Title: "Quantium Virtual Internship - Retail Strategy and Analytics - Task 1"

Stage: Data Cleaning

step 1. Preview the data

```
-- Preview transaction_data
SELECT *
FROM `river-hold-450804-s3.product_sales.transaction_data`
LIMIT 10;


-- Preview purchase data
SELECT *
FROM `river-hold-450804-s3.product_sales.purchase`
LIMIT 10;
```

step 2. Check for missing/null values

```
-- Count missing/null values in key columns
SELECT
  SUM(CASE WHEN PROD_NAME IS NULL THEN 1 ELSE 0 END) AS missing_prod_name,
  SUM(CASE WHEN Date IS NULL THEN 1 ELSE 0 END) AS missing_date,
  SUM(CASE WHEN PROD_QTY IS NULL THEN 1 ELSE 0 END) AS missing_quantity,
  SUM(CASE WHEN TOT_SALES IS NULL THEN 1 ELSE 0 END) AS missing_sales_value
FROM `river-hold-450804-s3.product_sales.transaction_data`;
```

step 3. Summary statistics for numeric columns

```
SELECT
  COUNT(*) AS total_rows,
  MIN(PROD_QTY) AS min_quantity,
  MAX(PROD_QTY) AS max_quantity,
  AVG(PROD_QTY) AS avg_quantity,
  MIN(TOT_SALES) AS min_sales,
  MAX(TOT_SALES) AS max_sales,
  AVG(TOT_SALES) AS avg_sales
FROM `river-hold-450804-s3.product_sales.transaction_data`;
```

step 4. Filter out non-chip products (remove salsa)

```
SELECT *
FROM `river-hold-450804-s3.product_sales.transaction_data`
WHERE LOWER(PROD_NAME) NOT LIKE '%salsa%';
```

step 5. Detect outlier transactions (e.g. quantity > 100)

```
SELECT *
FROM `river-hold-450804-s3.product_sales.transaction_data`
WHERE PROD_QTY > 100
ORDER BY PROD_QTY DESC;
```

step 6. Join transaction_data with purchase

```
SELECT
  t.TXN_ID,
  t.Date,
  t.PROD_NAME,
  t.PROD_QTY,
  t.TOT_SALES,
  p.LIFESTAGE,
  p.`PREMIUM CUSTOMER`
FROM `river-hold-450804-s3.product_sales.transaction_data` t
LEFT JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR;
```

step 7. Total sales by LIFESTAGE & PREMIUM CUSTOMER

```
SELECT
  p.LIFESTAGE,
  p.`PREMIUM CUSTOMER` AS premium_customer,
  SUM(t.TOT_SALES) AS total_sales
FROM `river-hold-450804-s3.product_sales.transaction_data` t
JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR
GROUP BY p.LIFESTAGE, premium_customer
ORDER BY total_sales DESC;
```

step 8. Number of customers by segment

```
SELECT
  p.LIFESTAGE,
  p.`PREMIUM CUSTOMER` AS premium_customer,
  COUNT(DISTINCT p.LYLTYCARD_NBR) AS customer_count
FROM `river-hold-450804-s3.product_sales.transaction_data` t
JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR
GROUP BY p.LIFESTAGE, premium_customer
ORDER BY customer_count DESC;
```

step 9. Average units per customer

```
SELECT
  p.LIFESTAGE,
  p.`PREMIUM CUSTOMER` AS premium_customer,
  AVG(t.PROD_QTY) AS avg_units_per_customer
FROM `river-hold-450804-s3.product_sales.transaction_data` t
JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR
GROUP BY p.LIFESTAGE, premium_customer
ORDER BY avg_units_per_customer DESC;
```

step 10. Average price per unit

```
SELECT
  p.LIFESTAGE,
  p.`PREMIUM CUSTOMER` AS premium_customer,
  SUM(t.TOT_SALES) / SUM(t.PROD_QTY) AS avg_price_per_unit
FROM `river-hold-450804-s3.product_sales.transaction_data` t
JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR
GROUP BY p.LIFESTAGE, premium_customer
ORDER BY avg_price_per_unit DESC;
```

