR (10) NOT NULL ,

Grade NUMBER (12,2)

) ;

ALTER TABLE Enrollment ADD CONSTRAINT Enrollment\_PK PRIMARY KEY ( Offering\_OfferNo, Student\_StdNo ) ;

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1500, 'S2210-0655', 97.85);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1501, 'S2210-0656', 87.25);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (2347, 'S2210-0657', 85.15);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (2181, 'S2210-0658', 99.10);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (7820, 'S2210-0659', 88.75);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (3350, 'S2210-0660', 83.13);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1468, 'S2210-0661', 78.69);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1579, 'S2210-0662', 84.97);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1580, 'S2210-0663', 85.85);

INSERT INTO Enrollment (Offering\_OfferNo, Student\_StdNo, Grade) VALUES (1602, 'S2210-0664', 91.35);

CREATE TABLE Offering

(

OfferNo INTEGER NOT NULL ,

Course\_CourseNo CHAR (7) NOT NULL ,

Location VARCHAR2 (10) ,

Days CHAR (6) ,

Term CHAR (6) ,

Year INTEGER

) ;

ALTER TABLE Offering ADD CONSTRAINT Offering\_PK PRIMARY KEY ( OfferNo ) ;

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1500, 'BUS4120', 'CAR 0100', 'M W', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1501, 'BUS4330', 'CAR 0220', 'T R', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (2347, 'WIS3934', 'NZH 0112', 'M W', 'Fall', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (2181, 'WIS3934', 'NZH 3086', 'T R', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (7820, 'FOR4120', 'MCC 0222', 'M W', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (3350, 'ARC2180', 'FAB 0105', 'M', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1468, 'GEO2220', 'TUR 3006', 'F', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1579, 'STA2023', 'CBD 0137', 'M W', 'Spring', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1580, 'STA2023', 'CBD 1215', 'M', 'Fall', 2015);

INSERT INTO Offering (OfferNo, Course\_CourseNo, Location, Days, Term, Year) VALUES (1602, 'STA4905', 'FLO 0112', 'M W', 'Spring', 2015);

CREATE TABLE Quiz

(

QuizID CHAR (7) NOT NULL ,

Course\_CourseNo CHAR (7) NOT NULL ,

QuizName VARCHAR2 (50) ,

QuizMaxPoints NUMBER (5,2) ,

QuizScore NUMBER (5,2) ,

QuizTimeLimit CHAR (2) ,

QuizResponses CHAR (3) ,

QuizAnswers CHAR (3) ,

QuizAvaiDate DATE ,

QuizSubDate DATE

) ;

ALTER TABLE Quiz ADD CONSTRAINT Quiz\_PK PRIMARY KEY ( QuizID ) ;

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('507427', 'BUS4120','Career Analysis', 10, 7, 20, 'Yes', 'Yes', TO\_DATE ('12/February/16','DD/MM/YY'), TO\_DATE ('12/February/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('519283', 'BUS4330','Persuasion', 15, 8, 15, 'Yes', 'Yes', TO\_DATE ('12/February/16','DD/MM/YY'), TO\_DATE ('14/February/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('583727', 'WIS3934','Predation', 20, 17, 15, 'Yes', 'Yes', TO\_DATE ('22/January/16','DD/MM/YY'), TO\_DATE ('23/January/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('594728', 'FOR4120','Pine Forest', 15, 15, 20, 'Yes', 'Yes', TO\_DATE ('18/January/16','DD/MM/YY'), TO\_DATE ('18/January/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('573849', 'ARC2180','The Cube', 10, 3, 15, 'Yes', 'Yes', TO\_DATE ('25/February/16','DD/MM/YY'), TO\_DATE ('26/February/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('528474', 'GEO2220','Mexico Gulf', 20, 10, 30, 'Yes', 'Yes', TO\_DATE ('10/February/16','DD/MM/YY'), TO\_DATE ('11/February/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('538278', 'STA4905','Model Assumptions', 10, 10, 20, 'Yes', 'Yes', TO\_DATE ('18/January/16','DD/MM/YY'), TO\_DATE ('19/January/16','DD/MM/YY'));

INSERT INTO Quiz (QuizID, Course\_CourseNo, QuizName, QuizMaxPoints, QuizScore, QuizTimeLimit, QuizResponses, QuizAnswers, QuizAvaiDate, QuizSubDate) VALUES ('517438', 'STA2023','Normality Test', 10, 5, 15, 'Yes', 'Yes', TO\_DATE ('22/February/16','DD/MM/YY'), TO\_DATE ('22/February/16','DD/MM/YY'));

