Database Design and Implementation
HW 05

Team 08
Henriod, Terence
Santoyo, Jorge
Singh, Raja
February 26, 2015

```
/**
 * IS675 - HW05
 * Using SQL Operations on Multiple Tables
 *
 * Division of Labor
 * Raja: exercises where (i % 3) == 1
 * Jorge: exercises where (i % 3) == 2
 * Terence: exercises where (i % 3) == 0
 */

/**
 * Pre-homework table modification
 *
 * Before starting this homework assignment, I want you to change the data
 * type of two fields in one table – the EmployeePay table. In the
 * EmployeePay table, I want the DateStartPay and DateEnd fields to
 * be datetime data types rather than date data types. The easiest way
 * to complete this change is to just drop the table, create it, and
 * repopulate it. Some of the DateEnd fields need to have a time in them,
 * so it seemed most expedient if I just created a SQL script file that
 * would accomplish those goals for you. The file is called
 * BuildEmployeePayHW5.sql and is located on the K: drive in the
 * IS475\CutGlassS15 folder.
 */
-- Just used the given file.

/**
 * Exercise 01
 *
 * Modify the Job table:
 */
-- Part a
-- Add a column to the table for DateDue. It should be a date data type.
ALTER TABLE
       [Job]
ADD
       [DateDue] [date] --NOT NULL?
;

-- Part b
-- UPDATE the new column with the DateDue data given below. I list the
-- JobID and the new DateDue for each job. If using the UPDATE command,
-- you will have to write 10 UPDATE statements.
UPDATE
       [Job]
SET
       [DateDue] = '2015-01-15'
WHERE
       [JobID] = 16885
;
UPDATE
       [Job]
SET
       [DateDue] = '2014-08-01'
WHERE
       [JobID] = 32687
;
UPDATE
       [Job]
SET
       [DateDue] = '2013-08-01'
WHERE
```

```
	[JobID] = 55841
;
UPDATE
	[Job]
SET
	[DateDue] = '2013-10-01'
WHERE
	[JobID] = 55873
;
UPDATE
	[Job]
SET
	[DateDue] = '2013-09-15'
WHERE
	[JobID] = 55878
;
UPDATE
	[Job]
SET
	[DateDue] = '2014-11-20'
WHERE
	[JobID] = 62254
;
UPDATE
	[Job]
SET
	[DateDue] = '2014-11-18'
WHERE
	[JobID] = 62257
;
UPDATE
	[Job]
SET
	[DateDue] = '2015-02-06'
WHERE
	[JobID] = 78431
;
UPDATE
	[Job]
SET
	[DateDue] = '2014-04-01'
WHERE
	[JobID] = 91584
;
--uncomment to run select query to see results
SELECT
	[JobID],
	[DateDue]
from [JOB] ;
```

Results | Messages

	JobID	DateDue
1	15771	NULL
2	16885	2015-01-15
3	32687	2014-08-01
4	55841	2013-08-01
5	55873	2013-10-01
6	55878	2013-09-15
7	62254	2014-11-20
8	62257	2014-11-18
9	78431	2015-02-06
10	91584	2014-04-01

```
/**
 * Exercise 02
 *
 * List all jobs with a DateDue in the previous year (i.e. 2014 if the code
 * is run in 2015). Use GETDATE() to determine the previous year. Calculate
 * the difference in days between the Date a job was accepted and the date
 * it was due.
 */
 SELECT
       JobId,
       JobName,
       ClientName,
       ClientZip,
       DateAccepted,
       DateDue,
       DATEDIFF(DAY,DateAccepted,DateDue) 'Days To Complete',
       EmpManagerID
From
       Job INNER JOIN Client ON
             Job.ClientID = Client.ClientID
WHERE
       DATEDIFF(YEAR,DateDue,GETDATE()) = 1
ORDER BY
       JobID;
```

Results | Messages

	JobId	JobName	ClientName	ClientZip	DateAccepted	DateDue	Days To Complete	EmpManagerID
1	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	2014-07-25	2014-08-01	7	NULL
2	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	2014-10-04	2014-11-20	47	4702
3	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	2014-10-04	2014-11-18	45	4702
4	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	2014-03-20	2014-04-01	12	4702

```sql
/**
 * Exercise 03
 *
 * Modify the query written for question #2 to replace the EmpManagerID
 * in the result table with the name of the employee who was the manager
 * for the purchase order. If the manager is null, display the message
 * "No Manager" in the column. If the manager is not null, then concatenate
 * the FirstName and LastName of the employee into a single column with
 * a space between the first and last names.
 */
SELECT
	[JobID] AS 'JobID',
	[JobName] AS 'Job Name',
	[ClientName] AS 'Client Name',
	[ClientZip] AS 'Client Zip',
	CONVERT(varchar(12), [DateAccepted],107 )AS 'Date Accepted',
	CONVERT ( varchar(12), [DateDue], 107)  AS 'Date Due',
	DATEDIFF(DAY, ISNULL([DateAccepted], GETDATE()), ISNULL([DateDue], GETDATE()))
		AS 'Days To Complete',
	CASE
		WHEN [EmpManagerId] IS NULL
			THEN 'No Manager'
		ELSE
			[Employee].[FirstName] + ' ' + [Employee].[LastName]
	END AS 'Manager Name'
FROM
	[Job]
	INNER JOIN [Client] ON
		[Job].[ClientID] = [Client].[ClientID]
	LEFT OUTER JOIN [Employee] ON
		[Job].[EmpManagerID] = [Employee].[EmpID]
WHERE
	DATEDIFF(YEAR, [DateDue], GETDATE()) = 1
ORDER BY
	[JobID]
;
```

