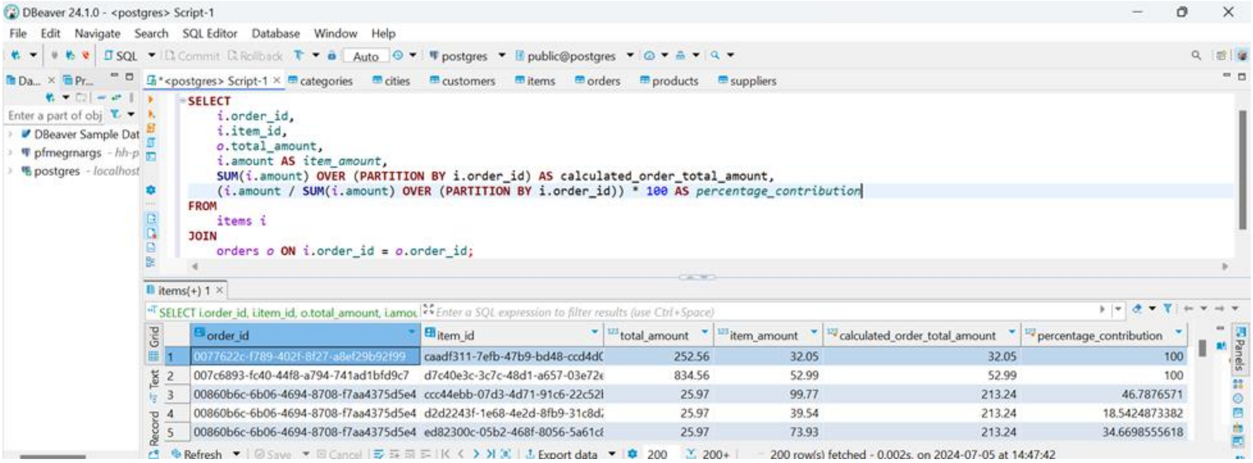
BWT Task-04 Exercise

Submitted By: ABRAR SAEED

Q1

Write a query to calculate the percentage contribution of each item's amount to its order's total amount, grouped by order_id. (Topics: Partition BY)



```
SELECT
    i.order_id,
    i.item_id,
    o.total_amount,
    i.amount AS item_amount,
    SUM(i.amount) OVER (PARTITION BY i.order_id) AS calculated_order_total_amount,
    (i.amount / SUM(i.amount) OVER (PARTITION BY i.order_id)) * 100 AS percentage_contribution
FROM
    items i
JOIN
    orders o ON i.order_id = o.order_id;
```

SELECT i.order_id, i.item_id, o.total_amount, i.amou

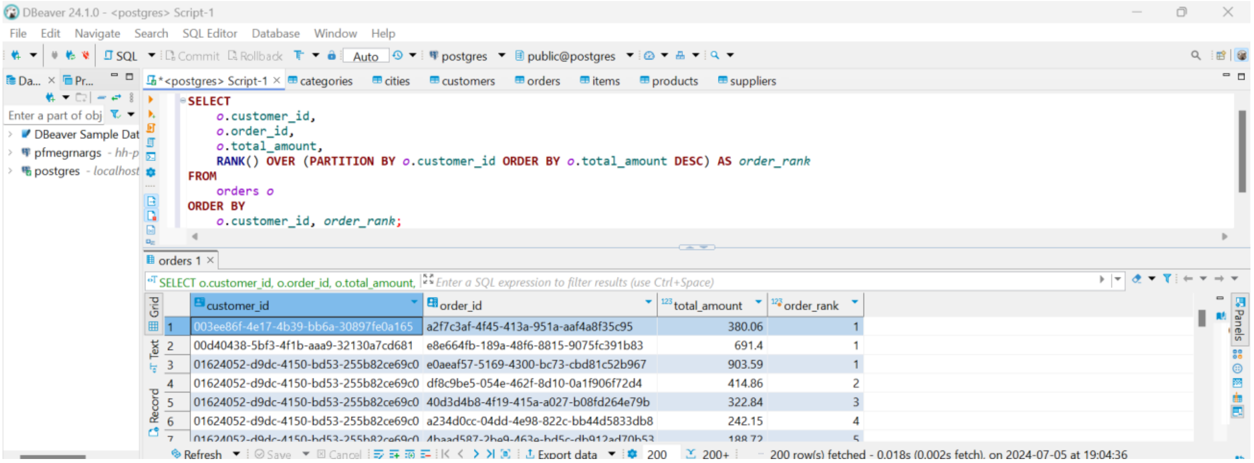
	order_id	item_id	total_amount	item_amount	calculated_order_total_amount	percentage_contribution
1	0077622c-f789-402f-8f27-a8ef29b92f99	caadf311-7efb-47b9-bd48-ccd4d(	252.56	32.05	32.05	100
2	007c6893-fc40-44f8-a794-741ad1bfd9c7	d7c40e3c-3c7c-48d1-a657-03e72e	834.56	52.99	52.99	100
3	00860b6c-6b06-4694-8708-f7aa4375d5e4	ccc44ebb-07d3-4d71-91c6-22c52l	25.97	99.77	213.24	46.7876571
4	00860b6c-6b06-4694-8708-f7aa4375d5e4	d2d2243f-1e68-4e2d-8fb9-31c8d.	25.97	39.54	213.24	18.5424873382
5	00860b6c-6b06-4694-8708-f7aa4375d5e4	ed82300c-05b2-468f-8056-5a61c	25.97	73.93	213.24	34.6698555618