CREATE TABLE Student

(

StdNo CHAR (10) NOT NULL ,

FirstName VARCHAR2 (20) ,

LastName VARCHAR2 (30) ,

City VARCHAR2 (30) ,

State CHAR (2) ,

Zip CHAR (5) ,

GPA NUMBER (12,2)

) ;

ALTER TABLE Student ADD CONSTRAINT Student\_PK PRIMARY KEY ( StdNo ) ;

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0655', 'John', 'Smith', 'Gainesville', 'FL', '32605', 3.95);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0656', 'Mark', 'Lennon', 'Ocala', 'FL', '32903', 2.95);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0657', 'William', 'Carter', 'Orlando', 'FL', '32806', 2.95);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0658', 'Marcus', 'Benhamin', 'Fulton', 'GA', '30004', 2.84);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0659', 'Carlos', 'Venero', 'Atlanta', 'GA', '30340', 3.83);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0660', 'Jesus', 'Moreno', 'Aurora', 'NC', '27806', 3.23);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0661', 'Nicole', 'Alsina', 'Barco', 'NC', '27917', 2.75);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0662', 'Adriana', 'Lopez', 'Hurley', 'MS', '39560', 3.65);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0663', 'Aurora', 'Noreli', 'Hurley', 'MS', '39560', 2.74);

INSERT INTO Student (StdNo, FirstName, LastName, City, State, Zip, GPA) VALUES ('S2210-0664', 'Martha', 'Smith', 'Arthur', 'TN', '37707', 3.15);

ALTER TABLE Assignment ADD CONSTRAINT Assignment\_Course\_FK FOREIGN KEY ( Course\_CourseNo ) REFERENCES Course ( CourseNo ) ;

ALTER TABLE Demographic1 ADD CONSTRAINT Demographic1\_Student\_FK FOREIGN KEY ( Student\_StdNo ) REFERENCES Student ( StdNo ) ;

ALTER TABLE Demographic2 ADD CONSTRAINT Demographic2\_Student\_FK FOREIGN KEY ( Student\_StdNo ) REFERENCES Student ( StdNo ) ;

ALTER TABLE Discussion ADD CONSTRAINT Discussion\_Course\_FK FOREIGN KEY ( Course\_CourseNo ) REFERENCES Course ( CourseNo ) ;

ALTER TABLE Enrollment ADD CONSTRAINT Enrollment\_Offering\_FK FOREIGN KEY ( Offering\_OfferNo ) REFERENCES Offering ( OfferNo ) ;

ALTER TABLE Enrollment ADD CONSTRAINT Enrollment\_Student\_FK FOREIGN KEY ( Student\_StdNo ) REFERENCES Student ( StdNo ) ;

ALTER TABLE Offering ADD CONSTRAINT Offering\_Course\_FK FOREIGN KEY ( Course\_CourseNo ) REFERENCES Course ( CourseNo ) ;

ALTER TABLE Quiz ADD CONSTRAINT Quiz\_Course\_FK FOREIGN KEY ( Course\_CourseNo ) REFERENCES Course ( CourseNo ) ;

## Normalization of Relational Tables

The following steps were carried out to normalize the Learning Analytics tables:

1. Making sure that all attributes had unique names following rules to achieve consistency.
2. Avoiding to use prefix as much as possible in attribute names to eliminate redundancies.
3. Checking tables to eliminate potential FDs. The constraint was: X determined Y (X 🡪 Y). X, the determinant, was in the left-hand side (LHS) of the expression, and Y in the right-hand side (RHS). The criterion was that a FD did not exist if two rows had the same value for the left-hand side (LHS), but different values for the right-hand side (RHS). In other words, I falsified a potential FD by proving that it did not exist. This step helps to clearly identify the FDs.
4. Simple synthesis procedure:
   * List of functional dependencies:
     + StdNo 🡪 FirstName
     + StudentStdNo 🡪 Age
     + StudentStdNo 🡪 SatVerbal,
     + Offering\_OfferNo, Student\_StdNo 🡪 Grade
     + OfferNo 🡪 Location
     + CourseNo 🡪 CourseName, CourseDesc, Creation
     + AssigID 🡪 AssigName, AssigDesc
     + DiscID 🡪 DiscName, DiscDesc
     + QuizID 🡪 QuizName
   * Because all FDs involve a PK on the left-hand side, there is no need of normalization work. There are no violations of Boyce-Codd Normal Form (BCNF) because all determinants are candidate keys.
   * There is no need of:
     + Deleting extraneous columns from the LHS of FDs.
     + Removing derived FDs from the FD list.
     + Arranging FDs into groups with each group having the same determinant
   * Tables were already created with the determinant as the PK and their respective FKs and PFKs.
   * Merging Demographic1 and Demographic2 into Demographic. Student\_StdNo was chosen as the PK.

