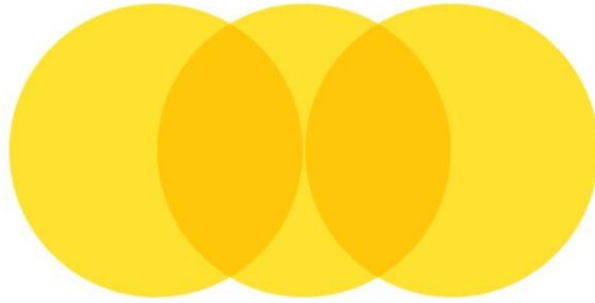


Bytewise Fellowship



Fellowship.

Powered By Bytewise

Task 6: JOINS & Window Functions on Sales Data

Name	Muhammad Ahsan Saleem
Date	8 th July, 2024
Submitted to:	Muhammad Bilal

Introduction:

Task 6 of the Bytewise Fellowship focused on comprehensive data retrieval techniques using SQL. This task aimed to write queries that fetch detailed information from the sales database by utilizing joins and window functions. The goal was to retrieve orders placed by customers along with their details, fetch product and supplier details, retrieve order details with product information, and fetch suppliers' location and products. Additionally, a query was crafted to retrieve the most recent order placed by each customer, showcasing the use of window functions alongside joins.

Task 6 – JOINS & Window Functions

- Using the same database you previously created to perform the following tasks:

Questions:

1. Write a query to retrieve all orders placed by customers, including customer details (name, phone), order details (order ID, timestamp), and item details (product, amount).
2. Write a query to fetch all products along with their suppliers' details (name, phone) and the corresponding category name.
3. Write a query to retrieve details of all orders including the product name and amount ordered for each item.
4. Write a query to retrieve all suppliers along with the city and country where they are located, and the products they supply.
5. Write a query to fetch details of the most recent order (by timestamp) placed by each customer, including the product details for each item in the order. [This question will use a Window Function alongside Joins]

Queries and Their Documentation:

Each query addresses a specific question regarding the data.

Question 1

Statement: Write a query to retrieve all orders placed by customers, including customer details (name, phone), order details (order ID, timestamp), and item details (product, amount).

Query:

```
SELECT
    c.name,
    c.phone,
    o.order_id,
    o.order_time,
    i.product_id,
    p.name AS product_name
FROM
    sales.customers AS c
RIGHT JOIN
    sales.orders AS o
ON
    c.customer_id = o.customer_id
LEFT JOIN
    sales.items AS i
ON
    o.order_id = i.order_id
LEFT JOIN
    sales.products AS p
ON
    i.product_id = p.product_id
```

```

SELECT
|   c.name,
|   c.phone,
|   o.order_id,
|   o.order_time,
|   i.product_id,
|   p.name AS product_name
FROM
|   sales.customers AS c
RIGHT JOIN
|   sales.orders AS o
ON
|   c.customer_id = o.customer_id
LEFT JOIN
|   sales.items AS i
ON
|   o.order_id = i.order_id
LEFT JOIN
|   sales.products AS p
ON
|   i.product_id = p.product_id

```

Explanation: This query retrieves all orders placed by customers, including customer details, order details, and item details using multiple JOIN statements.

Output:

	name character varying (100)	phone character varying (50)	order_id character varying (100)	order_time timestamp without time zone	product_id character varying (100)	product_name character varying (50)
1	Kimberly Rogers	+1-460-516-4009x4053	ef78adf6-ea9e-42f2-9f02-86782915e7c2	2024-01-28 17:57:53.634748	4e594d9f-d40a-48a5-8abb-0732f14ebe...	apply
2	Richard Wilcox	(724)299-4304x101	d48434e8-3f2e-456d-8ccd-86477a3e88eb	2024-04-22 05:32:40.292983	63196dc5-8ed6-4c45-a41d-edd7bbac79...	level
3	Loretta Hawkins	+1-493-923-3484x3483	6181e475-ac1f-4a12-acd1-67f5f3bca720	2024-03-28 18:33:28.498313	74fb7255-c081-4c15-95e5-9b37ae89a8...	station
4	Mr. Bryce Ortiz	706.967.6060x404	5c93f294-c443-440e-9c4a-4cf3d95ef459	2024-05-12 13:24:36.588333	99ff9991-7613-4fbf-a0a7-9423aa64edde	rich
5	Rebecca Miles	001-791-569-9566x5950	2b69e4ad-7629-420c-82e3-b10521d2e169	2024-01-11 00:11:21.409286	19d17104-4db0-48f5-a2eb-aad2879421...	head
6	Kayla Lee	501-803-2939	545eece9-d0c2-45f7-9a34-d862d9957791	2024-06-20 23:01:39.004452	44a8bf00-54c9-4239-8a7e-d406f4007c...	on
7	Kelly Austin	803.817.5409	bf04f9fb-7bb5-4d68-8730-579f7a2f8799	2024-02-21 05:58:08.131231	f222f904-0262-4036-b3dd-48297bdd8c...	into
8	Jennifer Stevens	307-800-7458x1537	c75508bd-3fa0-44f1-9db2-6ac2ad52bc1b	2024-04-16 21:37:48.714873	6c652e39-ac53-4ea6-9dda-94d0b9f545...	bit
9	Jared Evans	+1-636-519-6529	8f423864-a56c-46d0-8d91-7f8cd9f7c144	2024-01-08 03:23:11.383554	a7410066-9ba9-44b4-8ee7-7b9e17858...	his
10	Timothy Schaefer	452-270-7540	d4f3dada-01b2-4bba-c67f-edeeac31221b	2024-06-10 09:40:41.420451	b2a6a2a1-7206-4d8a-9d8a-9d03f5d470...	agreement

Question 2

Statement: Write a query to fetch all products along with their suppliers' details (name, phone) and the corresponding category name.

Query:

```
SELECT
    p.name AS product_name,
    p.category AS product_category,
    s.name AS supplier_name,
    s.phone AS supplier_phone,
    s.category AS supplier_category
FROM
    sales.products AS p
INNER JOIN
    sales.suppliers AS s
ON
    p.supplier_id = s.supplier_id
```

```
SELECT
    p.name AS product_name,
    p.category AS product_category,
    s.name AS supplier_name,
    s.phone AS supplier_phone,
    s.category AS supplier_category
FROM
    sales.products AS p
INNER JOIN
    sales.suppliers AS s
ON
    p.supplier_id = s.supplier_id
```

Explanation: This query fetches all products along with their suppliers' details and corresponding category name using an INNER JOIN.

Output:

	product_name character varying (50) 🔒	product_category character varying (50) 🔒	supplier_name character varying (100) 🔒	supplier_phone character varying (100) 🔒	supplier_category character varying (50) 🔒
1	talk	different	Turner, Newman and Rodriguez	001-650-863-5785x75794	accept
2	letter	write	Hall, Garcia and Wiggins	888-551-8790x931	decision
3	decade	decision	Acosta-Freeman	+1-949-497-5202	participant
4	major	believe	Reed, Gonzalez and Moore	+1-999-431-0825	common
5	various	speech	Alvarado-Cannon	6822144562	participant
6	security	trouble	Hill PLC	(534)558-4604x4870	offer
7	cost	rest	Smith, Burton and Smith	389-455-1435	difference
8	we	defense	Walker, Graham and Hernandez	001-978-752-3677	something
9	themselves	statement	Delacruz, Cummings and Silva	470-752-0877x6294	space

Question 3

Statement: Write a query to retrieve details of all orders including the product name and amount ordered for each item.

Query:

```
SELECT
    o.order_id,
    o.status,
    o.order_time,
    o.total_amount AS order_amount,
    p.name AS product_name,
    i.amount AS item_amount
FROM
    sales.orders AS o
LEFT JOIN
    sales.items AS i
ON
    o.order_id = i.order_id
LEFT JOIN
    sales.products AS p
ON
    i.product_id = p.product_id
ORDER BY
    o.order_id
```

```

SELECT
    o.order_id,
    o.status,
    o.order_time,
    o.total_amount AS order_amount,
    p.name AS product_name,
    i.amount AS item_amount
FROM
    sales.orders AS o
LEFT JOIN
    sales.items AS i
ON
    o.order_id = i.order_id
LEFT JOIN
    sales.products AS p
ON
    i.product_id = p.product_id
ORDER BY
    o.order_id

```

Explanation: This query retrieves details of all orders, including product name and amount ordered for each item using LEFT JOIN.

