HACKTIV8

DATA WAREHOUSE FOR SALES ANALYSIS

COMPREHENSIVE DATA ANALYTICS PROGRAM

amandarizkipati@gmail.com

❯ Start Slide

REFERENCE

DATASET SOURCE

[The Look - BigQuery]

Background of the **Analysis**

Company Profile

The Look is a global e-commerce platform offering fashion and lifestyle products. As the business grows, the company needs a data warehouse system to analyze sales and customer behavior for better decision-making and growth.

The Objective

To develop a data warehouse that enables efficient analysis of sales performance, customer behavior, and return patterns. This will support data-driven decision-making to enhance profitability, optimize operations, and improve customer service through actionable insights.

Business Understanding

Business Context

The Look is a rapidly expanding e-commerce platform, where the need for comprehensive data analysis has become critical to support strategic decision-making and sustain growth.

Main Problem

Running queries directly on the operational (OLTP) database is not feasible due to limitations and potential disruptions to the performance of transactional systems.

Need

Develop a separate data warehouse system that enables efficient and reliable analysis without affecting operational processes.

HACKTIV8

Business **Problem**

As The Look continues to scale, the demand for data-driven insights increases. However, relying on the operational database for analytical queries poses risks to system performance and lacks flexibility for in-depth analysis. To overcome these limitations, a separate data warehouse is essential for supporting strategic decision-making through reliable and efficient data access.

HACKTIV8

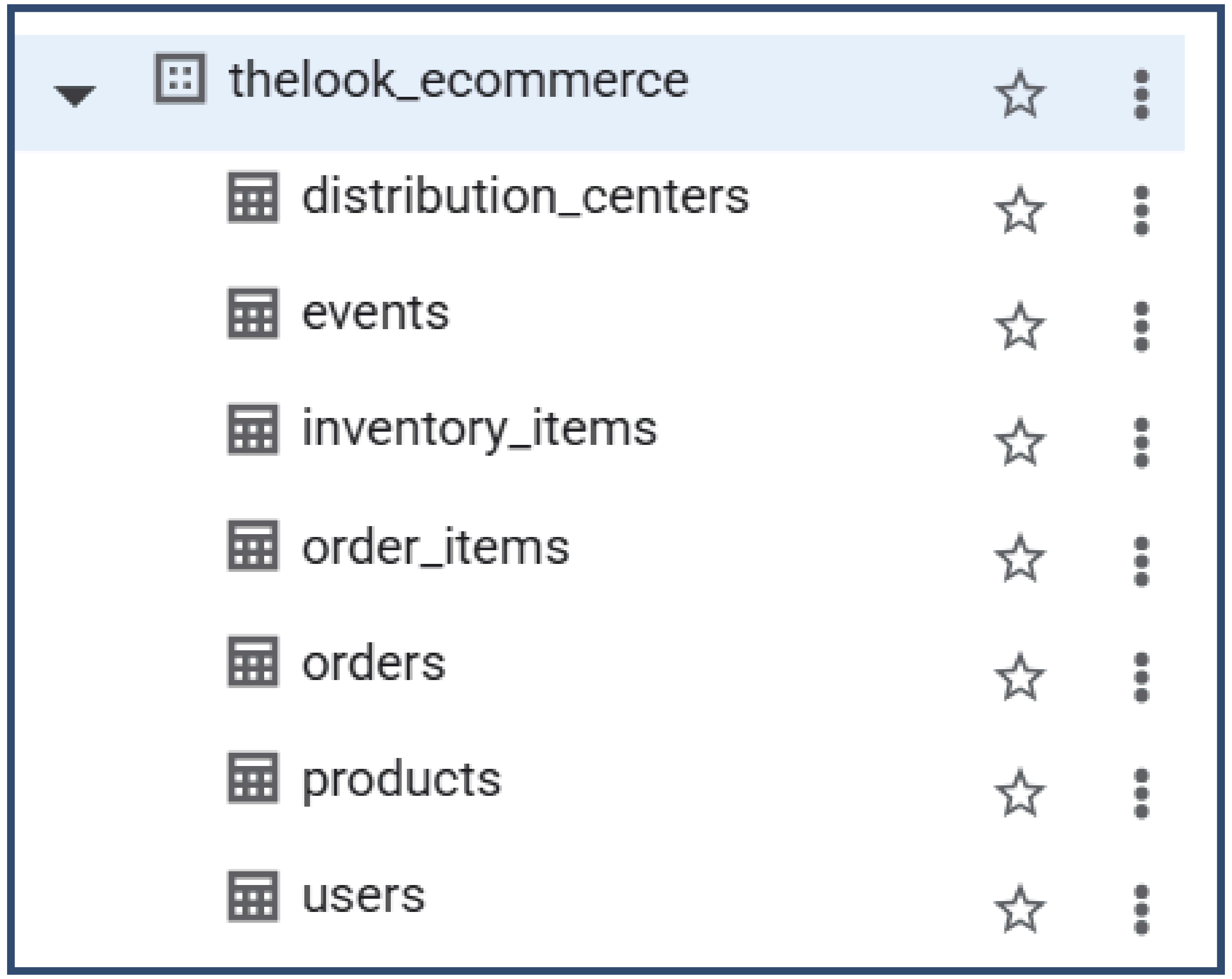
Business Process

Users make product purchases through the platform, and the related transaction data (such as who made the purchase, what product was bought, when it was bought, the price, shipping status, and so on) is recorded and can be used to analyze sales performance, product profitability, customer behavior, and seasonal trends.

HACKTIV8

BigQuery

The Look dataset has 7 tables, but we will only use data that is relevant to sales analysis.