Results | Messages

	JobID	Job Name	Client Name	Client Zip	Date Accepted	Date Due	Days To Complete	Manager Name
1	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager
2	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker
3	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker
4	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker

```sql
/**
 * Exercise 04
 *''
 * Modify the query written for question #3 to include information about
 * the tasks on each of the jobs listed in query #3, as shown below. The
 * data about each task on a job includes the taskID, DateStarted, and
 * DateCompleted for the task. The result table should be sorted by taskID
 * within JobID. JobID should be the primary sort key.
 */
 SELECT
                [Job].JobID AS 'JobID',
                [JobName] AS 'Job Name',
                [ClientName] AS 'Client Name',
                [ClientZip] AS 'Client Zip',
                CONVERT (varchar(12), [DateAccepted],107 )AS 'Date Accepted',
                CONVERT ( varchar(12), [DateDue], 107)  AS 'Date Due',
                DATEDIFF(DAY, ISNULL([DateAccepted], GETDATE()), ISNULL([DateDue], GETDATE()))
                AS 'Days To Complete',
                CASE
                     WHEN [EmpManagerId] IS NULL
                            THEN 'No Manager'
                     ELSE
                            [Employee].[FirstName] + ' ' + [Employee].[LastName]
                END AS 'Manager Name',
                [JobTask].TaskID As 'taskID',
                CONVERT (varchar(12), [JobTask].DateStarted, 107) AS 'DateStarted ',
                CONVERT (varchar (12), [JobTask].DateCompleted, 107) AS 'DateCompleted'
FROM
                [Job]
INNER
JOIN      [Client]
ON              [Job].[ClientID] = [Client].[ClientID]
LEFT
OUTER
JOIN      [Employee]
ON              [Job].[EmpManagerID] = [Employee].[EmpID]
INNER
JOIN      [JobTask]
ON              [Job].JobID = [JobTask].JobID
WHERE
                DATEDIFF(YEAR, [DateDue], GETDATE()) = 1
ORDER BY  [JobID]
```

Results | Messages

	JobID	Job Name	Client Name	Client Zip	Date Accepted	Date Due	Days To Complete	Manager Name	taskID	DateStarted	DateCompleted
1	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	110	Jul 31, 2014	Jul 31, 2014
2	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	130	Aug 04, 2014	Aug 04, 2014
3	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	150	Aug 04, 2014	Aug 07, 2014
4	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	160	Aug 11, 2014	Aug 14, 2014
5	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	170	Aug 18, 2014	Aug 18, 2014
6	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	180	Aug 22, 2014	Aug 22, 2014
7	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	150	Oct 14, 2014	Oct 24, 2014
8	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	160	Oct 16, 2014	Oct 30, 2014
9	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	170	Oct 21, 2014	Oct 31, 2014
10	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	180	Oct 30, 2014	Nov 03, 2014
11	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	150	Oct 30, 2014	Nov 03, 2014
12	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	160	Nov 03, 2014	Nov 07, 2014
13	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	170	Nov 07, 2014	Nov 12, 2014
14	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	180	Nov 13, 2014	Nov 18, 2014
15	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	150	Mar 31, 2014	Apr 02, 2014
16	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	160	Apr 07, 2014	Apr 11, 2014
17	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	170	Apr 14, 2014	Apr 14, 2014
18	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	180	Apr 18, 2014	Apr 21, 2014
19	91584	Restaurant Remodel	Adam's Rib Restaurant	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	230	Apr 03, 2014	Apr 04, 2014

```
/**
 * Exercise 05
 *
 * Modify the query written for question #4 to remove the JobName and
 * ClientName and include the TaskDescription.
 */
 SELECT
       Job.JobId,
       Client.ClientZip,
       convert(varchar, Job.DateAccepted, 109) 'Date Accepted',
       convert(varchar, Job.DateDue, 109) 'Date Due',
       DATEDIFF(DAY,Job.DateAccepted,Job.DateDue) 'Days To Complete',
       CASE
              WHEN Job.EmpManagerID is null
                     THEN 'No Manager'
              ELSE
                     Employee.FirstName + ' ' + Employee.LastName
              END 'Manager Name',
       Task.TaskId,
       Task.TaskDescription,
       convert(varchar,JobTask.DateStarted,109) 'Date Started',
       convert(varchar,JobTask.DateCompleted,109) 'Date Completed'
From
       Job INNER JOIN Client ON
              Job.ClientID = Client.ClientID
       LEFT OUTER JOIN Employee ON
              Employee.EmpID = Job.EmpManagerID
       INNER JOIN JobTask ON
              Job.JobID = JobTask.JobID
       INNER JOIN Task ON
              JobTask.TaskID = Task.TaskID
WHERE
       DATEDIFF(YEAR,Job.DateDue,GETDATE()) = 1
ORDER BY
       JOb.JobID;
```

