Database Design and Implementation

HW 06

Team 08

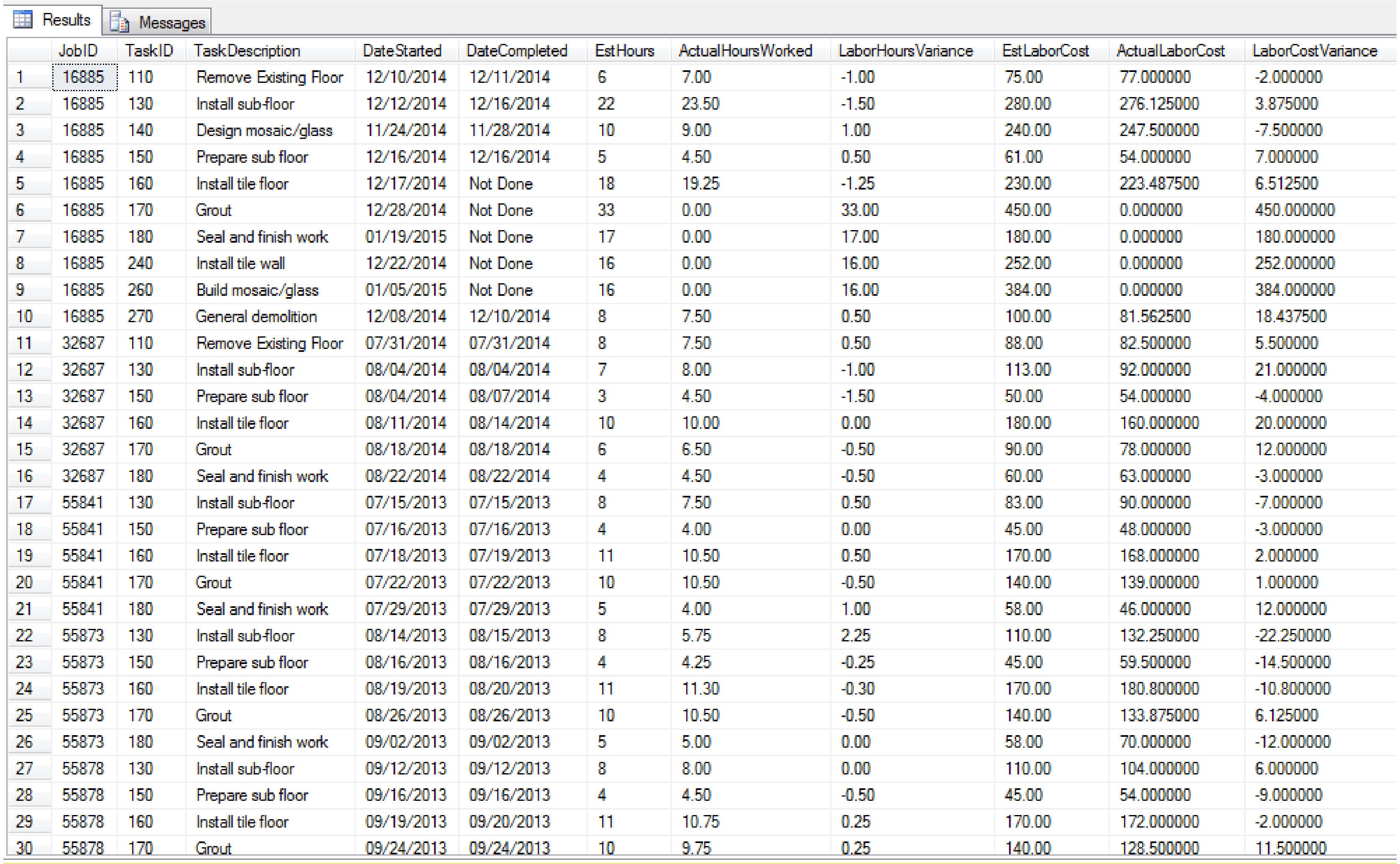
Henriod, Terence
Santoyo, Jorge
Singh, Raja

March 11, 2015

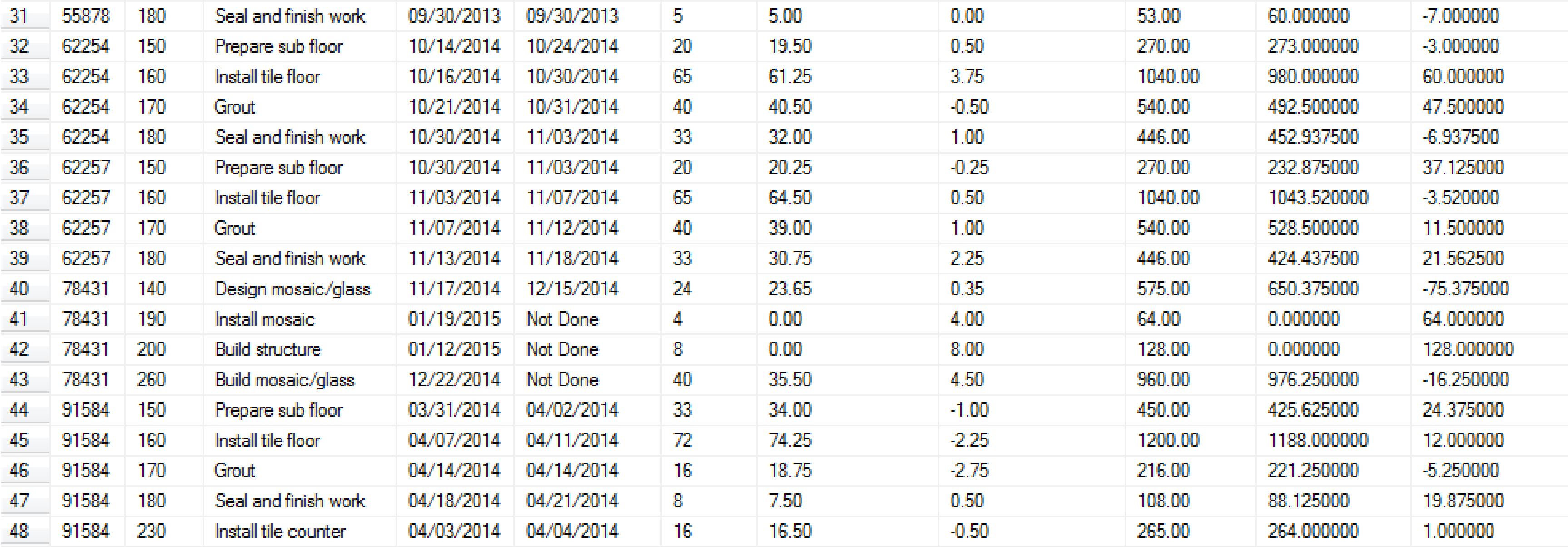
```sql
/**
 * IS675 - HW06 - Team08
 * Using SQL Operations with More Complex Queries
 *
 * Division of Labor
 * Well...
 */

/**
 * Exercise 01
 *
 * Summarize the actual hours worked and labor cost (hours worked * hourly
 * pay rate) by JobID and TaskID for all rows in the JobTask table. Remember
 * that an employee's pay rate changes by date, so it is necessary to locate
 * the correct pay rate by date as you did for question #15 for HW#5. If you
 * were not able to get question #15 on HW#5 to work, I recommend doing that
 * before starting on this question. Compare the actual labor hours to the
 * estimated labor hours, and the actual labor costs to the estimated
 * labor costs.
 */
DROP VIEW [v_TimeSheetsWithPay];
CREATE VIEW [v_TimeSheetsWithPay] AS
SELECT
       [TimeSheet].[EmpID] AS 'EmpID',
       [Employee].[LastName] + ', ' + [Employee].[FirstName] AS 'Employee Name',
    [TimeSheet].[JobID] AS 'JobID',
       [TimeSheet].[TaskID] AS 'TaskID',
    [Job].[JobCompleted] AS 'JobCompleted',
       [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]
        INNER JOIN [Job] ON
            [TimeSheet].[JobID] = [Job].[JobID]
WHERE
       [EmployeePay].[DateStartPay] <= [TimeSheet].[StartWork] AND
       [TimeSheet].[StartWork] < ISNULL([EmployeePay].[DateEnd], GETDATE()) AND
    [TimeSheet].[JobID] IS NOT NULL
;

DROP VIEW [v_ActualLaborByJobTask];
CREATE VIEW [v_ActualLaborByJobTask] AS
SELECT
    [v_TimeSheetsWithPay].[JobID] AS 'JobID',
    [v_TimeSheetsWithPay].[TaskID] AS 'TaskID',
    SUM([v_TimeSheetsWithPay].[HoursWorked]) AS 'HoursWorked',
    SUM(
        [v_TimeSheetsWithPay].[HoursWorked] * [v_TimeSheetsWithPay].[HourlyPay]
    ) AS 'LaborCost'
FROM
    [v_TimeSheetsWithPay]
GROUP BY
    [v_TimeSheetsWithPay].[JobID],
    [v_TimeSheetsWithPay].[TaskID]
;

DROP VIEW [v_LaborCostComparisons];
CREATE VIEW [v_LaborCostComparisons] AS
SELECT
    [JobTask].[JobID] AS 'JobID',
    [JobTask].[TaskID] AS 'TaskID',
    [Task].[TaskDescription] AS 'TaskDescription',
    CONVERT(varchar, [JobTask].[DateStarted], 101) AS 'DateStarted',
    ISNULL(
        CONVERT(varchar, [JobTask].[DateCompleted], 101),
        'Not Done'
    ) AS 'DateCompleted',
    [JobTask].[EstHours] AS 'EstHours',
    ISNULL([v_ActualLaborByJobTask].[HoursWorked], 0.00) AS 'ActualHoursWorked',
    [JobTask].[EstHours] - ISNULL([v_ActualLaborByJobTask].[HoursWorked], 0.00)
         AS 'LaborHoursVariance',
    [JobTask].[EstLaborCost] AS 'EstLaborCost',
    ISNULL([v_ActualLaborByJobTask].[LaborCost], 0.00) AS 'ActualLaborCost',
    [JobTask].[EstLaborCost] -
        ISNULL([v_ActualLaborByJobTask].[LaborCost], 0.00) AS 'LaborCostVariance'
FROM
    [JobTask]
        INNER JOIN [Task] ON
```

```
            [JobTask].[TaskID] = [Task].[TaskID]
        LEFT OUTER JOIN [v_ActualLaborByJobTask] ON
            [JobTask].[JobID] = [v_ActualLaborByJobTask].[JobID] AND
            [JobTask].[TaskID] = [v_ActualLaborByJobTask].[TaskID]
;

SELECT
    *
FROM
    [v_LaborCostComparisons]
ORDER BY
    [v_LaborCostComparisons].[JobID],
    [v_LaborCostComparisons].[TaskID]
;
```

