

# Unit 05 Problem Set Submission Form

## Overview

Your Name	Hendi Kushta
Your SU Email	hkushta@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. How many item types are there? Perform an analysis of each item type. For each item type, provide the count of items in that type, the minimum, average, and maximum item reserve prices for that type. Sort the output by item type.

```
SELECT COUNT(DISTINCT item_type) AS nr_of_item_types
FROM vb_items
```

```
GO
```

```
SELECT
    item_type,
    COUNT(*) AS count_items_in_a_type, -- or count(item_reserve)
    MIN(item_reserve) AS minimum_item_reserve_price,
    MAX(item_reserve) AS maximum_item_reserve_price,
    AVG(item_reserve) AS average_item_reserve_price
FROM vb_items
GROUP BY item_type
ORDER BY item_type
```

	nr_of_item_types
1	8

Open ▾

1 hkushta

Plain Text ▾ Tab Width: 8 ▾ Ln 1, Col 8 ▾ INS

	item_type	count_items_in_a_type	minimum_item_reserve_price	maximum_item_reserve_price	average_item_reserve_price
1	All Other	4	0.99	10000000.00	2500004.86
2	Antiques	6	9.00	250.00	81.5833
3	Books	3	4.50	10.99	8.48
4	Collectables	14	5.00	500.00	105.3828
5	Electronics	1	15.00	15.00	15.00
6	Jewelry	2	6.95	599.99	303.47
7	Sporting Goods	2	12.50	12.75	12.625
8	Tickets	2	5.00	750.00	377.50

- Perform an analysis of each item in the “Antiques” and “Collectables” item types. For each item display the name, item type and item reserve. Include the min, max and average item reserve over each item type so that the current item reserve can be compared to these values.

```
SELECT
    item_name,
    item_type,
    item_reserve,
    MIN(item_reserve) OVER (PARTITION BY item_type) AS minimum_item_reserve_price,
    MAX(item_reserve) OVER (PARTITION BY item_type) AS maximum_item_reserve_price,
    AVG(item_reserve) OVER (PARTITION BY item_type) AS average_item_reserve_price
FROM vb_items
WHERE item_type = 'Antiques' OR item_type = 'Collectables'
GROUP BY item_name, item_type, item_reserve
```

Open ▾

1 hkushta

Plain Text ▾

	item_name	item_type	item_reserve	minimum_item_reserve_price	maximum_item_reserve_price	average_item_reserve_price
1	a Toaster	Antiques	20.00	9.00	250.00	81.5833
2	Antique Desk	Antiques	250.00	9.00	250.00	81.5833
3	Brass French Press	Antiques	45.50	9.00	250.00	81.5833
4	case of vintage tube socks	Antiques	9.00	9.00	250.00	81.5833
5	Dusty Vase	Antiques	100.00	9.00	250.00	81.5833
6	Original Coke Bottle from...	Antiques	65.00	9.00	250.00	81.5833
7	Alf Alarm Clock	Collectables	5.00	5.00	500.00	105.3828
8	Autographed Mik Jagger Po...	Collectables	75.00	5.00	500.00	105.3828
9	Carlos Villalba BobbleHead	Collectables	49.95	5.00	500.00	105.3828
10	Dukes Of Hazard ashtray	Collectables	149.99	5.00	500.00	105.3828
11	Farrah Fawcet poster	Collectables	50.00	5.00	500.00	105.3828
12	Joe Montana Figurine	Collectables	200.00	5.00	500.00	105.3828
13	Kleenex used by Dr. Dre	Collectables	500.00	5.00	500.00	105.3828
14	Mike Fudge BobbleHead	Collectables	49.95	5.00	500.00	105.3828
15	PacMan Fever lunchbox	Collectables	29.99	5.00	500.00	105.3828
16	Pez dispensers	Collectables	10.00	5.00	500.00	105.3828
17	Rare Mint Snow Globe	Collectables	30.50	5.00	500.00	105.3828
18	Shatner's old Toupee	Collectables	199.99	5.00	500.00	105.3828
19	Smurf TV Tray	Collectables	25.00	5.00	500.00	105.3828
20	Some Beanie Babies, New w...	Collectables	99.99	5.00	500.00	105.3828

