Equipment Store - SQL EDA Project

Overview

This documents a complete SQL-based exploratory analysis of an equipment store's database, covering customer demographics, product catalog, and sales .transactions from 2010-2014

EDA Approach

- Database Exploration
- Dimension Exploration
- Date Exploration
- Measure Exploration
- Magnitude Exploration
- Ranking Exploration

Database Structure

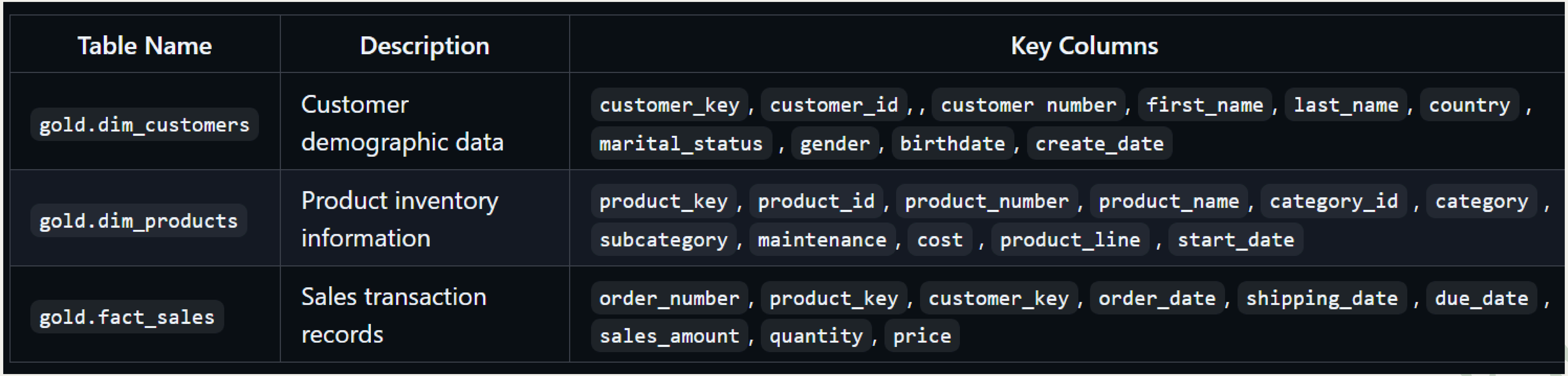


Table Name	Description	Key Columns
`gold.dim_customers`	Customer demographic data	`customer_key`, `customer_id`, , `customer number`, `first_name`, `last_name`, `country` , `marital_status` , `gender`, `birthdate`, `create_date`
`gold.dim_products`	Product inventory information	`product_key`, `product_id`, `product_number`, `product_name`, `category_id` , `category` , `subcategory`, `maintenance`, `cost` , `product_line` , `start_date`
`gold.fact_sales`	Sales transaction records	`order_number`, `product_key`, `customer_key`, `order_date`, `shipping_date` , `due_date` , `sales_amount`, `quantity`, `price`

Table idefintied

1. dim_customers : 18,484 records
2. dim_products : 295 records
3. fact_sales : 27,659 orders

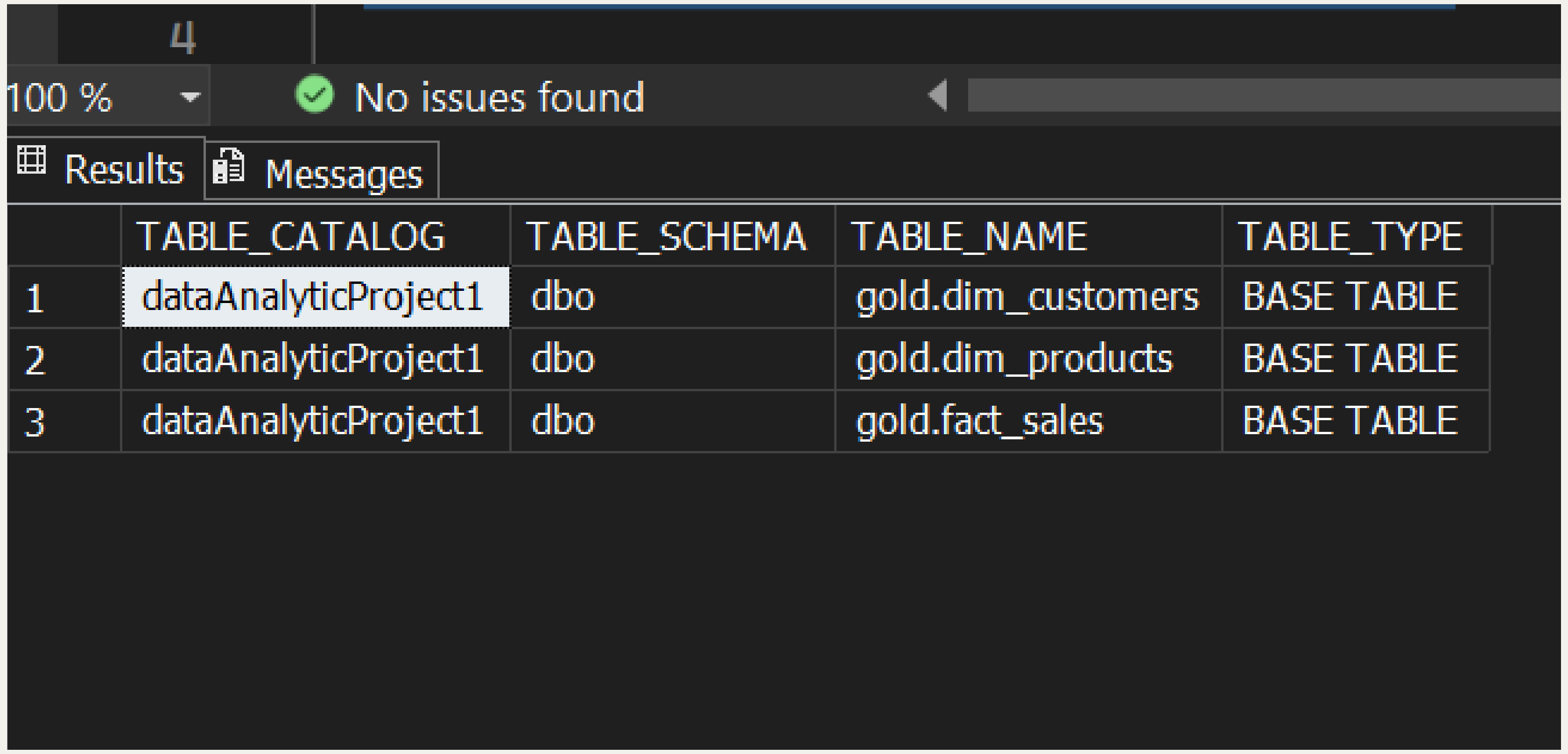
Database Exploration

1.Explore all object in database

input

```
SELECT * FROM
INFORMATION_SCH
EMA.TABLES;
```

output



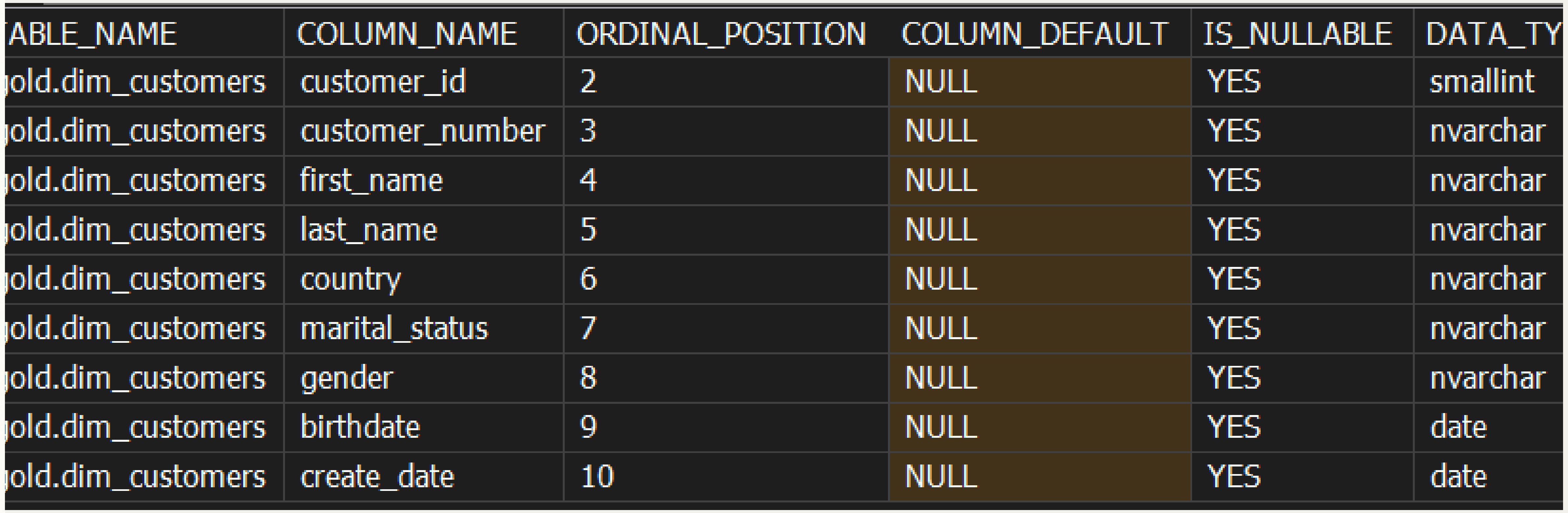
	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	dataAnalyticProject1	dbo	gold.dim_customers	BASE TABLE
2	dataAnalyticProject1	dbo	gold.dim_products	BASE TABLE
3	dataAnalyticProject1	dbo	gold.fact_sales	BASE TABLE