Output:

	order_id character varying (100) 🔒	status character varying (50) 🔒	order_time timestamp without time zone 🔒	order_amount double precision 🔒	product_name character varying (50) 🔒	item_amount double precision 🔒
1	002466f4-e2d2-41b6-908f-b12418db9f1d	Cancelled	2024-06-23 10:28:57.545177	123.93	[null]	[null]
2	0077622c-f789-402f-8f27-a8ef29b92f99	Completed	2024-01-04 07:44:52.702814	252.56	will	32.05
3	007c6893-fc40-44f8-a794-741ad1bfd9c7	Cancelled	2024-06-09 00:25:09.386567	834.56	difference	52.99
4	00860b6c-6b06-4694-8708-f7aa4375d5e4	Pending	2024-03-08 06:58:18.346432	25.97	build	73.93
5	00860b6c-6b06-4694-8708-f7aa4375d5e4	Pending	2024-03-08 06:58:18.346432	25.97	difficult	39.54
6	00860b6c-6b06-4694-8708-f7aa4375d5e4	Pending	2024-03-08 06:58:18.346432	25.97	evidence	99.77
7	00871b7a-6c23-4916-aeff-e22ee8e27ea9	Completed	2024-05-29 11:27:53.906636	315.29	citizen	65.86
8	00aa2373-5f66-4165-8ebf-7724b314028a	Pending	2024-02-06 17:35:35.482997	862.98	major	26.53
9	00aa2373-5f66-4165-8ebf-7724b314028a	Pending	2024-02-06 17:35:35.482997	862.98	address	10.07

Question 4

Statement: Write a query to retrieve all suppliers along with the city and country where they are located, and the products they supply.

Query:

```
SELECT
    s.name AS supplier_name,
    s.phone, s.location AS city,
    c.country,
    p.name AS product_name
FROM
    sales.suppliers AS s
LEFT JOIN
    sales.cities AS c
ON
    s.location = c.name
LEFT JOIN
    sales.products AS p
ON
    s.supplier_id = p.supplier_id
```

```
SELECT
|   s.name AS supplier_name,
|   s.phone, s.location AS city,
|   c.country,
|   p.name AS product_name
FROM
|   sales.suppliers AS s
LEFT JOIN
|   sales.cities AS c
ON
|   s.location = c.name
LEFT JOIN
|   sales.products AS p
ON
|   s.supplier_id = p.supplier_id
```


Explanation: This query retrieves all suppliers along with the city and country where they are located and the products they supply using LEFT JOIN.

Output:

	supplier_name character varying (100)	phone character varying (100)	city character varying (50)	country character varying (100)	product_name character varying (50)
1	Wilson, Freeman and Ware	+1-573-920-0632x8160	South Christine	Israel	[null]
2	Stein-Adams	724-831-9559x9164	New Frank	Belgium	outside
3	Stein-Adams	724-831-9559x9164	New Frank	Belgium	investment
4	Stein-Adams	724-831-9559x9164	New Frank	Belgium	should
5	Hall, Garcia and Wiggins	888-551-8790x931	Lake Michael	Slovenia	letter
6	Hall, Garcia and Wiggins	888-551-8790x931	Lake Michael	Slovenia	his
7	Hall, Garcia and Wiggins	888-551-8790x931	Lake Michael	Slovenia	rock
8	Mcconnell-Edwards	576-782-8489	Lake Michael	Slovenia	[null]
9	Conzales Head	(205)380-5207x846	Wrightton	Cuvana	[null]

Question 5

Statement: Write a query to fetch details of the most recent order (by timestamp) placed by each customer, including the product details for each item in the order. [This question will use a Window Function alongside Joins]