ORDER_ITEMS

- id
- order_id
- user_id
- product_id
- inventory_item_id
- status
- created_at
- shipped_at
- delivered_at
- returned_at
- sale_price

ORDERS

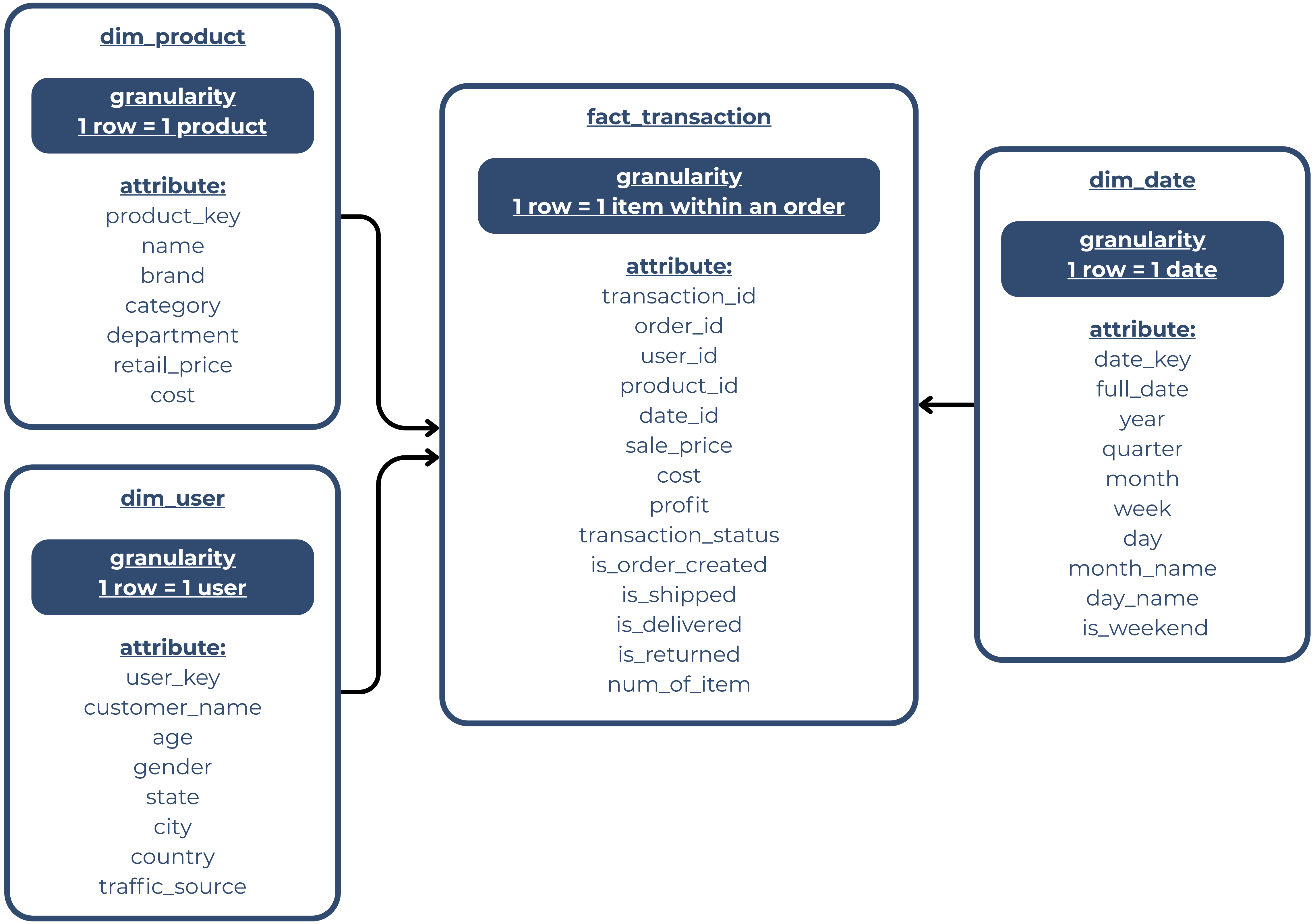
- order_id
- user_id
- status
- gender
- created_at
- returned_at
- shipped_at
- delivered_at
- num_of_item

PRODUCTS

- id
- cost
- category
- name
- brand
- retail_price
- department
- sku
- distribution_center_id

USERS

- id
- first_name
- last_name
- email
- age
- gender
- state
- street_address
- postal_code
- city
- country
- latitude
- longitude
- traffic_source
- created_at
- user_geom



Data Modelling

The data warehouse model employs a **Star Schema** approach, featuring a central fact table named **'transaction'** that stores sales information at a granularity of one row per item within each order. This fact table is linked to three dimension tables:

- **'dim_product'**
- **'dim_user'**
- **'dim_date'**

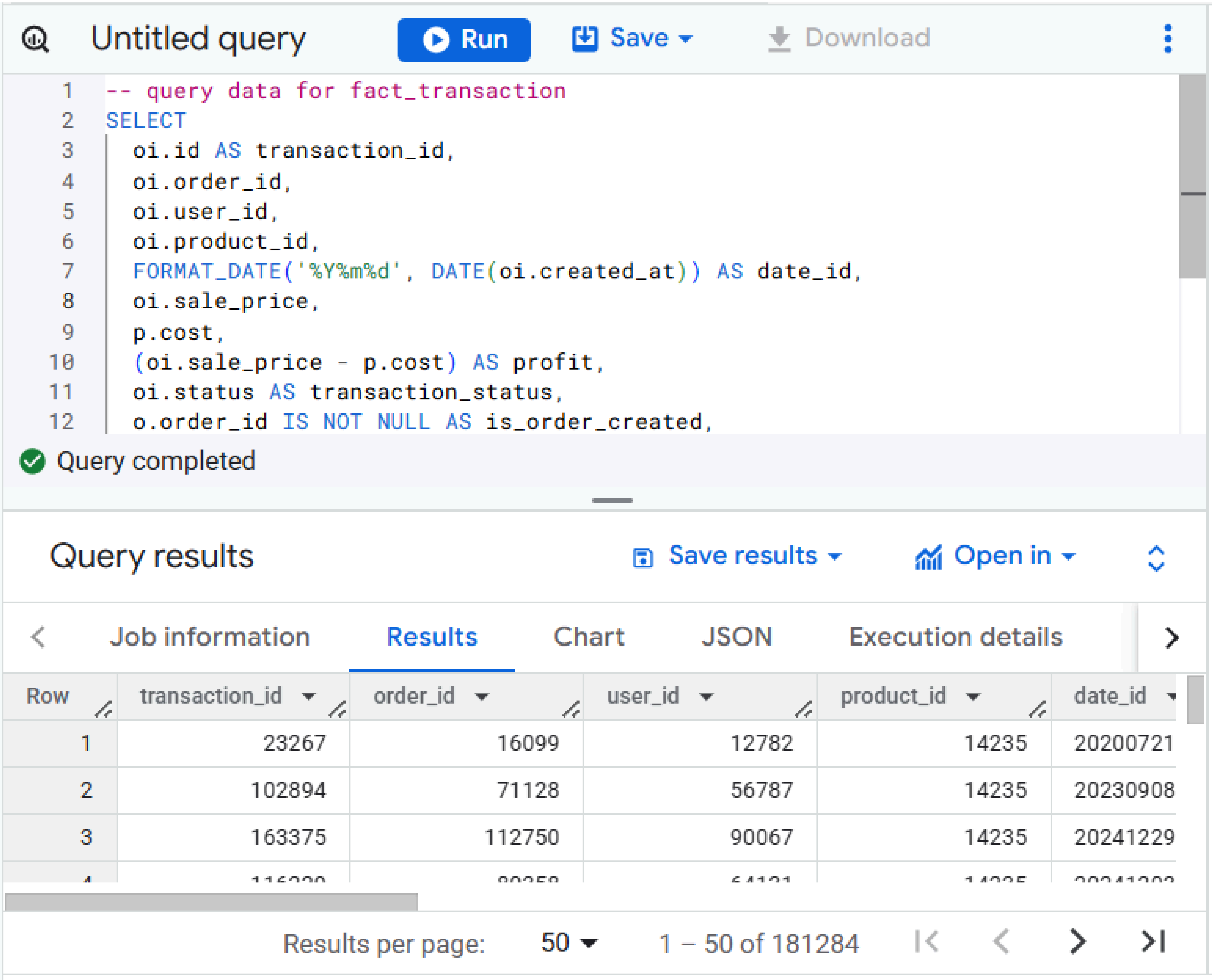
Extract Data

Source
bigquery-public-data.thelook_ecommerce dataset

Extraction method
SQL queries run directly in BigQuery

Output
CSV files for each dimension/fact table

Data Source (BigQuery)   SQL Queries   CSV Files



HACKTIV8

```sql
-- query data for fact_transaction
SELECT
  oi.id AS transaction_id,
  oi.order_id,
  oi.user_id,
  oi.product_id,
  FORMAT_DATE('%Y%m%d', DATE(oi.created_at)) AS date_id,
  oi.sale_price,
  p.cost,
  (oi.sale_price - p.cost) AS profit,
  oi.status AS transaction_status,
  o.order_id IS NOT NULL AS is_order_created,
  oi.shipped_at IS NOT NULL AS is_shipped,
  oi.delivered_at IS NOT NULL AS is_delivered,
  oi.returned_at IS NOT NULL AS is_returned,
  o.num_of_item
FROM bigquery-public-data.thelook_ecommerce.order_items oi
LEFT JOIN bigquery-public-data.thelook_ecommerce.products p ON oi.product_id = p.id
LEFT JOIN bigquery-public-data.thelook_ecommerce.orders o ON oi.order_id = o.order_id;
```

Extract Data

fact_transaction

The transaction data is combined from multiple tables (order_items, products, and orders) to form the fact_transaction table.

Extract Data

```sql
-- query data for dim_user
SELECT
  id AS user_key,
  first_name || ' ' || last_name AS customer_name,
  age,
  gender,
  state,
  city,
  country,
  traffic_source
FROM bigquery-public-data.thelook_ecommerce.users;
```

```sql
-- query data for dim_product
SELECT
  id AS product_key,
  name,
  brand,
  category,
  department,
  retail_price,
  cost
FROM bigquery-public-data.thelook_ecommerce.products;
```

dim_user

This query builds the dim_user table by selecting user ID, full name, age, gender, location, and traffic source from the users table to create detailed customer profiles.

dim_product

This query creates the dim_product table by extracting product ID, name, brand, category, department, retail price, and cost from the products table.

HACKTIV8

```sql
-- query data for dim_date
WITH unique_dates AS (
  SELECT DISTINCT DATE(created_at) AS transaction_date
  FROM `bigquery-public-data.thelook_ecommerce.order_items`
)
SELECT
  FORMAT_DATE('%Y%m%d', transaction_date) AS date_key,
  transaction_date AS full_date,
  EXTRACT(YEAR FROM transaction_date) AS year,
  EXTRACT(QUARTER FROM transaction_date) AS quarter,
  EXTRACT(MONTH FROM transaction_date) AS month,
  EXTRACT(WEEK FROM transaction_date) AS week,
  EXTRACT(DAY FROM transaction_date) AS day,
  FORMAT_DATE('%B', transaction_date) AS month_name,
  FORMAT_DATE('%A', transaction_date) AS day_name,
  CASE
    WHEN EXTRACT(DAYOFWEEK FROM transaction_date) IN (1, 7) THEN TRUE
    ELSE FALSE
  END AS is_weekend
FROM unique_dates
ORDER BY transaction_date;
```

Extract Data

dim_date

This query generates the dim_date table from unique transaction dates, adding fields like year, month, week, day, month/day names, and weekend indicator.

Transform Data

01 Checking Data Types

Based on the analysis, the data types of all attributes in each DataFrame are already appropriate and correctly assigned. Therefore, no data type conversion or modification is necessary. The current format is suitable for further data processing and analysis.

```
root
 |-- transaction_id: integer (nullable = true)
 |-- order_id: integer (nullable = true)
 |-- user_id: integer (nullable = true)
 |-- product_id: integer (nullable = true)
 |-- date_id: integer (nullable = true)
 |-- sale_price: double (nullable = true)
 |-- cost: double (nullable = true)
 |-- profit: double (nullable = true)
 |-- transaction_status: string (nullable = true)
 |-- is_order_created: boolean (nullable = true)
 |-- is_shipped: boolean (nullable = true)
 |-- is_delivered: boolean (nullable = true)
 |-- is_returned: boolean (nullable = true)
 |-- num_of_item: integer (nullable = true)
```

```
root
 |-- product_key: integer (nullable = true)
 |-- name: string (nullable = true)
 |-- brand: string (nullable = true)
 |-- category: string (nullable = true)
 |-- department: string (nullable = true)
 |-- retail_price: double (nullable = true)
 |-- cost: double (nullable = true)
```

```
root
 |-- date_key: integer (nullable = true)
 |-- full_date: date (nullable = true)
 |-- year: integer (nullable = true)
 |-- quarter: integer (nullable = true)
 |-- month: integer (nullable = true)
 |-- week: integer (nullable = true)
 |-- day: integer (nullable = true)
 |-- month_name: string (nullable = true)
 |-- day_name: string (nullable = true)
 |-- is_weekend: boolean (nullable = true)
```

```
root
 |-- user_key: integer (nullable = true)
 |-- customer_name: string (nullable = true)
 |-- age: integer (nullable = true)
 |-- gender: string (nullable = true)
 |-- state: string (nullable = true)
 |-- city: string (nullable = true)
 |-- country: string (nullable = true)
 |-- traffic_source: string (nullable = true)
```