2. Explore culomns in database ‘customer’

input

```
Select * from
INFORMATION_SCHEMA.COLUMNS
WHERE
TABLE_NAME =
'gold.dim_customers'
```

output



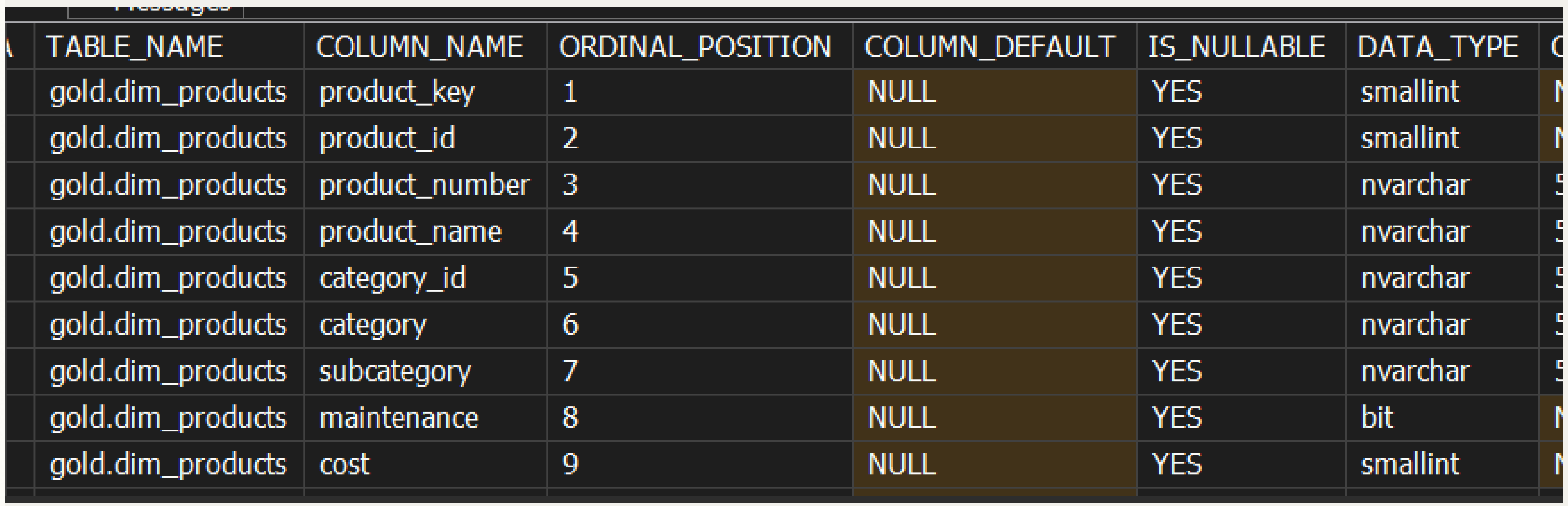
ABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TY
old.dim_customers	customer_id	2	NULL	YES	smallint
old.dim_customers	customer_number	3	NULL	YES	nvarchar
old.dim_customers	first_name	4	NULL	YES	nvarchar
old.dim_customers	last_name	5	NULL	YES	nvarchar
old.dim_customers	country	6	NULL	YES	nvarchar
old.dim_customers	marital_status	7	NULL	YES	nvarchar
old.dim_customers	gender	8	NULL	YES	nvarchar
old.dim_customers	birthdate	9	NULL	YES	date
old.dim_customers	create_date	10	NULL	YES	date

3. *Explore culomns in database 'product'*

input

```
Select * from
INFORMATION_SCHEMA.COLUMNS
WHERE
TABLE_NAME =
'gold.dim_products';
```

output



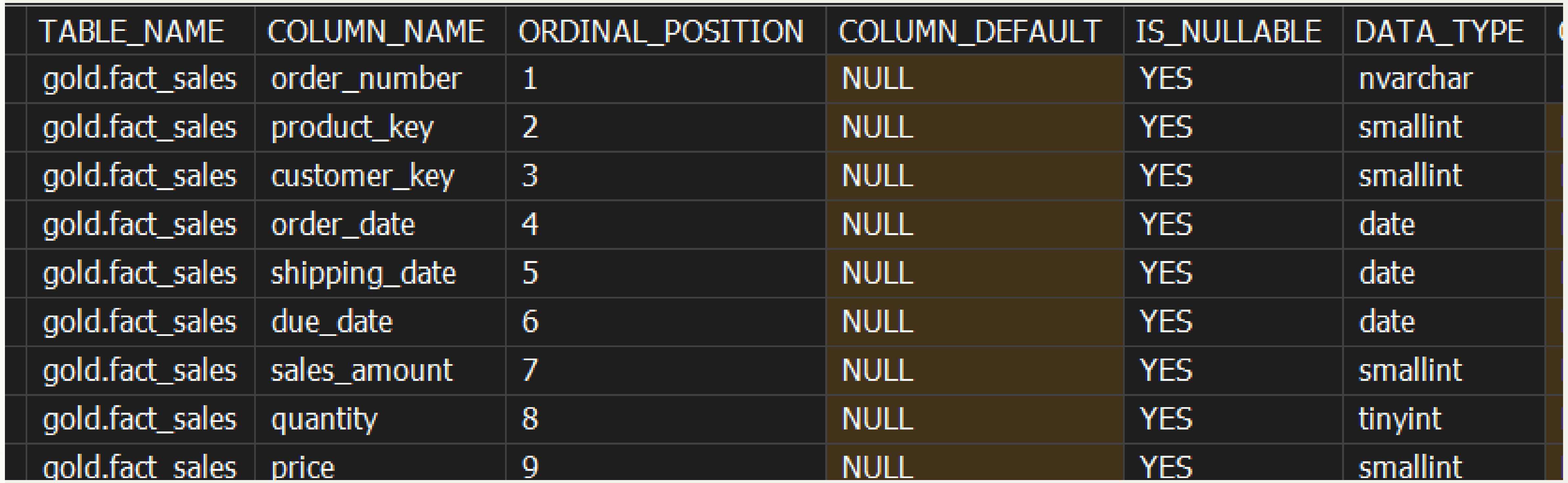
TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	C
gold.dim_products	product_key	1	NULL	YES	smallint	N
gold.dim_products	product_id	2	NULL	YES	smallint	N
gold.dim_products	product_number	3	NULL	YES	nvarchar	5
gold.dim_products	product_name	4	NULL	YES	nvarchar	5
gold.dim_products	category_id	5	NULL	YES	nvarchar	5
gold.dim_products	category	6	NULL	YES	nvarchar	5
gold.dim_products	subcategory	7	NULL	YES	nvarchar	5
gold.dim_products	maintenance	8	NULL	YES	bit	N
gold.dim_products	cost	9	NULL	YES	smallint	N

4. Explore culomns in database ‘sales’

input

```
Select * from
INFORMATION_SCHEMA.COLUMNS
WHERE
TABLE_NAME =
'gold.fact_sales';
```

output



TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE
gold.fact_sales	order_number	1	NULL	YES	nvarchar
gold.fact_sales	product_key	2	NULL	YES	smallint
gold.fact_sales	customer_key	3	NULL	YES	smallint
gold.fact_sales	order_date	4	NULL	YES	date
gold.fact_sales	shipping_date	5	NULL	YES	date
gold.fact_sales	due_date	6	NULL	YES	date
gold.fact_sales	sales_amount	7	NULL	YES	smallint
gold.fact_sales	quantity	8	NULL	YES	tinyint
gold.fact_sales	price	9	NULL	YES	smallint

