IS 475/675 HW#4

The objectives for this assignment are to: 1) learn how to write simple, 1 table SQL queries; 2) become more familiar with the CutGlass job costing database; and 3) learn about each component of the SQL SELECT statement. Each of the questions in this homework assignment requires you to create a SELECT statement to satisfy the request. There are 15 questions that should produce 15 SELECT queries for this assignment. None of the questions require that you join tables to create the answer. Questions start on pg. 3 of this document.

Deliverable

The deliverable is paper-based. For each query, turn in the SQL code and then the output generated from that code for that question directly under the code. I want to first see the code, and then see the output directly under the code. Please **DO NOT first list all the code for all the questions and then all the output for all the questions!** I want code and output integrated.

You will have to cut and paste to get the output to look readable in this format. I require that you capture the result table in graphics format from SQL Server using Windows' Snipping Tool (or a Snipping Tool like product) and paste the image of the result table into a Word document (or word processing package of your choice). Print the word processing document and turn in the pages.

Do not modify the result table in Word beyond resizing the image for better readability. You may have to use landscape output in MS Word for some of the wider reports.

For example, let's say that one of the questions is to find the client name, phone, and email address for all clients who are located in the state of Nevada. Let's assume that this is question #2 (it isn't - this is just an example to show what I want turned in for the assignment). The deliverable would be:

Question #2.

```
SELECT     ClientName       'Client Name',
           phone            'Client Phone Number',
           email            'Client Email'
FROM       tblClient
WHERE      ClientState = 'NV'
ORDER BY   ClientName;
```

	Client Name	Client Phone Number	Client Email
1	3 Gals From Verona	7754283191	SophiaM@mail.yahoo.inv
2	Aero Professional Corp.	7757157677	amelia777@gmail.inv
3	AO Reid Construction	7757225646	Harry@AOReid.inv
4	Casem, Brokaw and Stuart, LLC	7757261335	CCasem@gmail.inv
5	Dew Drop Inn Luxury Suites	7754316404	MarySmith@dewdropproperties.inv
6	Fran and Harrold Meyers	7758652686	franny47@gmail.inv
7	Kelly Property Development	7758642950	Dave.Kelley@KPDLLC.inv
8	Less Furniture Company	7757089540	bruce2@mail.yahoo.inv
9	Lloyds Casino Properties, LLC	7755004010	Guillermo@gmail.inv
10	Ms. Catherine Hampstead	7759180022	Champstead@abner.rr.inv

Please format the SQL code following the guidelines given in class. Here are some guidelines to follow about your SQL code:

- Start SQL reserved words in the first column of the line.
- Put SQL reserved words in all uppercase.
- Put attribute and table names in all lower case, or a combination of upper and lower case.
- Place no more than one SQL command (SELECT, FROM, etc.) per line. Feel free to combine functions such as SUBSTRING, POSITION, AVG on a single line.
- Leave spaces after the SQL reserved word and the directives of the command. Use the TAB key to improve readability.

The code and output for each question on each part of the project should be identifiable - please write down which question of the assignment is satisfied with the code and output. Again, note the sample question #2 above. The number of the question is clearly labeled. There will be many pages that must be turned in for the assignment, so please clip them with a paper clip.

Please make the result tables and SQL code large enough to be readable. The samples provided on this document are readable – please do not make yours smaller. Use landscape formatting in Word if necessary to keep it readable.

Your query should produce accurate output. Make sure your query will work correctly with any input data. Our test dataset, like all test datasets, is not comprehensive. Try and think of situations where your query could produce an incorrect result table and make sure your SQL code will handle those situations.

Make queries that continue to work when a year changes. For example, if a query asks to see all jobs accepted in January of the current year, don't limit it to the jobs accepted in January of 2015. Make your SQL code work for **any** current year so that the query will work in the future without modification.

Here are the questions:

This database is created to store estimates and actual direct job costs (labor and materials) for a hard surface subcontracting company called CutGlass Mosaic and Tile. Full result tables are provided for all questions to help you test your SQL code.

