

DATA WAREHOUSING PROJECT — ERP & CRM INTEGRATION USING MEDALLION ARCHITECTURE

PRESENTED BY:

- GANGARAJU
- VISHWAJEET
- KARTIK



INTRODUCTION

- Purpose: Build a complete DWH from raw ERP+CRM data.
- Goal: Convert messy CSVs into analytics-ready structured data.
- Approach: Used Medallion Architecture + SQL + Star Schema.
- Outcome: A unified, clean, query-optimized dataset.

SOURCE SYSTEMS (CRM + ERP)

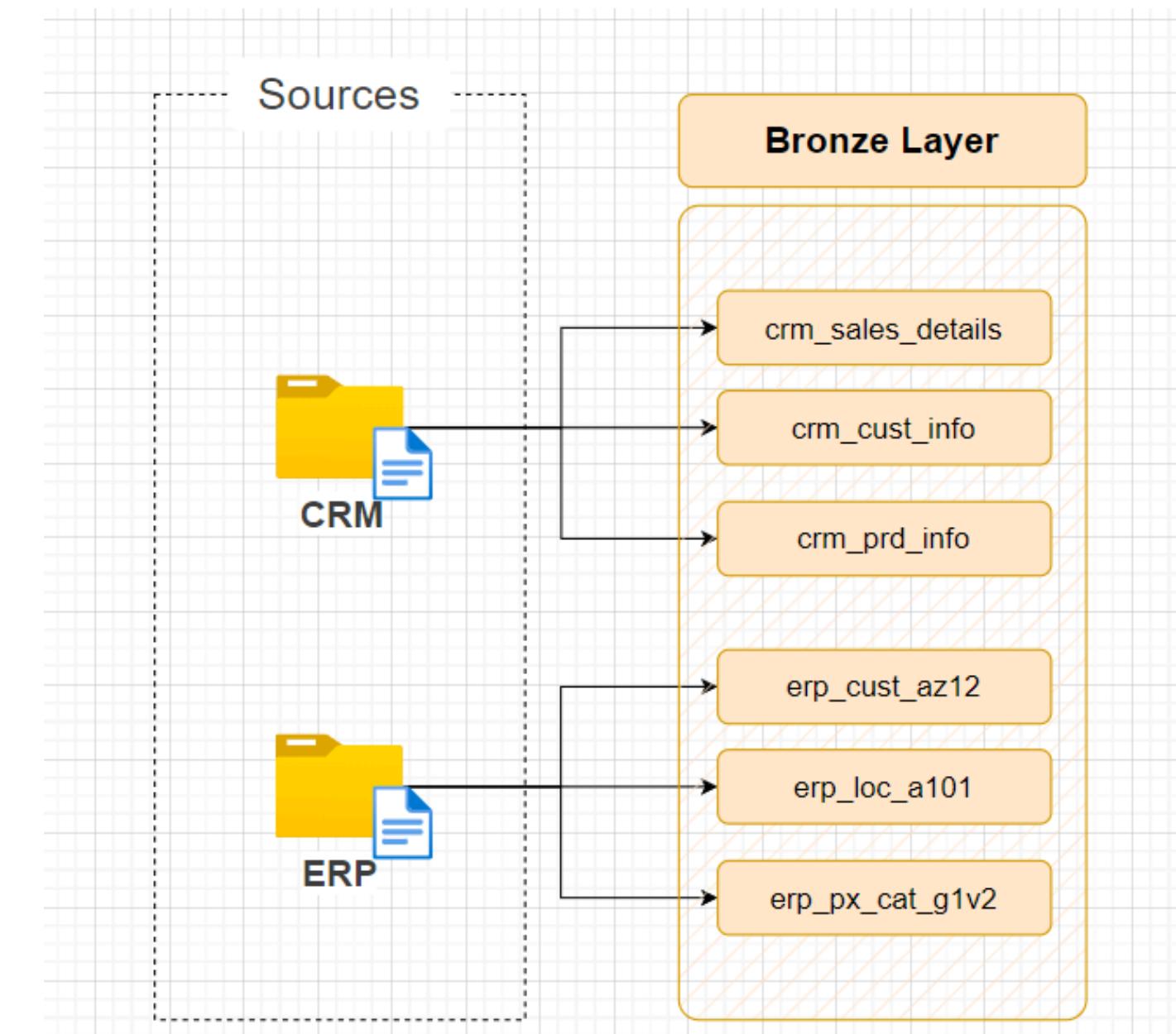
CRM PROVIDES:

- CUSTOMER INFO (CUST_INFO.CSV)
- PRODUCT INFO (PRD_INFO.CSV)
- SALESTRANSACtIONS(SALES_DETAILS.CSV)

ERP PROVIDES:

- CUSTOMER DOB + GENDER (CUST_AZ12.CSV)
- CUSTOMER LOCATION (LOC_A101.CSV)
- PRODUCT CATEGORIES (PX_CAT_G1V2.CSV)

TOTAL RAW FILES: 6



SQL IMPLEMENTATION

1. DDL (DATA DEFINITION LANGUAGE) – STRUCTURE & INTEGRITY

- CREATED TABLES FOR ALL SOURCE DATASETS.
- DESIGNED CLEANED TABLES WITH CORRECTED DATA TYPES AND STANDARDIZED COLUMNS.
- DEVELOPED DIMENSIONAL SCHEMA WITH PRIMARY & FOREIGN KEY CONSTRAINTS.
- PURPOSE: ENSURES DATA INTEGRITY AND A SOLID WAREHOUSE STRUCTURE.

2. DML (DATA MANIPULATION LANGUAGE) – DATA LOADING

- LOADED RAW CSV DATA INTO THE TABLES USING INSERT / COPY.
- APPLIED TRANSFORMATIONS AND INSERTED CLEANED RECORDS VIA SELECT...INSERT QUERIES.
- POPULATED DIMENSION & FACT TABLES WITH SURROGATE KEYS FOR ANALYTICS.
- PURPOSE: IMPLEMENTS A STRUCTURED ETL-LIKE WORKFLOW FOR DATA PREPARATION.

3. DQL (DATA QUERY LANGUAGE) – DATA RETRIEVAL & ANALYSIS

- SELECT + WHERE: FILTER SALES BY DATE, PRODUCT, OR COUNTRY.
- ORDER BY: SORT BY HIGHEST SALES OR LATEST TRANSACTIONS.
- JOIN: LINK FACT TABLES WITH DIMENSION TABLES FOR RICHER INSIGHTS.
- AGGREGATES: COMPUTE SUM, COUNT, AVG FOR MONTHLY OR ANNUAL KPIs.
- PURPOSE: ENABLES EXPLORATION AND OLAP-STYLE ANALYTICS ON THE WAREHOUSE DATA.

CUSTOMER SEGMENTATION

- PURPOSE: SEGMENT CUSTOMERS BASED ON TOTAL SALES.
- BUSINESS QUESTION: IDENTIFY HIGH-VALUE, MEDIUM-VALUE, AND LOW-VALUE CUSTOMERS.
- LOGIC: JOIN CUSTOMER DETAILS WITH SALES, COMPUTE SUM(SALES_AMOUNT), CLASSIFY WITH CASE.
- OUTPUT: CUSTOMER_ID, FIRST_NAME, LAST_NAME, COUNTRY, TOTAL_SALES, MONETARY_SEGMENT.
- INSIGHTS: TARGET HIGH-VALUE CUSTOMERS FOR MARKETING AND RETENTION.

```
498  SELECT
499      c.customer_id,
500      c.first_name,
501      c.last_name,
502      c.country,
503      SUM(f.sales_amount) AS total_sales,
504      CASE
505          WHEN SUM(f.sales_amount) >= 10000 THEN 'High Value'
506          WHEN SUM(f.sales_amount) >= 5000 THEN 'Medium Value'
507          ELSE 'Low Value'
508      END AS monetary_segment
509  FROM gold.fact_sales f
510  JOIN gold.dim_customers c
511      ON f.customer_key = c.customer_key
512  GROUP BY c.customer_id, c.first_name, c.last_name, c.country
513  ORDER BY c.customer_id ASC;
```