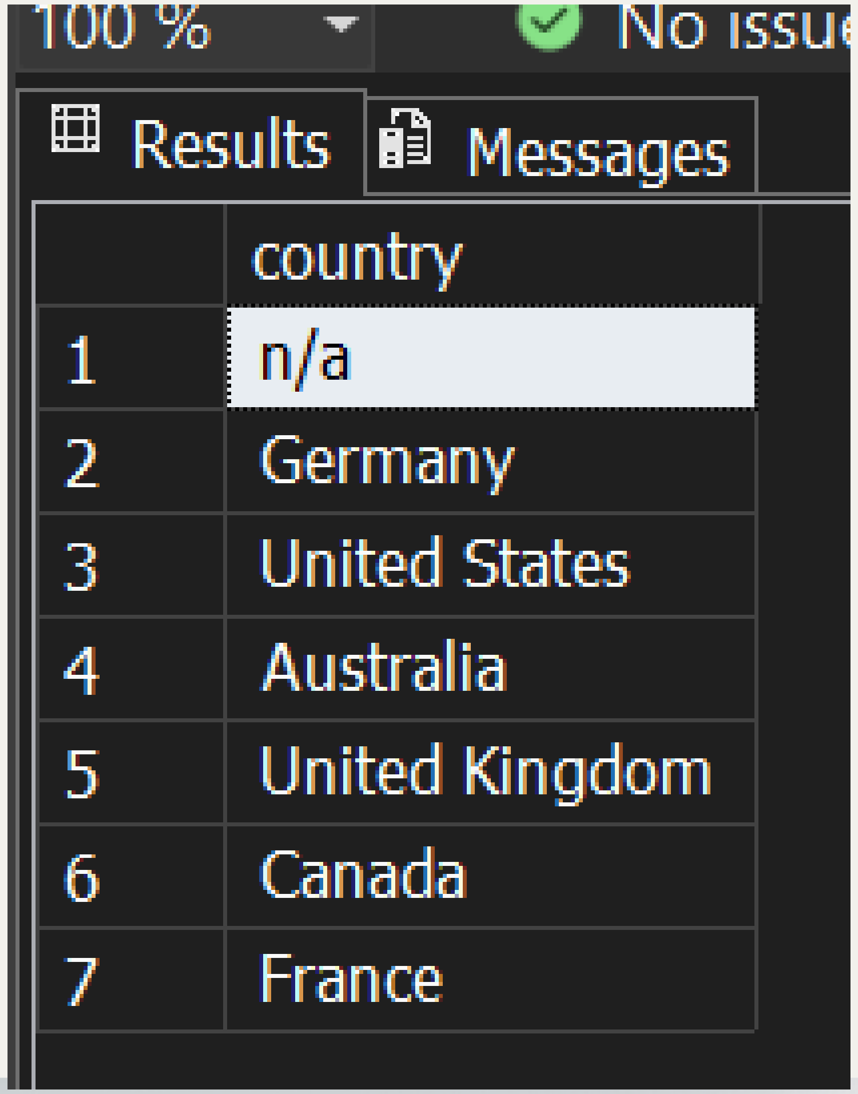
Dimension
Exploration

1. Explore all countries our customre come from

input

```
select DISTINCT
country from
[gold.dim_customer
s]
```

output



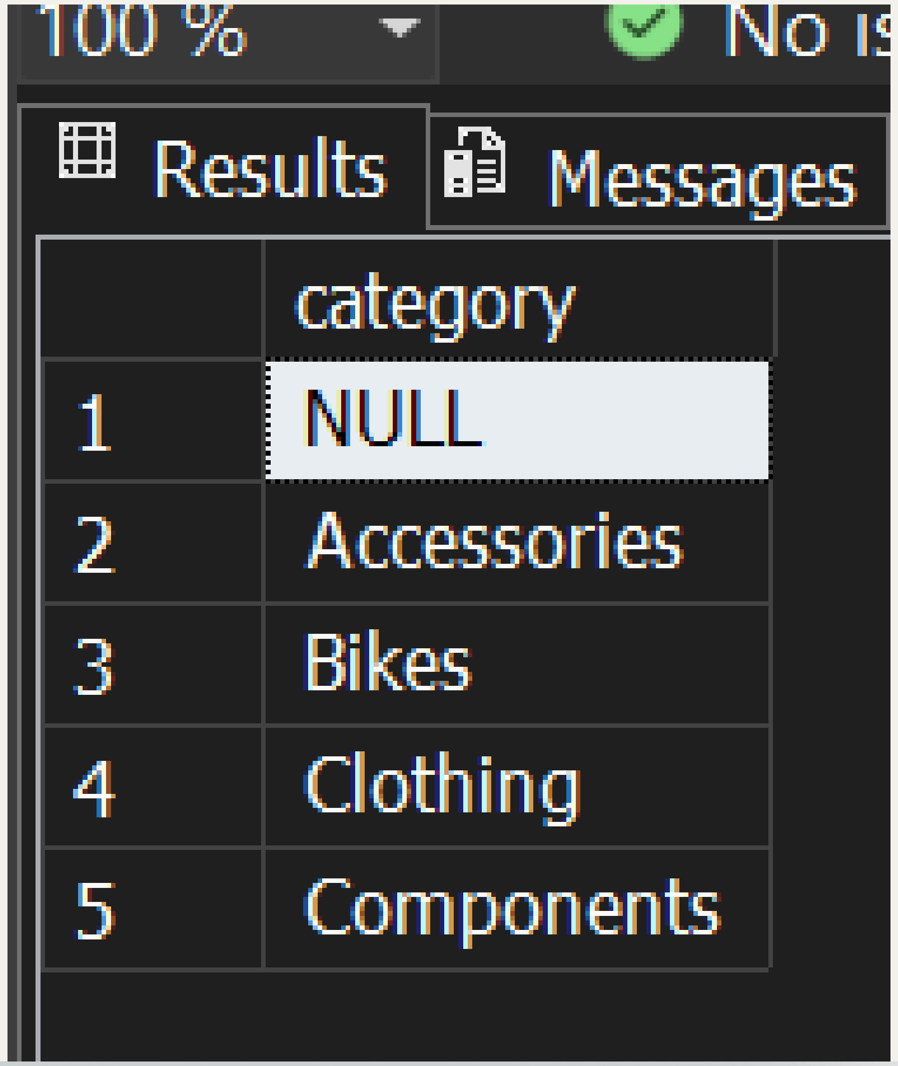
	country
1	n/a
2	Germany
3	United States
4	Australia
5	United Kingdom
6	Canada
7	France

2. explore all categories 'the major Divisions ‘category’

input

```
select DISTINCT
category   from
[gold.dim_products]
```

output



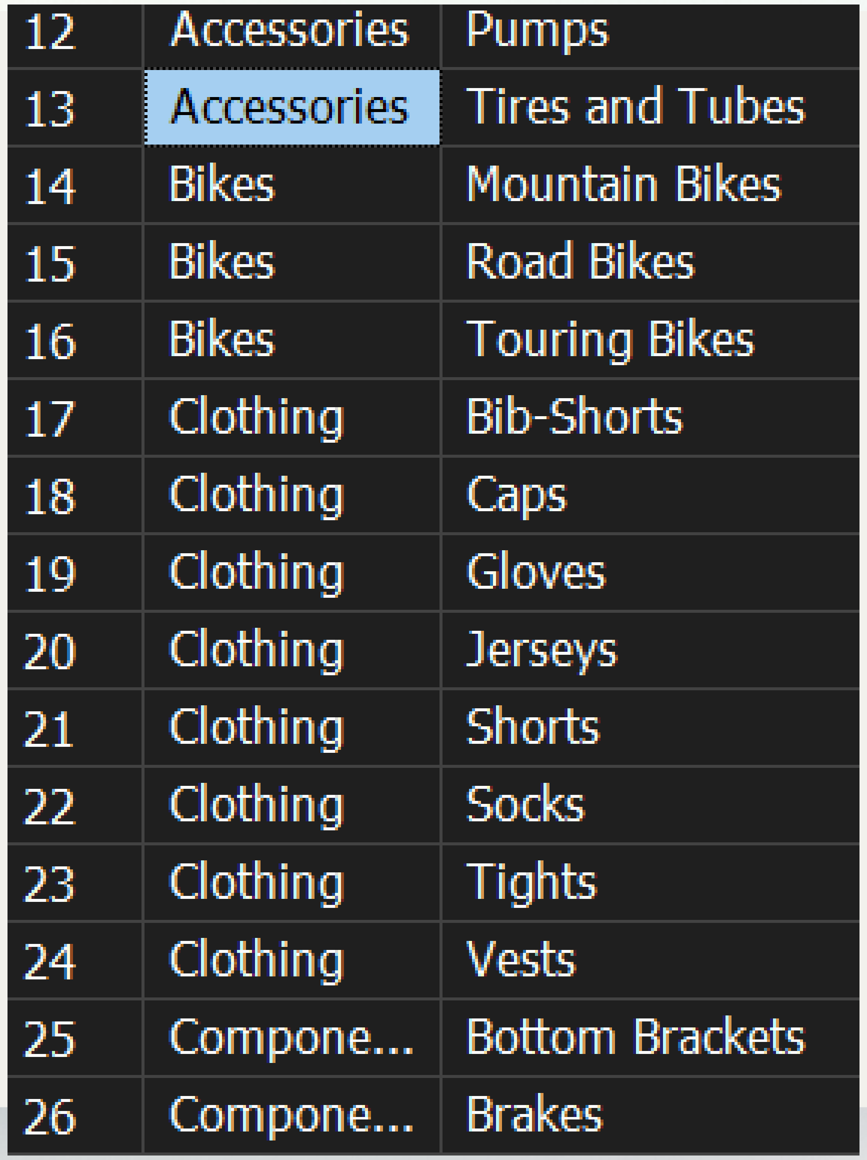
	category
1	NULL
2	Accessories
3	Bikes
4	Clothing
5	Components

3. explore all categories 'the major Divisions ‘subgategory’

input

```
select DISTINCT
category  ,
subcategory  from
[gold.dim_products]
```

output



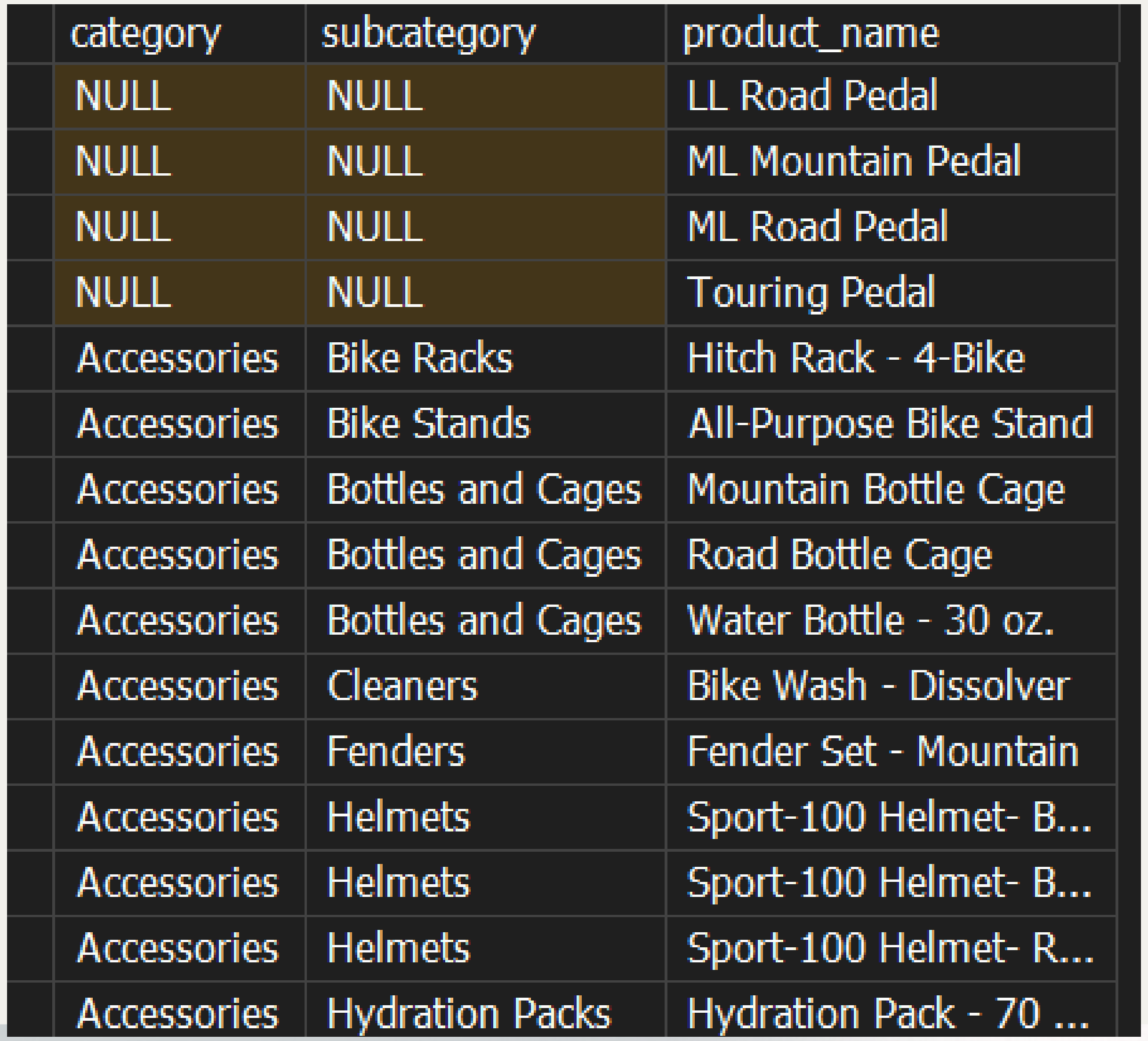
12	Accessories	Pumps
13	Accessories	Tires and Tubes
14	Bikes	Mountain Bikes
15	Bikes	Road Bikes
16	Bikes	Touring Bikes
17	Clothing	Bib-Shorts
18	Clothing	Caps
19	Clothing	Gloves
20	Clothing	Jerseys
21	Clothing	Shorts
22	Clothing	Socks
23	Clothing	Tights
24	Clothing	Vests
25	Compone...	Bottom Brackets
26	Compone...	Brakes

