

HASIL RUNNING EXPLARATORY DATA PADA DBEAVER

Data Ingestion:

Challenge_DS_RakaminXKkalbe (Tableau Public)

Case Study - Transaction.csv+ (Multiple Connections)

Extract contains all data. Sep 30, 2023, 8:48:35 PM

Filters: 0 Add

Case Study - Transaction.csv

Case Study - Customer.csv

Case Study - Product.csv

Case Study - Store.csv

Case Study - Transaction.csv

8 fields 5020 rows

100 rows

Name	Transaction ID	Customer ID	Date	Product ID	Price
Case Study - Transaction.csv	TR11369	328	1/1/2022	P3	123
Case Study - Transaction.csv	TR16356	165	1/1/2022	P9	123

Query 1:

```
/**
QUERY UNTUK MENENTUKAN RATA-RATA UMUR PELANGGAN BERDASARKAN MARITAL STATUS
**/
SELECT
CASE
WHEN "Marital Status" = '' THEN 'Unknown'
ELSE "Marital Status"
END AS marital_status,
FLOOR(AVG(age)) AS avg_age
FROM kalbe.customer
GROUP BY marital_status;
```

ECT CASE WHEN "Marital Status" = '' T | Enter a SQL

ABC marital_status	123 avg_age
Unknown	31
Married	43
Single	29

Query 2:

HASIL RUNNING EXPLARATORY DATA PADA DBEAVER

```

--
--
-- QUERY UNTUK MENENTUKAN RATA-RATA UMUR PELANGGAN BERDASARKAN GENDER
--
SELECT
    CASE
        WHEN gender = 0 THEN 'Female'
        WHEN gender = 1 THEN 'Male'
    END AS customer_gender,
    FLOOR(AVG(age)) AS avg_age
FROM kalbe.customer
GROUP BY gender;

```

Results 2 | Execution plan - 1

SELECT CASE WHEN gender = 0 THEN 'Fe' | Enter a SQL expression to filter results (use Ctrl+Space)

	customer_gender	avg_age
1	Female	40
2	Male	39

Query 3:

```

--
--
-- QUERY UNTUK MENENTUKAN NAMA STORE DENGAN TOTAL QTY TERBANYAK
--
SELECT s.storename, sum(t.qty) AS total_qty
FROM kalbe.store s
JOIN kalbe.transaction t ON t.storeid = s.storeid
GROUP BY s.storename
ORDER BY total_qty desc
LIMIT 1;

```

store 1 | Execution plan - 1

SELECT s.storename, sum(t.qty) AS total_q | Enter a SQL expression to filter results (use Ctrl+Space)

	storename	total_qty
1	Lingga	2,777

Query 4:

HASIL RUNNING EXPLARATORY DATA PADA DBEAVER

SQL Query Editor:

```
--  
QUERY UNTUK MENENTUKAN NAMA PRODUK TERLARIS DENGAN TOTAL AMT TERBANYAK  
**/  
SELECT p."Product Name", sum(t.totalamount) AS "Total Amount"  
FROM kalbe.product p  
JOIN kalbe.transaction t ON t.productid = p.productid  
GROUP BY p."Product Name"  
ORDER BY "Total Amount" desc  
LIMIT 1;
```

Execution plan - 1

SQL Editor: `SELECT p."Product Name", sum(t.totalamo`

Grid:

	Product Name	Total Amount
1	Cheese Stick	27,615,000