5 Team 08



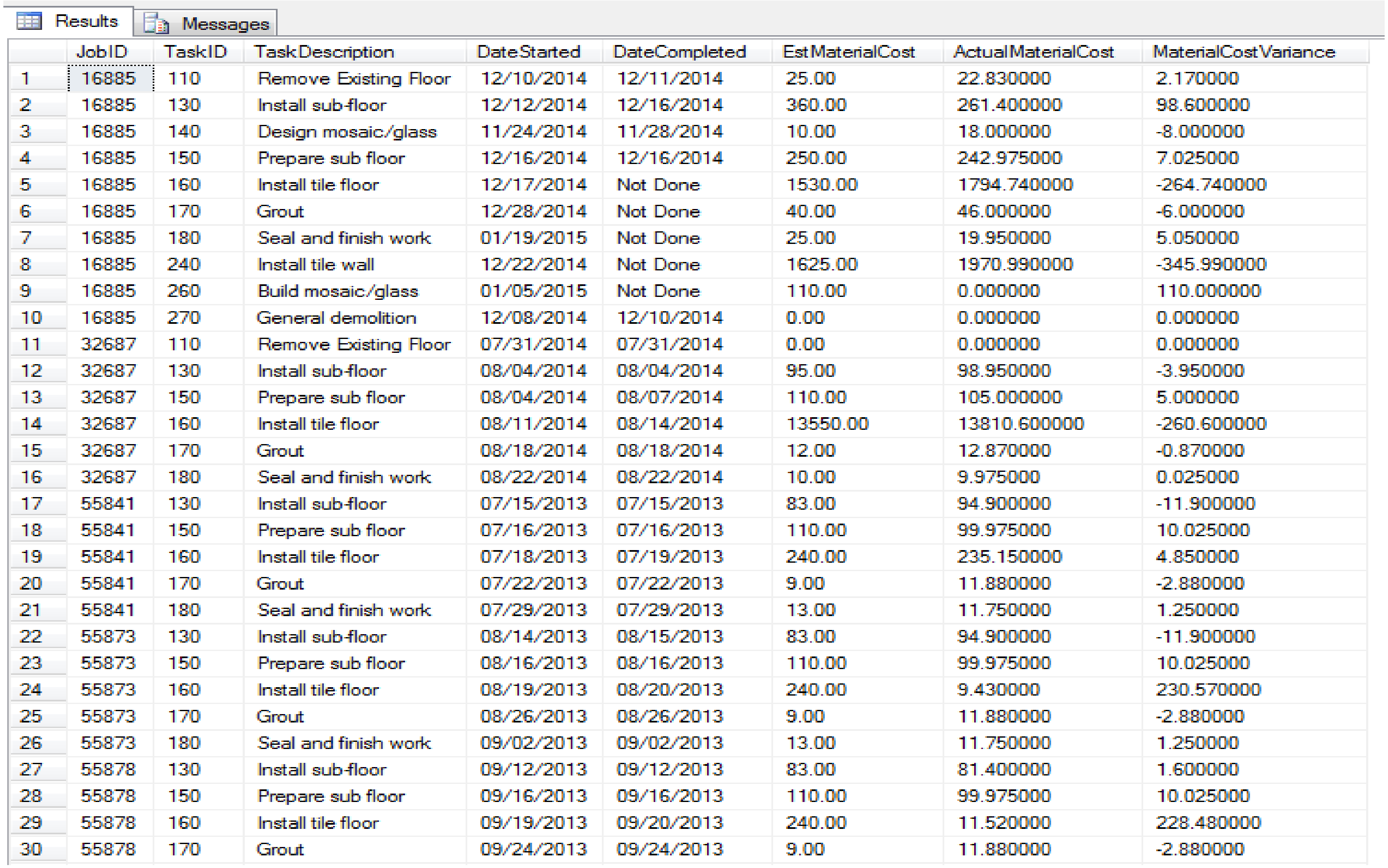
	JobID	TaskID	TaskDescription	DateStarted	DateCompleted	EstHours	ActualHoursWorked	LaborHoursVariance	EstLaborCost	ActualLaborCost	LaborCostVariance
1	16885	110	Remove Existing Floor	12/10/2014	12/11/2014	6	7.00	-1.00	75.00	77.000000	-2.000000
2	16885	130	Install sub-floor	12/12/2014	12/16/2014	22	23.50	-1.50	280.00	276.125000	3.875000
3	16885	140	Design mosaic/glass	11/24/2014	11/28/2014	10	9.00	1.00	240.00	247.500000	-7.500000
4	16885	150	Prepare sub floor	12/16/2014	12/16/2014	5	4.50	0.50	61.00	54.000000	7.000000
5	16885	160	Install tile floor	12/17/2014	Not Done	18	19.25	-1.25	230.00	223.487500	6.512500
6	16885	170	Grout	12/28/2014	Not Done	33	0.00	33.00	450.00	0.000000	450.000000
7	16885	180	Seal and finish work	01/19/2015	Not Done	17	0.00	17.00	180.00	0.000000	180.000000
8	16885	240	Install tile wall	12/22/2014	Not Done	16	0.00	16.00	252.00	0.000000	252.000000
9	16885	260	Build mosaic/glass	01/05/2015	Not Done	16	0.00	16.00	384.00	0.000000	384.000000
10	16885	270	General demolition	12/08/2014	12/10/2014	8	7.50	0.50	100.00	81.562500	18.437500
11	32687	110	Remove Existing Floor	07/31/2014	07/31/2014	8	7.50	0.50	88.00	82.500000	5.500000
12	32687	130	Install sub-floor	08/04/2014	08/04/2014	7	8.00	-1.00	113.00	92.000000	21.000000
13	32687	150	Prepare sub floor	08/04/2014	08/07/2014	3	4.50	-1.50	50.00	54.000000	-4.000000
14	32687	160	Install tile floor	08/11/2014	08/14/2014	10	10.00	0.00	180.00	160.000000	20.000000
15	32687	170	Grout	08/18/2014	08/18/2014	6	6.50	-0.50	90.00	78.000000	12.000000
16	32687	180	Seal and finish work	08/22/2014	08/22/2014	4	4.50	-0.50	60.00	63.000000	-3.000000
17	55841	130	Install sub-floor	07/15/2013	07/15/2013	8	7.50	0.50	83.00	90.000000	-7.000000
18	55841	150	Prepare sub floor	07/16/2013	07/16/2013	4	4.00	0.00	45.00	48.000000	-3.000000
19	55841	160	Install tile floor	07/18/2013	07/19/2013	11	10.50	0.50	170.00	168.000000	2.000000
20	55841	170	Grout	07/22/2013	07/22/2013	10	10.50	-0.50	140.00	139.000000	1.000000
21	55841	180	Seal and finish work	07/29/2013	07/29/2013	5	4.00	1.00	58.00	46.000000	12.000000
22	55873	130	Install sub-floor	08/14/2013	08/15/2013	8	5.75	2.25	110.00	132.250000	-22.250000
23	55873	150	Prepare sub floor	08/16/2013	08/16/2013	4	4.25	-0.25	45.00	59.500000	-14.500000
24	55873	160	Install tile floor	08/19/2013	08/20/2013	11	11.30	-0.30	170.00	180.800000	-10.800000
25	55873	170	Grout	08/26/2013	08/26/2013	10	10.50	-0.50	140.00	133.875000	6.125000
26	55873	180	Seal and finish work	09/02/2013	09/02/2013	5	5.00	0.00	58.00	70.000000	-12.000000
27	55878	130	Install sub-floor	09/12/2013	09/12/2013	8	8.00	0.00	110.00	104.000000	6.000000
28	55878	150	Prepare sub floor	09/16/2013	09/16/2013	4	4.50	-0.50	45.00	54.000000	-9.000000
29	55878	160	Install tile floor	09/19/2013	09/20/2013	11	10.75	0.25	170.00	172.000000	-2.000000
30	55878	170	Grout	09/24/2013	09/24/2013	10	9.75	0.25	140.00	128.500000	11.500000



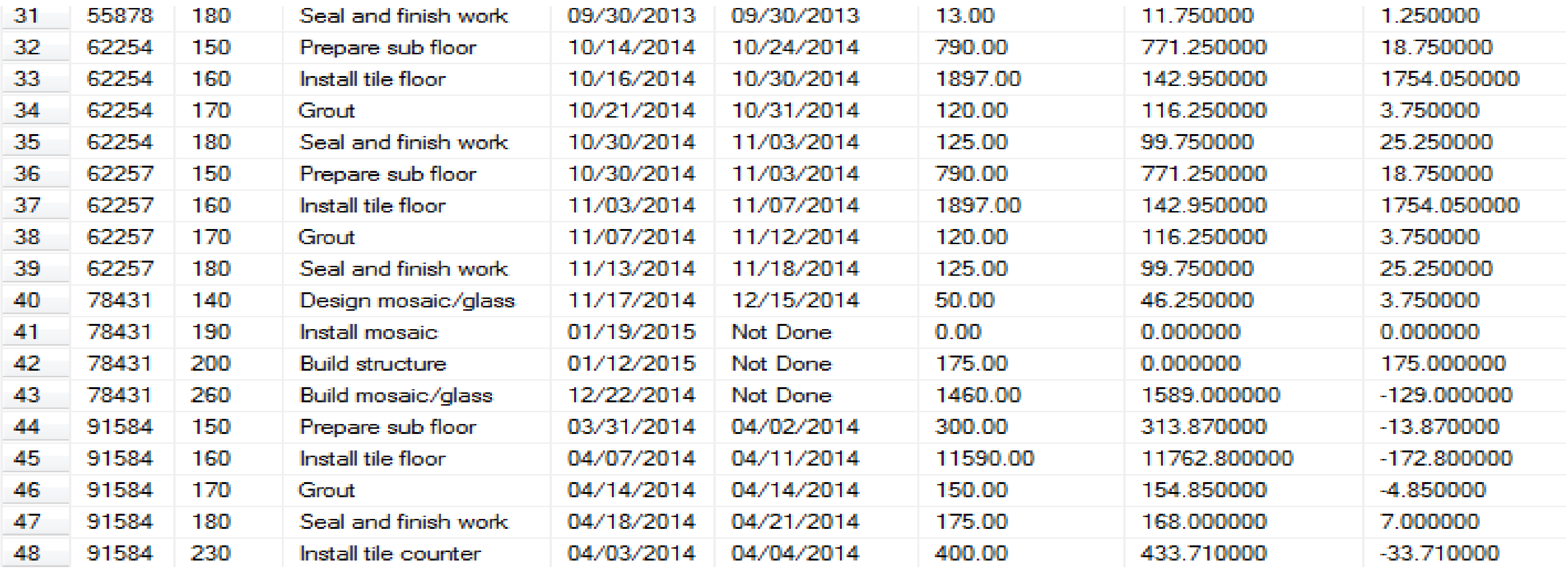
31	55878	180	Seal and finish work	09/30/2013	09/30/2013	5	5.00	0.00	53.00	60.000000	-7.000000
32	62254	150	Prepare sub floor	10/14/2014	10/24/2014	20	19.50	0.50	270.00	273.000000	-3.000000
33	62254	160	Install tile floor	10/16/2014	10/30/2014	65	61.25	3.75	1040.00	980.000000	60.000000
34	62254	170	Grout	10/21/2014	10/31/2014	40	40.50	-0.50	540.00	492.500000	47.500000
35	62254	180	Seal and finish work	10/30/2014	11/03/2014	33	32.00	1.00	446.00	452.937500	-6.937500
36	62257	150	Prepare sub floor	10/30/2014	11/03/2014	20	20.25	-0.25	270.00	232.875000	37.125000
37	62257	160	Install tile floor	11/03/2014	11/07/2014	65	64.50	0.50	1040.00	1043.520000	-3.520000
38	62257	170	Grout	11/07/2014	11/12/2014	40	39.00	1.00	540.00	528.500000	11.500000
39	62257	180	Seal and finish work	11/13/2014	11/18/2014	33	30.75	2.25	446.00	424.437500	21.562500
40	78431	140	Design mosaic/glass	11/17/2014	12/15/2014	24	23.65	0.35	575.00	650.375000	-75.375000
41	78431	190	Install mosaic	01/19/2015	Not Done	4	0.00	4.00	64.00	0.000000	64.000000
42	78431	200	Build structure	01/12/2015	Not Done	8	0.00	8.00	128.00	0.000000	128.000000
43	78431	260	Build mosaic/glass	12/22/2014	Not Done	40	35.50	4.50	960.00	976.250000	-16.250000
44	91584	150	Prepare sub floor	03/31/2014	04/02/2014	33	34.00	-1.00	450.00	425.625000	24.375000
45	91584	160	Install tile floor	04/07/2014	04/11/2014	72	74.25	-2.25	1200.00	1188.000000	12.000000
46	91584	170	Grout	04/14/2014	04/14/2014	16	18.75	-2.75	216.00	221.250000	-5.250000
47	91584	180	Seal and finish work	04/18/2014	04/21/2014	8	7.50	0.50	108.00	88.125000	19.875000
48	91584	230	Install tile counter	04/03/2014	04/04/2014	16	16.50	-0.50	265.00	264.000000	1.000000

```
/**
 * Exercise 02
 *
 * Summarize the actual material costs by jobID and task ID and compare
 * them to the estimated material cost for each row in the JobTask table.
 * This is very similar to what you did for question #11 in HW#5, so this
 * should be fairly easy if you got question #11 to work.
 */
DROP VIEW [v_ActualMaterialsByTask];
CREATE VIEW [v_ActualMaterialsByTask] AS
SELECT
    [MaterialAssigned].[JobID] AS 'JobID',
    [MaterialAssigned].[TaskID] AS 'TaskID',
    SUM([MaterialAssigned].[Quantity] * [MaterialPurchased].[CostPerUOM]) AS 'MaterialCost'
FROM
    [MaterialAssigned]
        INNER JOIN [MaterialPurchased] ON
            [MaterialAssigned].[POID] = [MaterialPurchased].[POID]
GROUP BY
    [MaterialAssigned].[JobID],
    [MaterialAssigned].[TaskID]
;

DROP VIEW [v_MaterialCostComparisons];
CREATE VIEW [v_MaterialCostComparisons] AS
SELECT
    [JobTask].[JobID] AS 'JobID',
    [JobTask].[TaskID] AS 'TaskID',
    [Task].[TaskDescription] AS 'TaskDescription',
    CONVERT(varchar, [JobTask].[DateStarted], 101) AS 'DateStarted',
    ISNULL(
        CONVERT(varchar, [JobTask].[DateCompleted], 101),
        'Not Done'
    ) AS 'DateCompleted',
    [JobTask].[EstMaterialCost] AS 'EstMaterialCost',
    ISNULL([v_ActualMaterialsByTask].[MaterialCost], 0.00) AS 'ActualMaterialCost',
    (
        [JobTask].[EstMaterialCost] -
        ISNULL([v_ActualMaterialsByTask].[MaterialCost], 0.00)
    ) AS 'MaterialCostVariance'
FROM
    [JobTask]
        INNER JOIN [Task] ON
            [JobTask].[TaskID] = [Task].[TaskID]
        LEFT OUTER JOIN [v_ActualMaterialsByTask] ON
            [JobTask].[JobID] = [v_ActualMaterialsByTask].[JobID] AND
            [JobTask].[TaskID] = [v_ActualMaterialsByTask].[TaskID]
;

SELECT
    *
FROM
    [v_MaterialCostComparisons]
ORDER BY
    [v_MaterialCostComparisons].[JobID],
    [v_MaterialCostComparisons].[TaskID]
;
```