Transform **Data**

Missing values in 'fact_transaction':

transaction_id	order_id	user_id	product_id	date_id	sale_price	cost	profit	transaction_status	is_order_created	is_shipped	is_delivered	is_returned	num_of_item
0	0	0	0	0	0	0	0	0	0	0	0	0	0

Missing values in 'dim_product':

product_key	name	brand	category	department	retail_price	cost
0	2	24	0	0	0	0

Missing values in 'dim_user':

user_key	customer_name	age	gender	state	city	country	traffic_source
0	0	0	0	0	0	0	0

Missing values in 'dim_date':

date_key	full_date	year	quarter	month	week	day	month_name	day_name	is_weekend
0	0	0	0	0	0	0	0	0	0

02

Handling Missing Value

All dimension and fact tables, fact_transaction, dim_user, and dim_datecontain no missing values and are ready for analysis, while dim_product has less than 30% missing values per column, which are addressed through imputation to ensure data completeness.

Transform **Data**

No duplicate rows found.
After Checking and Dropping Duplicate Rows:

transaction_id	order_id	user_id	product_id	date_id	sale_price	cost	profit	transaction_status	is_order_created	is_shipped	is_delivered	is_returned	num_of_item
177152	122160	98069	14235	20250412	0.0199999995529651	0.0082999997779726	0.011699999774992502	Cancelled	true	false	false	false	1
28849	19870	15771	14235	20240529	0.0199999995529651	0.0082999997779726	0.011699999774992502	Complete	true	true	true	false	4
27428	18901	15036	14235	20240514	0.0199999995529651	0.0082999997779726	0.011699999774992502	Processing	true	false	false	false	1
79317	54742	43944	14235	20250203	0.0199999995529651	0.0082999997779726	0.011699999774992502	Shipped	true	true	false	false	2
158000	108931	87429	14235	20250118	0.0199999995529651	0.0082999997779726	0.011699999774992502	Shipped	true	true	false	false	3

only showing top 5 rows

No duplicate rows found.
After Checking and Dropping Duplicate Rows:

user_key	customer_name	age	gender	state	city	country	traffic_source
25075	Clifford Johnson	36	M	Acre	null	Brasil	Search
97806	Angela Lopez	50	F	Acre	null	Brasil	Search
41816	Susan Kelley	55	F	Acre	null	Brasil	Search
82351	Jacqueline Zhang	62	F	Acre	null	Brasil	Search
70916	Marie Parker	66	F	Acre	null	Brasil	Search

only showing top 5 rows

No duplicate rows found.
After Checking and Dropping Duplicate Rows:

product_key	name	brand	category	department	retail_price	cost
13842	Low Profile Dyed ...	MG	Accessories	Women	6.25	2.518749990849756
13928	Low Profile Dyed ...	MG	Accessories	Women	5.949999809265137	2.3383499148894105
14115	Enzyme Regular So...	MG	Accessories	Women	10.989999771118164	4.879559879379869
14157	Enzyme Regular So...	MG	Accessories	Women	10.989999771118164	4.648769887297898
14273	Washed Canvas Ivy...	MG	Accessories	Women	15.989999771118164	6.507929886473045

only showing top 5 rows

No duplicate rows found.
After Checking and Dropping Duplicate Rows:

date_key	full_date	year	quarter	month	week	day	month_name	day_name	is_weekend
20190110	2019-01-10	2019	1	1	1	10	January	Thursday	false
20190111	2019-01-11	2019	1	1	1	11	January	Friday	false
20190117	2019-01-17	2019	1	1	2	17	January	Thursday	false
20190118	2019-01-18	2019	1	1	2	18	January	Friday	false
20190122	2019-01-22	2019	1	1	3	22	January	Tuesday	false