Results | Messages

	JobId	ClientZip	Date Accepted	Date Due	Days To Complete	Manager Name	TaskId	TaskDescription	Date Started	Date Completed
1	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	110	Remove Existing Floor	Jul 31 2014	Jul 31 2014
2	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	130	Install sub-floor	Aug 4 2014	Aug 4 2014
3	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	150	Prepare sub floor	Aug 4 2014	Aug 7 2014
4	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	160	Install tile floor	Aug 11 2014	Aug 14 2014
5	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	170	Grout	Aug 18 2014	Aug 18 2014
6	32687	89541	Jul 25 2014	Aug 1 2014	7	No Manager	180	Seal and finish work	Aug 22 2014	Aug 22 2014
7	62254	89509	Oct 4 2014	Nov 20 2014	47	Vance Martin Walker	150	Prepare sub floor	Oct 14 2014	Oct 24 2014
8	62254	89509	Oct 4 2014	Nov 20 2014	47	Vance Martin Walker	160	Install tile floor	Oct 16 2014	Oct 30 2014
9	62254	89509	Oct 4 2014	Nov 20 2014	47	Vance Martin Walker	170	Grout	Oct 21 2014	Oct 31 2014
10	62254	89509	Oct 4 2014	Nov 20 2014	47	Vance Martin Walker	180	Seal and finish work	Oct 30 2014	Nov 3 2014
11	62257	89509	Oct 4 2014	Nov 18 2014	45	Vance Martin Walker	150	Prepare sub floor	Oct 30 2014	Nov 3 2014
12	62257	89509	Oct 4 2014	Nov 18 2014	45	Vance Martin Walker	160	Install tile floor	Nov 3 2014	Nov 7 2014
13	62257	89509	Oct 4 2014	Nov 18 2014	45	Vance Martin Walker	170	Grout	Nov 7 2014	Nov 12 2014
14	62257	89509	Oct 4 2014	Nov 18 2014	45	Vance Martin Walker	180	Seal and finish work	Nov 13 2014	Nov 18 2014
15	91584	96161	Mar 20 2014	Apr 1 2014	12	Vance Martin Walker	150	Prepare sub floor	Mar 31 2014	Apr 2 2014
16	91584	96161	Mar 20 2014	Apr 1 2014	12	Vance Martin Walker	160	Install tile floor	Apr 7 2014	Apr 11 2014
17	91584	96161	Mar 20 2014	Apr 1 2014	12	Vance Martin Walker	170	Grout	Apr 14 2014	Apr 14 2014
18	91584	96161	Mar 20 2014	Apr 1 2014	12	Vance Martin Walker	180	Seal and finish work	Apr 18 2014	Apr 21 2014
19	91584	96161	Mar 20 2014	Apr 1 2014	12	Vance Martin Walker	230	Install tile counter	Apr 3 2014	Apr 4 2014

```sql
/**
 * Exercise 06
 *
 * Final modification. Modify the query written for question #5 to include
 * a self-join with the Job table. The goal of the self-join is to include
 * the JobName of the primary job that is related to the job – I named the
 * JobName associated with the PrimaryJobID as PrimaryJobName in the result
 * table. I used the ISNULL function to put the message "No Primary Job
 * Name" into the PrimaryJobName column.
 */
SELECT
	[Job].[JobID] AS 'JobID',
	[Client].[ClientZip] AS 'Client Zip',
	CONVERT (varchar(12), [Job].[DateAccepted], 107) AS 'Date Accepted',
	CONVERT ( varchar(12), [Job].[DateDue], 107) AS 'Date Due',
	ISNULL(
		DATEDIFF(
			DAY,
			ISNULL([Job].[DateAccepted], GETDATE()),
			[Job].[DateDue]
		),
		9999 --Assumed that jobs without deadlines have no practical timeline
	) AS 'Days To Complete',
	CASE
		WHEN [Job].[EmpManagerId] IS NULL
			THEN 'No Manager'
		ELSE
			[Employee].[FirstName] + ' ' + [Employee].[LastName]
	END AS 'Manager Name',
	ISNULL([PrimaryJob].[JobName], 'No Primary Job Name') AS 'Primary Job Name',
	[JobTask].[TaskID] AS 'TaskID',
	[Task].[TaskDescription] AS 'Task Description',
	CONVERT(varchar(12), [JobTask].[DateStarted], 107) AS 'Date Started',
	CONVERT (varchar(12), [JobTask].[DateCompleted], 107) AS 'Date Completed'
FROM
	[Job]
		INNER JOIN [Client] ON
			[Job].[ClientID] = [Client].[ClientID]
		LEFT OUTER JOIN [Employee] ON
			[Job].[EmpManagerID] = [Employee].[EmpID]
		LEFT OUTER JOIN [Job] [PrimaryJob] ON
			[Job].[PrimaryJobID] = [PrimaryJob].[JobID],
	[JobTask]
		INNER JOIN [Task] ON
			[JobTask].[TaskID] = [Task].[TaskID]
WHERE
	[Job].[JobID] = [JobTask].[JobID] AND
	DATEDIFF(YEAR, [Job].[DateDue], GETDATE()) = 1
ORDER BY
	[JobID],
	[TaskID]
;
```