3. Write a query to include the names, counts (number of ratings) and average seller ratings (as a decimal) of users. For reference, User Carrie Dababbbi has 4 seller ratings and an average rating of 4.75.

```

SELECT
    u.user_firstname + ' ' + u.user_lastname AS usr_full_name,
    COUNT(*) AS number_of_ratings,
    AVG(CAST(rating_value AS FLOAT)) AS average_seller_ratings
FROM vb_users u
    JOIN vb_user_ratings ur ON ur.rating_for_user_id = u.user_id
WHERE rating_astype = 'Seller'
GROUP BY user_firstname, user_lastname

```

	usr_full_name	number_of_ratings	average_seller_ratings
1	Rose Abov-Duresst	3	1
2	Ty Anott	2	2.5
3	Barb Barion	2	3.5
4	Carrie Dababbbi	4	4.75
5	Martin Eyezing	2	2.5
6	Les Ismoore	2	2.5
7	Anita Job	1	3
8	Abby Kuss	3	4.333333333333333
9	Victor Rhee	1	4

4. Create a list of "Collectable" item types with more than 1 bid. Include the name of the item and the number of bids making sure the item with the most bids appear first.

```

SELECT
    i.item_name,
    COUNT(b.bid_item_id) AS number_of_bids_per_item
FROM vb_items i
    JOIN vb_bids b ON b.bid_item_id = i.item_id
WHERE i.item_type = 'Collectables'
GROUP BY i.item_name, i.item_type
HAVING COUNT(b.bid_item_id) > 1
ORDER BY number_of_bids_per_item DESC

```

	item_name	number_of_bids_per_item	Open
1	Dukes Of Hazard ashtray	9	1 hkushta
2	Autographed Mik Jagger Po...	6	Plain Text
3	Shatner's old Toupee	5	
4	Rare Mint Snow Globe	3	
5	Farrah Fawcet poster	3	
6	Pez dispensers	2	

5. Generate a valid bidding history for any given item of your choice. Display the item id, item name a number representing the order the bid was placed, the bid amount and the bidder's name. Here's an example showing the first 3 bids on item 11.

item_id	item_name	bid_order	bid_amount	bidder
11	Dukes Of Hazard ashtray	1	150.0000	Dan Delyons
11	Dukes Of Hazard ashtray	2	175.0000	Al Fresco
11	Dukes Of Hazard ashtray	3	200.0000	Carrie Dababbi

```
SELECT
    i.item_id,
    i.item_name,
    ROW_NUMBER() OVER(ORDER BY bid_amount) AS bid_order,
    b.bid_amount,
    u.user_firstname + ' ' + u.user_lastname AS bidder
FROM vb_items i
JOIN vb_bids b ON b.bid_item_id = i.item_id
JOIN vb_users u ON u.user_id = b.bid_user_id
WHERE item_name = 'Antique Desk' and bid_status = 'ok'
```

	item_id	item_name	bid_order	bid_amount	bidder	Open
1	18	Antique Desk	1	251.00	Ray Ovligh	1 hkushta
2	18	Antique Desk	2	252.00	Barb Barion	Plain Text
3	18	Antique Desk	3	253.00	Ray Ovligh	
4	18	Antique Desk	4	254.00	Barb Barion	
5	18	Antique Desk	5	255.00	Ray Ovligh	

6. Re-Write your query in the previous question to include the names of the next and previous bidders, like this example again showing the first 3 bids for item 11.

item_name	bid_order	bid_amount	prev_bidder	bidder	next_bidder
Dukes Of Hazard ashtray	1	150.0000	NULL	Dan Delyons	Al Fresco
Dukes Of Hazard ashtray	2	175.0000	Dan Delyons	Al Fresco	Carrie Dababbi
Dukes Of Hazard ashtray	3	200.0000	Al Fresco	Carrie Dababbi	Gus Toffwind

```

SELECT
    i.item_id,
    i.item_name,
    ROW_NUMBER() OVER(ORDER BY i.item_id) AS bid_order,
    b.bid_amount,
    LAG(u.user_firstname + ' ' + u.user_lastname) OVER(ORDER BY i.item_id) AS prev_bidder,
    u.user_firstname + ' ' + u.user_lastname AS bidder,
    LEAD(u.user_firstname + ' ' + u.user_lastname) OVER(ORDER BY i.item_id) AS next_bidder
FROM vb_items i
JOIN vb_bids b ON b.bid_item_id = i.item_id
JOIN vb_users u ON u.user_id = b.bid_user_id
WHERE item_name = 'Antique Desk' and bid_status = 'ok'