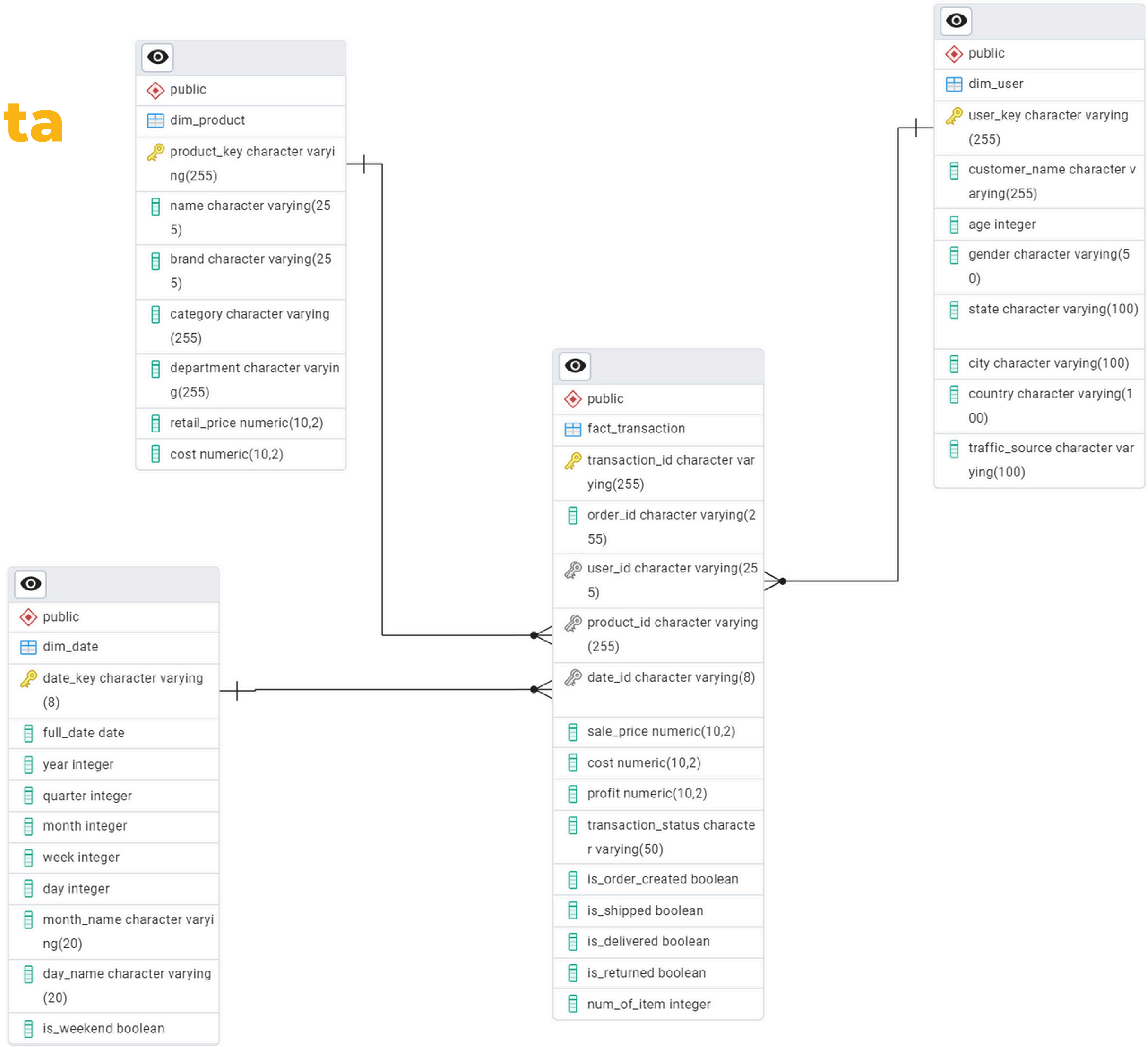
only showing top 5 rows

03

Handling Duplicate Data

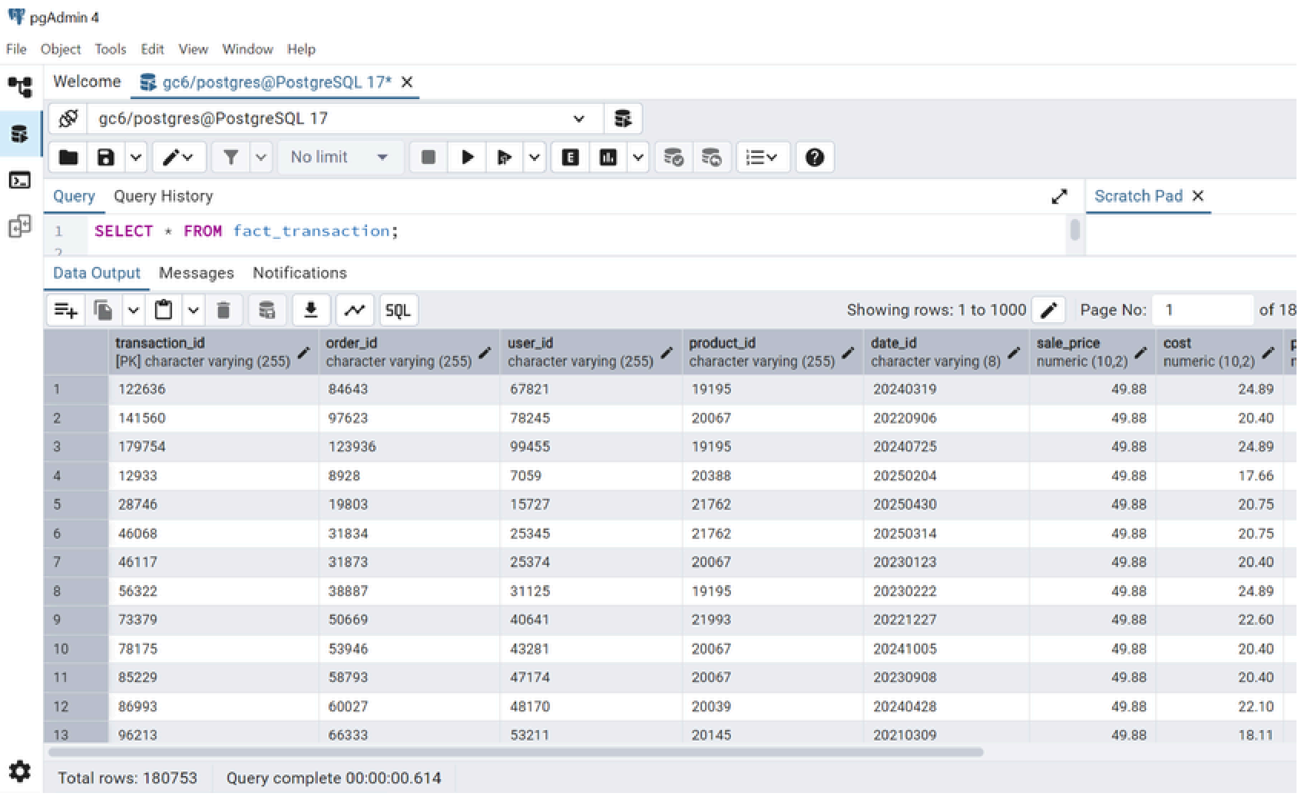
All tables, fact_transaction, dim_user, dim_date, and dim_product, have been validated and are free of duplicate rows, ensuring high data quality and reliable foundation for analysis and modeling.

Load Data



public
dim_product
product_key character varying(255)
name character varying(255)
brand character varying(255)
category character varying (255)
department character varying(255)
retail_price numeric(10,2)
cost numeric(10,2)
public
dim_user
user_key character varying (255)
customer_name character varying(255)
age integer
gender character varying(50)
state character varying(100)
city character varying(100)
country character varying(100)
traffic_source character varying(100)
public
fact_transaction
transaction_id character varying(255)
order_id character varying(255)
user_id character varying(255)
product_id character varying (255)
date_id character varying(8)
sale_price numeric(10,2)
cost numeric(10,2)
profit numeric(10,2)
transaction_status character varying(50)
is_order_created boolean
is_shipped boolean
is_delivered boolean
is_returned boolean
num_of_item integer
public
dim_date
date_key character varying (8)
full_date date
year integer
quarter integer
month integer
week integer
day integer
month_name character varying(20)
day_name character varying (20)
is_weekend boolean

