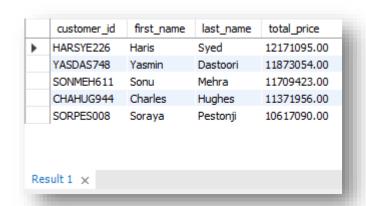# Customer Data Analysis

1. Top 5 Customers by Total Spending

MySQL Query

```sql
1    -- Q1 : Top 5 customers by Total Spending
2
3 •  SELECT
4        c.customer_id,
5        c.first_name,
6        c.last_name,
7        SUM(oi.total_price) AS total_price
8    FROM
9        customer_data_analysis.customers c
10           JOIN
11       customer_data_analysis.orders o ON o.customer_id = c.customer_id
12           JOIN
13       customer_data_analysis.order_items oi ON o.order_id = oi.order_id
14   GROUP BY c.customer_id , c.first_name , c.last_name
15   ORDER BY total_price DESC
16   LIMIT 5;
17
18
```

Its Result

	customer_id	first_name	last_name	total_price
▶	HARSYE226	Haris	Syed	12171095.00
	YASDAS748	Yasmin	Dastoori	11873054.00
	SONMEH611	Sonu	Mehra	11709423.00
	CHAHUG944	Charles	Hughes	11371956.00
	SORPES008	Soraya	Pestonji	10617090.00

Result 1 ✕

2. Repeat Customers in the Last 6 Months

MySQL Query

```sql
1    -- Q2 : Repeat Customers in the Last 6 Months
2
3    SELECT
4        c.customer_id,
5        c.first_name,
6        c.last_name,
7        COUNT(o.customer_id) AS order_placed
8    FROM
9        customer_data_analysis.customers c
10           JOIN
11       orders o ON c.customer_id = o.customer_id
12   WHERE
13       o.order_date >= CURDATE() - INTERVAL 6 MONTH
14   GROUP BY c.customer_id , c.first_name , c.last_name
15   HAVING order_placed > 1
16   ORDER BY order_placed DESC;
17
18   |
```

Its Result

customer_id	first_name	last_name	order_placed
LUKHAY434	Luke	Hayes	12
SUNKAU147	Sunil	Kaur	12
NIRSIN632	Nirvair	Singh	11
SURKAU492	Surajpal	Kaur	11
MOIAHM287	Moiz	Ahmed	11
AHSMAL563	Ahsan	Malik	11
MARHAR207	Mark	Harris	11
JAVSHA037	Javed	Shaikh	11

Result 1 ×

3. Product Performance: Total Sales for Each Product

MySQL Query

Its Result

```sql
1    -- Q3 Product Performance: Total Sales for Each Product
2
3 ●  SELECT
4        p.product_name,
5        p.category,
6        SUM(oi.quantity * p.price) AS total_sale,
7        SUM(oi.quantity) AS total_quantity
8    FROM
9        customer_data_analysis.products p
10           JOIN
11       customer_data_analysis.order_items oi ON p.product_id = oi.product_id
12   GROUP BY p.product_name , p.category
13   ORDER BY total_sale DESC;
14
15
```

product_name	category	total_sale	total_quantity
FreshBake Oven	Miscellaneous	17683821.00	1179
UltraView Monitor	Electronics	15208830.00	1170
AquaEase Purifier	Home & Living	10708810.00	1190
PaperEase Printer	Stationery & Office	9598800.00	1200
MiniExplore Telescope	Kids & Toys	8068655.00	1345
WonderBot Robot	Kids & Toys	5376305.00	1195
SprintWave Shoes	Fitness & Sports	5162709.00	1291
ChillBreeze Fan	Miscellaneous	4862784.00	1216

Result 1 ✕

4. Average Order Value per Customer

MySQL Query
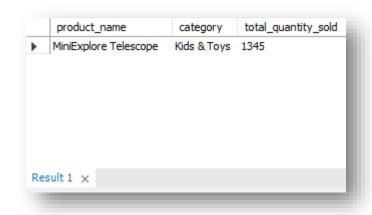
```sql
1    -- Q4 Average Order Value per Customer
2
3  • SELECT
4        c.customer_id,
5        c.first_name,
6        c.last_name,
7        ROUND(AVG(oi.total_price), 2) AS average_order_value
8    FROM
9        customer_data_analysis.customers c
10           JOIN
11       orders o ON c.customer_id = o.customer_id
12           JOIN
13       order_items oi ON o.order_id = oi.order_id
14   GROUP BY c.customer_id , c.first_name , c.last_name
15   ORDER BY average_order_value DESC;
16
17
```