## Presentation of the Learning Analytics Model

Since normalization merged Demographic1 and 2 into Demographic, the original ERDs did not change significantly. Figure 3 illustrates the ERDs model after normalization and Figure 4 presents the SQL script for Demographic.

**Figure 4. SQL script for Demographics after merging Demographic 1 & 2.**

CREATE TABLE Demographic

(

Student\_StdNo CHAR (10) NOT NULL ,

Age CHAR (2) ,

Gender VARCHAR2 (6) ,

Race VARCHAR2 (15) ,

Ethnicity VARCHAR2 (20) ,

MaritalStatus VARCHAR2 (15) ,

FamilySize CHAR (2) ,

EmpStatus VARCHAR2 (15) ,

AnnualIncome NUMBER (12,0) ,

SatVerbal NUMBER (12,2) ,

SatMath NUMBER (12,2) ,

Class CHAR (2) ,

Major VARCHAR2 (30) ,

NoCreditHours CHAR (2)

) ;

ALTER TABLE Demographic ADD CONSTRAINT "Demographic\_PK" PRIMARY KEY ( Student\_StdNo ) ;

ALTER TABLE Demographic ADD CONSTRAINT Demographic\_Student\_FK FOREIGN KEY ( Student\_StdNo ) REFERENCES Student ( StdNo ) ;

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0655', '25', 'Male', 'White', 'American', 'Married', '3', 'Employee', 48000, 660, 770,'SO', 'Business', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0656', '35', 'Male', 'White', 'American', 'Single', '1', 'Employee', 38000, 590, 680,'FR', 'Wildlife', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0657', '27', 'Male', 'Black', 'American', 'Living Together', '2', 'Employee', 35000, 560, 670,'JR', 'Forestry', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0658', '30', 'Male', 'Black', 'American', 'Married', '4', 'Employee', 48000, 530, 580,'SR', 'Architecture', '9');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0659', '28', 'Male', 'White', 'Latino', 'Married', '3', 'Employee', 48000, 565, 770,'FR', 'Geography', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0660', '29', 'Male', 'White', 'Latino', 'Single', '1', 'Unemployee', 25000, 537, 577,'SO', 'Statistics', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0661', '31', 'Female', 'White', 'American', 'Married', '2', 'Employee', 47000, 710, 680,'JR', 'Business', '9');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0662', '33', 'Female', 'White', 'Latino', 'Single', '1', 'Unemployee', 25000, 650, 760,'SR', 'Forestry', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0663', '23', 'Female', 'White', 'Latino', 'Single', '1', 'Unemployee', 27000, 620, 710,'FR', 'Statistics', '12');

INSERT INTO Demographic (Student\_StdNo, Age, Gender, Race, Ethnicity, MaritalStatus, FamilySize, EmpStatus, AnnualIncome, SatVerbal, SatMath, Class, Major, NoCreditHours) VALUES ('S2210-0664', '34', 'Female', 'Black', 'American', 'Living Together', '2', 'Employee', 32000, 592, 540,'SO', 'Wildlife', '12');

## Queries for the Learning Analytics Table

Figure 5 shows the query used to make the Learning Analytics Table. I joined eight tables by using the Cross Product Style. This is required table for conducting statistical analysis and for making predictions of students at risk. The selected columns are the variables that might influence a student to be at risk to fail a course at the end of the semester. The view was also created and it is included in Figure 5.

**Figure 5. Joining eight tables with the Cross Product Style to create the Learning Analytics table for conducting Statistical Analysis.**

CREATE VIEW LA\_View AS

SELECT StdNo, FirstName, LastName, GPA, Age, Gender, MaritalStatus, FamilySize, EmpStatus, SatVerbal, SatMath, NoCreditHours, Offering\_OfferNo, CourseNo, AssigScore, DiscScore, QuizScore

FROM Student, Demographic, Enrollment, Offering, Course, Assignment, Discussion, Quiz

WHERE Student.StdNo = Demographic.Student\_StdNo

AND Student.StdNo = Enrollment.Student\_StdNo

AND Enrollment.Offering\_OfferNo = Offering.OfferNo

AND Offering.Course\_CourseNo = Course.CourseNo

AND Course.CourseNo = Assignment.Course\_CourseNo

AND Course.CourseNo = Discussion.Course\_CourseNo

AND Course.CourseNo = Quiz.Course\_CourseNo