**Solutions**

**A)High Level Analysis**

**1) What was the total quantity sold for all products?**

**SELECT SUM(qty) AS total\_quantity**

**FROM sales;**

**2) What is the total generated revenue for all products before discounts?**

**SELECT SUM(qty \* price) AS revenue\_before\_discount**

**FROM sales;**

**3) What was the total discount amount for all products?**

**SELECT CAST(SUM(qty \* price \* discount/100.0) AS FLOAT) AS total\_discount**

**FROM sales;**

**B)Transactional Analysis**

**1) How many unique transactions were there?**

**SELECT COUNT(DISTINCT txn\_id) AS unique\_transactions**

**FROM sales;**

**2) What is the average unique products purchased in each transaction?**

**SELECT AVG(product\_count) AS avg\_unique\_products**

**FROM (**

**SELECT**

**txn\_id,**

**COUNT(DISTINCT prod\_id) AS product\_count**

**FROM sales**

**GROUP BY txn\_id**

**) temp;**

**3)** **WITH transaction\_revenue AS (**

**SELECT**

**txn\_id,**

**SUM(qty\*price) AS revenue**

**FROM sales**

**GROUP BY txn\_id)**

**SELECT**

**DISTINCT**

**PERCENTILE\_CONT(0.25) WITHIN GROUP (ORDER BY revenue) OVER () AS pct\_25th,**

**PERCENTILE\_CONT(0.50) WITHIN GROUP (ORDER BY revenue) OVER () AS pct\_50th,**

**PERCENTILE\_CONT(0.75) WITHIN GROUP (ORDER BY revenue) OVER () AS pct\_75th**

**FROM transaction\_revenue;**

**4)** **SELECT CAST(AVG(total\_discount) AS decimal(5, 1)) AS avg\_discount\_per\_transaction**

**FROM (**

**SELECT**

**txn\_id,**

**SUM(qty\*price\*discount/100.0) AS total\_discount**

**FROM sales**

**GROUP BY txn\_id**

**) temp;**

**5) What is the percentage split of all transactions for members vs non-members?**

**SELECT**

**CAST(100.0\*COUNT(DISTINCT CASE WHEN member = 1 THEN txn\_id END)**

**/ COUNT(DISTINCT txn\_id) AS FLOAT) AS members\_pct,**

**CAST(100.0\*COUNT(DISTINCT CASE WHEN member = 0 THEN txn\_id END)**

**/ COUNT(DISTINCT txn\_id) AS FLOAT) AS non\_members\_pct**

**FROM sales;**

**6) What is the average revenue for member transactions and non-member transactions?**

**WITH member\_revenue AS (**

**SELECT**

**member,**

**txn\_id,**

**SUM(qty\*price) AS revenue**

**FROM sales**

**GROUP BY member, txn\_id**

**)**

**SELECT**

**member,**

**CAST(AVG(1.0\*revenue) AS decimal(10,2)) AS avg\_revenue**

**FROM member\_revenue**

**GROUP BY member;**

**C)Product Analysis**

**1)  What are the top 3 products by total revenue before discount?**

**SELECT**

**TOP 3 pd.product\_name,**

**SUM(s.qty \* s.price) AS revenue\_before\_discount**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.product\_name**

**ORDER BY SUM(s.qty \* s.price) DESC;**

**2) What is the total quantity, revenue and discount for each segment?**

**SELECT**

**pd.segment\_name,**

**SUM(s.qty) total\_quantity,**

**SUM(s.qty \* s.price) AS total\_revenue\_before\_discount,**

**SUM(s.qty \* s.price \* discount) AS total\_discount**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.segment\_name;**

**3) What is the top selling product for each segment?**

**WITH segment\_product\_quantity AS (**

**SELECT**

**pd.segment\_name,**

**pd.product\_name,**

**SUM(s.qty) AS total\_quantity,**

**DENSE\_RANK() OVER (PARTITION BY pd.segment\_name ORDER BY SUM(s.qty) DESC) AS rnk**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.segment\_name, pd.product\_name**

**)**

**SELECT**

**segment\_name,**

**product\_name AS top\_selling\_product,**

**total\_quantity**

**FROM segment\_product\_quantity**

**WHERE rnk = 1;**

**4) What is the total quantity, revenue and discount for each category?**

**SELECT**

**pd.category\_name,**

**SUM(s.qty) AS total\_quantity,**

**SUM(s.qty \* s.price) AS total\_revenue,**

**SUM(s.qty \* s.price \* s.discount/100) AS total\_discount**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.category\_name;**