200 row(s) fetched - 0.002s, on 2024-07-05 at 14:47:42

```
SELECT
    i.order_id,
    i.item_id,
    o.total_amount,
    i.amount AS item_amount,
    SUM(i.amount) OVER (PARTITION BY i.order_id) AS calculated_order_total_amount,
    (i.amount / SUM(i.amount) OVER (PARTITION BY i.order_id)) * 100 AS
percentage_contribution
FROM
    items i
JOIN
    orders o ON i.order_id = o.order_id;
```

Q2

Write a query to rank orders by their total amount within each customer, ordering them from highest to lowest total amount. (Topics: Window functions like RANK, PARTITION BY, and ORDER BY)



```
SELECT
    o.customer_id,
    o.order_id,
    o.total_amount,
    RANK() OVER (PARTITION BY o.customer_id ORDER BY o.total_amount DESC) AS order_rank
FROM
    orders o
ORDER BY
    o.customer_id, order_rank;
```

orders 1

SELECT o.customer_id, o.order_id, o.total_amount,

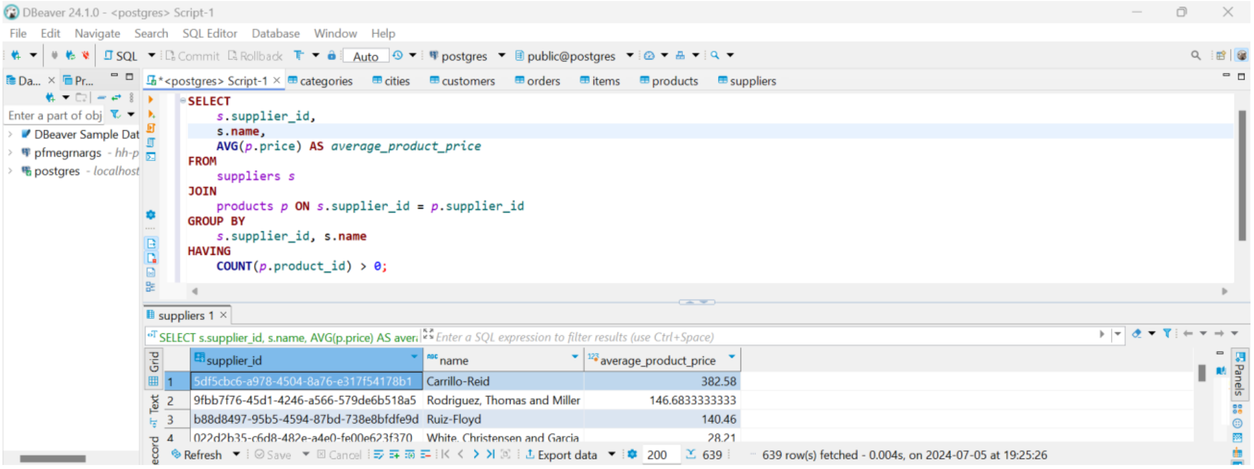
	customer_id	order_id	total_amount	order_rank
1	003ee86f-4e17-4b39-bb6a-30897fe0a165	a2f7c3af-4f45-413a-951a-aaf4a8f35c95	380.06	1
2	00d40438-5bf3-4f1b-aaa9-32130a7cd681	e8e664fb-189a-48f6-8815-9075fc391b83	691.4	1
3	01624052-d9dc-4150-bd53-255b82ce69c0	e0aeaf57-5169-4300-bc73-cbd81c52b967	903.59	1
4	01624052-d9dc-4150-bd53-255b82ce69c0	df8c9be5-054e-462f-8d10-0a1f906f72d4	414.86	2
5	01624052-d9dc-4150-bd53-255b82ce69c0	40d3d4b8-4f19-415a-a027-b08fd264e79b	322.84	3
6	01624052-d9dc-4150-bd53-255b82ce69c0	a234d0cc-04dd-4e98-822c-bb44d5833db8	242.15	4
7	01624052-d9dc-4150-bd53-255b82ce69c0	4baad587-2be9-463e-bd5c-db912ad70b53	188.72	5