Data Output Messages Notifications

Showing rows: 1 to 1000

	customer_id	first_name	last_name	country	total_sales	monetary_segment
1	11000	Jon	Yang	Australia	10583.00	High Value
2	11001	Eugene	Huang	Australia	8609.00	Medium Value
3	11002	Ruben	Torres	Australia	10479.00	High Value
4	11003	Christy	Zhu	Australia	10374.00	High Value
5	11004	Elizabeth	Johnson	Australia	10529.00	High Value
6	11005	Julio	Ruiz	Australia	10399.00	High Value
7	11006	Janet	Alvarez	Australia	10424.00	High Value

TOP CUSTOMERS ANALYSIS CONTENT

- PURPOSE: IDENTIFY HIGHEST-VALUE CUSTOMERS.
- BUSINESS QUESTION: WHICH CUSTOMERS CONTRIBUTE THE MOST REVENUE?
- SQL LOGIC: USE ORDER BY SALES DESC LIMIT 10.
- INSIGHTS: PRIORITIZE LOYALTY PROGRAMS & RETENTION STRATEGIES.
- OLAP FEATURE: AGGREGATE AND RANK DIMENSIONS EFFICIENTLY.

```
425     c.customer_id,  
426     c.first_name,  
427     c.last_name,  
428     c.country,  
429     SUM(f.sales_amount) AS total_sales  
430   FROM gold.fact_sales f  
431   JOIN gold.dim_customers c  
432     ON f.customer_key = c.customer_key  
433   GROUP BY c.customer_id, c.first_name, c.last_name, c.country  
434   ORDER BY total_sales DESC  
435   LIMIT 10;  
436
```

Data Output Messages Notifications

Showing rows 1-7 of 7

	customer_id integer	first_name text	last_name text	country text	total_sales numeric
1	11433	Maurice	Shan	France	19766.00
2	11439	Janet	Munoz	France	19443.00
3	12131	Randall	Dominguez	France	18232.00
4	12301	Nichole	Nara	France	18175.00
5	12300	Adriana	Gonzalez	France	18032.00
6	12124	Brandi	Gill	France	18032.00
7	12308	Margaret	He	France	18023.00

PRODUCT CATEGORY INSIGHTS

- PURPOSE: EVALUATE PRODUCT LINE PERFORMANCE.
- BUSINESS QUESTION: WHICH CATEGORIES GENERATE HIGHEST SALES?
- SQL LOGIC: AGGREGATE SALES BY CATEGORY, CALCULATE TOTALS & AVERAGES.
- INSIGHTS: OPTIMIZE INVENTORY & MARKETING STRATEGIES.
- OLAP FEATURE: MULTIDIMENSIONAL GROUPING & SUMMARIZATION.

```
437 -----  
438 -- Q4: Product Category Performance  
439 -----  
440 SELECT  
441     p.category,  
442     p.subcategory,  
443     SUM(f.sales_amount)AS total_sales,  
444     SUM(f.quantity)AS total_quantity,  
445     AVG(f.price)AS avg_unit_price  
446 FROM gold.fact_sales f  
447 JOIN gold.dim_products p  
448     ON f.product_key = p.product_key  
449 GROUP BY p.category, p.subcategory  
450 ORDER BY total_sales DESC;
```

Data Output Messages Notifications

Showing rows 1-7 of 7

	category text	subcategory text	total_sales numeric	total_quantity bigint	avg_unit_price numeric
1	Bikes	Road Bikes	20057472.00	11774	1703.2503821980635298
2	Bikes	Mountain Bikes	17895729.00	8527	2098.5330127829248270
3	Bikes	Touring Bikes	3848179.00	2169	1774.1719686491470724
4	Accessori...	Helmets	457485.00	13071	35.00000000000000000000
5	Clothing	Jerseys	346482.00	6813	50.9378580872631115
6	Clothing	Shorts	71330.00	1019	70.00000000000000000000
7	Clothing	Gloves	68658.00	2862	24.00000000000000000000