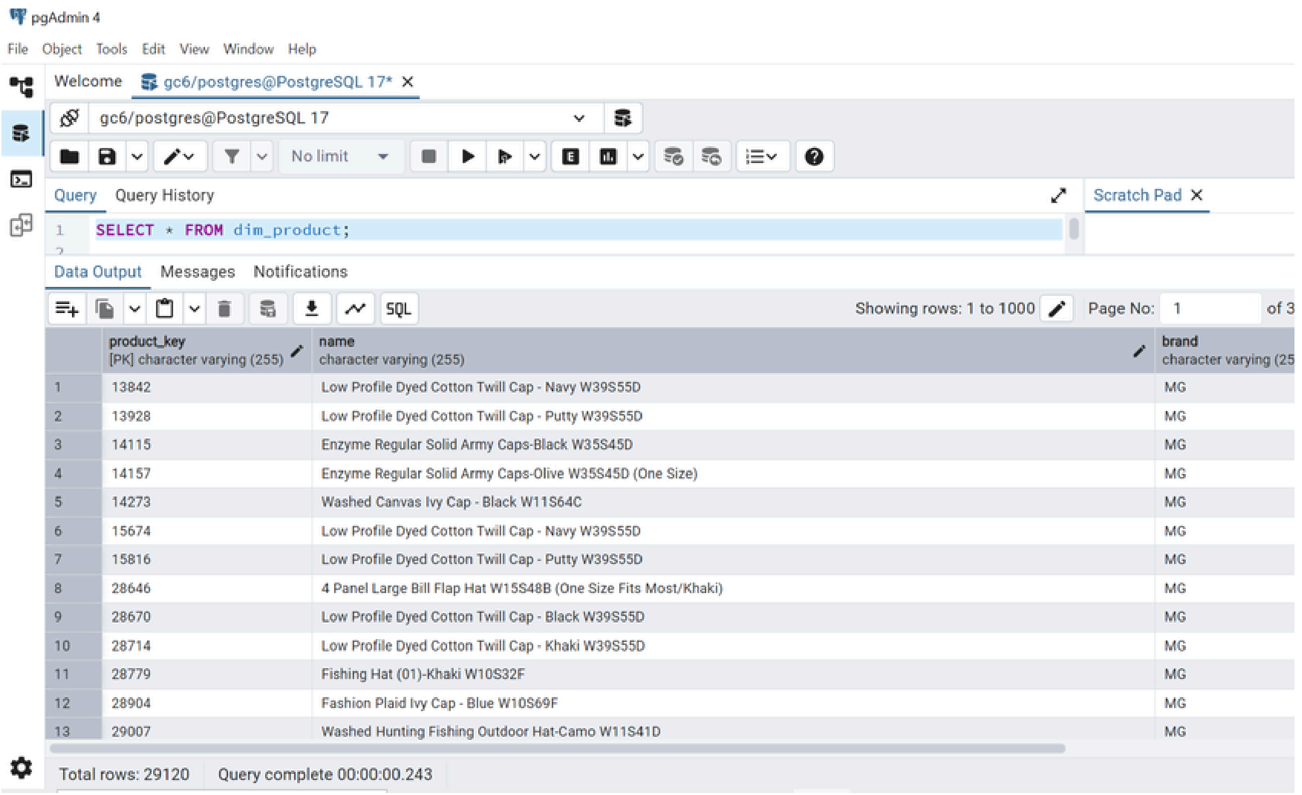
HACKTIV8



```
SELECT * FROM fact_transaction;
```

	transaction_id [PK] character varying (255)	order_id character varying (255)	user_id character varying (255)	product_id character varying (255)	date_id character varying (8)	sale_price numeric (10,2)	cost numeric (10,2)
1	122636	84643	67821	19195	20240319	49.88	24.89
2	141560	97623	78245	20067	20220906	49.88	20.40
3	179754	123936	99455	19195	20240725	49.88	24.89
4	12933	8928	7059	20388	20250204	49.88	17.66
5	28746	19803	15727	21762	20250430	49.88	20.75
6	46068	31834	25345	21762	20250314	49.88	20.75
7	46117	31873	25374	20067	20230123	49.88	20.40
8	56322	38887	31125	19195	20230222	49.88	24.89
9	73379	50669	40641	21993	20221227	49.88	22.60
10	78175	53946	43281	20067	20241005	49.88	20.40
11	85229	58793	47174	20067	20230908	49.88	20.40
12	86993	60027	48170	20039	20240428	49.88	22.10
13	96213	66333	53211	20145	20210309	49.88	18.11

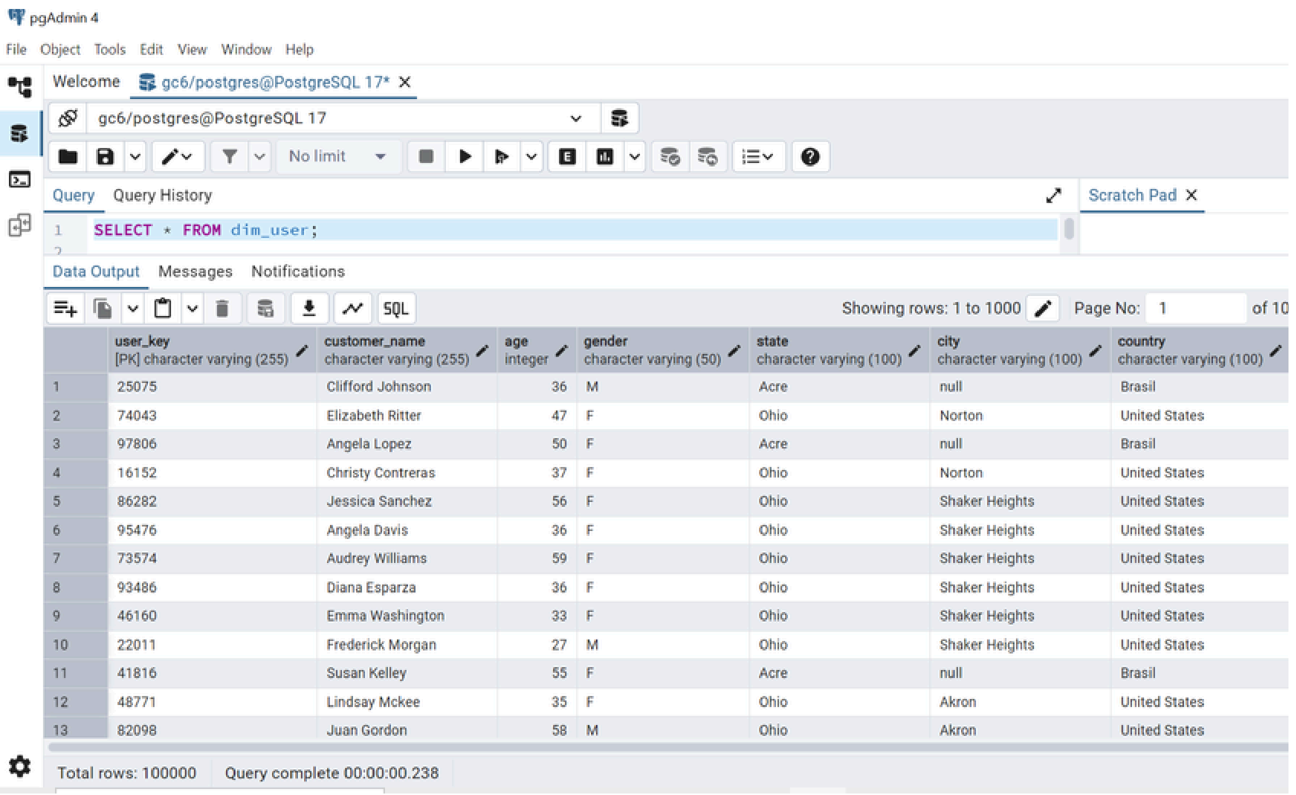
Total rows: 180753 Query complete 00:00:00.614



```
SELECT * FROM dim_product;
```