Results | Messages

	JobID	Client Zip	Date Accepted	Date Due	Days To Complete	Manager Name	Primary Job Name	TaskID	Task Description	Date Started	Date Completed
1	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	110	Remove Existing Floor	Jul 31, 2014	Jul 31, 2014
2	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	130	Install sub-floor	Aug 04, 2014	Aug 04, 2014
3	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	150	Prepare sub floor	Aug 04, 2014	Aug 07, 2014
4	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	160	Install tile floor	Aug 11, 2014	Aug 14, 2014
5	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	170	Grout	Aug 18, 2014	Aug 18, 2014
6	32687	89541	Jul 25, 2014	Aug 01, 2014	7	No Manager	No Primary Job Name	180	Seal and finish work	Aug 22, 2014	Aug 22, 2014
7	62254	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	No Primary Job Name	150	Prepare sub floor	Oct 14, 2014	Oct 24, 2014
8	62254	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	No Primary Job Name	160	Install tile floor	Oct 16, 2014	Oct 30, 2014
9	62254	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	No Primary Job Name	170	Grout	Oct 21, 2014	Oct 31, 2014
10	62254	89509	Oct 04, 2014	Nov 20, 2014	47	Vance Martin Walker	No Primary Job Name	180	Seal and finish work	Oct 30, 2014	Nov 03, 2014
11	62257	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	Dew Drop, Meadow Wood-1	150	Prepare sub floor	Oct 30, 2014	Nov 03, 2014
12	62257	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	Dew Drop, Meadow Wood-1	160	Install tile floor	Nov 03, 2014	Nov 07, 2014
13	62257	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	Dew Drop, Meadow Wood-1	170	Grout	Nov 07, 2014	Nov 12, 2014
14	62257	89509	Oct 04, 2014	Nov 18, 2014	45	Vance Martin Walker	Dew Drop, Meadow Wood-1	180	Seal and finish work	Nov 13, 2014	Nov 18, 2014
15	91584	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	No Primary Job Name	150	Prepare sub floor	Mar 31, 2014	Apr 02, 2014
16	91584	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	No Primary Job Name	160	Install tile floor	Apr 07, 2014	Apr 11, 2014
17	91584	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	No Primary Job Name	170	Grout	Apr 14, 2014	Apr 14, 2014
18	91584	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	No Primary Job Name	180	Seal and finish work	Apr 18, 2014	Apr 21, 2014
19	91584	96161	Mar 20, 2014	Apr 01, 2014	12	Vance Martin Walker	No Primary Job Name	230	Install tile counter	Apr 03, 2014	Apr 04, 2014

```sql
/**
 * Exercise 07
 *
 * Time for a new, but similar type of query! List information about all
 * JobTasks in the JobTask table that have a JobTask DateCompleted that
 * is greater than the DateDue for the Job (in the Job table). Sort the
 * output by TaskID within JobID.
 */
SELECT
              [JobTask].JobID AS 'JobID',
              [Job].JobName AS 'JobName',
              [Client].ClientName AS 'ClientName',
              [JobTask].TaskID AS 'TaskID',
              [Task].TaskDescription AS 'TaskDescription',
              [Job].DateDue AS 'DateDue',
              [JobTask].DateCompleted AS 'DateCompleted',
              DATEDIFF (DAY,  [Job].DateDue,[JobTask].[DateCompleted])  AS 'DaysOverdue'
FROM    [JobTask]
INNER
JOIN    [Job]
ON             [JobTask].JobID = [Job].JobID
INNER
JOIN    [Client]
ON             [Job].ClientID = [Client].ClientID
INNER
JOIN    [Task]
ON             [JobTask].TaskID = [Task].TaskID
WHERE   DATEDIFF (DAY,  [Job].DateDue,[JobTask].[DateCompleted])  > 0
ORDER BY [JobTask].JobID, [JobTask].TaskID;
```

