

# Unit 05 Problem Set Submission Form

## Overview

|               |                  |
|---------------|------------------|
| Your Name     | Jainish Savaliya |
| Your SU Email | jsavaliy@syr.edu |

## Instructions

Put your name and SU email at the top. Answer these questions all from the lab. When asked to include screenshots, please follow the screen shot guidelines from the first lab.

Remember as you complete the problem sets it is not only about getting it right / correct. We will discuss the answers in class so it's important to articulate anything you would like to contribute to the discussion in your answer:

- If you feel the question is vague, include any assumptions you've made.
- If you feel the answer requires interpretation or justification provide it.
- If you do not know the answer to the question, articulate what you tried and how you are stuck.

This how you receive credit for answering questions which might not be correct.

## Questions

Answer these questions using the problem set submission template. You will need to consult the logical model in the overview section for details. For any screenshots provided, please follow the guidelines for submitting a screenshot.

Write the following as SQL queries. If the query is ambiguous, fill in the gaps yourself and justify your reasoning. For each, include the SQL as a screenshot with the output of the query.

1. The GIS department would like to get a list of latitudes and longitudes of those vBay users who make valid bids on items and review users also. These are considered active participants in the platform and they would like to plot them on a map for a real-time dashboard.

SQLQuery\_3 - localhost.vbay (sa) - Azure Data Studio [Administrator]

```

1 --1
2 Select z.zip_lat, z.zip_lng, u.user_firstname + ' ' + u.user_lastname as username
3 From vb_users u
4 Join vb_zip_codes z on u.user_zip_code = z.zip_code
5 Join vb_bids b on b.bid_user_id = u.user_id
6 Join vb_user_ratings r on r.rating_by_user_id = u.user_id
7 Where bid_status = 'ok'
8 Group by z.zip_lat, z.zip_lng, u.user_firstname + ' ' + u.user_lastname
9 Order by u.user_firstname + ' ' + u.user_lastname, z.zip_lat, z.zip_lng
10
11

```

| zip_lat | zip_lng   | username             |
|---------|-----------|----------------------|
| 33.3800 | -111.8700 | Barb Barion          |
| 37.3200 | -121.9100 | Barry DeHatchett     |
| 33.1000 | -84.3400  | Bo Enarreau          |
| 42.8800 | -78.8500  | Carrie Dababbi       |
| 44.0100 | -92.4700  | Dan Deljons          |
| 38.3200 | -122.6400 | Gus Toffwind         |
| 43.0400 | -76.1400  | Hank Erchieff        |
| 37.3200 | -121.9100 | Isabelle Gunnering   |
| 39.8400 | -82.6000  | Jean Poole           |
| 35.0600 | -85.2500  | Les Ismoore          |
| 43.0400 | -76.1400  | Martin Eyzing        |
| 43.0400 | -76.1400  | Oliver Stuffsmission |
| 40.7100 | -73.9900  | Otto Moni            |
| 42.2500 | -77.7800  | Pete Moss            |
| 44.3300 | -75.9100  | Ray Ovligh           |
| 40.7100 | -73.9900  | Rose Abrov-Duresst   |
| 40.7100 | -73.9900  | Seymour Ofewa        |

Ln 1, Col 17 100% Windows (CRLF) UTF-8

Untitled - Notepad  
File Edit Format View Help  
jsavaliy@syr.edu

2. Northwind Traders would like to send out holiday cards to its Employees, Customers and Suppliers. Create a single mailing list from these sources. The mailing address should have 4 lines:
- 0) a label of the source of the address: Employee, Customer, or Supplier
  - 1) customer name and title,
  - 2) company,
  - 3) address, City, Region, Country, postal cod

SQLQuery\_4 - localhost.northwind (sa) - Azure Data Studio [Administrator]

```

1 Select 'Employee' as Emp_type, Firstname + ' ' + Lastname as Username, Title, 'random' as CompanyName, Address, City, Region, Country, PostalCode
2 From Employees
3 Union all
4 Select 'customer' as Cust_type, ContactName, ContactTitle, CompanyName, Address, City, Region, Country, PostalCode
5 From Customers
6 Union all
7 Select 'Suppliers' as Supp_type, ContactName, ContactTitle, CompanyName, Address, City, Region, Country, PostalCode
8 From Suppliers
9