```

	item_id	item_name	bid_order	bid_amount	prev_bidder	bidder	next_bidder
1	18	Antique Desk	1	251.00	NULL	Ray Ovligh	Barb Barion
2	18	Antique Desk	2	252.00	Ray Ovligh	Barb Barion	Ray Ovligh
3	18	Antique Desk	3	253.00	Barb Barion	Ray Ovligh	Barb Barion
4	18	Antique Desk	4	254.00	Ray Ovligh	Barb Barion	Ray Ovligh
5	18	Antique Desk	5	255.00	Barb Barion	Ray Ovligh	NULL

7. Find the names and emails of the users who give out the worst ratings (lower than the overall average rating) to either buyers or sellers (no need to differentiate whether the user rated a buyer or seller), and only include those users who have submitted more than 1 rating.

```

with ctel as (
    SELECT
        b.user_firstname + ' ' + b.user_lastname AS buyer_full_name,
        b.user_email,
        count(ur.rating_value) AS number_of_reviews,
        AVG(cast(ur.rating_value AS float)) AS average_of_reviews
    FROM vb_user_ratings ur
    JOIN vb_users b ON b.user_id = ur.rating_by_user_id
    GROUP BY
        b.user_firstname, b.user_lastname, b.user_email
    HAVING count(ur.rating_value) > 1
)
SELECT *
FROM ctel
WHERE average_of_reviews < (SELECT AVG(cast(rating_value AS FLOAT)) AS overall_average_rating
                           FROM vb_user_ratings ur)

```

	buyer_full_name	user_email	number_of_reviews	average_of_reviews
1	Anne Dewey	adewey@mail.org	2	2.5
2	Rose Abov-Duresst	rabovdu@mail.org	2	3
3	Ray Ovligh	rovligh@mail.org	2	3
4	Pete Moss	pmoss@mail.org	2	3
5	Oliver Stuffission	ostuff@mail.org	2	1
6	Carrie Dababbi	cdababbi@mail.org	4	3.25

8. Produce a report of the KPI (key performance indicator) user bids per item. Show the user's name and email total number of valid bids, total count of items bid upon and then the ratio of bids to items. As a check, Anne Dewey's bids per item ratio is 1.666666

```

SELECT
    u.user_firstname + ' ' + u.user_lastname AS usr_fullname,
    u.user_email,
    COUNT(bid_id) AS total_number_of_valid_bids,
    COUNT(DISTINCT bid_item_id) AS total_count_of_items_bid_upon,
    cast(COUNT(bid_id)AS float)/COUNT(DISTINCT bid_item_id)
FROM vb_users u
    JOIN vb_bids b ON b.bid_user_id = u.user_id
    JOIN vb_items i ON i.item_id = b.bid_item_id
WHERE b.bid_status = 'ok'
GROUP BY u.user_firstname, u.user_lastname, u.user_email

```

Open ▾

1 hkushta

Plain Text ▾

	usr_fullname ▾	user_email ▾	total_number_of_valid_bids ▾	total_count_of_items_bid_upon ▾	(No column name) ▾
1	Abby Kuss	abuss@mail.org	3	1	3
2	Anne Dewey	adewey@mail.org	5	3	1.6666666666666667
3	Barb Barion	bbarion@mail.org	3	2	1.5
4	Barry DeHatchett	bdehatchett@mail.org	5	1	5
5	Bo Enarreau	benarreau@mail.org	2	2	1
6	Gus Toffwind	gtoffwind@mail.org	2	2	1
7	Isabelle Gunnering	igunner@mail.org	7	2	3.5
8	Les Ismoore	lismoore@mail.org	3	3	1
9	Martin Eyezing	meveyzing@mail.org	1	1	1
10	Rose Abov-Duresst	rabovdu@mail.org	2	2	1
11	Ray Ovlight	rovlight@mail.org	6	3	2
12	Victor Rhee	vrhee@mail.org	2	2	1
13	Seymour Ofewe	sofewe@mail.org	2	1	2
14	Pete Moss	pmoss@mail.org	2	2	1
15	Otto Moni	omonni@mail.org	2	2	1
16	Oliver Stuffission	ostuff@mail.org	2	1	2
17	Bank Eschke	heskies@mail.org	2	1	2

