**IST 220 – Intro to DB**

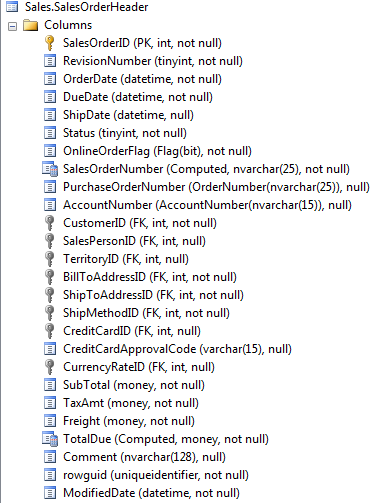
**Project 2 – Generating Desired Information from Raw Data**

Now you have learned how to use a SELECT statement with up to 6 clauses, various kinds of table joins, as well as built-in functions that can help data analysts to calculated and summarize data. Now let’s put all these into hands-on practice with the AdventureWorks database that mimics a real world business setting.

For your convenience, the questions are listed in two groups, for aggregate and scalar functions. Of course, questions in the second group may also use a group function so as to give you more exercises. For each group, you need to make up at least 2 more queries that require functions from the relevant group.

**Group 1 – Queries Requiring Aggregate Functions**

In the following 6 queries, only one table, the SalesOrderHeader table is used, whose schema is shown below. You are encouraged to use one or more additional tables in the queries that you are to make up.



Q1. Determine how many entries are in the table.

Q2. Determine how many (non-null) ShipDate values are in the table. Also display the earlies and latest ship dates.

Additional question: are all the order handled and shipped?

Q3. Determine how many orders involved a credit card and how many different credit cards are used.

Additional questions: are the two number the same; how do the numbers compare with customer count and do this make sense?

Q4. Determine total amounts for SubTotal, TaxAmt, Freight, and TotalDue values.

Additional questions: is the book balanced? How to check it without manual calculation?

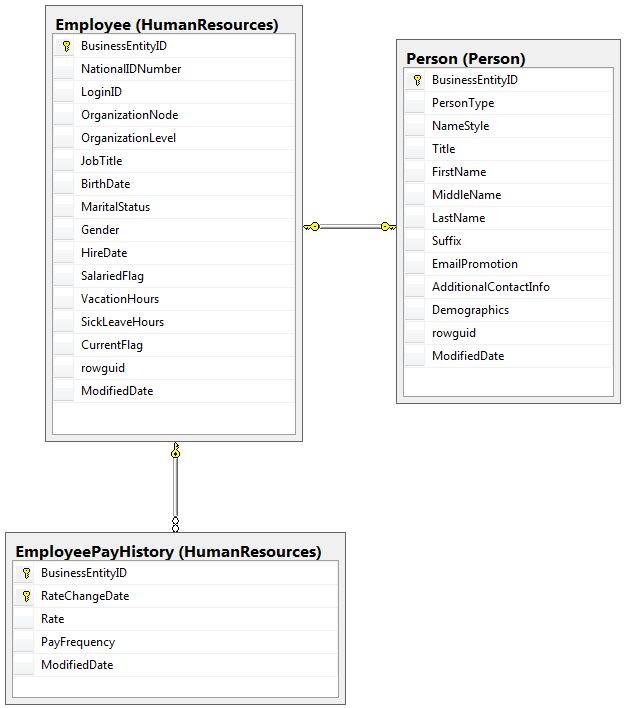
Q5. Generate a report that display order count and total SubTotal amount, and group the aggregate terms by customer and by then OnlineOrderFlag.

Additional questions: what can you tell about the trends? Suggest other breakdowns that can help find some hidden information.

Q6. Generate a report that display order count and total TotalDue amount by territory, but only display those total amounts that are higher than $10M.

**Group 2 – Queries Requiring Scalar (and possibly aggregate) Functions**

Four queries are listed here that deal with data in three tables as shown below: employee’s personal data are in the Person table, while pay rate and raise history data in another table.



SFQ1. Greetings from the CEO to the employees celebrating birthday this month. Generate a line in the pattern: Happy Birthday, Amy A! for Amy Alberts.   
[Functions to consider: Month(), Left() or Substring(), and Concat().]

SFQ2. Generate a list that displays years of service for all employees. Then use the Count() function to tally up employee counts by service year.  
[Functions to consider: DateDiff() and Count().]

SFQ3. Congratulations from the CFO to all employees for having a 5% across-the-board pay raise. You need to generate a line in the pattern: Congratulations, Ms. Erickson for your upcoming pay raise! for Gail Erickson. The employee’s salute (such Ms.) will be retrieved from Title column in the Person table if it’s not null, or from the standard options Ms. and Mr. based on the Gender value in the Employee table. (*BTW, do you see any problem here in the design of the AW database?*)  
[Functions to consider: IIf() and IsNull(). You will need to used one IIF() function inside of another one. The salutes can be further fine-tuned by incorporating MaritalStatus data.]

SFQ4. A pay raise of 10% for every employee whose pay Rate has never been changed. Again, this can be carried out step by step so that you can follow what exactly happened.

1. Select BusinessEntityID, **MAX(**Rate**)**, and COUNT(RateChangeDate) from the HumanResources. EmployeePayHistory table. The result is already sorted. You will see some IDs correspond to a count of changes greater than 1, meaning those employees have got a raise (more accurately, a rate change, but it’s been always raises) sometimes already.
2. Exclude those lucky guys. You should know how to do it.
3. Modify the Rate column such that the value will be increased by 10%. To show the raise amount in monetary format, use the Str() function to impose 2 decimal places. You may remove the count column now since you don’t really it for the raise.  
   [Functions to consider: Count() and Str().]

***Please make up at least one additional query for each group***.

You may find a partner to work on this phase, and my suggestion is to find someone who is not in the same major program with you. That is, if you are a Business major, try to find one who majors in IST or CSC, and vice versa.

Pay attention to the way tables are used for different business entities, what columns are in which table (and try to find “bugs” in the design as you can), and how are the tables related to one another. We will be discussing “design” next, and you will be doing some design work soon.