```

| Emp_type | Username           | Title                    | CompanyName | Address                         | City           | Region | Country | PostalCode |  |
|----------|--------------------|--------------------------|-------------|---------------------------------|----------------|--------|---------|------------|--|
| Employee | Nancy Davolio      | Sales Representative     | random      | 908 W. Capital Way              | Tacoma         | WA     | USA     |            |  |
| Employee | Andrew Fuller      | Vice President, Sales    | random      | 722 Moss Bay Blvd.              | Kirkland       | WA     | USA     |            |  |
| Employee | Janet Leverling    | Sales Representative     | random      | 4110 Old Redmond Rd.            | Redmond        | WA     | USA     |            |  |
| Employee | Margaret Peacock   | Sales Representative     | random      | 14 Garrett Hill                 | London         | NULL   | UK      |            |  |
| Employee | Steven Buchanan    | Sales Manager            | random      | Coventry House Miner Rd.        | London         | NULL   | UK      |            |  |
| Employee | Michael Suyama     | Sales Representative     | random      | Edgeham Hollow Winchester Way   | London         | NULL   | UK      |            |  |
| Employee | Robert King        | Sales Representative     | random      | 4726 - 11th Ave. N.E.           | Seattle        | WA     | USA     |            |  |
| Employee | Laura Callahan     | Inside Sales Coordinator | random      | 7 Houndstooth Rd.               | London         | NULL   | UK      |            |  |
| Employee | Anne Dodsworth     | Sales Representative     | random      | Alfreds Futterkiste             | Berlin         | NULL   | Germany |            |  |
| customer | Maria Anders       | Sales Representative     | random      | Avenida de la Constitución 2222 | México D.F.    | NULL   | Mexico  |            |  |
| customer | Ana Trujillo       | Owner                    | random      | Mataderos 2312                  | México D.F.    | NULL   | Mexico  |            |  |
| customer | Antonio Moreno     | Owner                    | random      | 120 Hanover Sq.                 | London         | NULL   | UK      |            |  |
| customer | Thomas Hardy       | Sales Representative     | random      | Berguvsvägen 8                  | Luleå          | NULL   | Sweden  |            |  |
| customer | Christina Berglund | Order Administrator      | random      | Forsterstr. 57                  | Mannheim       | NULL   | Germany |            |  |
| customer | Hanna Moos         | Sales Representative     | random      | Blondesdél père et fils         | Strasbourg     | NULL   | France  |            |  |
| customer | Frédérique Citeaux | Marketing Manager        | random      | Bólidio Comidas preparadas      | C/ Araquil, 67 | Madrid | NULL    | Spain      |  |
| customer | Martin Sommer      | Owner                    | random      |                                 |                |        |         |            |  |

Ln 1, Col 17 100% Windows (CRLF) UTF-8

Untitled - Notepad  
File Edit Format View Help  
jsavaliy@syr.edu

3. vBay would like a list of items with item id name and type along with a count of bids with bid type (ok, low\_bid, item\_seller). There should be a column for each of the bid types.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, with 'northwind' selected under 'SERVERS'. The right pane shows a SQL query window with the following query:

```

--3
With pivot_source as (
  Select i.item_id, i.item_name, i.item_type, vb_bids.bid_status
  From vb_items i
  Join vb_bids on vb_bids.bid_item_id = i.item_id
)
Select * from pivot_source PIVOT (
  Count(bid_status) for bid_status in (ok, low_bid, item_seller)
) as pivot_table
  
```

The 'Results' pane displays the following data:

| item_id | item_name                       | item_type      | ok | low_bid | item_seller |
|---------|---------------------------------|----------------|----|---------|-------------|
| 1       | Used Pink Bathrobe              | All Other      | 2  | 1       | 0           |
| 2       | Rare Mint Snow Globe            | Collectables   | 1  | 0       | 0           |
| 3       | Smurf TV Tray                   | Collectables   | 1  | 0       | 0           |
| 4       | Alf Alarm Clock                 | Collectables   | 1  | 0       | 0           |
| 5       | Shatner's old Toupee            | Collectables   | 3  | 0       | 2           |
| 6       | Slightly-damaged Golf Bag       | Sporting Goods | 3  | 0       | 0           |
| 7       | Some Beanie Babies, New with... | Collectables   | 1  | 0       | 0           |
| 8       | Dukes Of Hazard ashtray         | Collectables   | 8  | 1       | 0           |
| 9       | case of vintage tube socks      | Antiques       | 2  | 0       | 0           |
| 10      | Kleenex used by Dr. Dre         | Collectables   | 1  | 0       | 0           |
| 11      | Farrah Faucet poster            | Collectables   | 3  | 0       | 0           |
| 12      | Pez dispensers                  | Collectables   | 1  | 1       | 0           |
| 13      | Antique Desk                    | Antiques       | 5  | 1       | 0           |
| 14      | SQL for Dummies                 | Books          | 1  | 0       | 0           |
| 15      | Fuzzy Logic                     | Books          | 1  | 0       | 0           |
| 16      | Dusty Vase                      | Antiques       | 6  | 0       | 0           |
| 17      | Old Diamond Ring                | Jewelry        | 2  | 0       | 0           |