step 11. Top products preferred by a specific segment

Example: **Mainstream Young Singles/Couples**

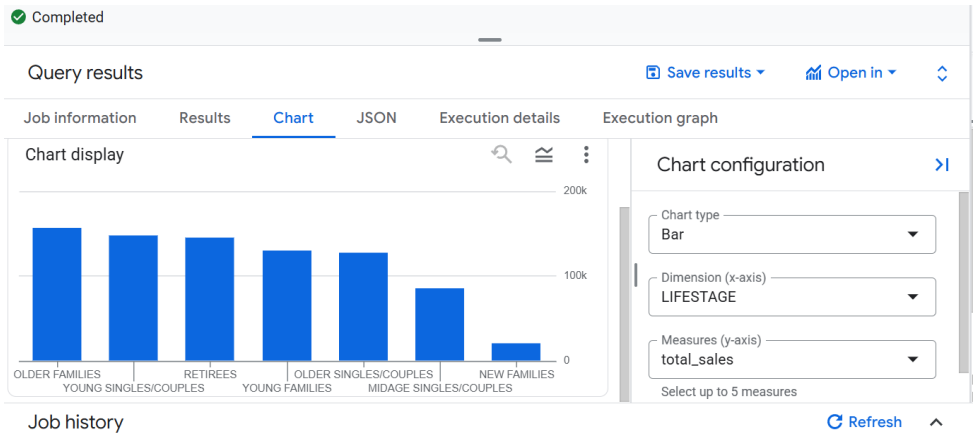
```
SELECT
  t.PROD_NAME,
  SUM(t.TOT_SALES) AS total_revenue,
  COUNT(*) AS transactions
FROM `river-hold-450804-s3.product_sales.transaction_data` t
JOIN `river-hold-450804-s3.product_sales.purchase` p
  ON t.LYLTY_CARD_NBR = p.LYLTYCARD_NBR
WHERE p.LIFESTAGE = 'YOUNG SINGLES/COUPLES'
  AND p.`PREMIUM CUSTOMER` = 'MAINSTREAM'
GROUP BY t.PROD_NAME
ORDER BY total_revenue DESC
LIMIT 10;
```

Answering questions about poducts

1. Who spends the most on chips?

Calculate total sales by LIFESTAGE and PREMIUM CUSTOMER

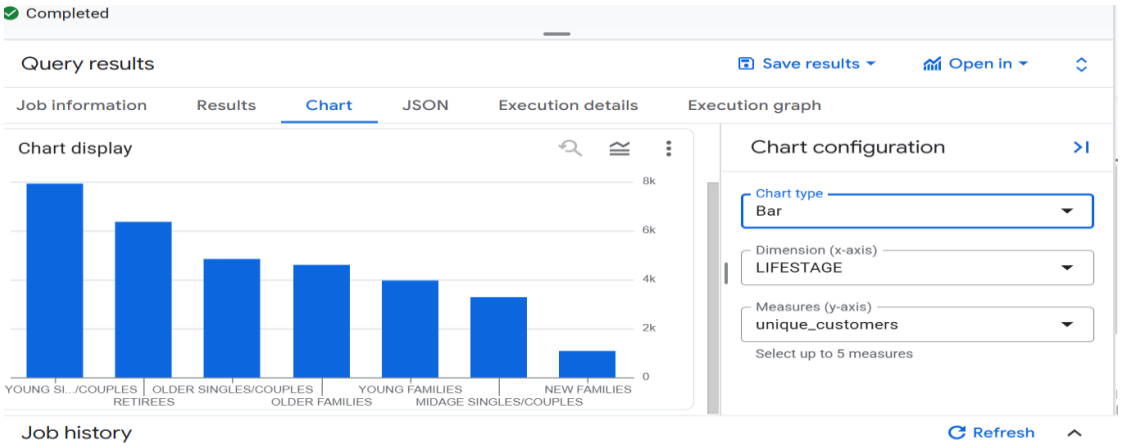
```
SELECT
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer,
  SUM(TOT_SALES) AS total_sales
FROM `river-hold-450804-s3.product_sales.cleaned_final`
GROUP BY LIFESTAGE, premium_customer
ORDER BY total_sales DESC;
```