4. explore all categories 'the major Divisions ‘product name’

input

```
select DISTINCT
category   ,
subcategory ,
product_name from
[gold.dim_products]
;
```

output



category	subcategory	product_name
NULL	NULL	LL Road Pedal
NULL	NULL	ML Mountain Pedal
NULL	NULL	ML Road Pedal
NULL	NULL	Touring Pedal
Accessories	Bike Racks	Hitch Rack - 4-Bike
Accessories	Bike Stands	All-Purpose Bike Stand
Accessories	Bottles and Cages	Mountain Bottle Cage
Accessories	Bottles and Cages	Road Bottle Cage
Accessories	Bottles and Cages	Water Bottle - 30 oz.
Accessories	Cleaners	Bike Wash - Dissolver
Accessories	Fenders	Fender Set - Mountain
Accessories	Helmets	Sport-100 Helmet- B...
Accessories	Helmets	Sport-100 Helmet- B...
Accessories	Helmets	Sport-100 Helmet- R...
Accessories	Hydration Packs	Hydration Pack - 70 ...

Date

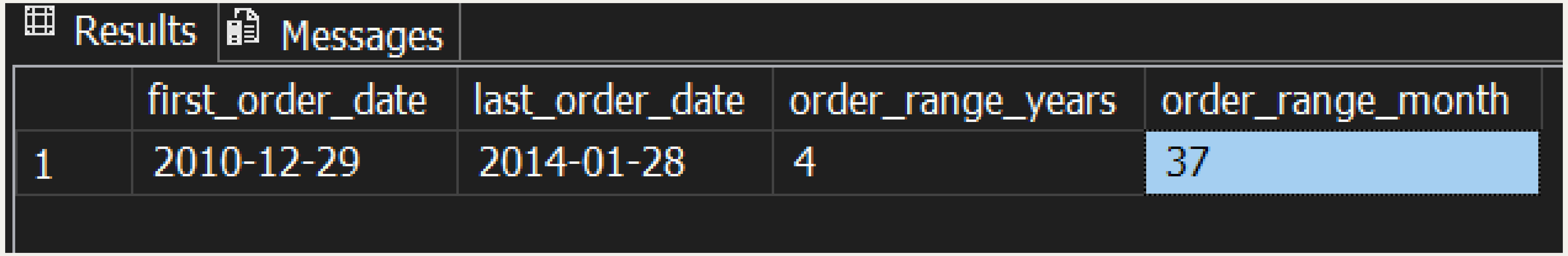
Exploration

1. The date of the first and last order and month , years of sales are available

input

```
select min(order_date) as
first_order_date ,
max(order_date)as
last_order_date  ,
DATEDIFF(year , min(order_date)
, max(order_date)) as
order_range_years,
DATEDIFF(month ,
min(order_date) ,
max(order_date)) as
order_range_month
from [gold.fact_sales]
```

output



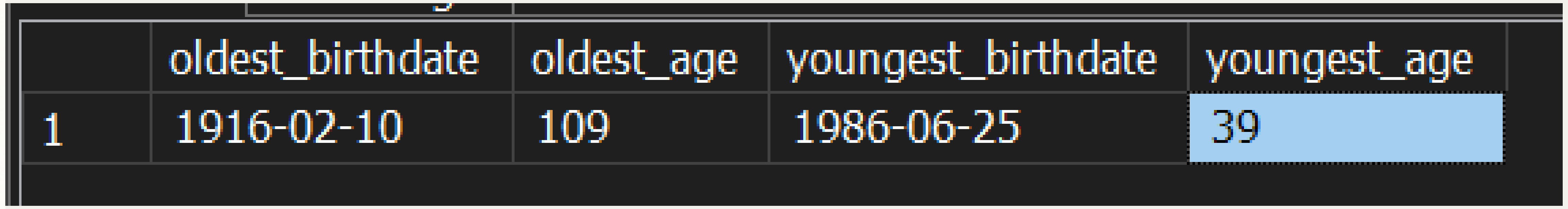
	first_order_date	last_order_date	order_range_years	order_range_month
1	2010-12-29	2014-01-28	4	37

2. youngest and oldest customres

input

```
select min(birthdate) as
oldest_birthdate,
datediff(year , min(birthdate) ,
getdate()) as oldest_age,
max(birthdate) as
youngest_birthdate,
datediff(year , max(birthdate) ,
getdate()) as youngest_age
from [gold.dim_customers]
```

output



	oldest_birthdate	oldest_age	youngest_birthdate	youngest_age
1	1916-02-10	109	1986-06-25	39

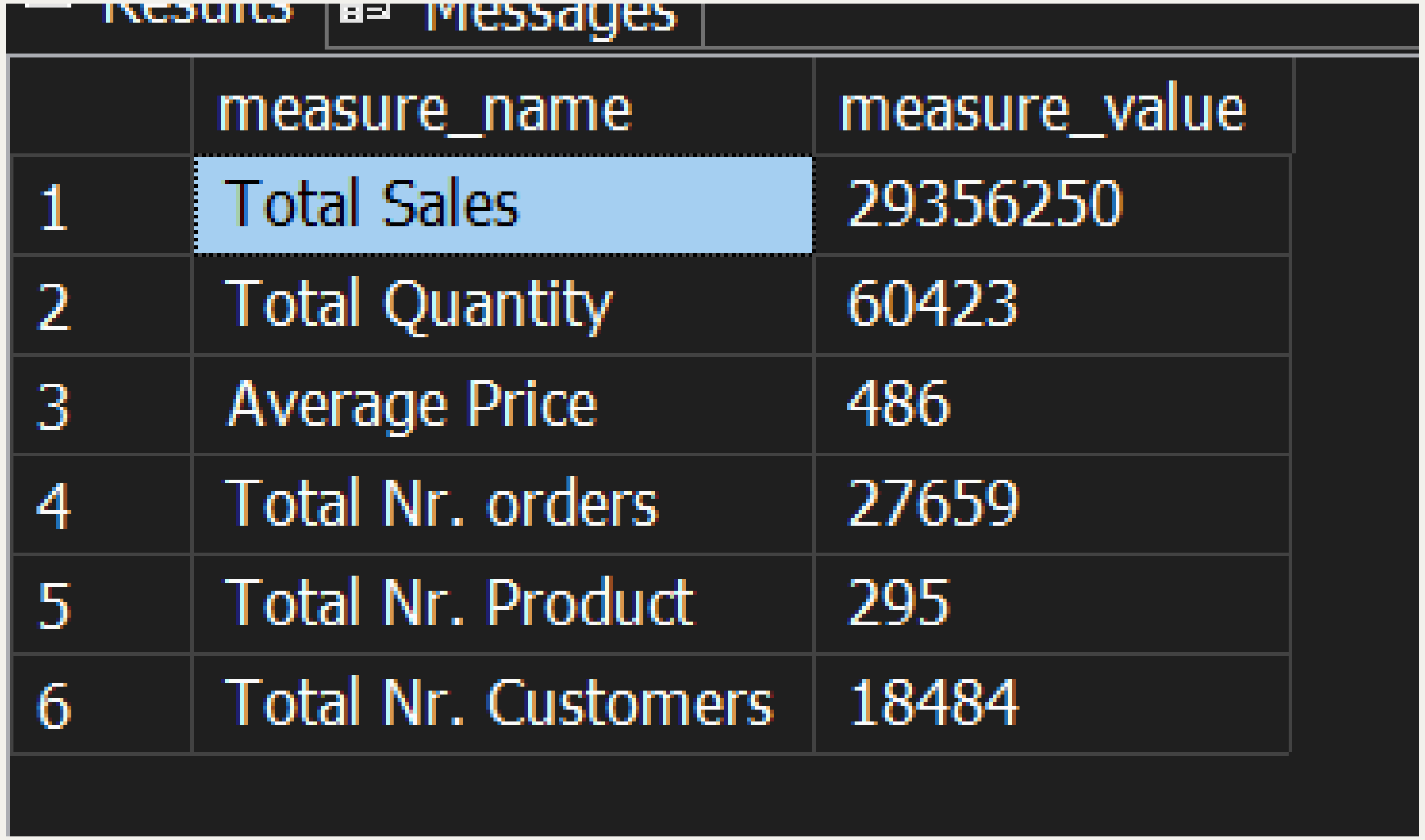
Measure
Exploration

mertics of business

input

```
select 'Total Sales' as measure_name ,
sum(sales_amount) as measure_value from
[gold.fact_sales]
union all
select'Total Quantity' as measure_name, sum(quantity)
as measure_value from [gold.fact_sales]
union all
select 'Average Price'  as measure_name , avg(price) as
measure_value from [gold.fact_sales]
union all
select'Total Nr. orders', count(DISTINCT order_number)
as measure_value from [gold.fact_sales]
union all
select 'Total Nr. Product ' ,count (product_key)as
measure_value from [gold.dim_products]
union all
select 'Total Nr. Customers' , count(DISTINCT
customer_key)as measure_value from [gold.fact_sales]
```

output



	measure_name	measure_value
1	Total Sales	29356250
2	Total Quantity	60423
3	Average Price	486
4	Total Nr. orders	27659
5	Total Nr. Product	295
6	Total Nr. Customers	18484