	product_key [PK] character varying (255)	name character varying (255)	brand character varying (25
1	13842	Low Profile Dyed Cotton Twill Cap - Navy W39S55D	MG
2	13928	Low Profile Dyed Cotton Twill Cap - Putty W39S55D	MG
3	14115	Enzyme Regular Solid Army Caps-Black W35S45D	MG
4	14157	Enzyme Regular Solid Army Caps-Olive W35S45D (One Size)	MG
5	14273	Washed Canvas Ivy Cap - Black W11S64C	MG
6	15674	Low Profile Dyed Cotton Twill Cap - Navy W39S55D	MG
7	15816	Low Profile Dyed Cotton Twill Cap - Putty W39S55D	MG
8	28646	4 Panel Large Bill Flap Hat W15S48B (One Size Fits Most/Khaki)	MG
9	28670	Low Profile Dyed Cotton Twill Cap - Black W39S55D	MG
10	28714	Low Profile Dyed Cotton Twill Cap - Khaki W39S55D	MG
11	28779	Fishing Hat (01)-Khaki W10S32F	MG
12	28904	Fashion Plaid Ivy Cap - Blue W10S69F	MG
13	29007	Washed Hunting Fishing Outdoor Hat-Camo W11S41D	MG

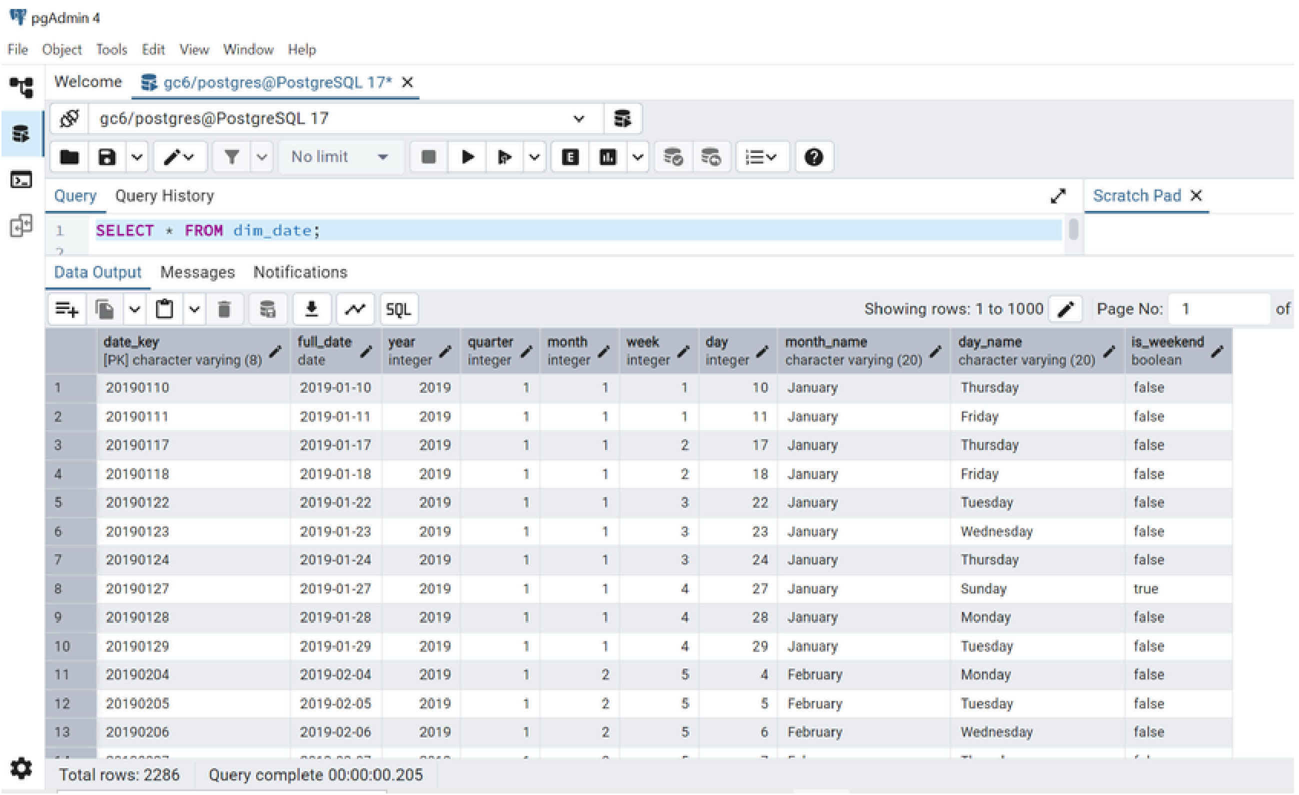
Total rows: 29120 Query complete 00:00:00.243



```
SELECT * FROM dim_user;
```

	user_key [PK] character varying (255)	customer_name character varying (255)	age integer	gender character varying (50)	state character varying (100)	city character varying (100)	country character varying (100)
1	25075	Clifford Johnson	36	M	Acre	null	Brasil
2	74043	Elizabeth Ritter	47	F	Ohio	Norton	United States
3	97806	Angela Lopez	50	F	Acre	null	Brasil
4	16152	Christy Contreras	37	F	Ohio	Norton	United States
5	86282	Jessica Sanchez	56	F	Ohio	Shaker Heights	United States
6	95476	Angela Davis	36	F	Ohio	Shaker Heights	United States
7	73574	Audrey Williams	59	F	Ohio	Shaker Heights	United States
8	93486	Diana Esparza	36	F	Ohio	Shaker Heights	United States
9	46160	Emma Washington	33	F	Ohio	Shaker Heights	United States
10	22011	Frederick Morgan	27	M	Ohio	Shaker Heights	United States
11	41816	Susan Kelley	55	F	Acre	null	Brasil
12	48771	Lindsay Mckee	35	F	Ohio	Akron	United States
13	82098	Juan Gordon	58	M	Ohio	Akron	United States