2. How many customers are in each segment?

Count unique customers in each segment

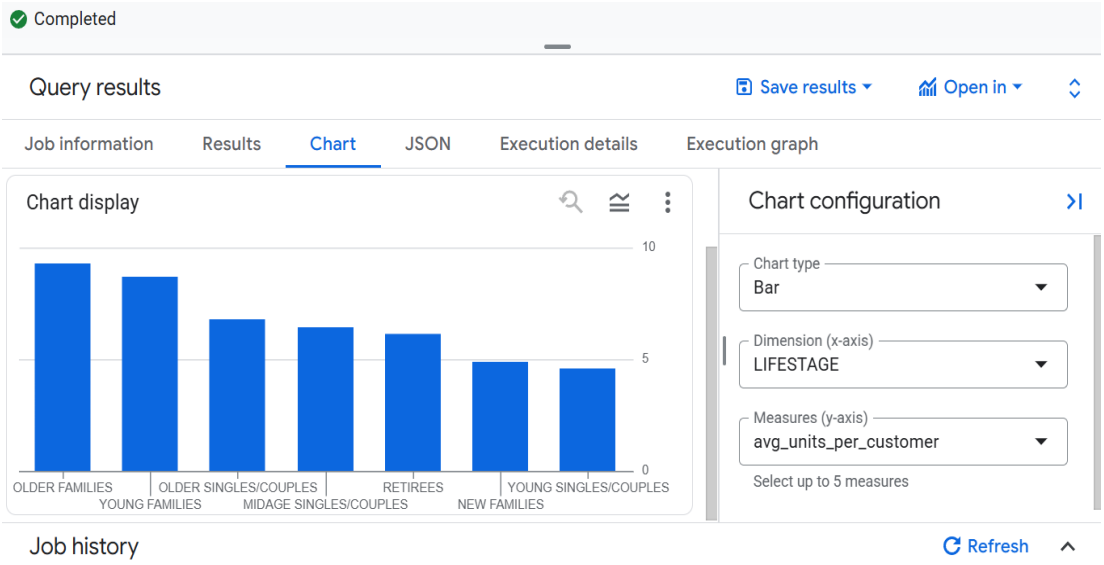
```
SELECT
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer,
  COUNT(DISTINCT LYLTY_CARD_NBR) AS unique_customers
FROM `river-hold-450804-s3.product_sales.cleaned_final`
GROUP BY LIFESTAGE, premium_customer
ORDER BY unique_customers DESC;
```



3. How many chips are bought per customer by segment?

Average units per customer per segment

```
SELECT
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer,
  ROUND(SUM(PROD_QTY) / COUNT(DISTINCT LYLTY_CARD_NBR), 2) AS avg_units_per_customer
FROM `river-hold-450804-s3.product_sales.cleaned_final`
GROUP BY LIFESTAGE, premium_customer
ORDER BY avg_units_per_customer DESC;
```



4. What's the average chip price by customer segment?

Average price per packet for each segment

```
SELECT
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer,
  ROUND(SUM(TOT_SALES) / SUM(PROD_QTY), 2) AS avg_price_per_unit
FROM `river-hold-450804-s3.product_sales.cleaned_final`
GROUP BY LIFESTAGE, premium_customer
ORDER BY avg_price_per_unit DESC;
```