8 Team 08



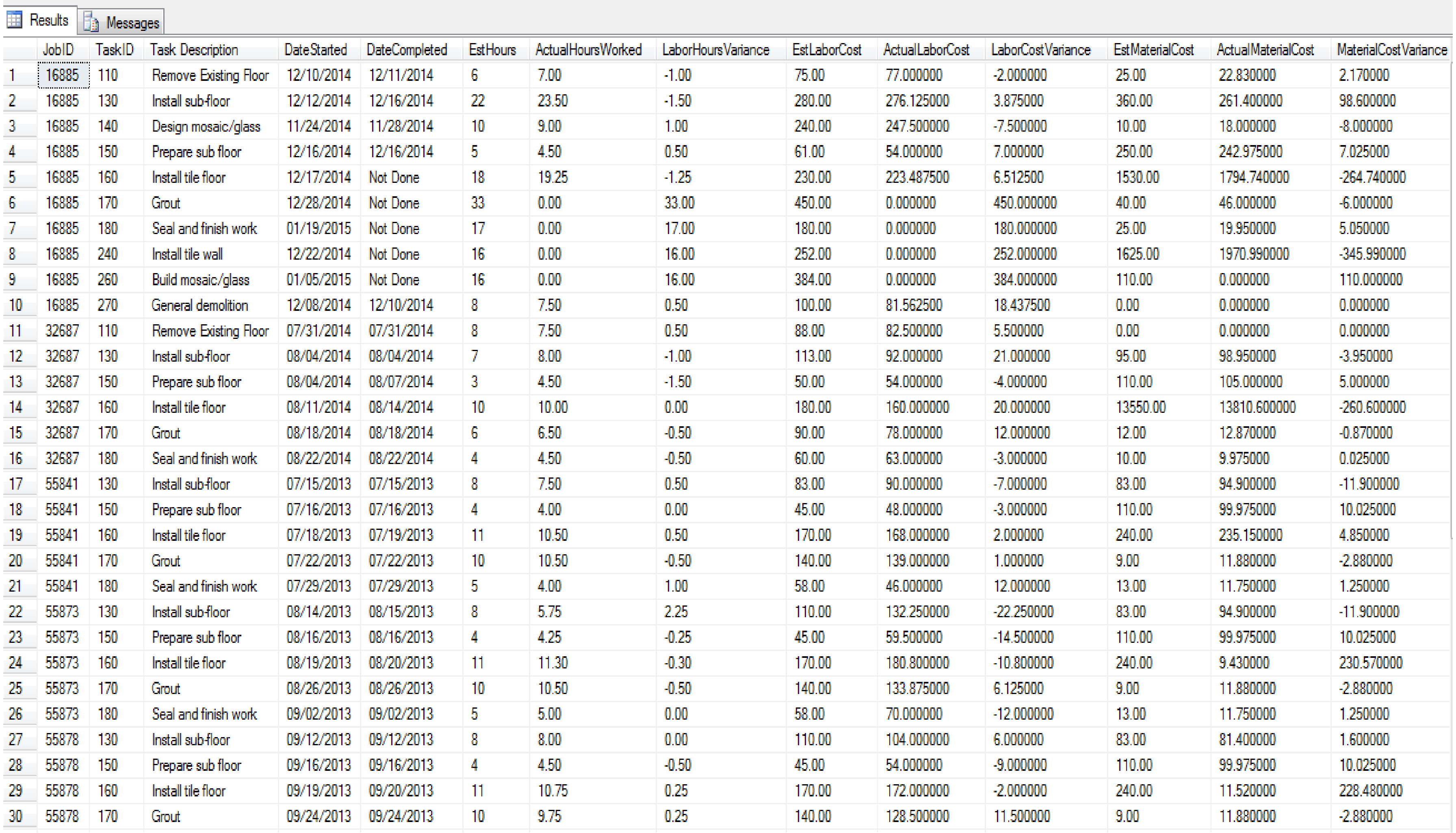
	JobID	TaskID	TaskDescription	DateStarted	DateCompleted	EstMaterialCost	ActualMaterialCost	MaterialCostVariance
1	16885	110	Remove Existing Floor	12/10/2014	12/11/2014	25.00	22.830000	2.170000
2	16885	130	Install sub-floor	12/12/2014	12/16/2014	360.00	261.400000	98.600000
3	16885	140	Design mosaic/glass	11/24/2014	11/28/2014	10.00	18.000000	-8.000000
4	16885	150	Prepare sub floor	12/16/2014	12/16/2014	250.00	242.975000	7.025000
5	16885	160	Install tile floor	12/17/2014	Not Done	1530.00	1794.740000	-264.740000
6	16885	170	Grout	12/28/2014	Not Done	40.00	46.000000	-6.000000
7	16885	180	Seal and finish work	01/19/2015	Not Done	25.00	19.950000	5.050000
8	16885	240	Install tile wall	12/22/2014	Not Done	1625.00	1970.990000	-345.990000
9	16885	260	Build mosaic/glass	01/05/2015	Not Done	110.00	0.000000	110.000000
10	16885	270	General demolition	12/08/2014	12/10/2014	0.00	0.000000	0.000000
11	32687	110	Remove Existing Floor	07/31/2014	07/31/2014	0.00	0.000000	0.000000
12	32687	130	Install sub-floor	08/04/2014	08/04/2014	95.00	98.950000	-3.950000
13	32687	150	Prepare sub floor	08/04/2014	08/07/2014	110.00	105.000000	5.000000
14	32687	160	Install tile floor	08/11/2014	08/14/2014	13550.00	13810.600000	-260.600000
15	32687	170	Grout	08/18/2014	08/18/2014	12.00	12.870000	-0.870000
16	32687	180	Seal and finish work	08/22/2014	08/22/2014	10.00	9.975000	0.025000
17	55841	130	Install sub-floor	07/15/2013	07/15/2013	83.00	94.900000	-11.900000
18	55841	150	Prepare sub floor	07/16/2013	07/16/2013	110.00	99.975000	10.025000
19	55841	160	Install tile floor	07/18/2013	07/19/2013	240.00	235.150000	4.850000
20	55841	170	Grout	07/22/2013	07/22/2013	9.00	11.880000	-2.880000
21	55841	180	Seal and finish work	07/29/2013	07/29/2013	13.00	11.750000	1.250000
22	55873	130	Install sub-floor	08/14/2013	08/15/2013	83.00	94.900000	-11.900000
23	55873	150	Prepare sub floor	08/16/2013	08/16/2013	110.00	99.975000	10.025000
24	55873	160	Install tile floor	08/19/2013	08/20/2013	240.00	9.430000	230.570000
25	55873	170	Grout	08/26/2013	08/26/2013	9.00	11.880000	-2.880000
26	55873	180	Seal and finish work	09/02/2013	09/02/2013	13.00	11.750000	1.250000
27	55878	130	Install sub-floor	09/12/2013	09/12/2013	83.00	81.400000	1.600000
28	55878	150	Prepare sub floor	09/16/2013	09/16/2013	110.00	99.975000	10.025000
29	55878	160	Install tile floor	09/19/2013	09/20/2013	240.00	11.520000	228.480000
30	55878	170	Grout	09/24/2013	09/24/2013	9.00	11.880000	-2.880000