Its Result

customer_id	first_name	last_name	average_order_value
RASMAL011	Rashid	Malik	15120.75
MANKAU896	Maninderpal	Kaur	13726.90
AMAMAL002	Amaan	Malik	11503.00
JACHOW146	Jacob	Howell	9336.56
KAMJAM456	Kamran	Jamshid	8477.15
MEHKAU155	Mehar	Kaur	7978.40
BHAKAP217	Bhavana	Kapoor	7972.20
MALSIN587	Malkit	Singh	7891.11

Result 1 ✕

5. Most Popular Product in Terms of Quantity Sold

MySQL Query

Its Result
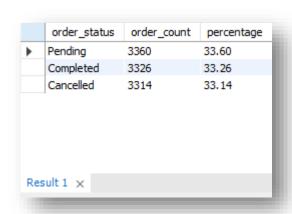
```
1    -- Q5 Most Popular Prouduct in Terms of Qunaity Sold
2
3 •  SELECT
4        p.product_name,
5        p.category,
6        SUM(oi.quantity) AS total_quantity_sold
7    FROM
8        customer_data_analysis.products p
9            JOIN
10       customer_data_analysis.order_items oi ON p.product_id = oi.product_id
11   GROUP BY p.product_name , p.category
12   ORDER BY total_quantity_sold DESC
13   LIMIT 1;
14
15
```

product_name	category	total_quantity_sold
MiniExplore Telescope	Kids & Toys	1345

Result 1 ×

6. Orders by Status: Percentage of Completed , Pending and Pending Orders

MySQL Query

Its Result

```
1      -- Q6 Orders by Status: Percentage of Completed, Pending and Cancelled Orders
2
3  •   SELECT
4          o.order_status,
5          COUNT(o.order_id) AS order_count,
6  ⊖      ROUND((COUNT(o.order_id) / (SELECT
7                      COUNT(*)
8                  FROM
9                      customer_data_analysis.orders) * 100),
10             2) AS percentage
11     FROM
12         customer_data_analysis.orders o
13     GROUP BY o.order_status;
14
15
```

order_status	order_count	percentage
Pending	3360	33.60
Completed	3326	33.26
Cancelled	3314	33.14

Result 1 ✕

7. Total Spending per Customer by State

MySQL Query
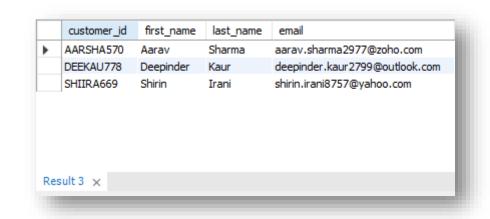
```
1    -- Q7 Total Spending per Customer by State
2
3 •  SELECT
4        c.state,
5        c.customer_id,
6        c.first_name,
7        c.last_name,
8        SUM(oi.total_price) AS total_spending
9    FROM
10       customer_data_analysis.customers c
11           JOIN
12       customer_data_analysis.orders o ON c.customer_id = o.customer_id
13           JOIN
14       customer_data_analysis.order_items oi ON o.order_id = oi.order_id
15   GROUP BY c.state , c.customer_id , c.first_name , c.last_name
16   ORDER BY total_spending DESC;
17
18
```

Its Result

state	customer_id	first_name	last_name	total_spending
▶ Mayur Vihar Phase 1	HARSYE226	Haris	Syed	12171095.00
Kolkata	YASDAS748	Yasmin	Dastoori	11873054.00
Visakhapatnam	SONMEH611	Sonu	Mehra	11709423.00
Madurai	CHAHUG944	Charles	Hughes	11371956.00
Aurangabad	SORPES008	Soraya	Pestonji	10617090.00
Udaipur	DILKAU676	Dilpreet	Kaur	10118921.00
Munger	MALSIN587	Malkit	Singh	9690288.00
Hoshiarpur	FFRWAD911	Feroze	Wadia	9256787.00

