Operations Reporting Analyst Test

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 join customer
On workorder.customerID=customer.customerID
Join worktype
On workoder.worktypeID=worktype.worktypeID
Where workorder.WorkRequestedDate between “10/01/2011” and “10/31/2011”;
```

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

```
Select worktype.description from worktype
Where worktype.worktypeID not in
(Select distinct worktypeID from workorder
Where WorkRequestedDate>”06/13/2017”);
```

3. 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
From workorder join customer
On workorder.customerID=customer.customerID
Join worktype
On workoder.worktypeID=worktype.worktypeID
Where workorder.TimeTaken -worktype.ExpectedTimeTaken>0;
```

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 customer.name, AVG(workorder.cost)
From customer join workorder
On customer.customerID=workorder.customerID
Where workorder.WorkRequestedDate>”10/13/2017”
Group by customer.id, customer.name;
```

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

```
Insert into customer (name, location)
Values(“Ed Edwards”, “Paris Maine”);

Insert into workorder(cutomerID, worktypeID, workrequestedDate, workDate, TimeTaken, Cost)
Values((select MAX(customerID) from customer), 1, “2012-02-06”, “2012-02-07”, 5, 6);
```

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

```
Insert into workorder(cutomerID, worktypeID, workrequestedDate, workDate, TimeTaken, Cost)
Values((select customerID from customer
            Where name=”Ed Edwards Jr” and location=”Paris Idaho”),
    1, “2012-01-06”, “2012-01-07”, 12, 24);
```

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 between “2012-01-01” and “2012-01-31”;
```

7. Find the total cost, by customer, for any customer with more than one order. Return the Customer Name and the Average Cost(??return average cost or total cost??)

```
Select customer.Name, SUM(workorder.cost)
From customer join workorder on customer.customerID=workorder.customerID
Group by customer.customerID, customer.Name
Having COUNT(workorder.workorderID)>1;
```

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

```
Update workorder
Set cost=12
Where WorkRequesteDate between “2012-02-01” and “2012-02-29”
And
customerID in (select customerID from customer where Location=”Paris Maine”);
```

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 count(distinct WorkType.WorkTypeID)
From
workorder join worktype
on
workorder.workTypeID=worktype.worktypeID
Group by workorder.worktypeID
Having AVG(workorder.TimeTaken)<worktype.ExpectedTimeTaken;
```

Step 1: count the work types

```
Select worktype.description, count(workorder.workorderID)
From
workorder join worktype
on
workorder.workTypeID=worktype.worktypeID
Group by workorder.worktypeID, worktype.description
Having AVG(workorder.TimeTaken)<AVG(worktype.ExpectedTimeTaken);
```

Step 2: return work type description and count of orders

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

```
Select SUM(cost) from workorder
Where WorkRequestedDate between “2018-02-01” and “2018-02-28”
Group by workDate
Order by WorkDate Desc;
```