200 row(s) fetched - 0.018s (0.002s fetch), on 2024-07-05 at 19:04:36

```
SELECT
    o.customer_id,
    o.order_id,
    o.total_amount,
    RANK() OVER (PARTITION BY o.customer_id ORDER BY o.total_amount DESC) AS
order_rank
FROM
    orders o
ORDER BY
    o.customer_id, order_rank;
```

Q3

Write a query to calculate the average price of products supplied by each supplier. Exclude suppliers who have no products in the result. (Topics: JOINS, AGGREGATE FUNCTIONS, GROUP BY)



```
SELECT
    s.supplier_id,
    s.name,
    AVG(p.price) AS average_product_price
FROM
    suppliers s
JOIN
    products p ON s.supplier_id = p.supplier_id
GROUP BY
    s.supplier_id, s.name
HAVING
    COUNT(p.product_id) > 0;
```

suppliers 1

SELECT s.supplier_id, s.name, AVG(p.price) AS aver

	supplier_id	name	average_product_price
1	5df5cbc6-a978-4504-8a76-e317f54178b1	Carrillo-Reid	382.58
2	9fbb7f76-45d1-4246-a566-579de6b518a5	Rodriguez, Thomas and Miller	146.6833333333
3	b88d8497-95b5-4594-87bd-738e8bfdfe9d	Ruiz-Floyd	140.46
4	022d2b35-c6d8-482e-a4e0-fe00e623f370	White, Christensen and Garcia	28.21

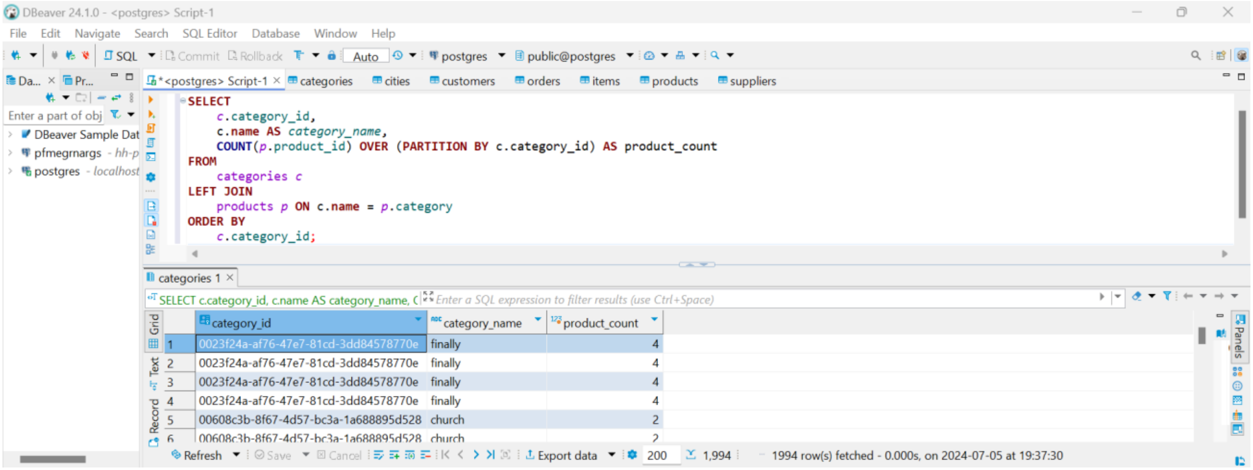
639 row(s) fetched - 0.004s, on 2024-07-05 at 19:25:26

```
SELECT
    s.supplier_id,
    s.name,
    AVG(p.price) AS average_product_price
FROM
```

```
    suppliers s
JOIN
    products p ON s.supplier_id = p.supplier_id
GROUP BY
    s.supplier_id, s.name
HAVING
    COUNT(p.product_id) > 0;
```

