Pizza Sales SQL Queries

A. KPIs

1. **Total Revenue**

   SELECT SUM(total_price) as total_revenue FROM pizza_sales;

	total_revenue numeric
1	817860.05

2. **Average Order Value**

```
WITH order_values AS (SELECT order_id, SUM(total_price) AS order_value
                      FROM pizza_sales
                      GROUP BY order_id)
SELECT AVG(order_value) as avg_order_value FROM order_values;

-- ALTERNATIVELY:

SELECT (SUM(total_price) / COUNT(DISTINCT order_id)) AS avg_order_value
FROM pizza_sales;
```

	avg_order_value numeric
1	38.3072622950819672

3. **Total Pizzas Sold**

   SELECT SUM(quantity) AS total_pizzas_sold FROM pizza_sales;

	total_pizzas_sold bigint
1	49574

4. **Total Orders Placed**

```
SELECT COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales;
```

	total_orders bigint
1	21350

5. **Average Pizzas Per Order**

```
WITH order_quants AS (SELECT order_id, SUM(quantity) AS order_quantity
                      FROM pizza_sales
                      GROUP BY order_id)
SELECT AVG(order_quantity) as avg_pizzas_per_order
FROM order_quants;

--ALTERNATIVELY:

SELECT (SUM(quantity) * 1.0 / COUNT(DISTINCT order_id)) AS
avg_pizzas_per_order
FROM pizza_sales;
```

	avg_pizzas_per_order numeric
1	2.3219672131147541

B. Trends / Insights

1. **Daily Orders Trend**

```
SELECT TO_CHAR(order_date, 'Day') AS day_name,
        COUNT(DISTINCT order_id) as total_orders
FROM pizza_sales
GROUP BY day_name
ORDER BY MIN(order_date)
```

	day_name text	total_orders bigint
1	Thursday	3239
2	Friday	3538
3	Saturday	3158
4	Sunday	2624
5	Monday	2794
6	Tuesday	2973
7	Wednesday	3024

2. **Hourly Orders Trend**

```
SELECT EXTRACT (HOUR FROM order_time) as hour_of_day,
        COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales
GROUP BY hour_of_day
ORDER BY hour_of_day;
```

	hour_of_day numeric	total_orders bigint
1	9	1
2	10	8
3	11	1231
4	12	2520
5	13	2455
6	14	1472
7	15	1468
8	16	1920
9	17	2336
10	18	2399
11	19	2009
12	20	1642
13	21	1198
14	22	663
15	23	28

3. **Monthly Orders Trend**

```
SELECT TO_CHAR(order_date, 'Month') AS month_name,
        COUNT(DISTINCT order_id) AS total_orders
FROM pizza_sales
GROUP BY month_name
ORDER BY MIN(order_date);
```

	month_name text	total_orders bigint
1	January	1845
2	February	1685
3	March	1840
4	April	1799
5	May	1853
6	June	1773
7	July	1935
8	August	1841
9	September	1661
10	October	1646
11	November	1792
12	December	1680

4. **Percentage of Sales by Category**

```
SELECT pizza_category,
        SUM(total_price) AS total_sales,
        ROUND((SUM(total_price) / (SELECT SUM(total_price) FROM
pizza_sales)) * 100, 2) AS percent_contribution
FROM pizza_sales
GROUP BY pizza_category;
```

	pizza_category character varying (30)	total_sales numeric	percent_contribution numeric
1	Supreme	208197.00	25.46
2	Chicken	195919.50	23.96
3	Veggie	193690.45	23.68
4	Classic	220053.10	26.91

Note: If you wish to filter for a specific month (January for example), do the following:

```
SELECT pizza_category,
        SUM(total_price) AS total_sales,
        ROUND((SUM(total_price) / (SELECT SUM(total_price) FROM
pizza_sales WHERE EXTRACT(MONTH FROM order_date) = 1)) * 100, 2) AS
percent_contribution
FROM pizza_sales
WHERE EXTRACT(MONTH FROM order_date) = 1
GROUP BY pizza_category;
```