SALES TREND ANALYSIS

- PURPOSE: TRACK SALES PROGRESSION OVER TIME.
- BUSINESS QUESTION: HOW DOES SALES ACCUMULATE MONTHLY?
- SQL LOGIC: USE SUM() OVER (PARTITION BY MONTH ORDER BY DATE ROWS BETWEEN UNBOUNDED PRECEDING).
- INSIGHTS: UNDERSTAND MOMENTUM & SEASONAL TRENDS.
- OLAP FEATURE: WINDOW FUNCTIONS ENABLE ADVANCED CUMULATIVE CALCULATIONS.

```
452
453 -- Q5: Sales Trend by Month + Running Total
454
455 SELECT
456     d.year,
457     d.month,
458     SUM(f.sales_amount) AS monthly_sales,
459     SUM(SUM(f.sales_amount)) OVER (
460         ORDER BY d.year, d.month
461         ROWS BETWEEN UNBOUNDED PRECEDING AND CURRENT ROW
462     ) AS running_total
463     FROM gold.fact_sales f
464     JOIN gold.dim_date d
465         ON f.order_date_key = d.date_key
466     GROUP BY d.year, d.month
467     ORDER BY d.year, d.month;
```

Data Output Messages Notifications

Showing rows:

	year integer 	month integer 	monthly_sales numeric 	running_total numeric 
1	2010	12	44817.00	44817.00
2	2011	1	479581.00	524398.00
3	2011	2	476792.00	1001190.00
4	2011	3	497048.00	1498238.00
5	2011	4	515323.00	2013561.00
6	2011	5	578328.00	2591889.00
7	2011	6	763416.00	3355305.00

SALES BY YEAR & MONTH

NOVEL SQL COMMAND (ROLLUP)

- PURPOSE: ANALYZE SALES TRENDS OVER TIME.
- BUSINESS QUESTION: HOW DO MONTHLY AND YEARLY SALES COMPARE?
- SQL LOGIC: USE GROUP BY WITH ROLLUP TO CREATE HIERARCHICAL AGGREGATES.
- INSIGHTS: REVEALS MONTHLY, YEARLY TOTALS & SUBTOTALS.
- OLAP FEATURE: HIERARCHICAL AGGREGATION SUPPORTS MULTIDIMENSIONAL ANALYSIS.

```
395  SELECT
396      d.year,
397      d.month,
398      SUM(f.sales_amount) AS total_sales
399  FROM gold.fact_sales f
400  JOIN gold.dim_date d
401      ON f.order_date_key = d.date_key
402  GROUP BY ROLLUP (d.year, d.month)
403  ORDER BY d.year, d.month;
```

Data Output Messages Notifications

	year integer 	month integer 	total_sales numeric 
3	2011	1	479581.00
4	2011	2	476792.00
5	2011	3	497048.00
6	2011	4	515323.00
7	2011	5	578328.00
8	2011	6	763416.00
9	2011	7	613486.00

SALES BY COUNTRY & GENDER

NOVEL SQL COMMAND (CUBE)

- PURPOSE: COMPARE SALES ACROSS REGIONS AND DEMOGRAPHICS.
- BUSINESS QUESTION: WHICH COUNTRY-GENDER SEGMENTS PERFORM BEST?
- SQL LOGIC: USE GROUP BY WITH CUBE FOR ALL COMBINATIONS.
- INSIGHTS: IDENTIFIES TOP-PERFORMING REGIONS & DEMOGRAPHIC GROUPS.
- OLAP FEATURE: MULTIDIMENSIONAL AGGREGATION WITH MULTIPLE DIMENSIONS.

The screenshot shows a SQL query editor interface with a code editor and a results grid.