1. Which clients are located in Reno? List the ClientID, ClientName, ClientCity, ClientState, ClientZip and Phone. Result table:

	ClientId	ClientName	ClientCity	ClientState	ClientZip	Phone
1	2924	Casem, Brokaw and Stuart, LLC	Reno	NV	89519	7757261335
2	4435	Dew Drop Inn Luxury Suites	Reno	NV	89509	7754316404
3	4469	Fran and Harrold Meyers	Reno	NV	89509	7758652686
4	5012	Less Furniture Company	Reno	NV	89509	7757089540

2. Change the format of the output from question #1 to add column aliases, format the phone number, and sort the result table by ClientZip. Result table:

	ClientId	Client Name	Client Billing City	Client Billing State	Client Billing Zip	Client Billing Phone
1	4435	Dew Drop Inn Luxury Suites	Reno	NV	89509	(775) 431-6404
2	4469	Fran and Harrold Meyers	Reno	NV	89509	(775) 865-2686
3	5012	Less Furniture Company	Reno	NV	89509	(775) 708-9540
4	2924	Casem, Brokaw and Stuart, LLC	Reno	NV	89519	(775) 726-1335

3. List the JobID, JobName, DateProposed, DateAccepted, EmpManagerID, and PrimaryJobID from the Job table. Create a message in a column called "JobCompletedMessage" that says "Job Not Finished" if the JobComplete flag is 0, and says "Job Finished" if the JobComplete flag is 1. If the DateAccepted is NULL, then put a message in the column that says "Not Accepted". Sort the result table by DateProposed so that the most recent date is the first row in the result table. Hints: Use the CASE statement to generated the JobCompletedMessage. Use the ISNULL function to generate the "Not Accepted" message in the DateAccepted field. Use the CONVERT function to change the date from a date field to a to a varchar field. Result table:

	JobID	JobName	Date Proposed	Date Accepted	EmpManagerID	PrimaryJobID	Job Completed Message
1	15771	Main Showroom, Entry Area	Feb 02, 2015	Not Accepted	NULL	NULL	Job Not Finished
2	78431	Custom Stained Glass Part.	Nov 01, 2014	Nov 18, 2014	6460	NULL	Job Not Finished
3	62254	Dew Drop, Meadow Wood-1	Sep 14, 2014	Oct 04, 2014	4702	NULL	Job Finished
4	62257	Dew Drop, Meadow Wood-2	Sep 14, 2014	Oct 04, 2014	4702	62254	Job Finished
5	16885	Hampstead, Bathroom #2	Jul 11, 2014	Jul 25, 2014	NULL	32687	Job Not Finished
6	32687	Hampstead, Bathroom 1	Jul 11, 2014	Jul 25, 2014	NULL	NULL	Job Finished
7	91584	Restaurant Remodel	Mar 18, 2014	Mar 20, 2014	4702	NULL	Job Finished
8	55841	AO Reid - Wonder Valley PH 2-2	May 20, 2013	Jun 17, 2013	7651	55873	Job Finished
9	55873	AO Reid - Wonder Valley PH 2-3	May 20, 2013	Jun 17, 2013	7651	55873	Job Finished
10	55878	AO Reid - Wonder Valley PH 2-4	May 20, 2013	Jun 17, 2013	7651	55873	Job Finished

4. Display information from the JobTask table about job tasks that were completed two years ago. List the JobID, TaskID, DateStarted, the estimated material cost per square foot, the estimated labor cost per square foot, and the estimated labor cost per estimated hour. Sort the result table by taskID within jobid. Make sure that your query determine which tasks were completed two years ago based on the current year. For example, we all know that the current year is 2015, so two years ago is 2013. However, I don't want you to code '2013' in the SQL WHERE clause. I want this query to be able to be run in 2016, and have it show tasks from two years ago as 2014 without having to change the actual query code. Result table:

	JobID	TaskID	DateStarted	EstMaterialCostPerSqFt	EstLaborCostPerSqFt	EstLaborCostPerHour
1	55841	130	2013-07-15	0.8829	0.8829	10.375
2	55841	150	2013-07-16	1.1702	0.4787	11.25
3	55841	160	2013-07-18	2.5531	1.8085	15.4545
4	55841	170	2013-07-22	0.0957	1.4893	14.00
5	55841	180	2013-07-29	0.1382	0.617	11.60
6	55873	130	2013-08-14	0.8829	1.1702	13.75
7	55873	150	2013-08-16	1.1702	0.4787	11.25
8	55873	160	2013-08-19	2.5531	1.8085	15.4545
9	55873	170	2013-08-26	0.0957	1.4893	14.00
10	55873	180	2013-09-02	0.1382	0.617	11.60
11	55878	130	2013-09-12	0.8829	1.1702	13.75
12	55878	150	2013-09-16	1.1702	0.4787	11.25
13	55878	160	2013-09-19	2.5531	1.8085	15.4545
14	55878	170	2013-09-24	0.0957	1.4893	14.00
15	55878	180	2013-09-30	0.1382	0.5638	10.60