**5) What is the top selling product for each category?**

**WITH category\_product\_quantity AS (**

**SELECT**

**pd.category\_name,**

**pd.product\_name,**

**SUM(s.qty) AS total\_quantity,**

**DENSE\_RANK() OVER (PARTITION BY pd.category\_name ORDER BY SUM(s.qty) DESC) AS rnk**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.category\_name, pd.product\_name**

**)**

**SELECT**

**category\_name,**

**product\_name AS top\_selling\_product,**

**total\_quantity**

**FROM category\_product\_quantity**

**WHERE rnk = 1;**

**6) What is the percentage split of revenue by product for each segment?**

**WITH segment\_product\_revenue AS (**

**SELECT**

**pd.segment\_name,**

**pd.product\_name,**

**SUM(s.qty \* s.price) AS product\_revenue**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.segment\_name, pd.product\_name**

**)**

**SELECT**

**segment\_name,**

**product\_name,**

**CAST(100.0 \* product\_revenue**

**/ SUM(product\_revenue) OVER (PARTITION BY segment\_name)**

**AS decimal (10, 2)) AS segment\_product\_pct**

**FROM segment\_product\_revenue;**

**7) What is the percentage split of revenue by segment for each category?**

**WITH segment\_category\_revenue AS (**

**SELECT**

**pd.segment\_name,**

**pd.category\_name,**

**SUM(s.qty \* s.price) AS category\_revenue**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.segment\_name, pd.category\_name**

**)**

**SELECT**

**segment\_name,**

**category\_name,**

**CAST(100.0 \* category\_revenue**

**/ SUM(category\_revenue) OVER (PARTITION BY category\_name)**

**AS decimal (10, 2)) AS segment\_category\_pct**

**FROM segment\_category\_revenue;**

**8) What is the percentage split of total revenue by category?**

**WITH category\_revenue AS (**

**SELECT**

**pd.category\_name,**

**SUM(s.qty \* s.price) AS revenue**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY pd.category\_name**

**)**

**SELECT**

**category\_name,**

**CAST(100.0 \* revenue / SUM(revenue) OVER () AS decimal (10, 2)) AS category\_pct**

**FROM category\_revenue;**

**9) What is the total transaction “penetration” for each product?**

**(hint: penetration = number of transactions where at least 1 quantity of a product was purchased divided by total number of transactions)**

**WITH product\_transations AS (**

**SELECT**

**DISTINCT s.prod\_id, pd.product\_name,**

**COUNT(DISTINCT s.txn\_id) AS product\_txn,**

**(SELECT COUNT(DISTINCT txn\_id) FROM sales) AS total\_txn**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**GROUP BY prod\_id, pd.product\_name**

**)**

**SELECT**

**\*,**

**CAST(100.0 \* product\_txn / total\_txn AS decimal(10,2)) AS penetration\_pct**

**FROM product\_transations;**

**10) What is the most common combination of at least 1 quantity of any 3 products in a 1 single transaction?**

**--Count the number of products in each transaction**

**WITH products\_per\_transaction AS (**

**SELECT**

**s.txn\_id,**

**pd.product\_id,**

**pd.product\_name,**

**s.qty,**

**COUNT(pd.product\_id) OVER (PARTITION BY txn\_id) AS cnt**

**FROM sales s**

**JOIN product\_details pd**

**ON s.prod\_id = pd.product\_id**

**),**

**--Filter transactions that have the 3 products and group them to a cell**

**combinations AS (**

**SELECT**

**STRING\_AGG(product\_id, ', ') WITHIN GROUP (ORDER BY product\_id) AS product\_ids,**

**STRING\_AGG(product\_name, ', ') WITHIN GROUP (ORDER BY product\_id) AS product\_names**

**FROM products\_per\_transaction**

**WHERE cnt = 3**

**GROUP BY txn\_id**

**),**

**--Count the number of times each combination appears**

**combination\_count AS (**

**SELECT**

**product\_ids,**

**product\_names,**

**COUNT (\*) AS common\_combinations**

**FROM combinations**

**GROUP BY product\_ids, product\_names**

**)**

**--Filter the most common combinations**

**SELECT**

**product\_ids,**

**product\_names**

**FROM combination\_count**

**WHERE common\_combinations = (SELECT MAX(common\_combinations)**

**FROM combination\_count);**