4. Northwind traders has 3 different shippers. For each customer ID and CompanyName list the total amount of shipping freight paid for each of the three different shippers. There should be a column for each shipper.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, with 'northwind' selected under 'SERVERS'. The right pane shows a SQL query window with the following query:

```

--4
With source_of_pivot as (
  Select c.CustomerID, c.CompanyName, s.CompanyName as Company, o.Freight
  From Customers c
  Join orders o on o.CustomerID = c.CustomerID
  Join Shippers s on s.ShipperID = o.ShipVia
)
Select * from source_of_pivot PIVOT(
  sum(Freight) for ShipperID in ([Speedy Express],[United Package],[Federal Shipping])
) as pivot_table
  
```

The 'Results' pane displays the following data:

| CustomerID | CompanyName | Speedy Express | United Package | Federal Shipping |
|------------|-------------|----------------|----------------|------------------|
| 1          | ALFKI       | 95.03          | 61.02          | 69.53            |
| 2          | ANATR       | 43.90          | NULL           | 53.52            |
| 3          | ANTON       | 63.09          | 116.56         | 88.87            |
| 4          | AROUT       | 41.95          | 358.54         | 71.46            |
| 5          | BERGS       | 189.44         | 1074.51        | 295.57           |
| 6          | BLAUS       | 0.15           | 126.19         | 41.92            |
| 7          | BONAP       | 217.96         | 215.70         | 190.00           |
| 8          | BOLID       | 16.16          | 175.01         | 597.14           |
| 9          | BONAP       | 341.16         | 419.57         | 502.36           |
| 10         | BOTTI       | 129.42         | 162.17         | 152.05           |
| 11         | BSBEV       | NULL           | 129.26         | 52.37            |
| 12         | CACTU       | 3.17           | 17.22          | 3.25             |
| 13         | CENTC       | NULL           | NULL           | 123.19           |
| 14         | CHOPS       | 171.68         | 72.37          | 66.28            |
| 15         | COMMI       | 121.62         | 47.45          | NULL             |
| 16         | CONSH       | 6.17           | 262.29         | NULL             |
| 17         | WANOK       | 170.58         | NULL           | NULL             |

- Unpivot all the dates in the Northwind orders table, creating a table output with three columns: order id, type of date (ship date, require date, order date) and the date itself.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

```

16 )
17 Select * from source_of_pivot PIVOT(
18 sum([ShipDate]) for [ShipDate] IN ([Speedy Express],[United Package],[Federal Shipping]))
19 as pivot_table
20
21 --$
22 With order_unpivot as (
23 Select orderID , order_Date , type_of_order
24 From orders UNPIVOT(
25 Order_Date for type_of_order IN(OrderDate , RequiredDate , ShippedDate)
26 ) As unpivot_table)
27 Select * from order_unpivot
28

```

The Results pane shows the output of the query, which is a table with three columns: orderID, order\_Date, and type\_of\_order. The data is as follows:

| orderID | order_Date              | type_of_order |
|---------|-------------------------|---------------|
| 1       | 1996-07-04 00:00:00.000 | OrderDate     |
| 2       | 1996-08-01 00:00:00.000 | RequiredDate  |
| 3       | 1996-07-16 00:00:00.000 | ShippedDate   |
| 4       | 1996-07-05 00:00:00.000 | OrderDate     |
| 5       | 1996-08-16 00:00:00.000 | RequiredDate  |
| 6       | 1996-07-10 00:00:00.000 | ShippedDate   |
| 7       | 1996-07-08 00:00:00.000 | OrderDate     |
| 8       | 1996-08-05 00:00:00.000 | RequiredDate  |
| 9       | 1996-07-12 00:00:00.000 | ShippedDate   |
| 10      | 1996-07-08 00:00:00.000 | OrderDate     |
| 11      | 1996-08-05 00:00:00.000 | RequiredDate  |
| 12      | 1996-07-15 00:00:00.000 | ShippedDate   |
| 13      | 1996-07-09 00:00:00.000 | OrderDate     |
| 14      | 1996-08-06 00:00:00.000 | RequiredDate  |
| 15      | 1996-07-11 00:00:00.000 | ShippedDate   |
| 16      | 1996-07-10 00:00:00.000 | OrderDate     |
| 17      | 1996-07-24 00:00:00.000 | RequiredDate  |

- Unpivot the FudgeFlix titles which are just Movies so that it is easier to query titles which are available in different formats such as instant, DVD, and Blu-ray. Flatten these three columns into a single column with types, but only include a row when the value is 1 in the column.