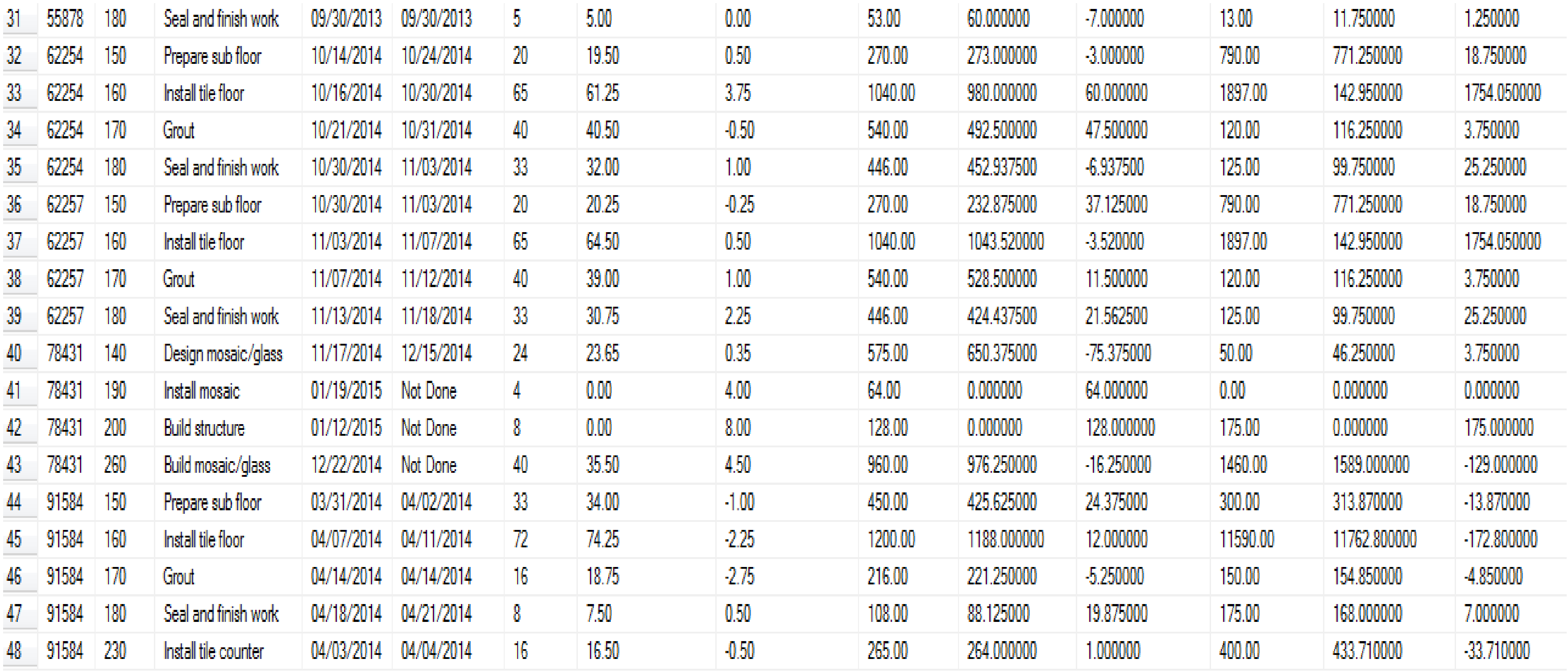
31	55878	180	Seal and finish work	09/30/2013	09/30/2013	13.00	11.750000	1.250000
32	62254	150	Prepare sub floor	10/14/2014	10/24/2014	790.00	771.250000	18.750000
33	62254	160	Install tile floor	10/16/2014	10/30/2014	1897.00	142.950000	1754.050000
34	62254	170	Grout	10/21/2014	10/31/2014	120.00	116.250000	3.750000
35	62254	180	Seal and finish work	10/30/2014	11/03/2014	125.00	99.750000	25.250000
36	62257	150	Prepare sub floor	10/30/2014	11/03/2014	790.00	771.250000	18.750000
37	62257	160	Install tile floor	11/03/2014	11/07/2014	1897.00	142.950000	1754.050000
38	62257	170	Grout	11/07/2014	11/12/2014	120.00	116.250000	3.750000
39	62257	180	Seal and finish work	11/13/2014	11/18/2014	125.00	99.750000	25.250000
40	78431	140	Design mosaic/glass	11/17/2014	12/15/2014	50.00	46.250000	3.750000
41	78431	190	Install mosaic	01/19/2015	Not Done	0.00	0.000000	0.000000
42	78431	200	Build structure	01/12/2015	Not Done	175.00	0.000000	175.000000
43	78431	260	Build mosaic/glass	12/22/2014	Not Done	1460.00	1589.000000	-129.000000
44	91584	150	Prepare sub floor	03/31/2014	04/02/2014	300.00	313.870000	-13.870000
45	91584	160	Install tile floor	04/07/2014	04/11/2014	11590.00	11762.800000	-172.800000
46	91584	170	Grout	04/14/2014	04/14/2014	150.00	154.850000	-4.850000
47	91584	180	Seal and finish work	04/18/2014	04/21/2014	175.00	168.000000	7.000000
48	91584	230	Install tile counter	04/03/2014	04/04/2014	400.00	433.710000	-33.710000

```
/**
 * Exercise 03
 *
 * Now it’s time to put them together. Compare actual to estimated costs
 * for each row in the JobTask table. The PercentVariance is the percentage
 * variance between the TotalEstCost and the TotalActualCost. The
 * general calculation is:
 * ((TotalEstCost – TotalActualCost)/TotalEstCost) * 100.
 */

CREATE VIEW [v_TotalCostsByTask] AS
SELECT
    [v_LaborCostComparisons].[JobID] AS 'JobID',
    [v_LaborCostComparisons].[TaskID] AS 'TaskID',
    [v_LaborCostComparisons].[TaskDescription] AS 'TaskDescription',
    [v_LaborCostComparisons].[DateStarted] AS 'DateStarted',
    [v_LaborCostComparisons].[DateCompleted] AS 'DateCompleted',
    (
        [v_LaborCostComparisons].[EstLaborCost] +
        [v_MaterialCostComparisons].[EstMaterialCost]
    ) AS 'TotalEstCost',
    (
        [v_LaborCostComparisons].[ActualLaborCost] +
        [v_MaterialCostComparisons].[ActualMaterialCost]
    ) AS 'TotalActualCost'
FROM
    [v_LaborCostComparisons]
        INNER JOIN [v_MaterialCostComparisons] ON
            [v_LaborCostComparisons].[JobID] = [v_MaterialCostComparisons].[JobID] AND
            [v_LaborCostComparisons].[TaskID] = [v_MaterialCostComparisons].[TaskID]
;

DROP VIEW [v_AllCostComparisonsByTask];
CREATE VIEW [v_AllCostComparisonsByTask] AS
SELECT
    [v_TotalCostsByTask].[JobID] AS 'JobID',
    [v_TotalCostsByTask].[TaskID] AS 'TaskID',
    [v_TotalCostsByTask].[TaskDescription] AS 'Task Description',
    [v_TotalCostsByTask].[DateStarted] AS 'DateStarted',
    [v_TotalCostsByTask].[DateCompleted] AS 'DateCompleted',
    [v_LaborCostComparisons].[EstHours] AS 'EstHours',
    [v_LaborCostComparisons].[ActualHoursWorked] AS 'ActualHoursWorked',
    [v_LaborCostComparisons].[LaborHoursVariance] AS 'LaborHoursVariance',
    [v_LaborCostComparisons].[EstLaborCost] AS 'EstLaborCost',
    [v_LaborCostComparisons].[ActualLaborCost] AS 'ActualLaborCost',
    [v_LaborCostComparisons].[LaborCostVariance] AS 'LaborCostVariance',
    [v_MaterialCostComparisons].[EstMaterialCost] AS 'EstMaterialCost',
    [v_MaterialCostComparisons].[ActualMaterialCost] AS 'ActualMaterialCost',
    [v_MaterialCostComparisons].[MaterialCostVariance] AS 'MaterialCostVariance',
    [v_TotalCostsByTask].[TotalEstCost] AS 'TotalEstCost',
    [v_TotalCostsByTask].[TotalActualCost] AS 'TotalActualCost',
    (
        [v_TotalCostsByTask].[TotalEstCost] -
        [v_TotalCostsByTask].[TotalActualCost]
    ) AS 'TotalCostVariance',
    (
        (
            [v_TotalCostsByTask].[TotalEstCost] -
            [v_TotalCostsByTask].[TotalActualCost]
        ) *
        (
            100.00 /
            [v_TotalCostsByTask].[TotalEstCost]
        )
    ) AS 'PercentVariance'
FROM
    [v_TotalCostsByTask]
        INNER JOIN [v_LaborCostComparisons] ON
            [v_TotalCostsByTask].[JobID] = [v_LaborCostComparisons].[JobID] AND
            [v_TotalCostsByTask].[TaskID] = [v_LaborCostComparisons].[TaskID]
        INNER JOIN [v_MaterialCostComparisons] ON
            [v_TotalCostsByTask].[JobID] = [v_MaterialCostComparisons].[JobID] AND
            [v_TotalCostsByTask].[TaskID] = [v_MaterialCostComparisons].[TaskID]
;

SELECT
    *
FROM
    [v_AllCostComparisonsByTask]
ORDER BY
    [v_AllCostComparisonsByTask].[JobID],
    [v_AllCostComparisonsByTask].[TaskID]
;
```

12 Team 08



	JobID	TaskID	Task Description	DateStarted	DateCompleted	EstHours	ActualHoursWorked	LaborHoursVariance	EstLaborCost	ActualLaborCost	LaborCostVariance	EstMaterialCost	ActualMaterialCost	MaterialCostVariance
1	16885	110	Remove Existing Floor	12/10/2014	12/11/2014	6	7.00	-1.00	75.00	77.000000	-2.000000	25.00	22.830000	2.170000
2	16885	130	Install sub-floor	12/12/2014	12/16/2014	22	23.50	-1.50	280.00	276.125000	3.875000	360.00	261.400000	98.600000
3	16885	140	Design mosaic/glass	11/24/2014	11/28/2014	10	9.00	1.00	240.00	247.500000	-7.500000	10.00	18.000000	-8.000000
4	16885	150	Prepare sub floor	12/16/2014	12/16/2014	5	4.50	0.50	61.00	54.000000	7.000000	250.00	242.975000	7.025000
5	16885	160	Install tile floor	12/17/2014	Not Done	18	19.25	-1.25	230.00	223.487500	6.512500	1530.00	1794.740000	-264.740000
6	16885	170	Grout	12/28/2014	Not Done	33	0.00	33.00	450.00	0.000000	450.000000	40.00	46.000000	-6.000000
7	16885	180	Seal and finish work	01/19/2015	Not Done	17	0.00	17.00	180.00	0.000000	180.000000	25.00	19.950000	5.050000
8	16885	240	Install tile wall	12/22/2014	Not Done	16	0.00	16.00	252.00	0.000000	252.000000	1625.00	1970.990000	-345.990000
9	16885	260	Build mosaic/glass	01/05/2015	Not Done	16	0.00	16.00	384.00	0.000000	384.000000	110.00	0.000000	110.000000
10	16885	270	General demolition	12/08/2014	12/10/2014	8	7.50	0.50	100.00	81.562500	18.437500	0.00	0.000000	0.000000
11	32687	110	Remove Existing Floor	07/31/2014	07/31/2014	8	7.50	0.50	88.00	82.500000	5.500000	0.00	0.000000	0.000000
12	32687	130	Install sub-floor	08/04/2014	08/04/2014	7	8.00	-1.00	113.00	92.000000	21.000000	95.00	98.950000	-3.950000
13	32687	150	Prepare sub floor	08/04/2014	08/07/2014	3	4.50	-1.50	50.00	54.000000	-4.000000	110.00	105.000000	5.000000
14	32687	160	Install tile floor	08/11/2014	08/14/2014	10	10.00	0.00	180.00	160.000000	20.000000	13550.00	13810.600000	-260.600000
15	32687	170	Grout	08/18/2014	08/18/2014	6	6.50	-0.50	90.00	78.000000	12.000000	12.00	12.870000	-0.870000
16	32687	180	Seal and finish work	08/22/2014	08/22/2014	4	4.50	-0.50	60.00	63.000000	-3.000000	10.00	9.975000	0.025000
17	55841	130	Install sub-floor	07/15/2013	07/15/2013	8	7.50	0.50	83.00	90.000000	-7.000000	83.00	94.900000	-11.900000
18	55841	150	Prepare sub floor	07/16/2013	07/16/2013	4	4.00	0.00	45.00	48.000000	-3.000000	110.00	99.975000	10.025000
19	55841	160	Install tile floor	07/18/2013	07/19/2013	11	10.50	0.50	170.00	168.000000	2.000000	240.00	235.150000	4.850000
20	55841	170	Grout	07/22/2013	07/22/2013	10	10.50	-0.50	140.00	139.000000	1.000000	9.00	11.880000	-2.880000
21	55841	180	Seal and finish work	07/29/2013	07/29/2013	5	4.00	1.00	58.00	46.000000	12.000000	13.00	11.750000	1.250000
22	55873	130	Install sub-floor	08/14/2013	08/15/2013	8	5.75	2.25	110.00	132.250000	-22.250000	83.00	94.900000	-11.900000
23	55873	150	Prepare sub floor	08/16/2013	08/16/2013	4	4.25	-0.25	45.00	59.500000	-14.500000	110.00	99.975000	10.025000
24	55873	160	Install tile floor	08/19/2013	08/20/2013	11	11.30	-0.30	170.00	180.800000	-10.800000	240.00	9.430000	230.570000
25	55873	170	Grout	08/26/2013	08/26/2013	10	10.50	-0.50	140.00	133.875000	6.125000	9.00	11.880000	-2.880000
26	55873	180	Seal and finish work	09/02/2013	09/02/2013	5	5.00	0.00	58.00	70.000000	-12.000000	13.00	11.750000	1.250000
27	55878	130	Install sub-floor	09/12/2013	09/12/2013	8	8.00	0.00	110.00	104.000000	6.000000	83.00	81.400000	1.600000
28	55878	150	Prepare sub floor	09/16/2013	09/16/2013	4	4.50	-0.50	45.00	54.000000	-9.000000	110.00	99.975000	10.025000
29	55878	160	Install tile floor	09/19/2013	09/20/2013	11	10.75	0.25	170.00	172.000000	-2.000000	240.00	11.520000	228.480000
30	55878	170	Grout	09/24/2013	09/24/2013	10	9.75	0.25	140.00	128.500000	11.500000	9.00	11.880000	-2.880000