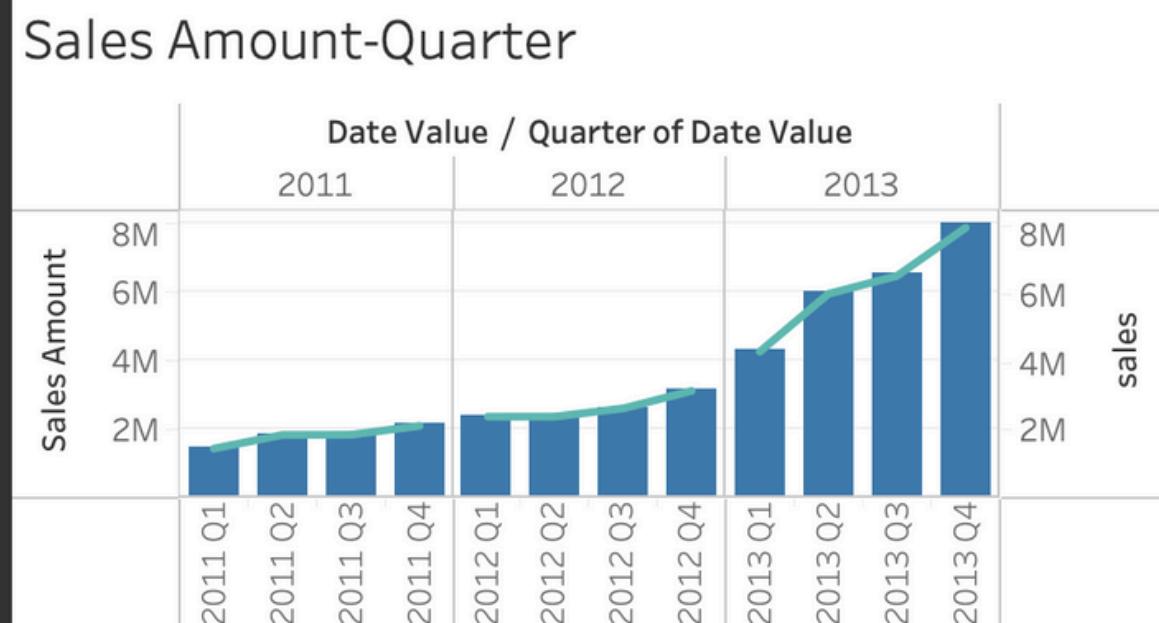
SQL Query:

```
410     c.country,
411     c.gender,
412     SUM(f.sales_amount) AS total_sales
413   FROM gold.fact_sales f
414   JOIN gold.dim_customers c
415     ON f.customer_key = c.customer_key
416   GROUP BY CUBE (c.country, c.gender)
417   HAVING c.country IS NOT NULL AND c.country <> '' AND c.gender IS NOT NULL AND c.gender <> ''
418   ORDER BY c.country, c.gender;
419
```

Results Grid:

	country text	gender text	total_sales numeric
1	Australia	Female	6800077.00
2	Australia	Male	6442913.00
3	Canada	F	105.00
4	Canada	Female	1380044.00
5	Canada	Male	1313880.00
6	DE	Female	1316231.00
7	DE	Male	0.00

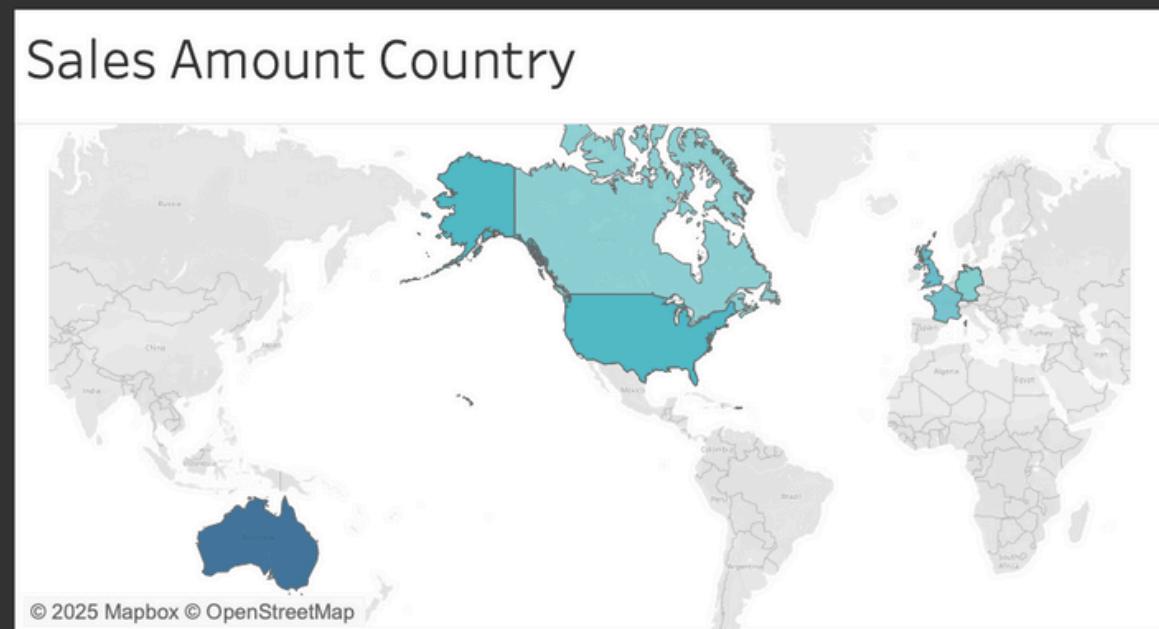
KPI	
Average	2,072
quantity	48,101
sales	42,826,730



