Database Design and Implementation
HW 04

Team 08
Henriod, Terence
Santoyo, Jorge
Singh, Raja

February 16, 2015

Abstract

An introductory assignment to become familiarized with writing queries using the SQL SELECT statement.

1 Assignment Background

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.

2 SQL Query Problems

1. *Solution*:

```
SELECT ClientID 'Clientid',
   ClientName,
   ClientCity,
   ClientState,
   ClientZip,
   Phone  'ClientPhone'
FROM Client
WHERE ClientCity = 'Reno';
```

	Clientid	ClientName	ClientCity	ClientState	ClientZip	ClientPhone
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

Figure 1: The solution to problem 1.

2. *Solution*:

```
SELECT  ClientID 'ClientId',
    ClientName 'Client Name ',
    ClientCity 'Client Billing City ',
    ClientState 'Client Billing State',
    ClientZip 'Client Billing Zip',
      STUFF(STUFF(STUFF(Phone,1,0,'('),6,0,') '),11,0,'-') as 'Client Billing Phone'
 FROM Client
 WHERE ClientCity = 'Reno'
 ORDER BY ClientZip;
```

	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	(7754) 316-404
2	4469	Fran and Harrold Meyers	Reno	NV	89509	(7758) 652-686
3	5012	Less Furniture Company	Reno	NV	89509	(7757) 089-540
4	2924	Casem, Brokaw and Stuart, LLC	Reno	NV	89519	(7757) 261-335

Figure 2: The solution to problem 2.

3. *Solution*:

```
 SELECT
    JobID 'Job ID'
  , JobName 'Job Name'
  , DateProposed  'Date Proposed'
  , ISNULL(CONVERT(VARCHAR, DateAccepted, 120), 'Not Accepted') 'Date Accepted'
  , EmpManagerID  'Employee Manager ID'
  , PrimaryJobID  'Primary Job ID'
  , CASE
    WHEN JobCompleted = 1
      THEN 'Job Finished'
    ELSE
      'Job Not Finished'
    END 'Job Completed Message'
FROM
  Job
ORDER BY
  DateProposed DESC
;
```

	Job ID	Job Name	Date Proposed	Date Accepted	Employee Manager ID	Primary Job ID	Job Completed Message
1	15771	Main Showroom, Entry Area	2015-02-02	Not Accepted	NULL	NULL	Job Not Finished
2	78431	Custom Stained Glass Part.	2014-11-01	2014-11-18	6460	NULL	Job Not Finished
3	62254	Dew Drop, Meadow Wood-1	2014-09-14	2014-10-04	4702	NULL	Job Finished
4	62257	Dew Drop, Meadow Wood-2	2014-09-14	2014-10-04	4702	62254	Job Finished
5	32687	Hampstead, Bathroom 1	2014-07-25	2014-07-25	NULL	NULL	Job Finished
6	16885	Hampstead, Bathroom #2	2014-07-11	2014-07-25	NULL	32687	Job Not Finished
7	91584	Restaurant Remodel	2014-03-18	2014-03-20	4702	NULL	Job Finished
8	55841	AO Reid - Wonder Valley PH 2-2	2013-05-20	2013-06-17	7651	55873	Job Finished
9	55873	AO Reid - Wonder Valley PH 2-3	2013-05-20	2013-06-17	7651	55873	Job Finished
10	55878	AO Reid - Wonder Valley PH 2-4	2013-05-20	2013-06-17	7651	55873	Job Finished

Figure 3: The solution to problem 3.

4. *Solution*:

```
SELECT JobID,
     TaskID,
     DateStarted,
     EstMaterialCost/SquareFeet 'EstMaterialCostperSqft',
     EstLaborCost/SquareFeet 'EstLaborCostperSqft',
     EstLaborCost/EstHours 'EstLaborCostperHr'
FROM JobTask
WHERE DATEDIFF(YYYY,  DateCompleted, GETDATE()) = 2
ORDER BY TaskID;
```

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

Figure 4: The solution to problem 4.

5. *Solution*:

```
SELECT
  ROUND(AVG(EstMaterialCost/SquareFeet),2) 'Average EstMaterialCostPerSqFt',
  ROUND(MAX(EstMaterialCost/SquareFeet),2) 'Largest EstMaterialCostPerSqFt',
  ROUND(MIN(EstMaterialCost/SquareFeet),2) 'Smallest EstMaterialCostPerSqFt',
  ROUND(AVG(EstLaborCost/SquareFeet),2) 'Average EstLaborCostPerSqFt',
  ROUND(MAX(EstLaborCost/SquareFeet),2) 'Largest EstLaborCostPerSqFt',
  ROUND(MIN(EstLaborCost/SquareFeet),2) 'Smallest EstLaborCostPerSqFt'
FROM
  JobTask
WHERE
  DateCompleted is not NULL AND
  DATEDIFF(YEAR, DateCompleted, GETDATE()) = 2;
```

	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

Figure 5: The solution to problem 5.

