Table of Contents

These are general rules to be followed for all assignments for this class

1. Files for Assignments

1. **Files to turn in:** For most assignments you turn in two files. One is the script file which contains the SQL statements. You are required to use the script template I posted for assignments. The other file is the spooled file which contains the result of running the script file; this will contain both the ZSQL statements and the result set for each SQL statement.

2. **File names:** The file names for the script and spool file must be named following this pattern. A03_ *yourLastName*.SQL and A03_*yourLastName*.LST where you use the 2 digit assignment number. If you have noticed that another person in class has the same last name, then please use the pattern A03_ *yourLastNameyourFirstName*. The zip file name will have the same naming pattern.

2. Definition of Terms

There are a few terms that I use in assignment that occasionally are confused. Use the following definitions for assignments. Ask questions in the forums in you have additional questions.

1. **Current Date:** this is the date that the server provides. In Oracle, we will use the date returned by sysdate as the current date. An Oracle date value always includes a time component.

2. **Current Year**: this is the year of the current date. If a task description uses the phrase "the current year" you must write the query in a way so that it will run correctly whenever it is run. So you will not hardcode in a year value such as 2015 for "the current year". You derive the current year from the system date.

3. **Current Month, Previous Month**: this is the full month of the current date. As I write this, it is 2015-01-04; so the current month would be Jan 1 through Jan 31, 2015.The previous month is the entire month before the current month- so as I write this, the previous month is Dec 1 through Dec 31, 2014. The next month would be Feb 1-28, 2015.

4. **List_Price, Sale_Price, Order_Price**: Some of the data tables store data for products that can be ordered. A product might have a list_price; this is sometimes called the "manufacturer's suggested retail price." This is stored in the table with the general product information.   When someone orders an item, they are given a price that might be called the sale price or order price or quoted price. This price is stored in the tables that contain the data for the items being ordered. (If you find this confusing- go to a web site such as Amazon- it generally shows a list price and an actual price- which price do you pay?)

5. **Extended Cost**: Extended cost is defined as the order price * quantity. If a customer orders 5 copies of a book that has an order price of $30.00, the extended cost is $150.00.

6. **Amount Due:** In this class, we can have three different meanings for amount due. Suppose a customer places an order for 2 pair of shoes at $80.00 each and 6 pair of socks **at** $4.00 each. The amount due per item is: for the shoes $160.00 (2 * 80); for the socks $24.00 (6 * 4). The amount due for the order is $184 (160 + 24). The total amount due for the customer would be the sum of the amount dues for all orders that customer placed. For this class, do not include sales tax, handling etc. in the amount due.

3. SQL Rules

1. The various parts of the assignment might be stated as questions to be answered or tasks to perform. In either case you create the SQL statements(s) needed to do that step. If you are not sure about the directions, ask. Some assignment tasks are intentionally a bit ambiguous; you should learn how to detect ambiguous task descriptions. If you are doing your work at the last minute and do not have time to get these ambiguities cleared up, then include a comment as to how you interpreted the task description- that may be considered in the grading. Do not give me alternate queries for tasks.

2. Unless other stated in the assignment, each task in the assignment is to be done with a single SQL statement.

3. Misc syntax requirements
    a. All joins must use the Ansi standard joins **in the From clause**; joins handled in the Where clause will lost points for the assignments.
    b. Use the fewest tables needed to accomplish a task. Do not use an outer join if an inner join will work; an inner join is more efficient. Do not use a full join unless it is the only solution; that is uncommon.
    c. If you use table aliases, the aliases must be meaningful; the use of aliases A, B, C, etc is not acceptable.
    d. Do not create views or temporary tables unless the assignment specifically requests them.
    e. Some tasks specify a sort order. If there is no sort specified, then you are allowed to sort if you wish.

4. As you develop the queries you may add additional columns in the Select to test your design. Be certain to remove any column not listed in the task as a column to be displayed in the final version. You might also insert additional rows for testing; be careful to remove those rows before you run your script for grading.

4. Documentation

Documentation should be appropriate for the task at hand. These script files are assignments to be graded. They are not script files that you would save for 5 years as documentation for a company project. So having extensive comments is inappropriate (wrong) for these scripts. I need a comment with your name, and the task comments. If you are writing SQL that is so unusual that you feel you need to explain it, then – first step back and think about why you are writing tricky code and could you find a way to do the task that does not need explaining. If you still feel that your way is the best way and it needs documentation, include a short comment that explains your technique.
Do **not** copy the task description from the assignment into the script.

5. Sample Output

Some assignments give you a display of sample output. This is intended to help you understand the assignment. The actual data values shown on the assignment sheets might not match the data that your program is supposed to produce or the data values in the tables provided. You must write assignments to meet the specifications, not to match the data values that are in a specific collection of data. Often I narrow the columns to fit the page; you should not attempt to do that in your queries. If I want a specific column width, I will state that.
You should follow the sample in terms of the column order and the column aliases.

6. SQL for Tables

Some assignments will ask you to download files from the class web page. You may not change the contents of the files unless directed to do so in the assignment. Specifically, you may not correct things that you perceive as

errors in the data. (Ask me if you think there is an error- but do not change the data without prior approval.)
I will generally provide files on my web page for creating the tables and inserting data into the tables. You will lose points on the assignment for using the wrong data file. You should not include the table creation or insertion statements in your spool file. I may ask for some specific queries at the start of the assignment to help check that you are using the right data set.
You can reload the datasets at any time during the semester

7. SQL Style

1. Your SQL needs to be written following the common convention of starting the main clauses on a new line. For example, this is an acceptable style

```
select z_name, z_price, z_dob
from zoo
where z_type = 'turtle'
order by z_dob;
```

This is also an acceptable style

```
select
  z_name
, z_price
, z_dob
from zoo
where z_type = 'turtle'
order by z_dob;
```

This sql statement is not in an acceptable style and will cost points on the assignment.

```
select z_name, z_price, z_dob from zoo
where z_type = 'turtle' order by z_dob;
```

This sql statement is not in an acceptable style and will cost points on the assignment.

```
select z_name, z_price, z_dob from zoo where z_type = 'turtle' order
by z_dob;
```

The main clauses start with the key words:

```
Select
From
Where
Group by
Order By
```

2. When you are writing joins, the words join, inner join, left join or right join start a new line. The joining phrase is on the same line as the join key word. This makes it easier to see the type of join you are using and the table names.

```
select  . . .
from table_1
join table_2  on table_1.colA = table_2.colA
join table_3  using (colC)
left join table_4  on table_3.colB1 = table_4.colB2;
```

Do *not* use this style with a new line for the On or Using phrase.

```
select  . . .
from table_1
join table_2
on table_1.colA = table_2.colA;
```

Do *not* use this style which puts the join word at the end of the line.
```
select  . . .
from table_1  join
table_2
on table_1.colA = table_2.colA;
```

3. You may use upper or lower case letter for key words such as Select and From.

4. Do not write query lines that are longer than 100 characters. It is hard to understand your work when I have to scroll horizontally..