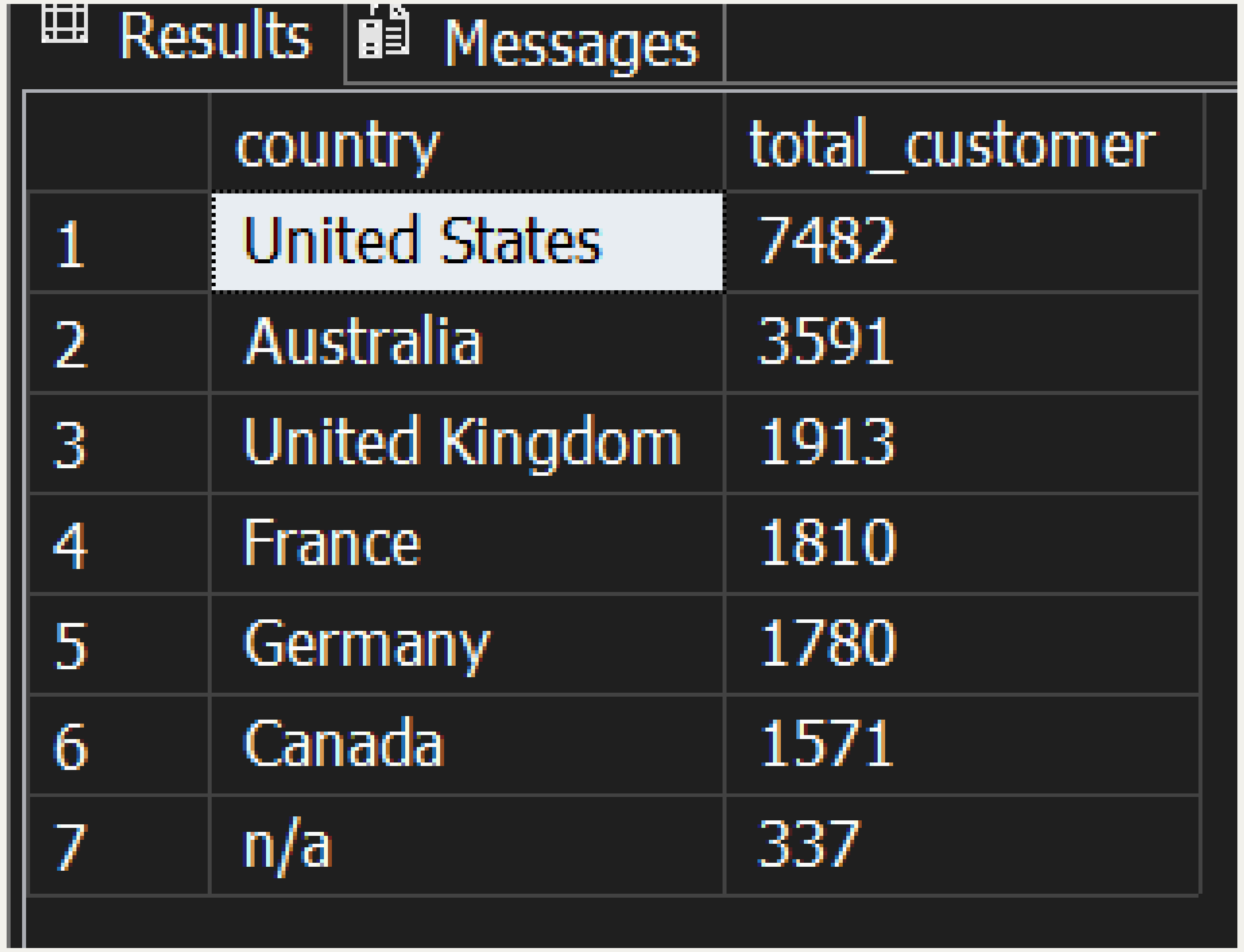
Measure
Exploration

1.total customer by countries

input

```
select country,
count(customer_key) as
total_customer from
[gold.dim_customers]
group by country
order by total_customer DESC
```

output



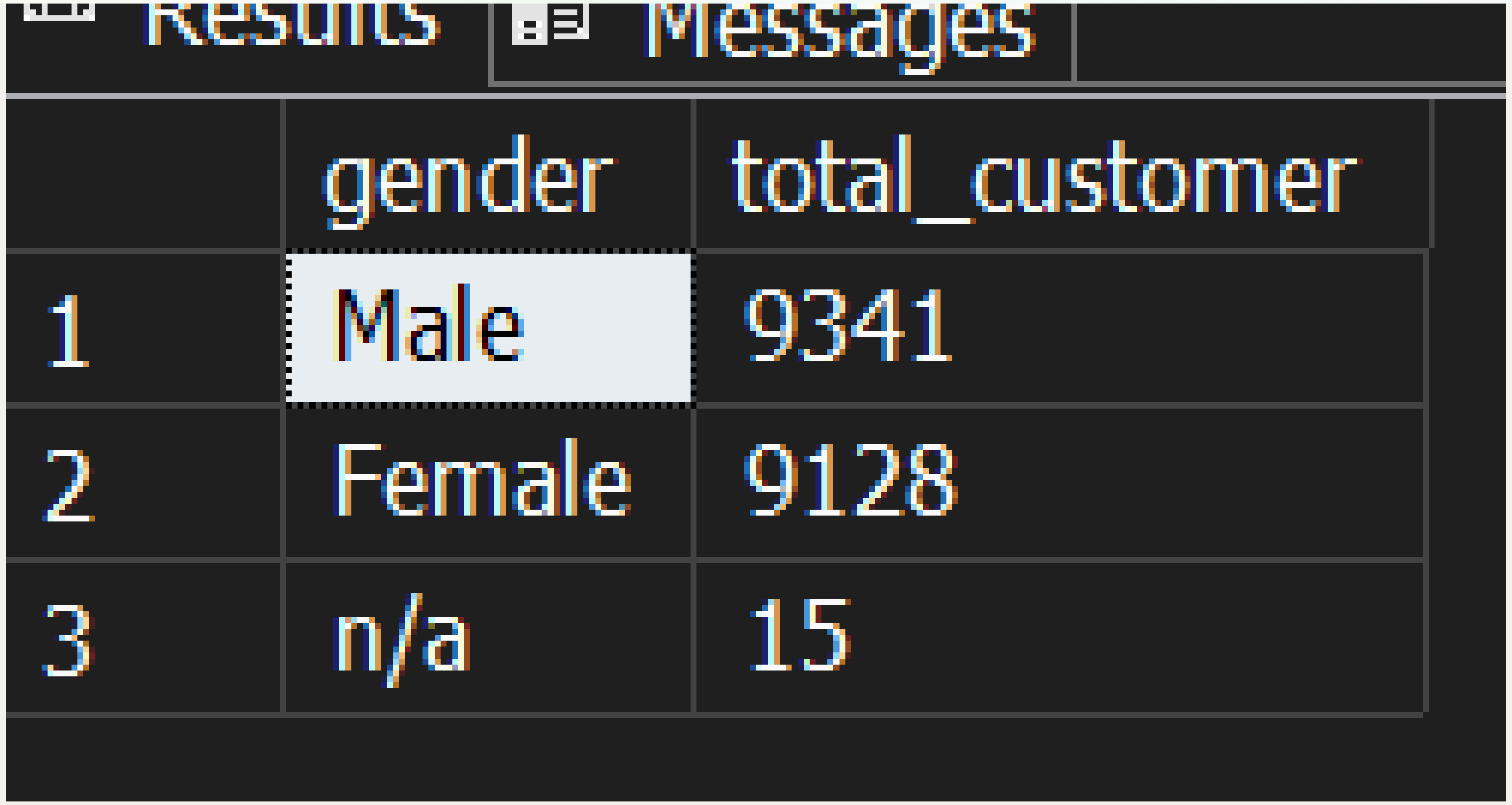
	country	total_customer
1	United States	7482
2	Australia	3591
3	United Kingdom	1913
4	France	1810
5	Germany	1780
6	Canada	1571
7	n/a	337