6. *Solution*:

```
/** Assumption: if a job is not currently complete, it makes
 * no sense to compute a completion time yet - let the result
 * be NULL. Also, don't consider incomplete jobs.
 */
SELECT
      JobID 'Job ID'
    , TaskID 'TaskID'
    , DateStarted 'Date Started'
    , DateCompleted 'Date Completed'
    , DATEDIFF(DAY, DateStarted, DateCompleted)
    , CASE
      WHEN DATEDIFF(DAY, DateStarted, DateCompleted) > 5
        THEN 'Late Completion - Investigate'
      ELSE
        '' -- No message for typical completion times was specified
      END 'Message'
FROM
  JobTask
WHERE
      DateCompleted IS NOT NULL
  AND DATEDIFF(DAY, DateStarted, DateCompleted) > 3
ORDER BY
    JobID
  , TaskID
;
```

	Job ID	TaskID	Date Started	Date Completed	(No column name)	Message
1	16885	130	2014-12-12	2014-12-16	4	
2	16885	140	2014-11-24	2014-11-28	4	
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	
7	62257	150	2014-10-30	2014-11-03	4	
8	62257	160	2014-11-03	2014-11-07	4	
9	62257	170	2014-11-07	2014-11-12	5	
10	62257	180	2014-11-13	2014-11-18	5	
11	78431	140	2014-11-17	2014-12-15	28	Late Completion - Investigate
12	91584	160	2014-04-07	2014-04-11	4	

Figure 6: The solution to problem 6.

7. *Solution*:

```
SELECT
  JobID,
  TaskID,
  Count(*) 'NumberOfTimeCards',
  SUM(HoursWorked) 'TotalHoursWorked'
FROM
  TimeSheet
GROUP BY
  JobID,TaskID
ORDER BY
  JobID,TaskID
```

	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

Figure 7: The solution to problem 7 (the first half).

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

Figure 8: The solution to problem 7 (the second half).

8. *Solution*:

```
SELECT
  ISNULL(CONVERT(VARCHAR, JobID, 120),'No JobID') 'JobID',
  ISNULL(CONVERT(VARCHAR, TaskID, 120),'No TaskID') 'TaskID',
  COUNT(*)  'NumberOfTimeCards',
  SUM(HoursWorked) 'TotalHoursWorked'
FROM
  TimeSheet
GROUP BY
  JobID,TaskID
HAVING
  COUNT(*) > 5
ORDER BY
  JobID ,TaskID
```

Results | Messages

	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

Figure 9: The solution to problem 8.

9. *Solution*:

```
SELECT
    EmpID 'Employee ID'
  , SUM(HoursWorked) 'Total Hours Not Assigned To Job'
  , COUNT(*)'Time Sheets Not Assigned To Job'
FROM
  TimeSheet
WHERE
  Activity IS NOT NULL
GROUP BY
  EmpID
;
```

	Employee ID	Total Hours Not Assigned To Job	Time Sheets Not Assigned To Job
1	2480	3.30	4
2	5291	1.00	1
3	5862	4.00	1
4	7656	1.30	1

Figure 10: The solution to problem 9.

10. *Solution*:

```
SELECT EmpID,
       HourlyPayRate
FROM EmployeePay
WHERE EmpID = 6460 AND( ((convert(date, 'June 18, 2013')) BETWEEN DateStartPay AND DateEnd) O
```

	EmpID	HourlyPayRate
1	6460	24.50

Figure 11: The solution to problem 10.

11. *Solution*:

```
SELECT
  DISTINCT YEAR(DatePurchased) 'Purchase Year',
  COUNT(*) 'Number of Purchase Orders',
  SUM(Quantity) 'Total Quantity Purchased',
  SUM(Quantity * CostPerUOM) 'Total Amount Purchased'
FROM
  MaterialPurchased
GROUP BY
  YEAR(DatePurchased);
```

	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

Figure 12: The solution to problem 11.

12. *Solution*:

```
SELECT
    YEAR(DateAssigned) 'Assigned Year'
  , JobID 'JobID'
  , SUM(Quantity) 'Total Material Quantity Assigned'
FROM
  MaterialAssigned
GROUP BY
    YEAR(DateAssigned)
  , JobID
ORDER BY
    YEAR(DateAssigned)
  , JobID
;
```

	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

Figure 13: The solution to problem 12.

13. *Solution*:

```
SELECT
  ClientName
FROM
  Client
WHERE
  ClientID not in (
    SELECT
      ClientID
    FROM
      JOB
  )
ORDER BY
  ClientName;
```

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

Figure 14: The solution to problem 13.

14. *Solution*:

```
SELECT ClientName,
     ClientZip,
     Email
From Client
WHERE ClientID In (
    SELECT ClientID
    FROM Job
    WHERE 2014 = DATEPART(YEAR, DateAccepted))
ORDER BY ClientName;
```

	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

Figure 15: The solution to problem 14.

15. *Solution*:

```
SELECT
  (LastName + ', ' + SUBSTRING(FirstName, 1, 1) + '.') 'Employee Name'
FROM
  Employee
WHERE
  EmpID NOT IN (
    SELECT
      EmpManagerID
    FROM
      Job
    WHERE
      EmpManagerID IS NOT NULL
  )
ORDER BY
  LastName
;
```

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

Figure 16: The solution to problem 15.