The screenshot shows the SQL Server Enterprise Manager interface. The query editor displays the following SQL code:

```

1 --6
2 Select title_name, type_of_movie_format from ff_titles
3 UNPIVOT(
4 Movie_format for type_of_movie_format IN (
5 title_instant_available, title_dvd_available, title_bluray_available
6 )
7 ) As unpivot_table
8 Where title_type = 'Movie' and movie_format = 1
9

```

The Results pane shows the output of the query, which is a table with two columns: title\_name and type\_of\_movie\_format. The data is as follows:

| title_name | type_of_movie_format                                |
|------------|---|
| 1          | Meet Joe Black title_dvd_available                  |
| 2          | Vampires title_dvd_available                        |
| 3          | Eyes Wide Shut title_dvd_available                  |
| 4          | Eyes Wide Shut title_bluray_available               |
| 5          | U.S. Marshals title_instant_available               |
| 6          | U.S. Marshals title_dvd_available                   |
| 7          | The Mighty title_dvd_available                      |
| 8          | Dark City title_dvd_available                       |
| 9          | 8 Monkey title_dvd_available                        |
| 10         | The Horse Whisperer title_dvd_available             |
| 11         | Armageddon title_dvd_available                      |
| 12         | Armageddon title_bluray_available                   |
| 13         | Pleasantville title_dvd_available                   |
| 14         | The Red Violin title_dvd_available                  |
| 15         | Apt Pupil title_dvd_available                       |
| 16         | Velvet Goldmine title_dvd_available                 |
| 17         | Sesame Street: Elmo Saves Ch... title_dvd_available |

7. Get a list of the employees from the payroll database as they were on May 31, 2018.

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, including the 'payroll' database. The right pane shows a SQL query window with the following query:

```
1 Select employee_id, employee_firstname + ' ' + employee_lastname as Name
2 From employees
3 For SYSTEM_TIME As of '2018-05-31'
4
```

The query results are displayed in a grid with the following data:

| employee_id | Name                |
|-------------|---------------------|
| 46          | June Nipper         |
| 54          | Alma Frienzengon    |
| 27          | Artie Choke         |
| 5           | Willie Pas-D'course |
| 47          | Sue Mii             |
| 1           | Michael Fudge       |
| 22          | Ajden Knowone       |
| 7           | Eileen Touda-Left   |
| 36          | Ally Gator          |
| 43          | Eura Lyre           |
| 52          | Jin Netics          |
| 6           | Bob Emweave         |
| 55          | Adam Antium         |
| 26          | Anita Favor         |
| 30          | Sandy Beeches       |
| 33          | Hugh Japple         |
| 39          | Tim Idd             |

8. Produce a report of Gus Toffwind's pay increases from the payroll database. Include the id, name SSN of the employee along with pay rate, previous pay rate and pay increase (difference between the current and previous pay rates).

The screenshot shows the SQL Server Enterprise Manager interface. The left pane displays the server hierarchy, including the 'payroll' database. The right pane shows a SQL query window with the following query:

```
1 Select employee_id, employee_firstname, employee_lastname, employee_ssn, employee_pay_rate as recent_pay,
2 LAG(employee_pay_rate)
3 Over (order by employee_pay_rate) as previous_pay, employee_pay_rate - LAG(employee_pay_rate,1)
4 as pay_difference
5 From employees_history
6 Where employee_firstname LIKE 'Gus%'
7
```

The query results are displayed in a grid with the following data:

| employee_id | employee_firstname | employee_lastname | employee_ssn | employee_pay_rate | previous_pay | pay_difference |
|-------------|--------------------|-------------------|--------------|-------------------|--------------|----------------|
| 25          | Gus                | Toffwind          | 638-22-3313  | 17.8193           | 17.25        | 0.5693         |
| 25          | Gus                | Toffwind          | 638-22-3313  | 18.6212           | 17.8193      | 0.8019         |

## Reflection

Use this section to reflect on your learning. To achieve the highest grade on the assignment you must be as descriptive and personal as possible with your reflection.

1. What are the key things you learned through the process of completing this assignment?  
-> **Use of joins, window functions, operations like union and intersection, and how to pivot and unpivot tables in databases are the main items discovered while working on the project.**
2. What were the challenges or roadblocks (if any) you encountered on the way to completing it?  
➔ **You must watch the video several times to fully grasp the ideas.**
3. Were you prepared for this assignment? What can you do to be better prepared?  
-> **I was ready for the homework after reading the material and seeing the videos.**
4. Now that you have completed the assignment rate your comfort level with this week's material. This should be an honest assessment: (choose one)  
  
**4 ==> I understand this material and can explain it to others.**