2. total customer by gender

input

```
select gender,
count(customer_key) as
total_customer from
[gold.dim_customers]
group by gender
order by total_customer DESC
```

output



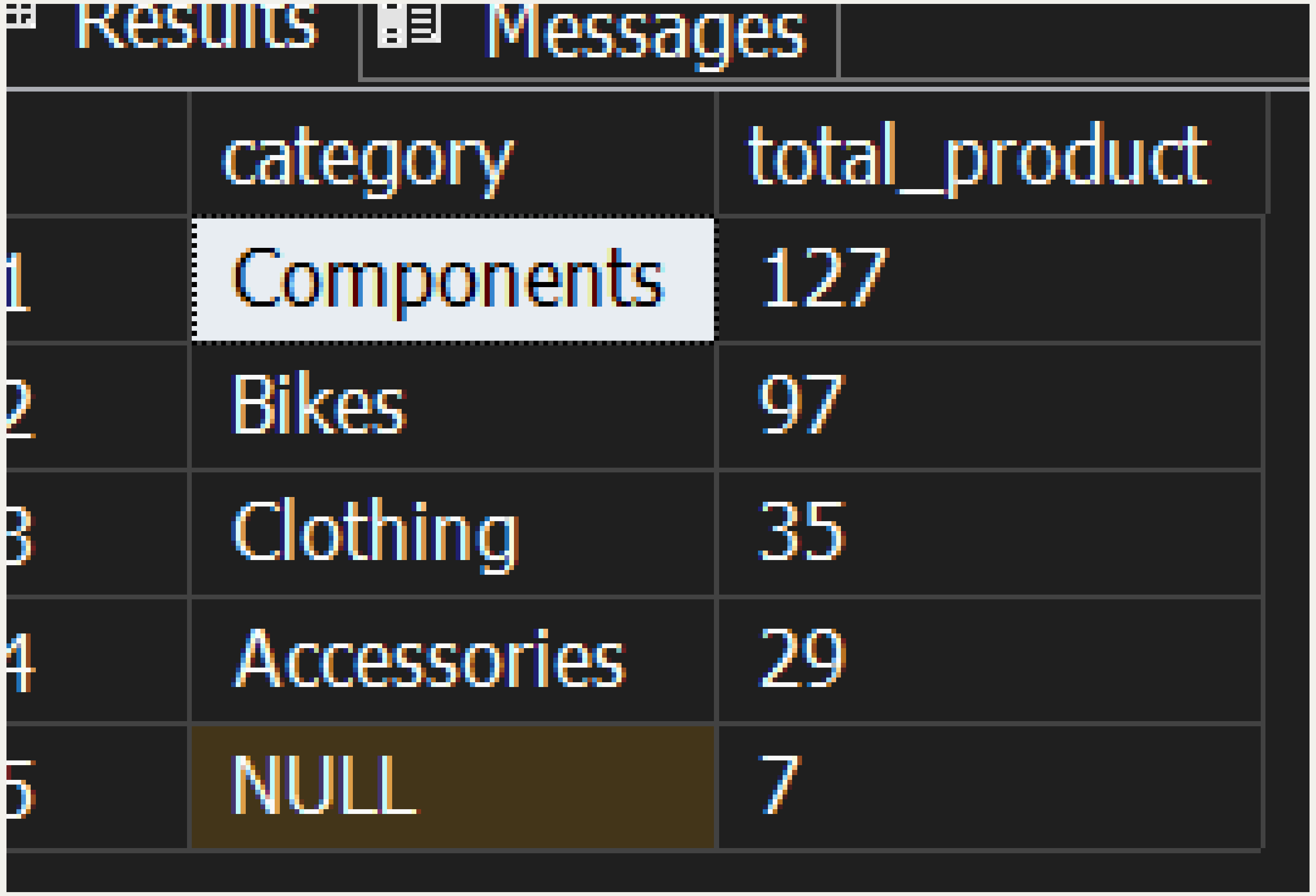
	gender	total_customer
1	Male	9341
2	Female	9128
3	n/a	15

3.total product by category

input

```
select category  ,
count(product_key) as
total_product
from [gold.dim_products]
group by category
order by total_product DESC
```

output



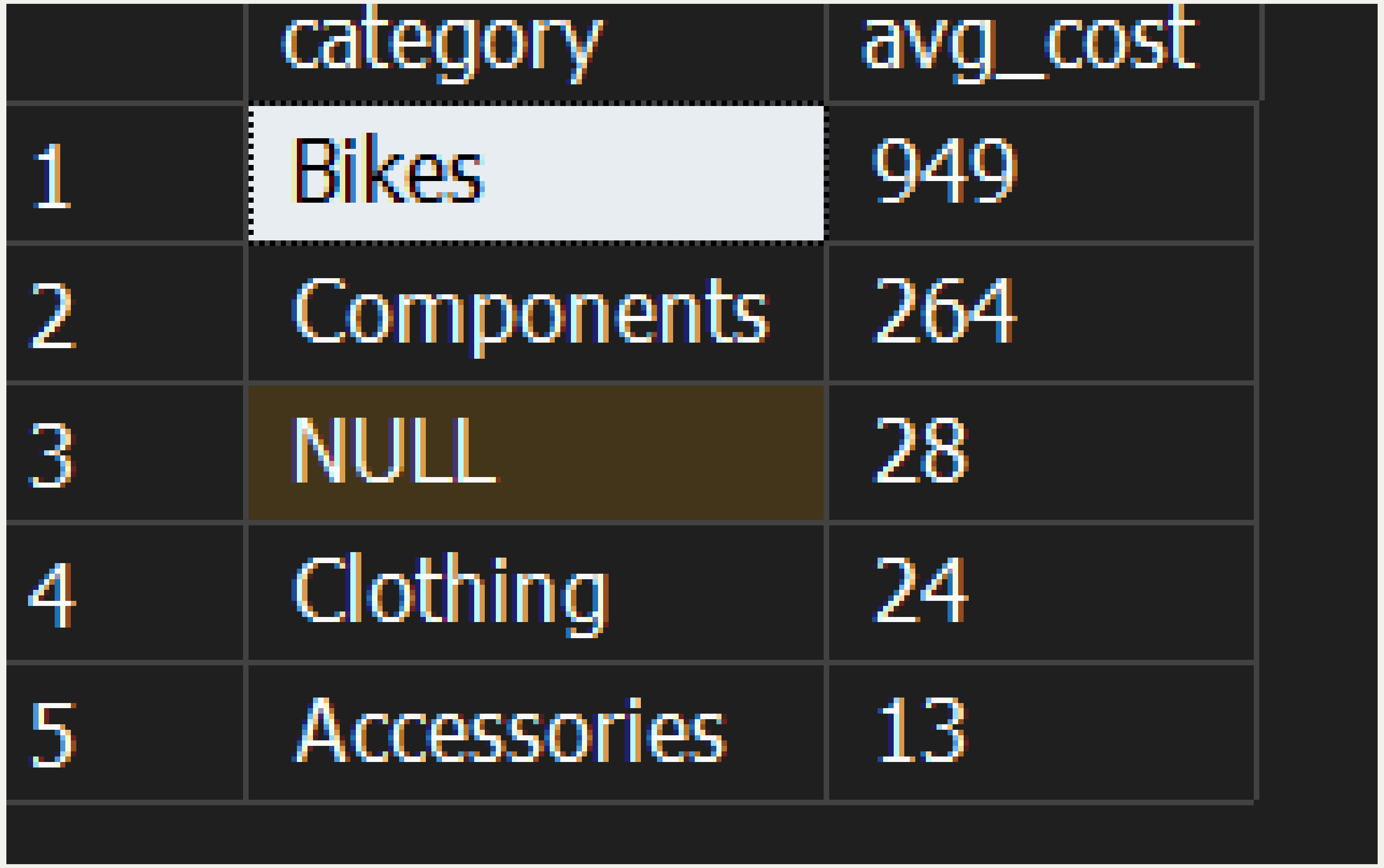
	category	total_product
1	Components	127
2	Bikes	97
3	Clothing	35
4	Accessories	29
5	NULL	7

4. average costs in each category

input

```
select category  ,avg(cost) as
avg_cost
from [gold.dim_products]
group by category
order by avg_cost DESC
```

output



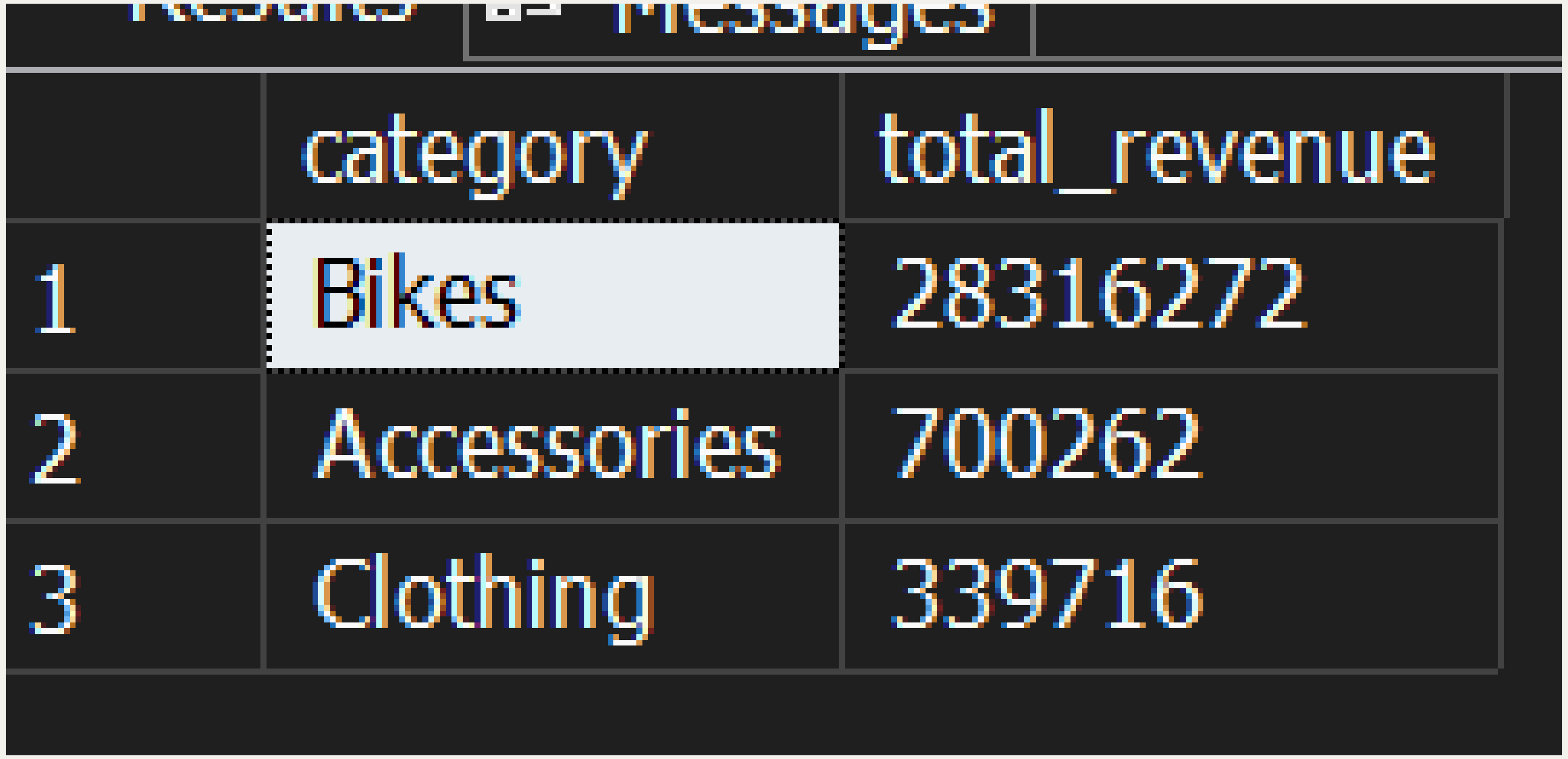
	category	avg_cost
1	Bikes	949
2	Components	264
3	NULL	28
4	Clothing	24
5	Accessories	13