Results | Messages

	JobID	JobName	ClientName	TaskID	TaskDescription	DateDue	DateCompleted	DaysOverdue
1	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	130	Install sub-floor	2014-08-01	2014-08-04	3
2	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	150	Prepare sub floor	2014-08-01	2014-08-07	6
3	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	160	Install tile floor	2014-08-01	2014-08-14	13
4	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	170	Grout	2014-08-01	2014-08-18	17
5	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	180	Seal and finish work	2014-08-01	2014-08-22	21
6	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	150	Prepare sub floor	2013-09-15	2013-09-16	1
7	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	160	Install tile floor	2013-09-15	2013-09-20	5
8	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	170	Grout	2013-09-15	2013-09-24	9
9	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	180	Seal and finish work	2013-09-15	2013-09-30	15
10	91584	Restaurant Remodel	Adam's Rib Restaurant	150	Prepare sub floor	2014-04-01	2014-04-02	1
11	91584	Restaurant Remodel	Adam's Rib Restaurant	160	Install tile floor	2014-04-01	2014-04-11	10
12	91584	Restaurant Remodel	Adam's Rib Restaurant	170	Grout	2014-04-01	2014-04-14	13
13	91584	Restaurant Remodel	Adam's Rib Restaurant	180	Seal and finish work	2014-04-01	2014-04-21	20
14	91584	Restaurant Remodel	Adam's Rib Restaurant	230	Install tile counter	2014-04-01	2014-04-04	3

```sql
/**
 * Exercise 08
 *
 * Modify query #7 so that it also shows the job tasks that haven't been
 * completed yet (DateCompleted is null) as of when your SQL code runs,
 * but are beyond the duedate of the job. I ran this SQL code on 2/18/2015,
 * so the DaysOverdue calculation reflect that current date. Your output
 * should reflect the current date when it is run.
 */
 SELECT
       Job.JobID,
       Job.JobName,
       Client.ClientName,
       JobTask.TaskId,
       Job.DateDue,
       ISNULL(CAST(JobTask.DateCompleted AS varchar),'Not Done') 'Date Completed',
       DATEDIFF(DAY,ISNULL(JobTask.DateCompleted,GETDATE()),Job.DateDue) * -1 'DaysOverDue'
FROM
       Job INNER JOIN CLient ON
              Job.ClientID = Client.ClientID
       INNER JOIN JobTask ON
              Job.JobID = JobTask.JobID
       INNER JOIN Task ON
              JobTask.taskID = Task.Taskid
WHERE
       DATEDIFF(DAY,JobTask.DateCompleted,Job.DateDue) < 0 or JobTask.DateCompleted is null
ORDER BY
       JobID;
```

Results | Messages

	JobID	JobName	ClientName	TaskId	DateDue	Date Completed	DaysOverDue
1	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	160	2015-01-15	Not Done	41
2	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	170	2015-01-15	Not Done	41
3	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	180	2015-01-15	Not Done	41
4	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	240	2015-01-15	Not Done	41
5	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	260	2015-01-15	Not Done	41
6	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	130	2014-08-01	2014-08-04	3
7	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	150	2014-08-01	2014-08-07	6
8	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	160	2014-08-01	2014-08-14	13
9	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	170	2014-08-01	2014-08-18	17
10	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	180	2014-08-01	2014-08-22	21
11	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	150	2013-09-15	2013-09-16	1
12	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	160	2013-09-15	2013-09-20	5
13	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	170	2013-09-15	2013-09-24	9
14	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	180	2013-09-15	2013-09-30	15
15	78431	Custom Stained Glass Part.	Casem, Brokaw and Stuart, LLC	190	2015-02-06	Not Done	19
16	78431	Custom Stained Glass Part.	Casem, Brokaw and Stuart, LLC	200	2015-02-06	Not Done	19
17	78431	Custom Stained Glass Part.	Casem, Brokaw and Stuart, LLC	260	2015-02-06	Not Done	19
18	91584	Restaurant Remodel	Adam's Rib Restaurant	150	2014-04-01	2014-04-02	1
19	91584	Restaurant Remodel	Adam's Rib Restaurant	160	2014-04-01	2014-04-11	10
20	91584	Restaurant Remodel	Adam's Rib Restaurant	170	2014-04-01	2014-04-14	13
21	91584	Restaurant Remodel	Adam's Rib Restaurant	180	2014-04-01	2014-04-21	20
22	91584	Restaurant Remodel	Adam's Rib Restaurant	230	2014-04-01	2014-04-04	3

```sql
/**
 * Exercise 09
 *
 * Time for a new query! List the material costs by task for JobID 91584.
 * The TotalCost is the CostPerUOM in the MaterialPurchased table
 * multiplied by the Quantity assigned in the MaterialAssigned table.
 * Cast or CONVERT the TotalCost to a money data type. Sort the result table
 * by materialID within taskID.
 */
SELECT
	[MaterialAssigned].[JobID] AS 'JobID',
	[MaterialAssigned].[TaskID] AS 'TaskID',
	[MaterialPurchased].[MaterialID] AS 'MaterialID',
	[MaterialAssigned].[Quantity] AS 'Quantity',
	[MaterialPurchased].[CostPerUOM] AS 'Cost Per UOM',
	CAST(
		([MaterialPurchased].[CostPerUOM] * [MaterialAssigned].[Quantity])
	AS MONEY)
		AS 'Actual Cost'
FROM
	[MaterialAssigned]
		INNER JOIN [MaterialPurchased] ON
			[MaterialAssigned].[POID] = [MaterialPurchased].[POID]
WHERE
	[MaterialAssigned].[JobID] = 91584
ORDER BY
	[MaterialAssigned].[TaskID],
	[MaterialPurchased].[MaterialID]
;
```