Total rows: 100000 Query complete 00:00:00.238



```
SELECT * FROM dim_date;
```

	date_key [PK] character varying (8)	full_date date	year integer	quarter integer	month integer	week integer	day integer	month_name character varying (20)	day_name character varying (20)	is_weekend boolean
1	20190110	2019-01-10	2019	1	1	1	10	January	Thursday	false
2	20190111	2019-01-11	2019	1	1	1	11	January	Friday	false
3	20190117	2019-01-17	2019	1	1	2	17	January	Thursday	false
4	20190118	2019-01-18	2019	1	1	2	18	January	Friday	false
5	20190122	2019-01-22	2019	1	1	3	22	January	Tuesday	false
6	20190123	2019-01-23	2019	1	1	3	23	January	Wednesday	false
7	20190124	2019-01-24	2019	1	1	3	24	January	Thursday	false
8	20190127	2019-01-27	2019	1	1	4	27	January	Sunday	true
9	20190128	2019-01-28	2019	1	1	4	28	January	Monday	false
10	20190129	2019-01-29	2019	1	1	4	29	January	Tuesday	false
11	20190204	2019-02-04	2019	1	2	5	4	February	Monday	false
12	20190205	2019-02-05	2019	1	2	5	5	February	Tuesday	false
13	20190206	2019-02-06	2019	1	2	5	6	February	Wednesday	false

Total rows: 2286 Query complete 00:00:00.205

Load Data

The dataset has been successfully integrated into PostgreSQL as a data warehouse, enabling efficient and seamless analysis.

Example of Data Warehouse Usage

Which product generated the highest total profit?

The North Face Apex Bionic Soft Shell Jacket - Men's' recorded a total profit of 9,174.47, making it the most profitable product.

```sql
SELECT
    dp.name AS product_name,
    SUM(ft.profit) AS total_profit
FROM fact_transaction ft
JOIN dim_product dp ON ft.product_id = dp.product_key
GROUP BY dp.name
ORDER BY total_profit DESC
LIMIT 1;
```

	product_name character varying (255)	total_profit numeric
1	The North Face Apex Bionic Soft Shell Jacket - Men's	9174.47

Which **country** has the **highest number** of transactions?

China recorded the highest number of transactions, totaling 60,891, making it the most active country in the dataset.

```sql
SELECT
    du.country,
    COUNT(ft.transaction_id) AS total_transactions
FROM fact_transaction ft
JOIN dim_user du ON ft.user_id = du.user_key
GROUP BY du.country
ORDER BY total_transactions DESC
LIMIT 1;
```

	country character varying (100)	total_transactions bigint
1	China	60891

HACKTIV8

What is the **monthly sales trend** over the year?

The analysis shows fluctuating monthly sales in 2025, with a peak in April at 673,711.19. February had the lowest sales, indicating potential for improvement or further investigation.

4o

```
SELECT
    dd.year,
    dd.month,
    dd.month_name,
    SUM(ft.sale_price) AS total_sales
FROM fact_transaction ft
JOIN dim_date dd ON ft.date_id = dd.date_key
WHERE dd.year = EXTRACT(YEAR FROM CURRENT_DATE)
GROUP BY dd.year, dd.month, dd.month_name
ORDER BY dd.month;
```

	year integer	month integer	month_name character varying (20)	total_sales numeric
1	2025	1	January	442026.66
2	2025	2	February	417630.86
3	2025	3	March	533005.98
4	2025	4	April	673711.19
5	2025	5	May	431135.83

What is the average profit per transaction by gender?

The analysis shows that male users have a higher average profit per transaction (32.87) compared to female users (29.01). This suggests that transactions by male users tend to generate slightly more profit.

```sql
SELECT
    du.gender,
    ROUND(AVG(ft.profit), 2) AS avg_profit_per_transaction
FROM fact_transaction ft
JOIN dim_user du ON ft.user_id = du.user_key
GROUP BY du.gender;
```

	gender character varying (50)	avg_profit_per_transaction numeric
1	F	29.01
2	M	32.87

HACKTIV8

Which product has the **highest return rate**?

The product with the highest return rate is 7 For All Mankind Women's The Skinny Jean, with a return rate of 46.67%, indicating potential issues with fit, quality, or customer expectations.

```
SELECT
    dp.name AS product_name,
    COUNT(ft.transaction_id) AS total_transactions,
    SUM(CASE WHEN ft.is_returned THEN 1 ELSE 0 END) AS returned_transactions,
    ROUND(SUM(CASE WHEN ft.is_returned THEN 1 ELSE 0 END) * 100.0 / COUNT(ft.transaction_id), 2)
    AS return_rate_percentage
FROM fact_transaction ft
JOIN dim_product dp ON ft.product_id = dp.product_key
GROUP BY dp.name
HAVING COUNT(ft.transaction_id) > 10
ORDER BY return_rate_percentage DESC
LIMIT 5;
```

	product_name character varying (255)	total_transactions bigint	returned_transactions bigint	return_rate_percentage numeric
1	7 For All Mankind Women's The Skinny Jean	15	7	46.67
2	Calvin Klein Sportswear Men's Dobby Two Tone Dylan Pant	13	6	46.15
3	Carhartt Men's Waterproof Breathable Acadia Pant	13	6	46.15
4	Hanes Sport Women's No Show Socks 6 Pack # 418/6	11	5	45.45
5	Mango Women's Velvet Short Jumpsuit - Ginette	14	6	42.86

THANK YOU

COMPREHENSIVE DATA ANALYTICS PROGRAM

amandarizkipati@gmail.com

❯ End Slide