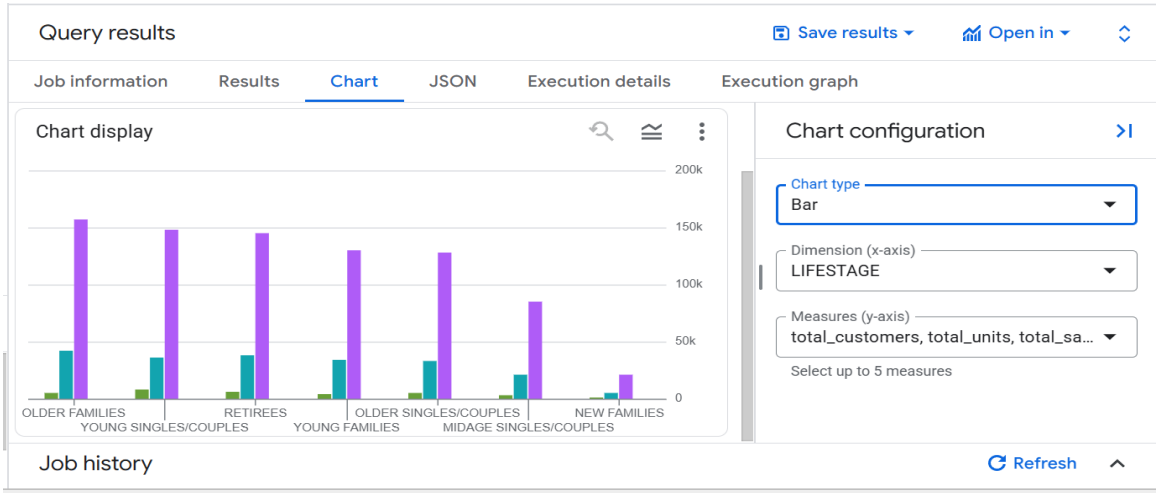
5. Are higher sales due to more customers or higher buying frequency?

Compare sales vs customers

```
SELECT
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer,
  COUNT(DISTINCT LYLTY_CARD_NBR) AS total_customers,
  SUM(PROD_QTY) AS total_units,
  SUM(TOT_SALES) AS total_sales,
  ROUND(SUM(PROD_QTY)/COUNT(DISTINCT LYLTY_CARD_NBR),2) AS avg_units_per_customer
FROM `river-hold-450804-s3.product_sales.cleaned_final`
```

GROUP BY LIFESTAGE, premium_customer

ORDER BY total_sales DESC;



6. Do Mainstream Young Singles/Couples prefer specific brands?

Top brands for that segment

SELECT

BRAND,

SUM(PROD_QTY) AS total_units,

ROUND(SUM(TOT_SALES),2) AS total_sales

FROM `river-hold-450804-s3.product_sales.cleaned_final`

WHERE LIFESTAGE = 'YOUNG SINGLES/COUPLES'

AND `PREMIUM CUSTOMER` = 'MAINSTREAM'

GROUP BY BRAND

ORDER BY total_sales DESC

LIMIT 10;

7. Do they prefer larger pack sizes?

Pack size preference for Mainstream Young Singles/Couples

SELECT

PACK_SIZE,

COUNT(*) AS total_transactions,

SUM(PROD_QTY) AS total_units,

ROUND(SUM(TOT_SALES),2) AS total_sales

FROM `river-hold-450804-s3.product_sales.cleaned_final`

```
WHERE LIFESTAGE = 'YOUNG SINGLES/COUPLES'
  AND `PREMIUM CUSTOMER` = 'MAINSTREAM'
GROUP BY PACK_SIZE
ORDER BY total_units DESC;
```

Compare with overall population

```
SELECT
  PACK_SIZE,
  COUNT(*) AS total_transactions,
  SUM(PROD_QTY) AS total_units,
  ROUND(SUM(TOT_SALES),2) AS total_sales
FROM `river-hold-450804-s3.product_sales.cleaned_final`
GROUP BY PACK_SIZE
ORDER BY total_units DESC;
```

8. Is the price difference statistically significant?

BigQuery can't do t-test directly → export price per unit for segments to Python/R:

```
SELECT
  ROUND(TOT_SALES / PROD_QTY, 2) AS price_per_unit,
  LIFESTAGE,
  `PREMIUM CUSTOMER` AS premium_customer
FROM `river-hold-450804-s3.product_sales.cleaned_final`
WHERE LIFESTAGE IN ('YOUNG SINGLES/COUPLES','MIDAGE SINGLES/COUPLES')
  AND `PREMIUM CUSTOMER` IN ('MAINSTREAM','BUDGET','PREMIUM');
```