Results | Messages

	JobID	TaskID	MaterialID	Quantity	Cost Per UOM	Actual Cost
1	91584	150	2000	313.87	1.00	313.87
2	91584	160	1010	7.00	13.00	91.00
3	91584	160	1080	1300.00	8.95	11635.00
4	91584	160	2000	36.80	1.00	36.80
5	91584	170	1150	163.00	0.95	154.85
6	91584	180	1170	7.00	24.00	168.00
7	91584	230	1080	45.00	9.55	429.75
8	91584	230	1150	4.00	0.99	3.96

```sql
/**
 * Exercise 10
 *
 *  Modify query #9 to summarize the costs of materials by taskID.
 */
SELECT
	[MaterialAssigned].[JobID] AS 'JobID',
	[MaterialAssigned].[TaskID] AS 'TaskID',
	SUM([MaterialAssigned].[Quantity]) AS 'Actual Quantity',
	CAST(
		SUM([MaterialPurchased].[CostPerUOM] * [MaterialAssigned].[Quantity])
	AS MONEY)
		AS 'Actual Cost'
FROM
	[MaterialAssigned]
		INNER JOIN [MaterialPurchased] ON
			[MaterialAssigned].[POID] = [MaterialPurchased].[POID]
WHERE
	[MaterialAssigned].[JobID] = 91584
GROUP BY
	[MaterialAssigned].[JobID],
	[MaterialAssigned].[TaskID]
ORDER BY
	[MaterialAssigned].[TaskID]
;
```

	JobID	TaskID	Actual Quantity	Actual Cost
1	91584	150	313.87	313.87
2	91584	160	1343.80	11762.80
3	91584	170	163.00	154.85
4	91584	180	7.00	168.00
5	91584	230	49.00	433.71

```sql
/**
 * Exercise 11
 *
 * Modify query #10 to compare the actual material costs generated in
 * that query to the estimated material costs in the JobTask table.
 * Calculate the difference between the estimated cost of materials for
 * a task to the actual cost for the materials for a task.
 */
SELECT
	[MaterialAssigned].[JobID] AS 'JobID',
	[MaterialAssigned].[TaskID] AS 'TaskID',
	MIN([JobTask].[DateCompleted]) AS 'Date Completed',
	([JobTask].[EstMaterialCost]) AS 'Estimated Material Cost',
	CAST(
		SUM([MaterialPurchased].[CostPerUOM] * [MaterialAssigned].[Quantity])
	AS MONEY)
		AS 'Actual Cost',
	CAST(
		([JobTask].[EstMaterialCost]) -
		SUM([MaterialPurchased].[CostPerUOM] * [MaterialAssigned].[Quantity])
	AS MONEY)
		AS 'Difference Estimated/Actual'
FROM
	[MaterialAssigned]
		INNER JOIN [MaterialPurchased] ON
			[MaterialAssigned].[POID] = [MaterialPurchased].[POID]
		INNER JOIN [JobTask] ON
			[MaterialAssigned].[JobID] = [JobTask].[JobID] AND
			[MaterialAssigned].[TaskID] = [JobTask].[TaskID]
WHERE
	[MaterialAssigned].[JobID] = 91584
GROUP BY
	[MaterialAssigned].[JobID],
	[MaterialAssigned].[TaskID],
	[JobTask].[JobID],
	[JobTask].[TaskID],
	[JobTask].[EstMaterialCost]
ORDER BY
	[MaterialAssigned].[TaskID]
;
```

