|  |  |
| --- | --- |
|  | **General Instructions** |
|  | Students may study together for the assignment and review each other’s completed work |
|  | Students must each complete the assignment by their own hand |
|  | Please use the provided word document template |
|  | Please save the completed word document into PDF format before uploading |
|  | Please submit the PDF file electronically through eLearning before the due date and time |
|  | Do not worry about variations among database vendors – you may write SQL to any vendor’s dialect |
|  | Do not include output – only the SQL |
|  | Use table aliases for all tables in all queries (unless otherwise specified) |
|  | Column aliases are required for all derived columns including aggregate columns (unless otherwise specified) |
|  | Do not use column aliases unless required as stated previously |
|  | If a problem does not ask for a specific sort order, use your best judgement to add a sort order |

**Chapter 7 Problems – Introduction to Structured Query Language (SQL)**

Do **Problems** 1-6 from Chapter 7 of our textbook at the end of the chapter.

**Chapter 8 Problems – Advanced Structured Query Language (SQL)**

Do **Problems** 1-7 from Chapter 8 of our textbook at the end of the chapter.

**Using MSSQL to answer the following seven (17-23) practical SQL questions using the same university data from Assignment 1 and 2. You will need to convert the MySQL Data to Microsoft SQL to the questions. You may use SQLLINES.COM at** [**http://www.sqlines.com/online**](http://www.sqlines.com/online)**.**

# Problem #14 – Aggregates that are grouped and subsetted (using a GROUP BY clause and a HAVING clause)

Retrieve the class name, minimum GPA, maximum GPA, average GPA, and average GPA plus 10% for each class but only for classes with an average GPA less than 3.5.

|  |
| --- |
| SELECT StdClass, SUM(TotalSale),AVG(StdGPA),MAX(StdGPA),  FROM Student  GROUP BY StdClass  WHERE StdGPA <=3.5; |

# Problem #15 – Aggregates of a subset of rows that are grouped and subsetted (using a WHERE clause, a GROUP BY clause, and a HAVING clause)

Retrieve the class name, minimum GPA, maximum GPA, average GPA, and average GPA plus 10% for each class but only for non-IS majors and only for classes with an average GPA greater than 3 for non-IS majors.

|  |
| --- |
| SELECT StdClass, SUM(TotalSale),AVG(StdGPA),MAX(StdGPA)  FROM SALES  GROUP BY Salesperson  HAVING Salesperson <>'Bennett' AND StdGPA>=3.5; |

# Problem #16 – Cartesian Products, how many rows expected

Perform a Cartesian Product between tables Student, Offering, Enrollment, Course, and Faculty How many columns are expected? 5

How many rows are expected? The Maximum of the 6

|  |
| --- |
| SELECT FactNo, CrsUnits, OfferNo, StdNo, EnrGrade  FROM Faculty, Course,Offering,Student,Enrollment; |

# Problem #17 – Cartesian Products, figuring out which rows match

Perform a Cartesian Product between tables Student, Offering, Enrollment, Course, and Faculty

Retrieve only the columns which are needed to show matching based on the relationship between the five tables and order in such a way as to tell the matching records.

|  |  |
| --- | --- |
| |  | | --- | | SELECT FactNo, CrsUnits, OfferNo, StdNo, EnrGrade  FROM Faculty, Course,Offering,Student,Enrollment; | |

# Problem #18 – Turning a Cartesian Product into an Inner Join by adding a WHERE clause to the Cross Product Syntax

Start with a Cartesian Product between tables Student, Offering, Enrollment, Course, and Faculty

Retrieve only the columns which are needed to show matching based on the relationship between the five tables and order in such a way as to tell the matching records

Add a WHERE clause to turn the Cartesian Product into an Inner Join.

|  |  |
| --- | --- |
| |  | | --- | | SELECT FactNo, CrsUnits, OfferNo, StdNo, EnrGrade  FROM Faculty, Course,Offering,Student,Enrollment  WHERE Student.StdNo==123456789; | |

# Problem #19 – Converting an Inner Join from Cross Product Syntax to Join Operator Syntax

Start with the Inner Join using Cross Product Syntax for the tables: Student, Offering, Enrollment, Course, and Faculty Convert to Join Operator Syntax.

|  |
| --- |
| SELECT FactNo, CrsUnits, OfferNo, StdNo, EnrGrade  FROM Faculty, Course,Offering,Student,Enrollment  WHERE Student.StdNo==123456789; |

# Problem #20 – Combining Inner Join and WHERE, GROUP BY, and HAVING clauses

List the course number, offer number, and average grade of students enrolled in fall 2010 IS course offerings in which more than one student is enrolled.

|  |
| --- |
| SELECT FactNo, CrsUnits, OfferNo, StdNo, EnrGrade  FROM Faculty, Course,Offering,Student,Enrollment  WHERE Student.StdNo==123456789;  HAVING Offering.OffYear=”fall 2010” AND Course.CrsDesc==”IS” |