Q4

Write a query to count the number of products in each category. Include categories with zero products in the result set. (WINDOW FUNCTIONS, AGGREGATE FUNCTIONS, JOINS, GROUP BY)



```
SELECT
    c.category_id,
    c.name AS category_name,
    COUNT(p.product_id) OVER (PARTITION BY c.category_id) AS product_count
FROM
    categories c
LEFT JOIN
    products p ON c.name = p.category
ORDER BY
    c.category_id;
```

SELECT c.category_id, c.name AS category_name, C | Enter a SQL expression to filter results (use Ctrl+Space)

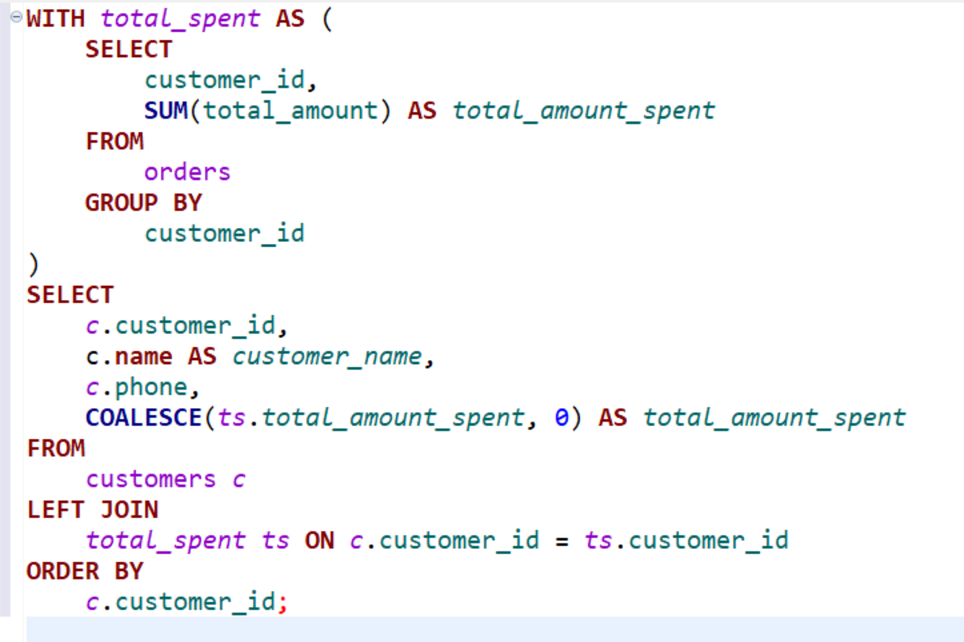
	category_id	category_name	product_count
1	0023f24a-af76-47e7-81cd-3dd84578770e	finally	4
2	0023f24a-af76-47e7-81cd-3dd84578770e	finally	4
3	0023f24a-af76-47e7-81cd-3dd84578770e	finally	4
4	0023f24a-af76-47e7-81cd-3dd84578770e	finally	4
5	00608c3b-8f67-4d57-bc3a-1a688895d528	church	2
6	00608c3b-8f67-4d57-bc3a-1a688895d528	church	2