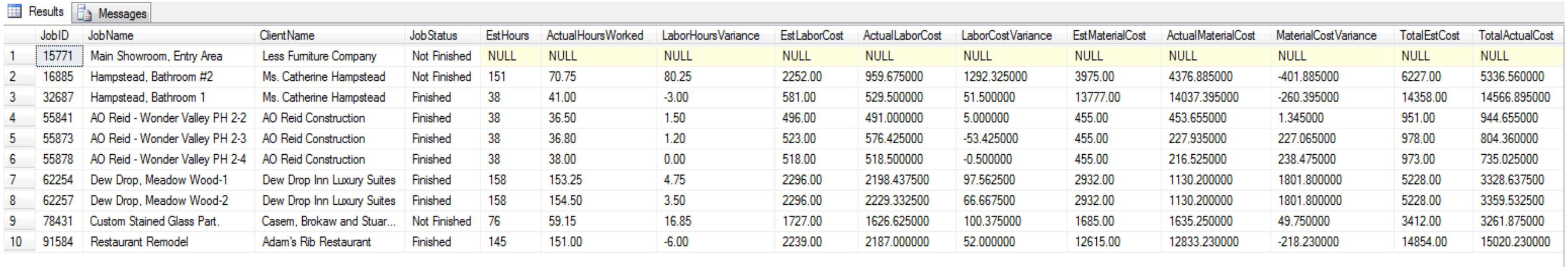


31	55878	180	Seal and finish work	09/30/2013	09/30/2013	5	5.00	0.00	53.00	60.000000	-7.000000	13.00	11.750000	1.250000
32	62254	150	Prepare sub floor	10/14/2014	10/24/2014	20	19.50	0.50	270.00	273.000000	-3.000000	790.00	771.250000	18.750000
33	62254	160	Install tile floor	10/16/2014	10/30/2014	65	61.25	3.75	1040.00	980.000000	60.000000	1897.00	142.950000	1754.050000
34	62254	170	Grout	10/21/2014	10/31/2014	40	40.50	-0.50	540.00	492.500000	47.500000	120.00	116.250000	3.750000
35	62254	180	Seal and finish work	10/30/2014	11/03/2014	33	32.00	1.00	446.00	452.937500	-6.937500	125.00	99.750000	25.250000
36	62257	150	Prepare sub floor	10/30/2014	11/03/2014	20	20.25	-0.25	270.00	232.875000	37.125000	790.00	771.250000	18.750000
37	62257	160	Install tile floor	11/03/2014	11/07/2014	65	64.50	0.50	1040.00	1043.520000	-3.520000	1897.00	142.950000	1754.050000
38	62257	170	Grout	11/07/2014	11/12/2014	40	39.00	1.00	540.00	528.500000	11.500000	120.00	116.250000	3.750000
39	62257	180	Seal and finish work	11/13/2014	11/18/2014	33	30.75	2.25	446.00	424.437500	21.562500	125.00	99.750000	25.250000
40	78431	140	Design mosaic/glass	11/17/2014	12/15/2014	24	23.65	0.35	575.00	650.375000	-75.375000	50.00	46.250000	3.750000
41	78431	190	Install mosaic	01/19/2015	Not Done	4	0.00	4.00	64.00	0.000000	64.000000	0.00	0.000000	0.000000
42	78431	200	Build structure	01/12/2015	Not Done	8	0.00	8.00	128.00	0.000000	128.000000	175.00	0.000000	175.000000
43	78431	260	Build mosaic/glass	12/22/2014	Not Done	40	35.50	4.50	960.00	976.250000	-16.250000	1460.00	1589.000000	-129.000000
44	91584	150	Prepare sub floor	03/31/2014	04/02/2014	33	34.00	-1.00	450.00	425.625000	24.375000	300.00	313.870000	-13.870000
45	91584	160	Install tile floor	04/07/2014	04/11/2014	72	74.25	-2.25	1200.00	1188.000000	12.000000	11590.00	11762.800000	-172.800000
46	91584	170	Grout	04/14/2014	04/14/2014	16	18.75	-2.75	216.00	221.250000	-5.250000	150.00	154.850000	-4.850000
47	91584	180	Seal and finish work	04/18/2014	04/21/2014	8	7.50	0.50	108.00	88.125000	19.875000	175.00	168.000000	7.000000
48	91584	230	Install tile counter	04/03/2014	04/04/2014	16	16.50	-0.50	265.00	264.000000	1.000000	400.00	433.710000	-33.710000

```
/**
 * Exercise 04
 *
 * Summarize the information created in question #3 by job. The result
 * table should have one row per job in the Job table. Add additional data
 * from the Job and Client tables to provide more information about each
 * job in the result table. Hint: \textbf{\underline{Calculate}} the
 * PercentVariance – you cannot sum that field.
 */
DROP VIEW [v_TotalCostsByJob];
CREATE VIEW [v_TotalCostsByJob] AS
SELECT
    [v_AllCostComparisonsByTask].[JobID] AS 'JobID',
```

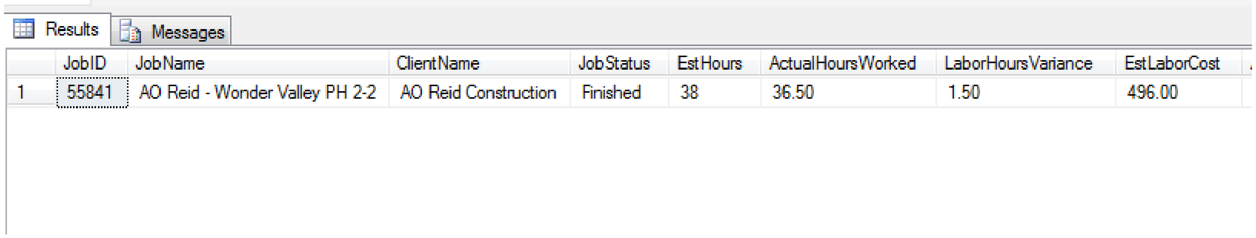
```sql
    SUM([v_AllCostComparisonsByTask].[EstHours]) AS 'EstHours',
    SUM([v_AllCostComparisonsByTask].[ActualHoursWorked]) AS 'ActualHoursWorked',
    SUM([v_AllCostComparisonsByTask].[LaborHoursVariance]) AS 'LaborHoursVariance',
    SUM([v_AllCostComparisonsByTask].[EstLaborCost]) AS 'EstLaborCost',
    SUM([v_AllCostComparisonsByTask].[ActualLaborCost]) AS 'ActualLaborCost',
    SUM([v_AllCostComparisonsByTask].[LaborCostVariance]) AS 'LaborCostVariance',
    SUM([v_AllCostComparisonsByTask].[EstMaterialCost]) AS 'EstMaterialCost',
    SUM([v_AllCostComparisonsByTask].[ActualMaterialCost]) AS 'ActualMaterialCost',
    SUM([v_AllCostComparisonsByTask].[MaterialCostVariance]) AS 'MaterialCostVariance',
    SUM([v_AllCostComparisonsByTask].[TotalEstCost]) AS 'TotalEstCost',
    SUM([v_AllCostComparisonsByTask].[TotalActualCost]) AS 'TotalActualCost',
    SUM([v_AllCostComparisonsByTask].[TotalCostVariance]) AS 'TotalCostVariance',
    (
        SUM([v_AllCostComparisonsByTask].[TotalCostVariance]) * 100 /
        SUM([v_AllCostComparisonsByTask].[TotalEstCost])
    ) AS 'PercentVariance'
FROM
    [v_AllCostComparisonsByTask]
GROUP BY
    [v_AllCostComparisonsByTask].[JobID]
;

DROP VIEW [v_JobInfo];
CREATE VIEW [v_JobInfo] AS
SELECT
    [Job].[JobID] AS 'JobID',
    [Job].[JobName] AS 'JobName',
    [Client].[ClientName] AS 'ClientName',
    CASE
        WHEN [Job].[JobCompleted] = 1
            THEN 'Finished'
        ELSE
            'Not Finished'
    END /**/ AS 'JobStatus'
FROM
    [Job]
        INNER JOIN [Client] ON
            [Job].[ClientID] = [Client].[ClientID]
;

DROP VIEW [v_JobSummary];
CREATE VIEW [v_JobSummary] AS
```

```sql
SELECT
    [v_JobInfo].[JobID] AS 'JobID',
    [v_JobInfo].[JobName] AS 'JobName',
    [v_JobInfo].[ClientName] AS 'ClientName',
    [v_JobInfo].[JobStatus] AS 'JobStatus',
    [v_TotalCostsByJob].[EstHours] AS 'EstHours',
    [v_TotalCostsByJob].[ActualHoursWorked] AS 'ActualHoursWorked',
    [v_TotalCostsByJob].[LaborHoursVariance] AS 'LaborHoursVariance',
    [v_TotalCostsByJob].[EstLaborCost] AS 'EstLaborCost',
    [v_TotalCostsByJob].[ActualLaborCost] AS 'ActualLaborCost',
    [v_TotalCostsByJob].[LaborCostVariance] AS 'LaborCostVariance',
    [v_TotalCostsByJob].[EstMaterialCost] AS 'EstMaterialCost',
    [v_TotalCostsByJob].[ActualMaterialCost] AS 'ActualMaterialCost',
    [v_TotalCostsByJob].[MaterialCostVariance] AS 'MaterialCostVariance',
    [v_TotalCostsByJob].[TotalEstCost] AS 'TotalEstCost',
    [v_TotalCostsByJob].[TotalActualCost] AS 'TotalActualCost',
    [v_TotalCostsByJob].[TotalCostVariance] AS 'TotalCostVariance',
    [v_TotalCostsByJob].[PercentVariance] AS 'PercentVariance'
FROM
    [v_JobInfo]
        LEFT OUTER JOIN [v_TotalCostsByJob] ON
            [v_JobInfo].[JobID] = [v_TotalCostsByJob].[JobID]
;

SELECT
    *
FROM
    [v_JobSummary]
ORDER BY
    [v_JobSummary].[JobID]
;
```

