Work Order SQL Assignment

**Please Note: Use full syntax to write the queries. (Preferred Microsoft SQL server 2016 Engine syntax)**

Please complete the following questions below **to the best of your ability.** Once returned based on your results, we will schedule a follow up interview. Table schemas have been provided below along with example data in those tables.

Answer as many questions as you can.

## SCHEMA

|  |  |  |
| --- | --- | --- |
| **Customer** | | |
| **Field** | **Type** | **Key** |
| CustomerId | (Int) | primary key |
| Name | (string) |  |
| Location | (string) |  |

|  |  |  |
| --- | --- | --- |
| **WorkOrder** | | |
| **Field** | **Type** | **Key** |
| WorkOrderId | (int) | primary key |
| CustomerId | (int) | foreign key - Customer.CustomerId |
| WorkTypeId | (int) | foreign key - WorkType.WorkTypeId |
| WorkRequestedDate | (date) |  |
| WorkDate | (date) |  |
| TimeTaken | (int) |  |
| Cost | (decimal:000.00) |  |

|  |  |  |
| --- | --- | --- |
| **WorkType** | | |
| **Field** | **Type** | **Key** |
| WorkTypeId | (int) | primary key / Auto Increment |
| Description | (string) |  |
| ExpectedTimeTaken | (int) |  |
| CostPerTimeUnit | (decimal:000.00) |  |

## Example Tables

|  |  |  |
| --- | --- | --- |
| **Customer** | | |
| CustomerId | Name | Location |
| 1 | William | Florida |
| 2 | Brian | Japan |
| 3 | Eric | Nowhere |
| 4 | Pete | Middle of Nowhere |
| 5 | George | Jungle |
| 6 | Andrew | Mazer |
| 7 | Harry | London |
| 8 | Ed Edwards Jr | Paris Idaho |

|  |  |  |  |
| --- | --- | --- | --- |
| **WorkType** | | | |
| WorkTypeId | Description | ExpectedTimeTaken | CostPerTimeUnit |
| 1 | Report Design | 1 | 1 |
| 2 | DB Design | 2 | 10 |
| 3 | UI Design | 5 | 20 |
| 4 | Flow Design | 10 | 5 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **WorkOrder** | | | | | | |
| WorkOrderId | CustomerId | WorkTypeId | WorkRequestedDate | WorkDate | TimeTaken | Cost |
| 1 | 3 | 2 | 03/26/2012 | 04/15/2012 | 4 | 40 |
| 2 | 5 | 4 | 01/21/2010 | 04/16/2010 | 8 | 50 |
| 3 | 2 | 1 | 07/26/2011 | 07/28/2011 | 1 | 2 |
| 4 | 7 | 3 | 06/20/2011 | 09/22/2011 | 10 | 220 |

## Questions

The questions below are not based on the data provided in the example tables above, you are not expected to return the values but the SQL statement used to generate questions below.

1. Find all orders created during October 2011. Return Customer Name, Location, Work Type Description, and Cost

SELECT Customer.Name, Customer.Location, WorkType.Description, WorkOrder.Cost

FROM WorkOrder

LEFT JOIN Customer ON Customer.CustomerID = WorkOrder.CustomerID

LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

WHERE WorkRequestedDate >= '10/01/2011' AND WorkRequestedDate <'11/01/2011'

2. Find all Work Types for which we have not billed an order in the last 12 months. Return Work Type Description

/\*I assumed orders are billed on the work date\*/

SELECT WorkType.Description

FROM WorkType

WHERE WorkType.Description NOT IN (

SELECT DISTINCT WorkType.Description

FROM WorkOrder LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

WHERE DATEDIFF(month,cast(getdate() as Date),WorkDate) <= 12)

1. Find all Work Types where the billed Time Taken is greater than the expected time taken. Return the Work Type Description, Customer Name, Time Taken, Expected Time Taken, and the difference between the Expected Time Taken and Time Taken.

SELECT WorkType.Description, Customer.Name, WorkOrder.TimeTaken, WorkType.ExpectedTimeTaken, (WorkOrder.TimeTaken-WorkType.ExpectedTimeTaken) AS DaysLate

FROM WorkOrder

LEFT JOIN Customer ON Customer.CustomerID = WorkOrder.CustomerID

LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

WHERE TimeTaken > ExpectedTimeTaken

4. Find the average cost by Customer for the last 8 months. Return the Customer Name and Average Cost. Ignore any customer that doesn't have a work order in the same time frame.