200 1,994 1994 row(s) fetched - 0.000s, on 2024-07-05 at 19:37:30

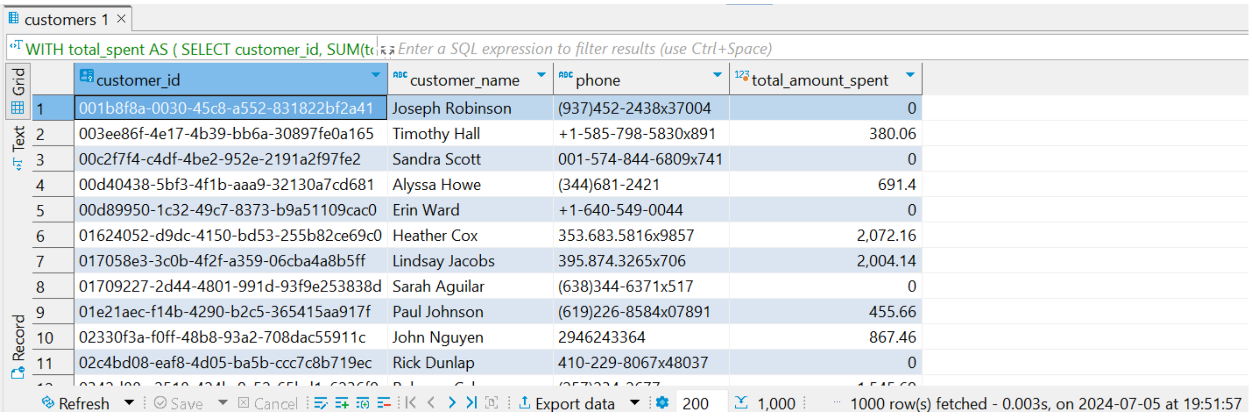
```
SELECT
    c.category_id,
    c.name AS category_name,
    COUNT(p.product_id) OVER (PARTITION BY c.category_id) AS product_count
FROM
    categories c
LEFT JOIN
    products p ON c.name = p.category
ORDER BY
    c.category_id;
```

Q5

Write a query to retrieve the total amount spent by each customer, along with their name and phone number. Ensure customers with no orders also appear with a total amount of 0. (WINDOW FUNCTIONS, AGGREGATE FUNCTIONS, JOINS, GROUP BY)



```sql
WITH total_spent AS (
    SELECT
        customer_id,
        SUM(total_amount) AS total_amount_spent
    FROM
        orders
    GROUP BY
        customer_id
)
SELECT
    c.customer_id,
    c.name AS customer_name,
    c.phone,
    COALESCE(ts.total_amount_spent, 0) AS total_amount_spent
FROM
    customers c
LEFT JOIN
    total_spent ts ON c.customer_id = ts.customer_id
ORDER BY
    c.customer_id;
```



WITH total_spent AS (SELECT customer_id, SUM(t
Enter a SQL expression to filter results (use Ctrl+Space)

	customer_id	customer_name	phone	total_amount_spent
1	001b8f8a-0030-45c8-a552-831822bf2a41	Joseph Robinson	(937)452-2438x37004	0
2	003ee86f-4e17-4b39-bb6a-30897fe0a165	Timothy Hall	+1-585-798-5830x891	380.06
3	00c2f7f4-c4df-4be2-952e-2191a2f97fe2	Sandra Scott	001-574-844-6809x741	0
4	00d40438-5bf3-4f1b-aaa9-32130a7cd681	Alyssa Howe	(344)681-2421	691.4
5	00d89950-1c32-49c7-8373-b9a51109cac0	Erin Ward	+1-640-549-0044	0
6	01624052-d9dc-4150-bd53-255b82ce69c0	Heather Cox	353.683.5816x9857	2,072.16
7	017058e3-3c0b-4f2f-a359-06cba4a8b5ff	Lindsay Jacobs	395.874.3265x706	2,004.14
8	01709227-2d44-4801-991d-93f9e253838d	Sarah Aguilar	(638)344-6371x517	0
9	01e21aec-f14b-4290-b2c5-365415aa917f	Paul Johnson	(619)226-8584x07891	455.66
10	02330f3a-f0ff-48b8-93a2-708dac55911c	John Nguyen	2946243364	867.46
11	02c4bd08-eaf8-4d05-ba5b-ccc7c8b719ec	Rick Dunlap	410-229-8067x48037	0

1000 row(s) fetched - 0.003s, on 2024-07-05 at 19:51:57

```sql
WITH total_spent AS (
    SELECT
        customer_id,
        SUM(total_amount) AS total_amount_spent
    FROM
        orders
    GROUP BY
        customer_id
)
SELECT
    c.customer_id,
    c.name AS customer_name,
    c.phone,
    COALESCE(ts.total_amount_spent, 0) AS total_amount_spent
FROM
    customers c
LEFT JOIN
    total_spent ts ON c.customer_id = ts.customer_id
ORDER BY
    c.customer_id;
```