	JobID	TaskID	Date Completed	Estimated Material Cost	Actual Cost	Difference Estimated/Actual
1	91584	150	2014-04-02	300.00	313.87	-13.87
2	91584	160	2014-04-11	11590.00	11762.80	-172.80
3	91584	170	2014-04-14	150.00	154.85	-4.85
4	91584	180	2014-04-21	175.00	168.00	7.00
5	91584	230	2014-04-04	400.00	433.71	-33.71

```
/**
 * Exercise 12
 *
 * Modify query #11 to summarize the estimated material cost and actual
 * material cost for the entire job. This may prove to be difficult because of
 * the way that joins work. The MaterialAssigned table is the child of the
 * JobTask table in the relationship between the two. Thus, a join between
 * the two will always produce a result table with the number of rows in
 * the child table (MaterialAssigned). This may yield an incorrect result
 * if you are trying to sum a quantity that is in the parent table, while
 * also summing a quantity that is in the child table. Hint: Use a
 * sub-query in the SELECT list to get the correct sum of the
 * EstMaterialCost in the parent table (JobTask).
 */
 GO
CREATE VIEW v_EstimatedMaterialCostSums AS
SELECT
	[JobTask].[JobID] AS 'JobID',
	SUM([JobTask].[EstMaterialCost]) AS 'EstimatedMaterialCost',
	SUM([JobTask].[EstLaborCost]) AS 'EstimatedLaborCost'
FROM
	[JobTask]
GROUP BY
	[JobTask].[JobID]
;
GO
CREATE VIEW v_ActualMaterialCostSums AS
SELECT
	[MaterialAssigned].[JobID] AS 'JobID',
	SUM(
		[MaterialAssigned].[Quantity] *
		[MaterialPurchased].[CostPerUOM]
	) AS 'ActualMaterialCost'
FROM
	[MaterialAssigned]
		INNER JOIN [MaterialPurchased] ON
			[MaterialAssigned].[POID] = [MaterialPurchased].[POID]
GROUP BY
	[MaterialAssigned].[JobID]
;
GO
SELECT
	[v_EstimatedMaterialCostSums].[JobID] AS 'JobID',
	CAST(
		SUM([v_EstimatedMaterialCostSums].[EstimatedMaterialCost])
		AS MONEY
	) AS 'Estimated Material Cost',
	CAST(
		SUM([v_ActualMaterialCostSums].[ActualMaterialCost])
		AS MONEY
	) AS 'Actual Material Cost',
	CAST(
		SUM([v_EstimatedMaterialCostSums].[EstimatedMaterialCost]) -
		SUM([v_ActualMaterialCostSums].[ActualMaterialCost])
		AS MONEY
	) AS 'Difference Estimated/Actual'
FROM
	[v_EstimatedMaterialCostSums]
		INNER JOIN [v_ActualMaterialCostSums] ON
			[v_EstimatedMaterialCostSums].[JobID] = [v_ActualMaterialCostSums].[JobID]

WHERE
	[v_EstimatedMaterialCostSums].[JobID] = 91584
GROUP BY
```

```
	[v_EstimatedMaterialCostSums].[JobID]
ORDER BY
	[v_EstimatedMaterialCostSums].[JobID]
;

DROP VIEW v_EstimatedMaterialCostSums;
DROP VIEW v_ActualMaterialCostSums;
```

Results | Messages

	JobID	Estimated Material Cost	Actual Material Cost	Difference Estimated/Actual
1	91584	12615.00	12833.23	-218.23

```sql
/**
 * Exercise 13
 *
 * Time for a new query! Summarize the total amount of time worked for
 * each employee in the database. In addition to summing the HoursWorked,
 * also count the number of TimeSheets for each employee.
 */
SELECT
	Employee.EmpID,
	Employee.LastName + ',' + Employee.firstName 'Employee Name',
	ISNULL(SUM(Timesheet.HoursWorked),0) 'TotalHoursWorked',
	(SELECT  COUNT(EmpID) FROM TimeSheet WHERE Employee.EmpID = TimeSheet.EmpID)  'Number OF
Timesheets'
From
	Employee  LEFT OUTER JOIN TIMESHEET ON
	Employee.EmpID = TimeSheet.EmpID
GROUP BY
	Employee.EmpID,
	Employee.LastName,
	Employee.FirstName
```

Results | Messages

	EmpID	Employee Name	TotalHoursWorked	Number OF Timesheets
1	2300	Riggs,Evelyn	88.75	24
2	2480	Hazelton,Blake	89.80	29
3	3155	Allen,Noel	54.25	14
4	4702	Walker,Vance Martin	22.00	6
5	5291	Kinney,Deneece	124.30	33
6	5862	Bridges,Carol	139.75	35
7	6460	Kane,Sylvia	68.15	21
8	7651	Galloway,Odessa	0.00	0
9	7656	Wiggins,Cody	101.80	27
10	8110	Burgess,DaraLee	33.25	9
11	8750	Fleming,Cathleen	28.50	8

```sql
/**
 * Exercise 14
 *
 * Another new query! Summarize all the time worked in the TimeSheet table
 * by task.
 */
SELECT
	Task.TaskID,
	Task.TaskDescription,
	ISNULL(SUM(TimeSheet.HoursWorked),0) 'Total Hours'
FROM
	Task LEFT OUTER JOIN TimeSheet ON
		Task.taskID = TImeSheet.taskID
GROUP BY
	Task.TaskID,
	Task.TaskDescription
ORDER BY
	Task.TaskID
```