Query:

```
WITH RankedOrders AS (  
    SELECT  
        o.customer_id,  
        RANK() OVER (PARTITION BY o.customer_id ORDER BY o.order_time  
DESC) AS order_rank_for_customer,  
        o.order_id,  
        o.order_time,  
        c.name,  
        c.location,  
        p.name AS product_name,  
        p.category AS product_category,  
        p.price AS product_price
```

```
FROM
    sales.orders AS o
LEFT JOIN
    sales.customers AS c
ON
    o.customer_id = c.customer_id
LEFT JOIN
    sales.items AS i
ON
    o.order_id = i.order_id
LEFT JOIN
    sales.products AS p
ON
    i.product_id = p.product_id
)
SELECT
    customer_id,
    name AS customer_name,
    location AS customer_location,
    order_id,
    order_time,
    product_name,
    product_category,
    product_price
FROM
    RankedOrders
WHERE
    order_rank_for_customer = 1
ORDER BY
    customer_id DESC;
```

```

WITH RankedOrders AS (
  SELECT
    o.customer_id,
    RANK() OVER (PARTITION BY o.customer_id ORDER BY o.order_time DESC) AS order_rank_for_customer,
    o.order_id,
    o.order_time,
    c.name,
    c.location,
    p.name AS product_name,
    p.category AS product_category,
    p.price AS product_price
  FROM
    sales.orders AS o
  LEFT JOIN
    sales.customers AS c
  ON
    o.customer_id = c.customer_id
  LEFT JOIN
    sales.items AS i
  ON
    o.order_id = i.order_id
  LEFT JOIN
    sales.products AS p
  ON
    i.product_id = p.product_id
)

```

```

SELECT
  customer_id,
  name AS customer_name,
  location AS customer_location,
  order_id,
  order_time,
  product_name,
  product_category,
  product_price
FROM
  RankedOrders
WHERE
  order_rank_for_customer = 1
ORDER BY
  customer_id DESC;

```

Explanation: This query fetches details of the most recent order placed by each customer, including product details, using a common table expression (CTE), JOINS and window functions.

Output:

	customer_id character varying (100) 🔒	customer_name character varying (100) 🔒	customer_location character varying (50) 🔒	order_id character varying (100) 🔒	order_time timestamp without time zone 🔒	product_name character varying	product_category character varying (product_price double precision 🔒
1	ffb7c3c7-705e-43b6-8a9...	Jesse Greer	East Timothyfort	6f32e006-cebd-43fa-bad...	2024-05-29 00:36:38.51086	customer	eight	40.59
2	ffb7c3c7-705e-43b6-8a9...	Jesse Greer	East Timothyfort	6f32e006-cebd-43fa-bad...	2024-05-29 00:36:38.51086	police	hour	411.02
3	ff878edc-47e7-4b39-a24...	Thomas Roberts	Coxside	c98cbffa-4918-4175-88...	2024-01-09 07:01:32.409216	suffer	property	316.08
4	ff845231-1e77-47eb-994...	Andrea Davis	Lake Stephanieside	9eda8c53-d1d8-4593-b9...	2024-05-30 07:55:43.06364	million	lose	310.71
5	ff71de97-5e4f-4749-bc7...	Shelly Haley	North Pamela	92e966a1-46e1-45ea-b0...	2024-04-17 13:40:05.036449	[null]	[null]	[null]
6	ff6dceba-1795-4271-ac3...	Stephanie Ferguson	West Michael	a293a612-e954-4496-a1...	2024-06-20 16:43:40.069428	program	talk	228.27
7	ff6dceba-1795-4271-ac3...	Stephanie Ferguson	West Michael	a293a612-e954-4496-a1...	2024-06-20 16:43:40.069428	big	race	173.49
8	ff52b73e-74b4-48f6-941...	John Rodriguez	New Natashamouth	d1ba21ce-33eb-49ca-80...	2024-04-13 13:23:09.472914	[null]	[null]	[null]
9	feea5f6d-b82f-4094-80e...	Tiffany Bishop	Benjaminfurt	d02cdb84-46d3-45d5-ac...	2024-05-10 16:35:24.849445	[null]	[null]	[null]
10	fee8f537-dcce-49c4-ba7...	Sydney Little	Petersfort	0468ffb8-8c01-42d5-81f...	2024-02-22 00:41:20.029938	smile	total	112.02
11	fea10b26-f4f3-4d51-8aa...	Courtney Cooley	Lake Jenniferstad	9d5bbff6-ad52-4f15-82c...	2024-03-04 07:12:35.396137	able	blood	76.61
12	fea10b26-f4f3-4d51-8aa...	Courtney Cooley	Lake Jenniferstad	9d5bbff6-ad52-4f15-82c...	2024-03-04 07:12:35.396137	chair	task	150.75

Conclusion:

This task focused on utilizing SQL JOINS and Window Functions to retrieve detailed and comprehensive data insights from the sales database. By completing this task, I further honed my skills in SQL query formulation and data manipulation, which are essential for advanced data engineering and analytics.