Result 1 ×

8. Find Customers Who Have Never Ordered

MySQL Query

Its Result

```
1    --  Q8 Find Customers Who Have Never Ordered
2
3  ●  SELECT
4        c.customer_id,
5        c.first_name,
6        c.last_name, c.email
7    FROM
8        customer_data_analysis.customers c
9            LEFT JOIN
10       customer_data_analysis.orders o ON c.customer_id = o.customer_id
11   WHERE
12       o.order_id IS NULL;
13
14
```

customer_id	first_name	last_name	email
AARSHA570	Aarav	Sharma	aarav.sharma2977@zoho.com
DEEKAU778	Deepinder	Kaur	deepinder.kaur2799@outlook.com
SHIIRA669	Shirin	Irani	shirin.irani8757@yahoo.com

Result 3 ✕

9. Products Not Sold in the Last 6 Months

MySQL Query

```sql
1    --  Q9 Products Not Sold in the last 6 Months
2
3 •  SELECT
4        p.product_id,
5        p.product_name,
6        p.category
7    FROM
8        customer_data_analysis.products p
9            LEFT JOIN
10       customer_data_analysis.order_items oi ON p.product_id = oi.product_id
11           LEFT JOIN
12       customer_data_analysis.orders o ON oi.order_id = o.order_id
13   WHERE
14       o.order_date < CURDATE() - INTERVAL 6 MONTH
15           OR o.order_id IS NULL;
16
17
```

Its Result

product_id	product_name	category
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office
AJ-VPF-1428	QuickNote Notebook	Stationery & Office

Result 26 ✕

10. Customer Retention: Customers Who Placed Orders in Both the Last and Previous Year

MySQL Query

Its Result

```sql
1    -- Q10. Customer Retention: Customers Who Placed Orders in Both the Last and Previous Year
2
3    SELECT
4        c.customer_id,
5        c.first_name,
6        c.last_name,
7        COUNT(DISTINCT YEAR(o.order_date)) AS year_with_orders
8    FROM
9        customer_data_analysis.customers c
10           JOIN
11       customer_data_analysis.orders o ON c.customer_id = o.customer_id
12   WHERE
13       YEAR(o.order_date) = YEAR(CURDATE())
14           OR YEAR(o.order_date) = YEAR(CURDATE()) - 1
15   GROUP BY c.customer_id , c.first_name , c.last_name
16   HAVING year_with_orders = 2;
17
18
```

customer_id	first_name	last_name	year_with_orders
AARGUP085	Aarav	Gupta	2
ABIMIT600	Abigail	Mitchell	2
AIDWEB891	Aiden	Webb	2
AMASIN360	Amarjeet	Singh	2
ANDROB208	Andrew	Roberts	2
ARMKHO253	Arman	Khorshed	2
BILAHM560	Bilal	Ahmed	2
BILMAL238	Bilal	Malik	2

Result 3 ✕

11. Revenue Breakdown by Product Category

MySQL Query

Its Result

```
1    -- Q11 Revenue Breakdown by Product Category
2
3  • SELECT
4        p.category,
5        SUM(oi.quantity * p.price) AS total_revenue
6    FROM
7        customer_data_analysis.products p
8            JOIN
9        customer_data_analysis.order_items oi ON p.product_id = oi.product_id
10   GROUP BY p.category
11   ORDER BY total_revenue DESC;
12
13
```

category	total_revenue
Miscellaneous	36423056.00
Electronics	22259062.00
Home & Living	19562650.00
Kids & Toys	16277789.00
Fitness & Sports	14921999.00
Fashion	13504169.00
Stationery & Office	11412529.00
Beauty & Personal Care	5064249.00