SELECT Name, AVG(Cost) as AverageCost

FROM WorkOrder

LEFT JOIN Customer ON Customer.CustomerID = WorkOrder.CustomerID

LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

WHERE DATEDIFF(month,cast(getdate() as Date),WorkDate) <= 8)

GROUP BY CustomerID

5-A Insert the following Work Order for a new customer

* Name: Ed Edwards
* Location: Paris Maine
* WorkTypeId: 1
* WorkRequestedDate: 2012-02-06
* WorkDate: 2012-02-07
* TimeTaken: 5
* Cost: 6.00

/\*I assumed the database would generate the CustomerID for the Customers table and the WorkOrderID for the WorkOrder table\*/

INSERT INTO Customer (Name, Location)

VALUES (Ed Edwards, Paris Maine)

Declare @CustomerIDNum as INT

SET @CustomerIDNum=(SELECT CustomerId FROM Customer WHERE Name=’Ed Edwards’ AND Location=’Paris Maine’)

INSERT INTO WorkOrder (CustomerId, WorkTypeId, WorkRequestedDate, WorkDate, TimeTaken, Cost)

VALUES (@CustomerIDNum , 1, ‘2012-02-06’, ‘2012-02-07’, 5, 6.00)

5-B Insert the following work order for an existing customer, assuming that no customer has the same name and location

* Name: Ed Edwards Jr
* Location: Paris Idaho
* WorkTypeId: 1
* WorkRequestedDate: 2012-01-06
* WorkDate: 2012-01-07
* TimeTaken: 12
* Cost: 24.00

Declare @CustomerIDNum as INT

SET @CustomerIDNum=(SELECT CustomerId FROM Customer WHERE Name= ‘Ed Edwards Jr’ AND Location= ‘Paris Idaho’)

INSERT INTO WorkOrder (CustomerId, WorkTypeId, WorkRequestedDate, WorkDate, TimeTaken, Cost)

VALUES (@CustomerIDNum , 1, ‘2012-01-06’, ‘2012-01-07’, 12, 24.00)

6. Delete the work orders for all customers in Paris Idaho in January 2012

DELETE \*

FROM WorkOrder

WHERE CustomerID IN (SELECT CustomerID FROM Customer WHERE Location = ‘Paris Idaho’)

AND WorkRequestedDate >= ‘01/01/2012’ AND WorkRequestedDate < ‘02/01/2012’

7. Find the total cost, by customer, for any customer with more than one order. Return the Customer Name and the Average Cost

SELECT Name, SUM(Cost) as TotalCost, AVG(Cost) as AverageCost

FROM WorkOrder

LEFT JOIN Customer ON Customer.CustomerID = WorkOrder.CustomerID

LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

GROUP BY CustomerId

HAVING COUNT(WorkOrderID) > 1

8. Update the Cost for all customers in Paris Maine to 12 for all orders created in February 2012

/\*I assumed orders were created on the Work Requested Date\*/

UPDATE WorkOrder

SET Cost=12

WHERE CustomerID IN (SELECT CustomerID FROM Customer WHERE Location=’Paris Maine’)

AND WorkRequestedDate>=’02/01/2012 AND WorkRequestedDate < ‘03/01/2012’

9. Count the Work Types where the average time taken on the order is less than the average expected time taken for the work type. Return the work type description and the count of the orders.

SELECT Description, COUNT(WorkOrderID) as Orders,

FROM WorkOrder

LEFT JOIN Customer ON Customer.CustomerID = WorkOrder.CustomerID

LEFT JOIN WorkType ON WorkType.WorkTypeID = WorkOrder.WorkTypeID

GROUP BY WorkTypeID

HAVING AVG(TimeTaken) < AVG(ExpectedTimeTaken)

10. Return the running total (cost) of all orders created in February, by Work Date, in descending order by Work Date.

/\*I assumed orders were created on the Work Requested Date\*/

SELECT WorkDate, SUM(Cost) OVER (ORDER BY WorkDate ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW) AS CumulativeCost

FROM WorkOrder

WHERE MONTH(WorkRequestedDate)=2

ORDER By WorkDate DESC