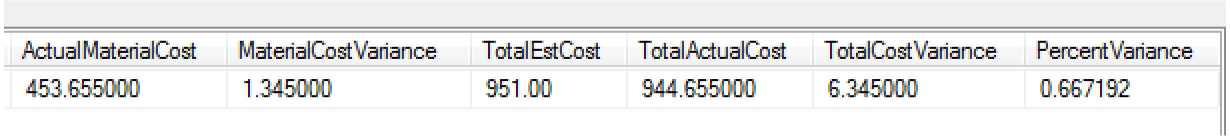


	JobID	JobName	ClientName	JobStatus	EstHours	ActualHoursWorked	LaborHoursVariance	EstLaborCost	ActualLaborCost	LaborCostVariance	EstMaterialCost	ActualMaterialCost	MaterialCostVariance	TotalEstCost	TotalActualCost
1	15771	Main Showroom, Entry Area	Less Furniture Company	Not Finished	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2	16885	Hampstead, Bathroom #2	Ms. Catherine Hampstead	Not Finished	151	70.75	80.25	2252.00	959.675000	1292.325000	3975.00	4376.885000	-401.885000	6227.00	5336.560000
3	32687	Hampstead, Bathroom 1	Ms. Catherine Hampstead	Finished	38	41.00	-3.00	581.00	529.500000	51.500000	13777.00	14037.395000	-260.395000	14358.00	14566.895000
4	55841	AO Reid - Wonder Valley PH 2-2	AO Reid Construction	Finished	38	36.50	1.50	496.00	491.000000	5.000000	455.00	453.655000	1.345000	951.00	944.655000
5	55873	AO Reid - Wonder Valley PH 2-3	AO Reid Construction	Finished	38	36.80	1.20	523.00	576.425000	-53.425000	455.00	227.935000	227.065000	978.00	804.360000
6	55878	AO Reid - Wonder Valley PH 2-4	AO Reid Construction	Finished	38	38.00	0.00	518.00	518.500000	-0.500000	455.00	216.525000	238.475000	973.00	735.025000
7	62254	Dew Drop, Meadow Wood-1	Dew Drop Inn Luxury Suites	Finished	158	153.25	4.75	2296.00	2198.437500	97.562500	2932.00	1130.200000	1801.800000	5228.00	3328.637500
8	62257	Dew Drop, Meadow Wood-2	Dew Drop Inn Luxury Suites	Finished	158	154.50	3.50	2296.00	2229.332500	66.667500	2932.00	1130.200000	1801.800000	5228.00	3359.532500
9	78431	Custom Stained Glass Part.	Casem, Brokaw and Stuar...	Not Finished	76	59.15	16.85	1727.00	1626.625000	100.375000	1685.00	1635.250000	49.750000	3412.00	3261.875000
10	91584	Restaurant Remodel	Adam's Rib Restaurant	Finished	145	151.00	-6.00	2239.00	2187.000000	52.000000	12615.00	12833.230000	-218.230000	14854.00	15020.230000

```
/**
 * Exercise 05
 *
 * Which job that is \underline{finished} had actual total costs that were
 * closest to the estimated total costs? (PercentVariance closest to zero)
 * Make sure that the query could select the correct job from any data
 * set - the query should not just work with our test data set.
 */
SELECT
    *
FROM
    [v_JobSummary]
WHERE
    [v_JobSummary].[PercentVariance] = (
        SELECT
            MIN(ABS([v_JobSummary].[PercentVariance]))
        FROM
            [v_JobSummary]
    )
;
```

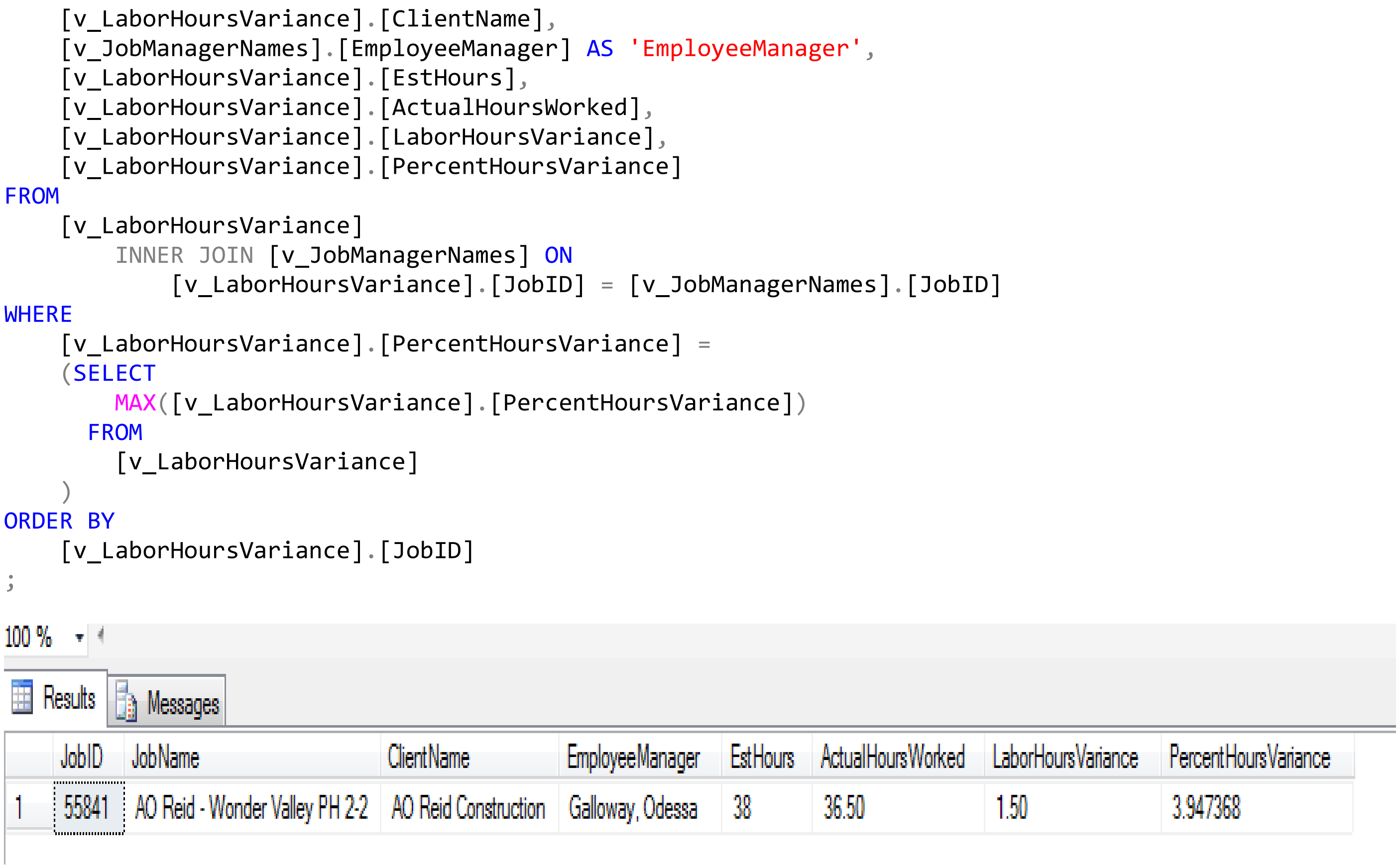


	JobID	JobName	ClientName	JobStatus	EstHours	ActualHoursWorked	LaborHoursVariance	EstLaborCost
1	55841	AO Reid - Wonder Valley PH 2-2	AO Reid Construction	Finished	38	36.50	1.50	496.00



ActualMaterialCost	MaterialCostVariance	TotalEstCost	TotalActualCost	TotalCostVariance	PercentVariance
453.655000	1.345000	951.00	944.655000	6.345000	0.667192

```
/**
 * Exercise 06
 *
 * Which job that is \underline{finished} had the largest percentage
 * positive labor hours variance? In other words, which finished job was
 * able to be completed with the least number of labor hours, when compared
 * to the estimated labor hours? The percentage labor hours variance is
 * calculated as the LaborHoursVariance/EstHours * 100. Add in the name of
 * the employee who served as the manager for the job.
 */
DROP VIEW [v_LaborHoursVariance];
CREATE VIEW [v_LaborHoursVariance] AS
SELECT
    [v_JobSummary].[JobID],
    [v_JobSummary].[JobName],
    [v_JobSummary].[ClientName],
    [v_JobSummary].[EstHours],
    [v_JobSummary].[ActualHoursWorked],
    [v_JobSummary].[EstHours] - [v_JobSummary].[ActualHoursWorked] AS 'LaborHoursVariance',
    ([v_JobSummary].[EstHours] - [v_JobSummary].[ActualHoursWorked]) * 100 / [v_JobSummary].[EstHours] AS
'PercentHoursVariance'
FROM
    [v_JobSummary]
WHERE
    [v_JobSummary].[JobStatus] = 'Finished'
;

DROP VIEW [v_JobManagerNames];
CREATE VIEW [v_JobManagerNames] AS
SELECT
    [Job].[JobID] AS 'JobID',
    [Employee].[LastName] + ', ' + [Employee].[FirstName] AS 'EmployeeManager'
FROM
    [Job]
        INNER JOIN [Employee] ON
            [Job].[EmpManagerID] = [Employee].[EmpID]
;


SELECT
    [v_LaborHoursVariance].[JobID],
    [v_LaborHoursVariance].[JobName],
```