Category Filter

Category

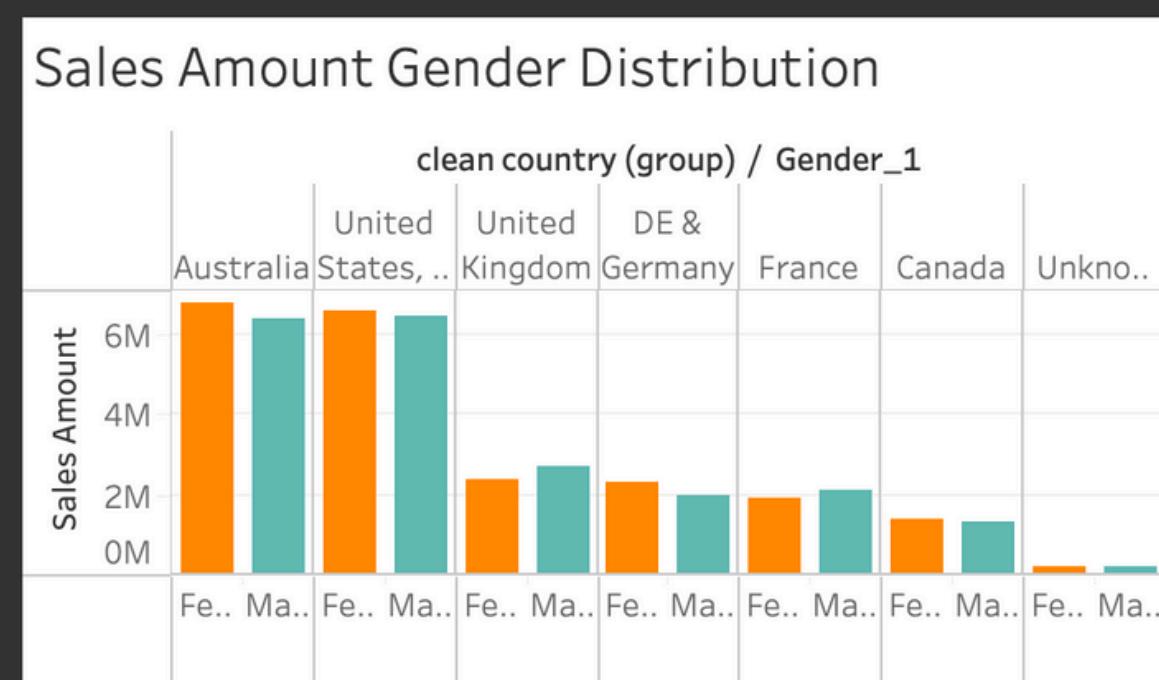
Accessories	Abc
Bikes	Abc
Clothing	Abc



Region Filter

clean country (gr.. ▾)

Australia	Abc
Canada	Abc
DE & Germany	Abc
France	Abc
United Kingdom	Abc
United States, US, USA	Abc



Gender Filter

Gender

Female	Abc
Male	Abc

THANKYOU

Q/A