Result 3 ✕

12. Customer Order History: How Many Orders Did Each Customer Make?

MySQL Query

Its Result

```
1    -- Q12 Customer Order History: How Many Orders Did Each Customer Make?
2
3
4  ● SELECT
5        c.customer_id,
6        c.first_name,
7        c.last_name,
8        COUNT(o.order_id) AS order_placed
9    FROM
10       customer_data_analysis.customers c
11          JOIN
12       customer_data_analysis.orders o ON c.customer_id = o.customer_id
13   GROUP BY c.customer_id , c.first_name , c.last_name
14   ORDER BY order_placed DESC;
15
16
```

customer_id	first_name	last_name	order_placed
JACHOW146	Jacob	Howell	22
JAMMIS276	Jamshed	Mistry	20
NILGHA690	Nilofer	Ghadiali	20
BHAKAP217	Bhavana	Kapoor	19
CHASIN554	Charanpreet	Singh	19
CHRRIC555	Christopher	Richardson	18
HASKHA627	Haseeb	Khan	18
JACCOO979	Jacob	Cooper	18

Result 6 ✕

13. Total Sales by Month for Each Product

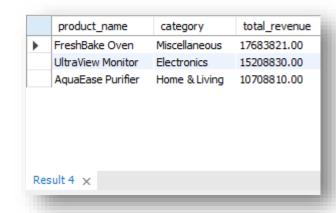**MySQL Query**

```
1    -- Q13 Total Sales by Month for Each Product
2
3
4  • SELECT
5        p.product_name,
6        p.category,
7        YEAR(o.order_date) AS order_year,
8        MONTH(o.order_date) AS order_month,
9        SUM(oi.quantity * p.price) AS total_sales
10  FROM
11       customer_data_analysis.products p
12           JOIN
13       customer_data_analysis.order_items oi ON p.product_id = oi.product_id
14           JOIN
15       customer_data_analysis.orders o ON oi.order_id = o.order_id
16  GROUP BY p.product_name , p.category , YEAR(o.order_date) , MONTH(o.order_date)
17  order by order_year asc, order_month asc;
18
19
```

Its Result

product_name	category	order_year	order_month	total_sales
QuickNote Notebook	Stationery & Office	2018	1	12699.00
AromaZen Candles	Home & Living	2018	1	52712.00
PetPals Collar	Miscellaneous	2018	1	42415.00
PlanPro Organizer	Stationery & Office	2018	1	80011.00
MiniExplore Telescope	Kids & Toys	2018	1	647892.00
UltraView Monitor	Electronics	2018	1	1234905.00
EcoBuzz Speaker	Electronics	2018	1	142443.00
GripMax Tools	Miscellaneous	2018	1	79239.00

Result 11 ✕

14. Top 3 Products Based on Total Revenue

MySQL Query

```
1    -- Q14 Top 3 Products Based on Total Revenue
2
3
4  ● SELECT
5        p.product_name,
6        p.category,
7        SUM(oi.quantity * p.price) AS total_revenue
8    FROM
9        customer_data_analysis.products p
10           JOIN
11       customer_data_analysis.order_items oi ON p.product_id = oi.product_id
12   GROUP BY p.product_name , p.category
13   ORDER BY total_revenue DESC
14   LIMIT 3;
15
16
```

Its Result

product_name	category	total_revenue
FreshBake Oven	Miscellaneous	17683821.00
UltraView Monitor	Electronics	15208830.00
AquaEase Purifier	Home & Living	10708810.00

Result 4 ×

15. Customers with the Highest Number of Different Products Ordered

MySQL Query

```
1     --    Q15 Customers with the Highest Number of Different Products Ordered
2
3
4  •  SELECT
5         c.customer_id,
6         c.first_name,
7         c.last_name,
8         COUNT(DISTINCT oi.product_id) AS distinct_product_count
9     FROM
10        customer_data_analysis.customers c
11            JOIN
12        customer_data_analysis.orders o ON c.customer_id = o.customer_id
13            JOIN
14        customer_data_analysis.order_items oi ON o.order_id = oi.order_id
15    GROUP BY c.customer_id , c.first_name , c.last_name
16    ORDER BY distinct_product_count DESC
17    LIMIT 1;
18
19
```

Its Result

customer_id	first_name	last_name	distinct_product_count
ALIKHA136	Aliyah	Khan	50

Result 5 ✕