Open ▾

1 hkushta

Plain Text ▾

9. Among items not sold, show highest bidder name and the highest bid for each item. Make sure to include only valid bids.

```

WITH cte1 AS (
    SELECT
        u.user_firstname + ' ' + u.user_lastname AS usr_fullname,
        i.item_name,
        MAX(b.bid_amount) AS highest_bid_amount,
        RANK() OVER(PARTITION BY item_name ORDER BY bid_amount DESC) AS desc_ranking
    FROM vb_items i
        JOIN vb_bids b ON b.bid_item_id = i.item_id
        JOIN vb_users u ON u.user_id = b.bid_user_id
    WHERE item_sold = 0 AND b.bid_status = 'ok'
    GROUP BY i.item_name, user_firstname, user_lastname, bid_amount, item_name
)
SELECT usr_fullname, item_name, highest_bid_amount
FROM cte1
WHERE desc_ranking = 1

```

Open ▾

1 hkushta

Plain Text ▾

	usr_fullname	item_name	highest_bid_amount
1	Les Ismoore	Alf Alarm Clock	5.01
2	Ray Ovligh	Antique Desk	255.00
3	Victor Rhee	case of vintage tube socks	9.02
4	Al Fresco	Client/Server Survival Gu..	11.00
5	Isabelle Gunnering	Dukes Of Hazard ashtray	325.00
6	Jean Poole	Dusty Vase	106.00
7	Ray Ovligh	Farrah Fawcet poster	515.00
8	Martin Eyezing	Joe Montanna Figurine	205.00
9	Ray Ovligh	Kleenex used by Dr. Dre	1000.00
10	Otto Moni	Old Diamond Ring	601.00
11	Gus Toffwind	Original Coke Bottle from..	70.00
12	Dan Delyons	Shatner's old Toupee	202.00
13	Seymour Ofewe	Slightly-damaged Golf Bag	14.50
14	Les Ismoore	Some Beanie Babies, New w..	250.00
15	Al Fresco	SQL for Dummies	11.00
16	Isabelle Gunnering	Ten Speed Bike	22.00

Open

1 hkushta

Plain Text

10. Write a query with output similar to question 3, but also includes the overall average seller rating, and the difference between each user's average rating and the overall average. For reference, the overall average seller rating should be 3.2.

```
WITH cte1 AS (
    SELECT
        u.user_firstname + ' ' + u.user_lastname AS usr_full_name,
        COUNT(*) AS number_of_ratings,
        AVG(CAST(rating_value AS FLOAT)) AS average_seller_ratings
    FROM vb_users u
    JOIN vb_user_ratings ur ON ur.rating_for_user_id = u.user_id
    WHERE rating_astype = 'Seller'
    GROUP BY user_firstname, user_lastname
),
cte2 AS (
    SELECT
        AVG(CAST(rating_value AS FLOAT)) AS sellers_overall_rating
    FROM vb_user_ratings WHERE rating_astype = 'Seller'
)
SELECT *,
    average_seller_ratings - sellers_overall_rating AS delta_change
FROM cte1
CROSS JOIN cte2
GROUP BY usr_full_name, number_of_ratings, average_seller_ratings, sellers_overall_rating
```

Open

1 hkushta

Plain Text

	usr_full_name	number_of_ratings	average_seller_ratings	sellers_overall_rating	delta_change
1	Abby Kuss	3	4.333333333333333	3.2	1.133333333333329
2	Anita Job	1	3	3.2	-0.20000000000000018
3	Barb Barion	2	3.5	3.2	0.2999999999999998
4	Carrie Dababbi	4	4.75	3.2	1.5499999999999998
5	Les Ismoore	2	2.5	3.2	-0.7000000000000002
6	Martin Eyezing	2	2.5	3.2	-0.7000000000000002
7	Rose Abov-Duresst	3	1	3.2	-2.2
8	Ty Anott	2	2.5	3.2	-0.7000000000000002
9	Victor Rhee	1	4	3.2	0.7999999999999998

Open

1 hkushta

Plain Text

## 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?

How to use window functions in queries.

2. What were the challenges or roadblocks (if any) you encountered on the way to completing it?

3. Were you prepared for this assignment? What can you do to be better prepared?

Yes, I was

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.**

3 ==> I understand this material.

2 ==> I somewhat understand the material but sometimes need guidance from others.

1 ==> I understand very little of this material and need extra help.