Note: Important to also include WHERE condition within the subquery, or else percentages won't be correct

	pizza_category character varying (30)	total_sales numeric	percent_contribution numeric
1	Chicken	16188.75	23.20
2	Classic	18619.40	26.68
3	Supreme	17929.75	25.69
4	Veggie	17055.40	24.44

5. **Percentage of Pizza Sales by Pizza Size**

```
SELECT pizza_size, SUM(total_price) AS total_sales,
        ROUND(SUM(total_price) / (SELECT SUM(total_price) FROM
pizza_sales) * 100, 2) AS percent_contribution
FROM pizza_sales
GROUP BY pizza_size
ORDER BY pizza_size;
```

	pizza_size character varying (5)	total_sales numeric	percent_contribution numeric
1	L	375318.70	45.89
2	M	249382.25	30.49
3	S	178076.50	21.77
4	XL	14076.00	1.72
5	XXL	1006.60	0.12

6. **Total Pizzas Sold by Category**

```
SELECT pizza_category,
        SUM(quantity) AS pizzas_sold
FROM pizza_sales
GROUP BY pizza_category
ORDER BY pizzas_sold DESC;
```

	pizza_category character varying (30)	pizzas_sold bigint
1	Classic	14888
2	Supreme	11987
3	Veggie	11649
4	Chicken	11050

7. **Top 5 Best Sellers by Total Pizzas Sold**

```
WITH quant_sold_rankings AS (SELECT pizza_name, SUM(quantity) AS total_sold,
                                    RANK() OVER(ORDER BY SUM(quantity) DESC) AS ranking
                             FROM pizza_sales
                             GROUP BY pizza_name)

SELECT * FROM quant_sold_rankings
WHERE ranking <= 5;
```

	pizza_name character varying	total_sold bigint	ranking bigint
1	The Classic Deluxe Pizza	2453	1
2	The Barbecue Chicken Pizza	2432	2
3	The Hawaiian Pizza	2422	3
4	The Pepperoni Pizza	2418	4
5	The Thai Chicken Pizza	2371	5

8. **Bottom 5 Worst Sellers by Total Pizzas Sold**

```
WITH quant_sold_rankings AS (SELECT pizza_name, SUM(quantity) AS total_sold,
                                    RANK() OVER(ORDER BY SUM(quantity)) AS ranking
                             FROM pizza_sales
                             GROUP BY pizza_name)

SELECT * FROM quant_sold_rankings
WHERE ranking <= 5;
```

	pizza_name character varying	total_sold bigint	ranking bigint
1	The Brie Carre Pizza	490	1
2	The Mediterranean Pizza	934	2
3	The Calabrese Pizza	937	3
4	The Spinach Supreme Pizza	950	4
5	The Soppressata Pizza	961	5

9. **Top 5 Best Sellers by Revenue**

```
WITH revenue_rank AS (SELECT pizza_name,
                             SUM(total_price) AS total_revenue,
                             RANK() OVER(ORDER BY
SUM(total_price) DESC) AS revenue_ranking
                      FROM pizza_sales
                      GROUP BY pizza_name)
SELECT *
FROM revenue_rank
WHERE revenue_ranking <= 5;
```

	pizza_name character varying	total_revenue numeric	revenue_ranking bigint
1	The Thai Chicken Pizza	43434.25	1
2	The Barbecue Chicken Pizza	42768.00	2
3	The California Chicken Pizza	41409.50	3
4	The Classic Deluxe Pizza	38180.50	4
5	The Spicy Italian Pizza	34831.25	5

10. **Bottom 5 Sellers by Revenue**

```
WITH revenue_rank AS (SELECT pizza_name,
                             SUM(total_price) AS total_revenue,
                             RANK() OVER(ORDER BY
SUM(total_price)) AS revenue_ranking
                      FROM pizza_sales
                      GROUP BY pizza_name)
SELECT *
FROM revenue_rank
WHERE revenue_ranking <= 5;
```

	pizza_name character varying	total_revenue numeric	revenue_ranking bigint
1	The Brie Carre Pizza	11588.50	1
2	The Green Garden Pizza	13955.75	2
3	The Spinach Supreme Pizza	15277.75	3
4	The Mediterranean Pizza	15360.50	4
5	The Spinach Pesto Pizza	15596.00	5