5. What is the average EstMaterialCostPerSqFt, and EstLaborCostPerSqFt for all tasks completed two years ago? What was the largest and smallest of those values? Round the result to two digits after the decimal point. Result table:

	Average EstMaterialCostPerSqFt	Largest EstMaterialCostPerSqFt	Smallest EstMaterialCostPerSqFt	Average EstLaborCostPerSqFt	Largest EstLaborCostPerSqFt	Smallest EstLaborCostPerSqFt
1	0.97	2.55	0.10	1.09	1.81	0.48

6. Most tasks on most jobs are finished within 3 days of the start date for the task. List only those tasks in the JobTask table that took more than 3 days to complete. For any task that took over 5 days to complete, put a message in a created column that says "Late Completion – Investigate". Sort the output by taskID within JobID. Result table:

	JobID	TaskID	DateStarted	DateCompleted	DaysToComplete	CompletionMessage
1	16885	130	2014-12-12	2014-12-16	4	NULL
2	16885	140	2014-11-24	2014-11-28	4	NULL
3	62254	150	2014-10-14	2014-10-24	10	Late Completion - Investigate
4	62254	160	2014-10-16	2014-10-30	14	Late Completion - Investigate
5	62254	170	2014-10-21	2014-10-31	10	Late Completion - Investigate
6	62254	180	2014-10-30	2014-11-03	4	NULL
7	62257	150	2014-10-30	2014-11-03	4	NULL
8	62257	160	2014-11-03	2014-11-07	4	NULL
9	62257	170	2014-11-07	2014-11-12	5	NULL
10	62257	180	2014-11-13	2014-11-18	5	NULL
11	78431	140	2014-11-17	2014-12-15	28	Late Completion - Investigate
12	91584	160	2014-04-07	2014-04-11	4	NULL

7. Summarize the amount of time worked in the TimeSheet table by job and task. Sort the result table by taskID within jobID. The result table should produce 43 rows. I have broken the result table below into two pieces to fit over two pages.

	JobID	TaskID	NumberOfTimeCards	TotalHoursWorked
1	NULL	NULL	7	9.60
2	16885	110	2	7.00
3	16885	130	6	23.50
4	16885	140	4	9.00
5	16885	150	1	4.50
6	16885	160	5	19.25
7	16885	270	2	7.50
8	32687	110	2	7.50
9	32687	130	2	8.00
10	32687	150	1	4.50
11	32687	160	3	10.00
12	32687	170	2	6.50
13	32687	180	1	4.50
14	55841	130	2	7.50
15	55841	150	1	4.00
16	55841	160	3	10.50
17	55841	170	3	10.50
18	55841	180	1	4.00

19	55873	130	1	5.75
20	55873	150	2	4.25
21	55873	160	3	11.30
22	55873	170	3	10.50
23	55873	180	1	5.00
24	55878	130	2	8.00
25	55878	150	1	4.50
26	55878	160	2	10.75
27	55878	170	3	9.75
28	55878	180	1	5.00
29	62254	150	5	19.50
30	62254	160	16	61.25
31	62254	170	11	40.50
32	62254	180	9	32.00
33	62257	150	5	20.25
34	62257	160	16	64.50
35	62257	170	12	39.00
36	62257	180	8	30.75
37	78431	140	8	23.65
38	78431	260	9	35.50
39	91584	150	9	34.00
40	91584	160	19	74.25
41	91584	170	6	18.75
42	91584	180	2	7.50
43	91584	230	4	16.50

8. Modify the query written for question #7 to display only those rows where the NumberofTimeCards is greater than 5. For any timecards with null values for the jobID or taskID, put a message in the JobID and TaskID columns. Sort the result table by TaskID within JobID. Result table:

	JobID	TaskID	NumberOfTimeCards	TotalHoursWorked
1	16885	130	6	23.50
2	62254	160	16	61.25
3	62254	170	11	40.50
4	62254	180	9	32.00
5	62257	160	16	64.50
6	62257	170	12	39.00
7	62257	180	8	30.75
8	78431	140	8	23.65
9	78431	260	9	35.50
10	91584	150	9	34.00
11	91584	160	19	74.25
12	91584	170	6	18.75
13	No JobID	No TaskID	7	9.60