Results | Messages

	TaskID	TaskDescription	Total Hours
1	110	Remove Existing Floor	14.50
2	120	Clean mold	0.00
3	130	Install sub-floor	52.75
4	140	Design mosaic/glass	32.65
5	150	Prepare sub floor	95.50
6	160	Install tile floor	261.80
7	170	Grout	135.50
8	180	Seal and finish work	88.75
9	190	Install mosaic	0.00
10	200	Build structure	0.00
11	210	Remove existing tile	0.00
12	220	Clean site	0.00
13	230	Install tile counter	16.50
14	240	Install tile wall	0.00
15	260	Build mosaic/glass	35.50
16	270	General demolition	7.50

```
/**
 * Exercise 15
 *
 * Another new query (important for HW6). Eventually, we want to be able
 * to compare the actual number of hours worked and the actual cost of
 * those hours to the estimated hours and estimated labor cost. However,
 * this data is spread over a number of related tables in the database,
 * making it necessary to join data and use sub-queries. Right now, all
 * we want to do is figure out the actual hours worked and the actual cost
 * of those hours for every row in the TimeSheet table where the jobID is
 * NOT NULL. Sort the result table by EmpID.
 *
 * The hourly pay rate for an employee is stored in the EmployeePay table,
 * and that payrate can change over time, so we need to determine the
 * correct pay rate for HoursWorked based on the date that work was performed
 * in the TimeSheet table and the time in the EmployeePay table. I recommend
 * that you look back at HW#4, question #10 to remember how to locate the
 * correct HourlyPayRate for a given period of time for a given employee.
 * There should be 199 rows in the result table, so I have broken it up
 * into multiple snips but included all rows for your reference. You do
 * not have to include all rows on your output. As shown and sorted below,
 * I'd like to see rows 1-5, 125-135, 180-184 on your output so that I can
 * verify your results.
 */


SELECT
	TimeSheet.EmpID AS 'EmpID',
	Employee.LastName + ', ' + Employee.FirstName AS 'Employee Name',
	TimeSheet.TaskID AS 'TaskID',
	TimeSheet.JobID AS 'JobID',
	TimeSheet.StartWork AS 'StartWork',
	TimeSheet.HoursWorked AS 'HoursWorked',
	EmployeePay.HourlyPayRate AS 'HourlyPay',
	TimeSheet.HoursWorked * EmployeePay.HourlyPayRate AS 'LaborCost'
FROM
	TimeSheet
		INNER JOIN EmployeePay ON
			TimeSheet.EmpID = EmployeePay.EmpID
		INNER JOIN Employee ON
			TimeSheet.EmpID = Employee.EmpID
WHERE
	EmployeePay.DateStartPay <= TimeSheet.StartWork AND
	TimeSheet.StartWork < ISNULL(EmployeePay.DateEnd, GETDATE()) AND
	TimeSheet.JobID IS NOT NULL
;
```

Results | Messages

	EmpID	Employee Name	TaskID	JobID	StartWork	HoursWorked	HourlyPay	LaborCost
1	2300	Riggs, Evelyn	130	55841	2013-07-15 08:00:00.000	3.50	12.00	42.000000
2	2300	Riggs, Evelyn	130	55841	2013-07-15 13:30:00.000	4.00	12.00	48.000000
3	2300	Riggs, Evelyn	150	55841	2013-07-16 08:30:00.000	4.00	12.00	48.000000
4	2300	Riggs, Evelyn	170	55841	2013-07-22 08:00:00.000	4.00	12.00	48.000000
5	2300	Riggs, Evelyn	130	55878	2013-09-12 08:00:00.000	4.00	12.00	48.000000

125	5862	Bridges, Carol	160	62254	2014-10-20 13:00:00.000	4.00	16.00	64.000000
126	5862	Bridges, Carol	160	62254	2014-10-21 08:00:00.000	4.00	16.00	64.000000
127	5862	Bridges, Carol	160	62254	2014-10-21 14:30:00.000	4.00	16.00	64.000000
128	5862	Bridges, Carol	160	62257	2014-11-03 08:00:00.000	4.50	16.00	72.000000
129	5862	Bridges, Carol	160	62257	2014-11-03 13:00:00.000	4.00	16.00	64.000000
130	5862	Bridges, Carol	160	62257	2014-11-04 08:00:00.000	4.00	16.00	64.000000
131	5862	Bridges, Carol	160	62257	2014-11-04 13:00:00.000	4.00	16.00	64.000000
132	5862	Bridges, Carol	160	62257	2014-11-05 08:00:00.000	4.00	16.72	66.880000
133	5862	Bridges, Carol	160	62257	2014-11-05 13:00:00.000	4.00	16.72	66.880000
134	5862	Bridges, Carol	160	62257	2014-11-06 08:00:00.000	4.00	16.72	66.880000
135	5862	Bridges, Carol	160	62257	2014-11-06 13:00:00.000	4.00	16.72	66.880000

180	7656	Wiggins, Cody	160	16885	2014-12-18 08:00:00.000	4.00	11.50	46.000000
181	7656	Wiggins, Cody	160	16885	2014-12-18 13:00:00.000	4.00	11.50	46.000000
182	7656	Wiggins, Cody	160	16885	2014-12-19 08:00:00.000	3.25	12.15	39.487500
183	8110	Burgess, DaraL...	110	32687	2014-08-01 08:00:00.000	4.00	11.00	44.000000
184	8110	Burgess, DaraL...	110	32687	2014-08-01 13:00:00.000	3.50	11.00	38.500000