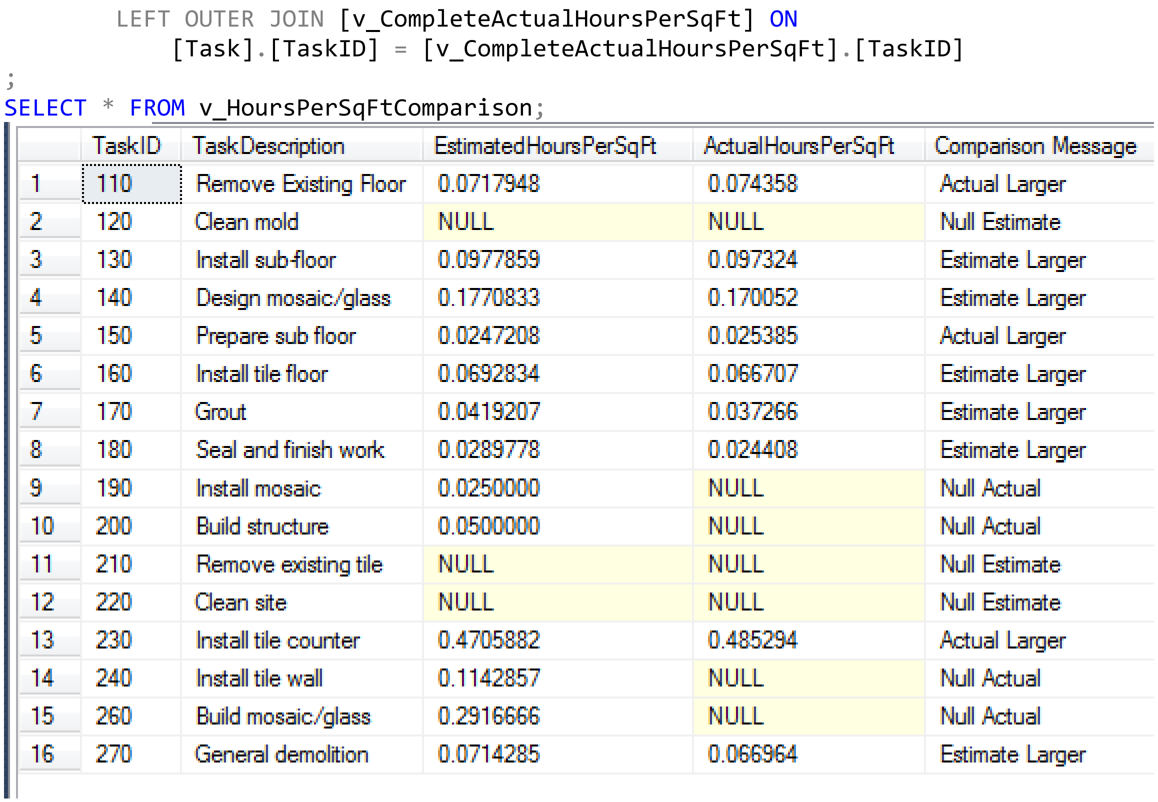
```
    [v_LaborHoursVariance].[ClientName],
    [v_JobManagerNames].[EmployeeManager] AS 'EmployeeManager',
    [v_LaborHoursVariance].[EstHours],
    [v_LaborHoursVariance].[ActualHoursWorked],
    [v_LaborHoursVariance].[LaborHoursVariance],
    [v_LaborHoursVariance].[PercentHoursVariance]
FROM
    [v_LaborHoursVariance]
        INNER JOIN [v_JobManagerNames] ON
            [v_LaborHoursVariance].[JobID] = [v_JobManagerNames].[JobID]
WHERE
    [v_LaborHoursVariance].[PercentHoursVariance] =
    (SELECT
        MAX([v_LaborHoursVariance].[PercentHoursVariance])
      FROM
        [v_LaborHoursVariance]
    )
ORDER BY
    [v_LaborHoursVariance].[JobID]
;
```

	JobID	JobName	ClientName	EmployeeManager	EstHours	ActualHoursWorked	LaborHoursVariance	PercentHoursVariance
1	55841	AO Reid - Wonder Valley PH 2-2	AO Reid Construction	Galloway, Odessa	38	36.50	1.50	3.947368

```sql
/**
 * Exercise 07
 *
 * What is the average amount of time (labor hours) spent on a
 * \underline{completed} job task per square foot, as compared to the
 * estimated amount of time that should be spent on a task per square foot?
 *
 * Use the data in the JobTask table to calculate the average amount of
 * EstHours/Squarefeet, but use the data in the TimeSheet table to calculate
 * the average amount of time that was actually worked on a completed task.
 * I recommend creating separate views for the estimated hours per square
 * feet and the actual hours per square feet. The estimate view is a little
 * easier to create because it doesn’t require a join. Include all rows in the
 * JobTask table to get the average EstHours/Squarefeet for a task. To get the
 * average \underline{actual} hours per square feet requires that you join the
 * TimeSheet table and the JobTask table to be able to use the square feet in
 * the JobTask table. Do not include data for incompleted tasks when
 * calculating the ActualHours/SquareFeet. Remember that you have to SUM the
 * HoursWorked in the TimeSheet table by JobID and TaskID to get the Actual
 * HoursWorked from the TimeSheet table. I rounded the final results to 6
 * digits after the decimal point. The result table is at the top of the
 * next page. There is one row in the result table for each row in the Task
 * table. Sort the result table by TaskID.
 *
 * The ComparisonMessage should be generated as shown on the result table
 * above; if both the EstimatedHours and ActualHours are null, then put the
 * message ``Null Estimate" in the ComparisonMessage column. Remember that
 * a CASE statement in the SELECT list executes sequentially, so whatever
 * WHEN statement is placed first will be executed first. The CASE
 * statement stops executing as soon as a WHEN condition is true.
 * Potential problem: EstHours and Squarefeet are integers and must
 * be converted to decimal data types before they can be used in a
 * calculation that could generate a decimal result.
 */
DROP VIEW  [v_EstHoursPerSqFt];
CREATE VIEW [v_EstHoursPerSqFt] AS
SELECT
    [JobTask].[TaskID] AS 'TaskID',
    SUM([JobTask].[EstHours]) /
        CAST(SUM([JobTask].[SquareFeet]) AS DECIMAL(6, 2)) AS 'EstimatedHoursPerSqFt'
FROM
```

```
    [JobTask]
GROUP BY
    [JobTask].[TaskID]
;

----------
CREATE VIEW v_CompleteSqFtSum AS
SELECT
    TaskID,
    SUM(SquareFeet) 'SquareFeet'
FROM
    JobTask
WHERE
    DateCompleted IS NOT NULL
GROUP BY
    TaskID
;

CREATE VIEW v_CompleteActualHours AS
SELECT
    TimeSheet.TaskID,
    SUM(TimeSheet.HoursWorked) 'Hours'
FROM
    TimeSheet
        INNER JOIN JobTask ON
            TimeSheet.JobID = JobTask.JobID AND TimeSheet.TaskID = JobTask.TaskID
WHERE
    JobTask.DateCompleted IS NOT NULL
GROUP BY
    TimeSheet.TaskID
;

CREATE VIEW v_CompleteActualHoursPerSqFt AS
SELECT
    A.TaskID,
    A.[Hours] / S.SquareFeet 'ActualHoursPerSqFt'
FROM
    v_CompleteActualHours A
        INNER JOIN v_CompleteSqFtSum S ON
            A.TaskID = S.TaskID
;
```

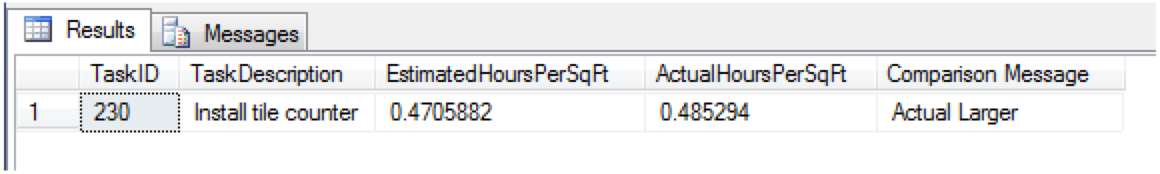
```sql
DROP VIEW [v_ActualHoursPerSqFt];
CREATE VIEW [v_ActualHoursPerSqFt] AS
SELECT
    [TimeSheet].[TaskID] AS 'TaskID',
    SUM([TimeSheet].[HoursWorked]) /
        SUM([JobTask].[SquareFeet]) AS 'ActualHoursPerSqFt'
FROM
    [v_TimeSheetsWithPay] [TimeSheet]
        INNER JOIN [JobTask] ON
            [TimeSheet].[JobID] = [JobTask].[JobID] AND
            [TimeSheet].[TaskID] = [JobTask].[TaskID]
WHERE
    [JobTask].[DateCompleted] IS NOT NULL
GROUP BY
    [TimeSheet].[TaskID]
;

------------------------------
DROP VIEW [v_HoursPerSqFtComparison];
CREATE VIEW [v_HoursPerSqFtComparison] AS
SELECT
    [Task].[TaskID] AS 'TaskID',
    [Task].[TaskDescription] AS 'TaskDescription',
    [v_EstHoursPerSqFt].[EstimatedHoursPerSqFt] AS 'EstimatedHoursPerSqFt',
    [v_CompleteActualHoursPerSqFt].[ActualHoursPerSqFt] AS 'ActualHoursPerSqFt',
    CASE
        WHEN [v_EstHoursPerSqFt].[EstimatedHoursPerSqFt] IS NULL
            THEN 'Null Estimate'
        WHEN [v_CompleteActualHoursPerSqFt].[ActualHoursPerSqFt] IS NULL
            THEN 'Null Actual'
        WHEN [v_EstHoursPerSqFt].[EstimatedHoursPerSqFt] > [v_CompleteActualHoursPerSqFt].[ActualHoursPerSqFt]
            THEN 'Estimate Larger'
        WHEN [v_CompleteActualHoursPerSqFt].[ActualHoursPerSqFt] > [v_EstHoursPerSqFt].[EstimatedHoursPerSqFt]
            THEN 'Actual Larger'
        ELSE
            'No Difference'
    END /**/ AS 'Comparison Message'
FROM
    [Task]
        LEFT OUTER JOIN [v_EstHoursPerSqFt] ON
            [Task].[TaskID] = [v_EstHoursPerSqFt].[TaskID]
```



```
        LEFT OUTER JOIN [v_CompleteActualHoursPerSqFt] ON
            [Task].[TaskID] = [v_CompleteActualHoursPerSqFt].[TaskID]
;
SELECT * FROM v_HoursPerSqFtComparison;
```

	TaskID	TaskDescription	EstimatedHoursPerSqFt	ActualHoursPerSqFt	Comparison Message
1	110	Remove Existing Floor	0.0717948	0.074358	Actual Larger
2	120	Clean mold	NULL	NULL	Null Estimate
3	130	Install sub-floor	0.0977859	0.097324	Estimate Larger
4	140	Design mosaic/glass	0.1770833	0.170052	Estimate Larger
5	150	Prepare sub floor	0.0247208	0.025385	Actual Larger
6	160	Install tile floor	0.0692834	0.066707	Estimate Larger
7	170	Grout	0.0419207	0.037266	Estimate Larger
8	180	Seal and finish work	0.0289778	0.024408	Estimate Larger
9	190	Install mosaic	0.0250000	NULL	Null Actual
10	200	Build structure	0.0500000	NULL	Null Actual
11	210	Remove existing tile	NULL	NULL	Null Estimate
12	220	Clean site	NULL	NULL	Null Estimate
13	230	Install tile counter	0.4705882	0.485294	Actual Larger
14	240	Install tile wall	0.1142857	NULL	Null Actual
15	260	Build mosaic/glass	0.2916666	NULL	Null Actual
16	270	General demolition	0.0714285	0.066964	Estimate Larger