9. Most of the employee time sheets in the TimeSheet table are for activities that are assigned directly to a task on a job. However, CutGlass occasionally has employee time sheets (time that they pay) for activities that are not directly assigned to a task on a job. When an employee's time sheet is assigned to a task on a job, then the "activity" column is null. For other activities, the activity column has a value. The goal of this query is to display summarized information about employee time sheets that do not have a null value in the activity column. This query should help you learn about how to display data that is not null. Display the EmpID, the total number of hours worked, and a count of the time sheets for that situation. Result table:

	EmpID	TotalHoursNotAssignedtoJob	NumberofTimeSheetsNotAssignedtoJob
1	2480	3.30	4
2	5291	1.00	1
3	5862	4.00	1
4	7656	1.30	1

10. Let's get familiar with the EmployeePay table in this query. This query is a little difficult to understand, but very important to the next two homework assignments. The EmployeePay table keeps track of the employee HourlyPayRate over time. The assumption is that an employee's HourlyPayRate may change based on date. In the EmployeePay table you created for HW#3, the HourlyPayRate always goes up over time. However, that may not always be true. It is also possible that an HourlyPayRate might go down, depending on the economy, an employee's performance, and an employee's work circumstances (i.e. moving from full-time to part-time). The ultimate goal for our job costing application is to make sure that the HourlyPayRate earned by an employee is correctly applied to the direct labor costs recorded on a time sheet. Since we aren't joining tables in this assignment, I just want to be sure that you know how to access the pay rate for a particular employee for a particular date. The same general query should work for all of the questions below, you should only have to change the date in the WHERE clause to make them work. You only have to turn in the SQL code for the first date, as long as that query will work for all 3.

What is the hourly pay rate for EmpID 6460 on January 15, 2014? Result table:

	EmpID	HourlyPayRate
1	6460	24.50

What is the hourly pay rate for EmpID 6460 on June 18, 2014? Result table:

	EmpID	HourlyPayRate
1	6460	27.50

What is the hourly pay rate for EmpID 6460 on November 8, 2011? Result table:

	EmpID	HourlyPayRate
1	6460	24.50

11. Summarize the MaterialPurchased in the MaterialPurchased table by the year of the date purchased. Include the year of the date purchased, the sum of the quantity, a count of the number of POID's, and the extended purchase cost (quantity * CostPerUOM). Result table:

	Purchase Year	Number of Purchase Orders	Total Quantity Purchased	Total Amount Purchased
1	2013	8	706.00	1730.300000
2	2014	26	5361.00	39006.190000

12. Summarize the MaterialAssigned in the MaterialAssigned table by JobID and the year of the dateassigned. Display the year of the dateassigned, the jobID and the sum of the quantity of the material assigned. Sort the result table by jobID within year of date assigned. Result table:

	Assigned Year	JobID	Total Material Quantity Assigned
1	2013	16885	157.00
2	2013	55841	146.00
3	2013	55873	38.00
4	2013	55878	34.50
5	2014	16885	488.33
6	2014	32687	144.00
7	2014	55878	0.50
8	2014	62254	260.00
9	2014	62257	260.00
10	2014	78431	226.25
11	2014	91584	1876.67
12	2015	16885	1.00
13	2015	78431	122.00

13. Which client(s) do not currently have a Job in the job table? Hint: this query requires the use of a non-correlated sub-query. Result table:

	clientname
1	3 Gals From Verona
2	Aero Professional Corp.
3	Fran and Harrold Meyers
4	Kelly Property Development
5	Lloyds Casino Properties, LLC

14. Which client(s) had a job in the Job table where the year of the dateaccepted for the job was in 2014? Hint: this query requires the use of a non-correlated sub-query. Result table:

	ClientName	ClientZip	Email
1	Adam's Rib Restaurant	96161	Jean@adamsribsrestaurant.inv
2	Casem, Brokaw and Stuart, LLC	89519	CCasem@gmail.inv
3	Dew Drop Inn Luxury Suites	89509	MarySmith@dewdropproperties.inv
4	Ms. Catherine Hampstead	89541	Champstead@abner.rr.inv

15. What are the names of the employees who have not worked as a manager for a job in the Job table? Hint: this query requires the use of a non-correlated sub-query, and you need to consider that some of the empmanagerID's in the Job table are null values. Result table:

	EmployeeName
1	Allen, N.
2	Bridges, C.
3	Burgess, D.
4	Fleming, C.
5	Hazelton, B.
6	Kinney, D.
7	Riggs, E.
8	Wiggins, C.