5. *total revenue generated for eacg category*

input

```
select  p.category  ,
sum(f.sales_amount ) as
total_revenue
from [gold.fact_sales] f
left join [gold.dim_products] p
on p.product_key =
f.product_key
group by p.category
order by total_revenue DESC
```

output



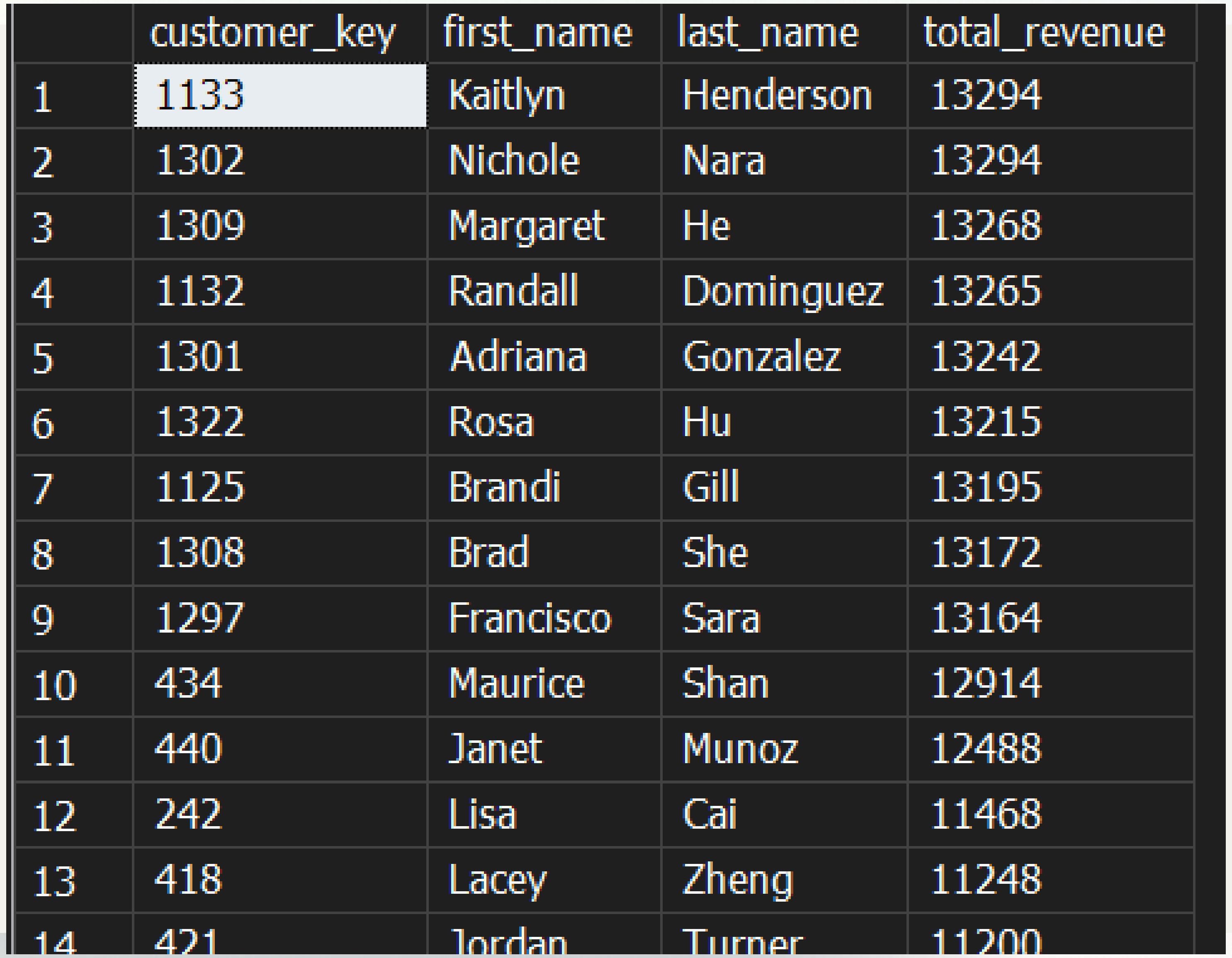
	category	total_revenue
1	Bikes	28316272
2	Accessories	700262
3	Clothing	339716

6. *total revneu is generated by each customer*

input

```
select c.customer_key ,
c.first_name , c.last_name , sum
(f.sales_amount) as
total_revenue
from [gold.fact_sales] f
left join [gold.dim_customers] c
on f.customer_key =
c.customer_key
group by c.customer_key ,
c.first_name , c.last_name
order by total_revenue DESC
```

output



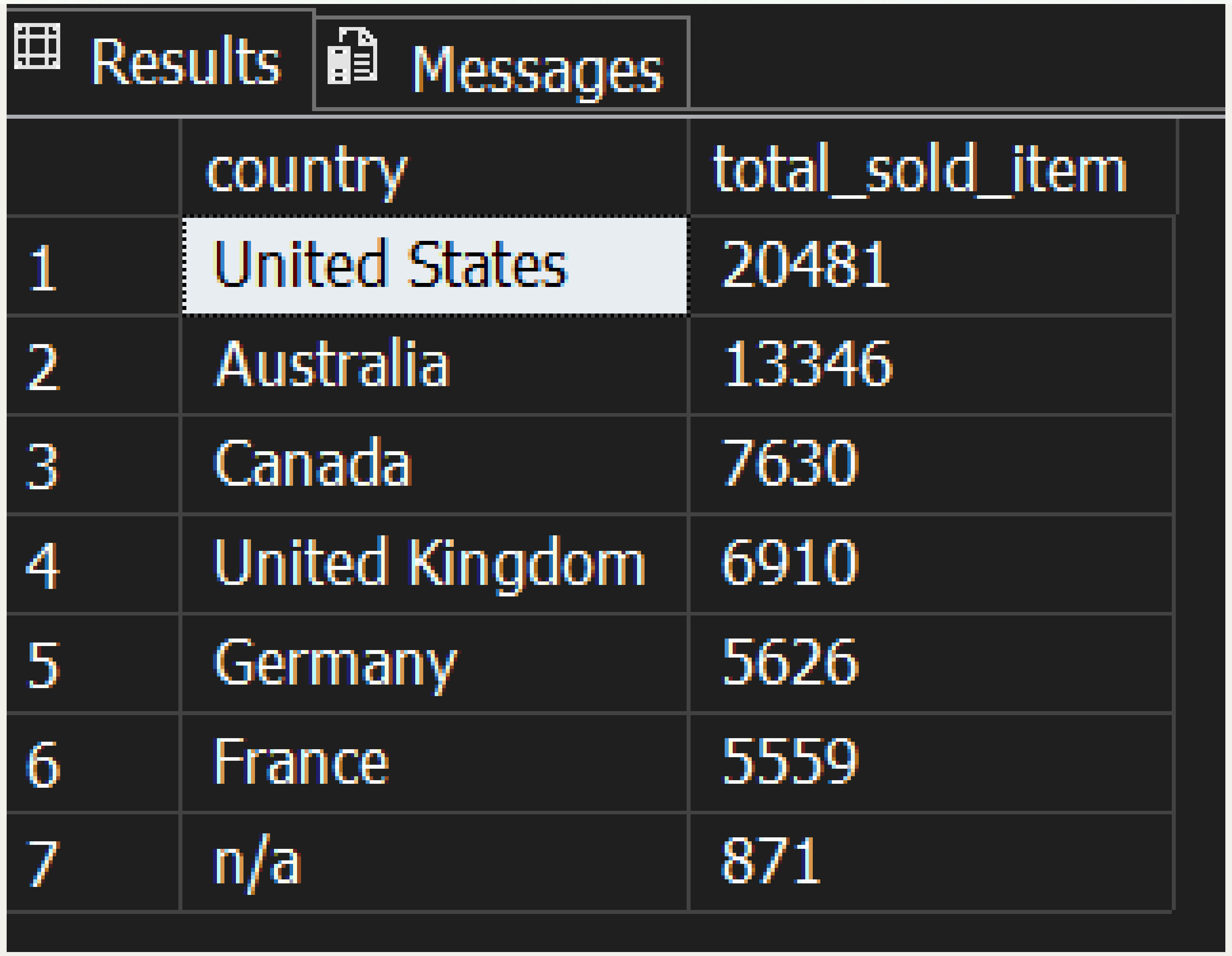
	customer_key	first_name	last_name	total_revenue
1	1133	Kaitlyn	Henderson	13294
2	1302	Nichole	Nara	13294
3	1309	Margaret	He	13268
4	1132	Randall	Dominguez	13265
5	1301	Adriana	Gonzalez	13242
6	1322	Rosa	Hu	13215
7	1125	Brandi	Gill	13195
8	1308	Brad	She	13172
9	1297	Francisco	Sara	13164
10	434	Maurice	Shan	12914
11	440	Janet	Munoz	12488
12	242	Lisa	Cai	11468
13	418	Lacey	Zheng	11248
14	421	Jordan	Turner	11200