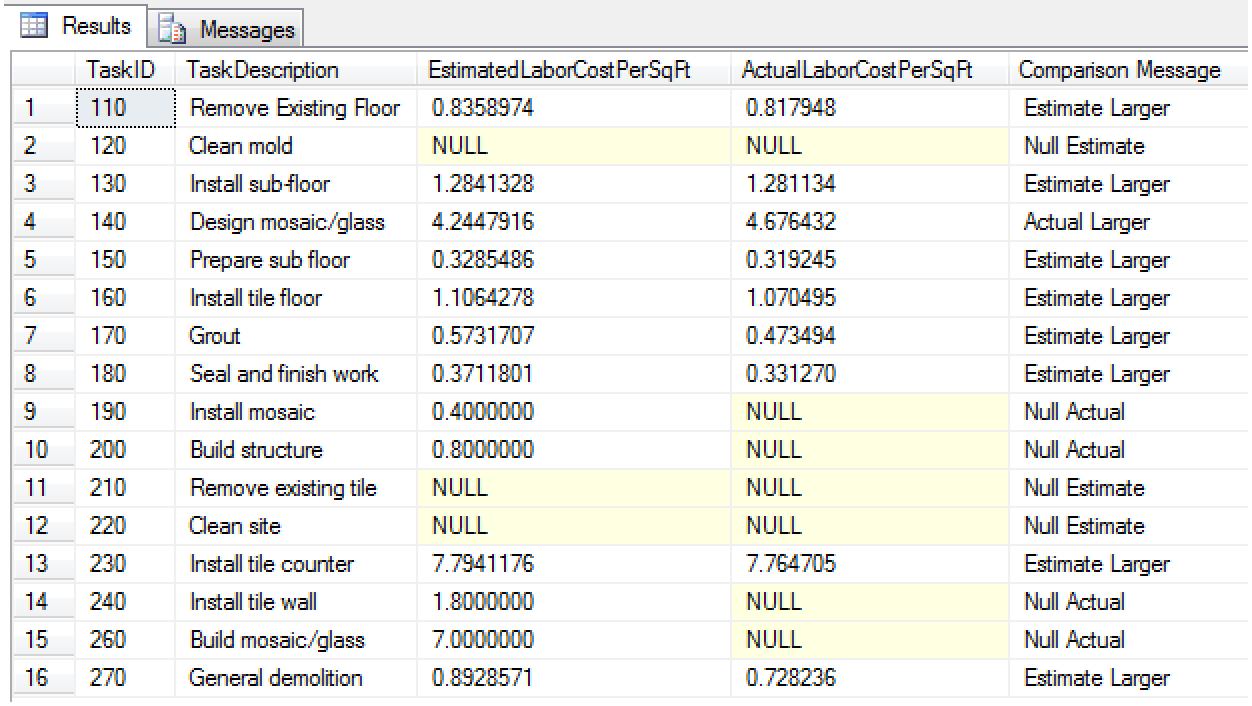
```
/**
 * Exercise 08
 *
 * Use the result table generated for question #7 to help you answer
 * this question. The goal of this query is to identify which task has
 * the largest negative difference between the EstimatedHoursPerSqFt and
 * ActualHoursPerSqFt (which estimate is the worst because the actual
 * is larger).
 */
SELECT
    *
FROM
    v_HoursPerSqFtComparison Comparison
WHERE
    Comparison.EstimatedHoursPerSqFt - Comparison.ActualHoursPerSqFt = (
        SELECT
            MIN(EstimatedHoursPerSqFt - ActualHoursPerSqFt)
        FROM
            v_HoursPerSqFtComparison
    )
;
```



	TaskID	TaskDescription	EstimatedHoursPerSqFt	ActualHoursPerSqFt	Comparison Message
1	230	Install tile counter	0.4705882	0.485294	Actual Larger

```
/**
 * Exercise 09
 *
 * The objective of this query is similar to that for question #7, except
 * this time we are going to look at labor costs rather than labor hours.
 * What is the average estimated labor cost per square foot as compared
 * to the actual labor cost per square foot for each task? I recommend
 * looking back at question #1, where you probably created a view to help
 * you calculate actual labor costs for a task on a job. That view will help
 * you with this question. Do \textbf{\underline{not}} include data for
 * incompleted tasks when calculating the actual labor cost/SquareFeet;
 * do include data for incompleted tasks when calculating the estimated
 * labor cost/squarefeet.
 */
DROP VIEW  [v_EstWagesPerSqFt];
CREATE VIEW [v_EstWagesPerSqFt] AS
SELECT
    [JobTask].[TaskID] AS 'TaskID',
    CAST(SUM([JobTask].[EstLaborCost]) AS DECIMAL) /
        CAST(SUM([JobTask].[SquareFeet]) AS DECIMAL(6, 2)) AS 'EstimatedWagesPerSqFt'
FROM
    [JobTask]
GROUP BY
    [JobTask].[TaskID]
;

----------
CREATE VIEW v_ActualPay AS
SELECT
    TS.TaskID,
    SUM(TS.LaborCost) 'ActualWagesPerSqFt'
FROM
    v_TimeSheetsWithPay TS
        INNER JOIN JobTask JT ON
            TS.JobID = JT.JobID AND TS.TaskID = JT.TaskID
WHERE
    JT.DateCompleted IS NOT NULL
GROUP BY
    TS.TaskID
;

DROP VIEW [v_ActualWagesPerSqFt];
```

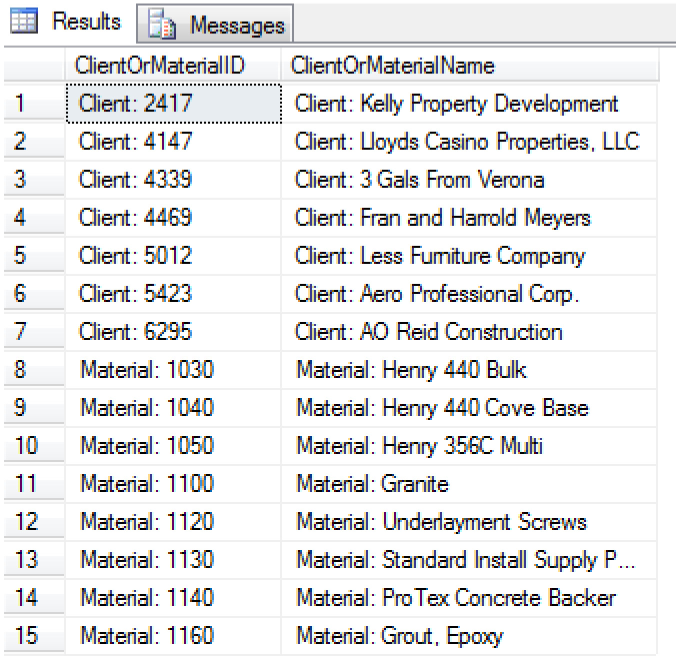
```
CREATE VIEW [v_ActualWagesPerSqFt] AS
SELECT
    [v_ActualPay].[TaskID] AS 'TaskID',
    v_ActualPay.ActualWagesPerSqFt /
        v_CompleteSqFtSum.SquareFeet AS 'ActualWagesPerSqFt'
FROM
    [v_ActualPay]
        INNER JOIN [v_CompleteSqFtSum] ON
            [v_ActualPay].[TaskID] = [v_CompleteSqFtSum].[TaskID]
;

------------------------------
SELECT
    [Task].[TaskID] AS 'TaskID',
    [Task].[TaskDescription] AS 'TaskDescription',
    [v_EstWagesPerSqFt].[EstimatedWagesPerSqFt] AS 'EstimatedLaborCostPerSqFt',
    [v_ActualWagesPerSqFt].[ActualWagesPerSqFt] AS 'ActualLaborCostPerSqFt',
    CASE
        WHEN [v_EstWagesPerSqFt].[EstimatedWagesPerSqFt] IS NULL
            THEN 'Null Estimate'
        WHEN [v_ActualWagesPerSqFt].[ActualWagesPerSqFt] IS NULL
            THEN 'Null Actual'
        WHEN [v_EstWagesPerSqFt].[EstimatedWagesPerSqFt] > [v_ActualWagesPerSqFt].[ActualWagesPerSqFt]
            THEN 'Estimate Larger'
        WHEN [v_ActualWagesPerSqFt].[ActualWagesPerSqFt] > [v_EstWagesPerSqFt].[EstimatedWagesPerSqFt]
            THEN 'Actual Larger'
        ELSE
            'No Difference'
    END /**/ AS 'Comparison Message'
FROM
    [Task]
        LEFT OUTER JOIN [v_EstWagesPerSqFt] ON
            [Task].[TaskID] = [v_EstWagesPerSqFt].[TaskID]
        LEFT OUTER JOIN [v_ActualWagesPerSqFt] ON
            [Task].[TaskID] = [v_ActualWagesPerSqFt].[TaskID]
;
```



	TaskID	TaskDescription	EstimatedLaborCostPerSqFt	ActualLaborCostPerSqFt	Comparison Message
1	110	Remove Existing Floor	0.8358974	0.817948	Estimate Larger
2	120	Clean mold	NULL	NULL	Null Estimate
3	130	Install sub-floor	1.2841328	1.281134	Estimate Larger
4	140	Design mosaic/glass	4.2447916	4.676432	Actual Larger
5	150	Prepare sub floor	0.3285486	0.319245	Estimate Larger
6	160	Install tile floor	1.1064278	1.070495	Estimate Larger
7	170	Grout	0.5731707	0.473494	Estimate Larger
8	180	Seal and finish work	0.3711801	0.331270	Estimate Larger
9	190	Install mosaic	0.4000000	NULL	Null Actual
10	200	Build structure	0.8000000	NULL	Null Actual
11	210	Remove existing tile	NULL	NULL	Null Estimate
12	220	Clean site	NULL	NULL	Null Estimate
13	230	Install tile counter	7.7941176	7.764705	Estimate Larger
14	240	Install tile wall	1.8000000	NULL	Null Actual
15	260	Build mosaic/glass	7.0000000	NULL	Null Actual
16	270	General demolition	0.8928571	0.728236	Estimate Larger

```sql
/**
 * Exercise 10
 *
 * Which clients did not have any jobs with a DateAccepted last year?
 * Which materials were not assigned (DateAssigned) to any job tasks last year?
 * Combine the clients and materials into a single result table (hint: Use the
 * UNION statement). Make sure that you use the GETDATE() function to
 * determine the correct year.
 */
DROP VIEW [v_ClientsWithoutJobsInPreviousYear];
CREATE VIEW [v_ClientsWithoutJobsInPreviousYear] AS
SELECT
    [Client].[ClientID] AS 'ClientID',
    [Client].[ClientName] AS 'ClientName'
FROM
    [Client]
WHERE
    [Client].[ClientID] NOT IN (
        SELECT
            ClientID
        FROM
               Job
        WHERE
            DATEDIFF(YEAR, DateAccepted, GETDATE()) = 1
    )
;

DROP VIEW v_MaterialsAssignedYearPrevious;
CREATE VIEW v_MaterialsAssignedYearPrevious AS
SELECT
    DISTINCT(Material.MaterialID)
FROM
    MaterialAssigned
        INNER JOIN MaterialPurchased ON
            MaterialAssigned.POID = MaterialPurchased.POID
        INNER JOIN Material ON
            MaterialPurchased.MaterialID = Material.MaterialID
WHERE
    DATEDIFF(YEAR, MaterialAssigned.DateAssigned, GETDATE()) = 1
;

DROP VIEW v_MaterialsNotAssignedYearPrevious;
```

```sql
CREATE VIEW v_MaterialsNotAssignedYearPrevious AS
SELECT
    MaterialID,
    MaterialName
FROM
    Material
WHERE
    MAterial.MaterialID NOT IN (SELECT * FROM v_MaterialsAssignedYearPrevious)


SELECT
    'Client: ' + CAST(ClientID AS VARCHAR) AS 'ClientOrMaterialID',
    'Client: ' + ClientName AS 'ClientOrMaterialName'
FROM
    v_ClientsWithoutJobsInPreviousYear
/**/
UNION
/**/
SELECT
    'Material: ' + CAST(MaterialID AS VARCHAR) AS 'ClientOrMaterialID',
    'Material: ' + MaterialName AS 'ClientOrMaterialName'
FROM
    v_MaterialsNotAssignedYearPrevious
;
```



	ClientOrMaterialID	ClientOrMaterialName
1	Client: 2417	Client: Kelly Property Development
2	Client: 4147	Client: Lloyds Casino Properties, LLC
3	Client: 4339	Client: 3 Gals From Verona
4	Client: 4469	Client: Fran and Harrold Meyers
5	Client: 5012	Client: Less Furniture Company
6	Client: 5423	Client: Aero Professional Corp.
7	Client: 6295	Client: AO Reid Construction
8	Material: 1030	Material: Henry 440 Bulk
9	Material: 1040	Material: Henry 440 Cove Base
10	Material: 1050	Material: Henry 356C Multi
11	Material: 1100	Material: Granite
12	Material: 1120	Material: Underlayment Screws
13	Material: 1130	Material: Standard Install Supply P...
14	Material: 1140	Material: ProTex Concrete Backer
15	Material: 1160	Material: Grout, Epoxy