7.distribution of sold item across countries

input

```
select c.country , sum
(f.quantity) as total_sold_item
from [gold.fact_sales] f
left join [gold.dim_customers] c
on f.customer_key =
c.customer_key
group by c.country
order by total_sold_item DESC
```

output



	country	total_sold_item
1	United States	20481
2	Australia	13346
3	Canada	7630
4	United Kingdom	6910
5	Germany	5626
6	France	5559
7	n/a	871

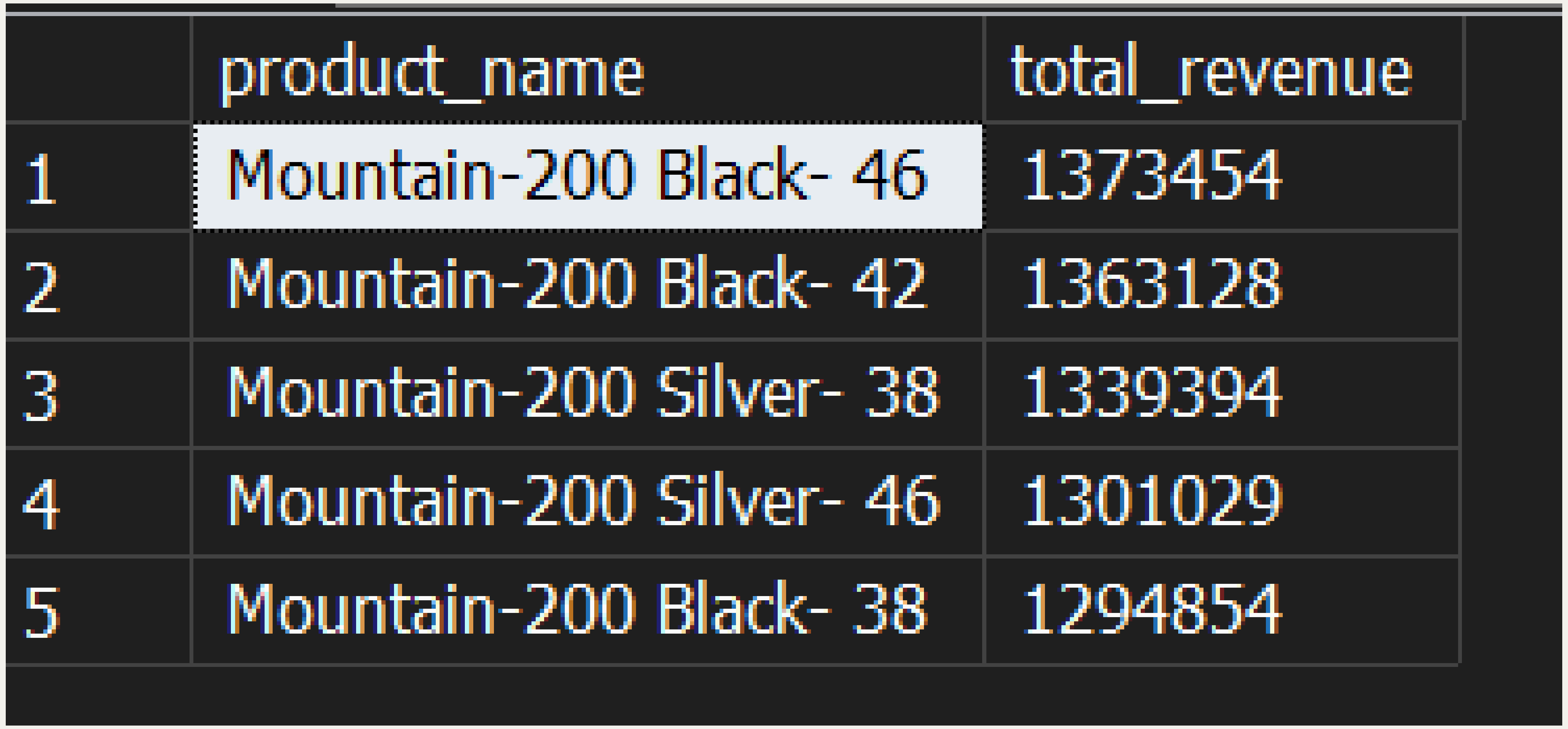
Ranking Exploration

1. the 5 product generate the highest revenue

input

```
select top 5 p.product_name ,
sum(f.sales_amount)total_revenue
from [gold.fact_sales] f
left join [gold.dim_products] p
on p.product_key =
f.product_key
group by p.product_name
order by total_revenue DESC
```

output



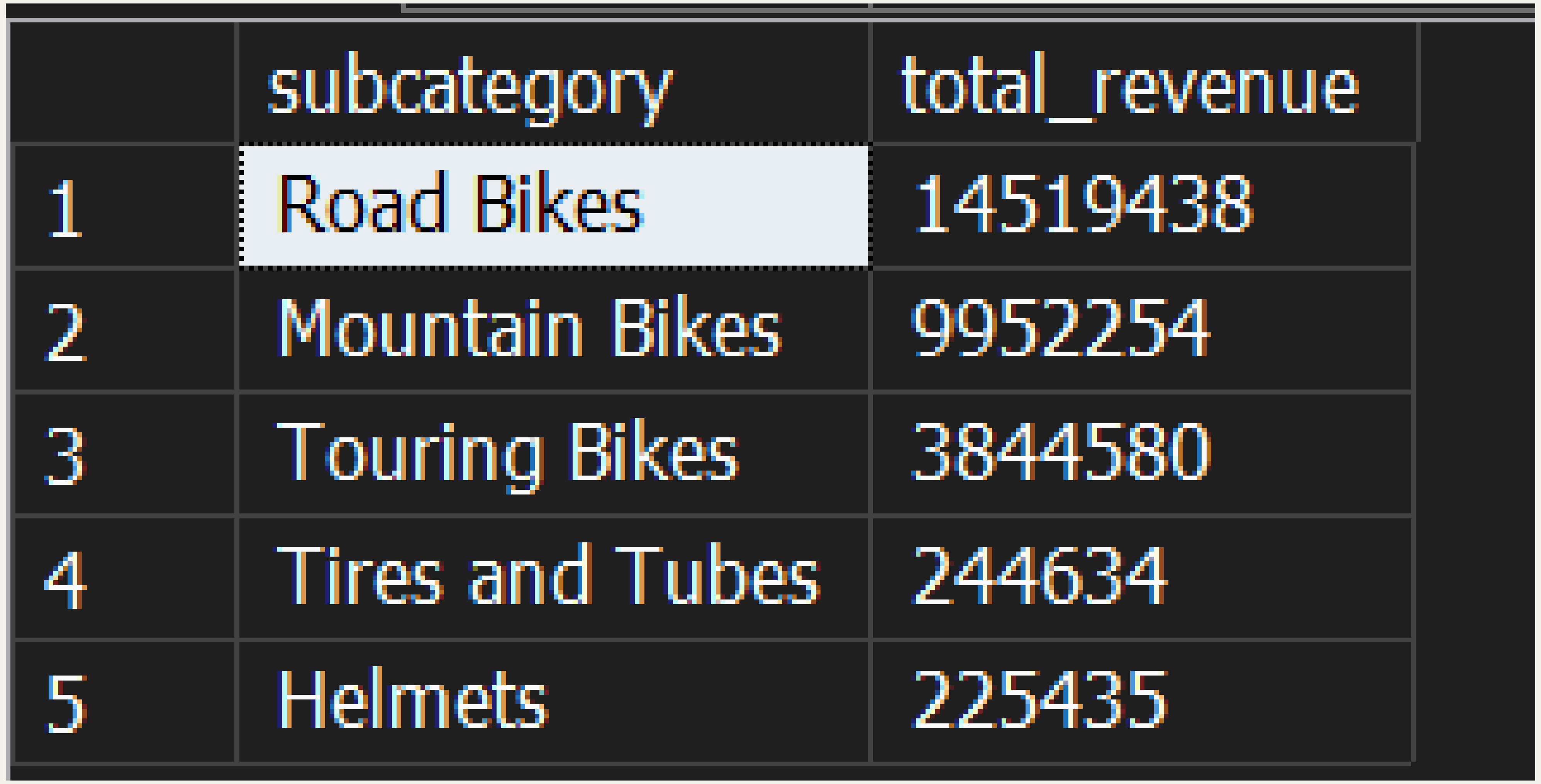
	product_name	total_revenue
1	Mountain-200 Black- 46	1373454
2	Mountain-200 Black- 42	1363128
3	Mountain-200 Silver- 38	1339394
4	Mountain-200 Silver- 46	1301029
5	Mountain-200 Black- 38	1294854

2. the 5 product generate the highest revenue subcategory

input

```
select top 5 p.subcategory ,
sum(f.sales_amount)total_revenue
from [gold.fact_sales] f
left join [gold.dim_products] p
on p.product_key = f.product_key
group by p.subcategory
order by total_revenue DESC
```

output



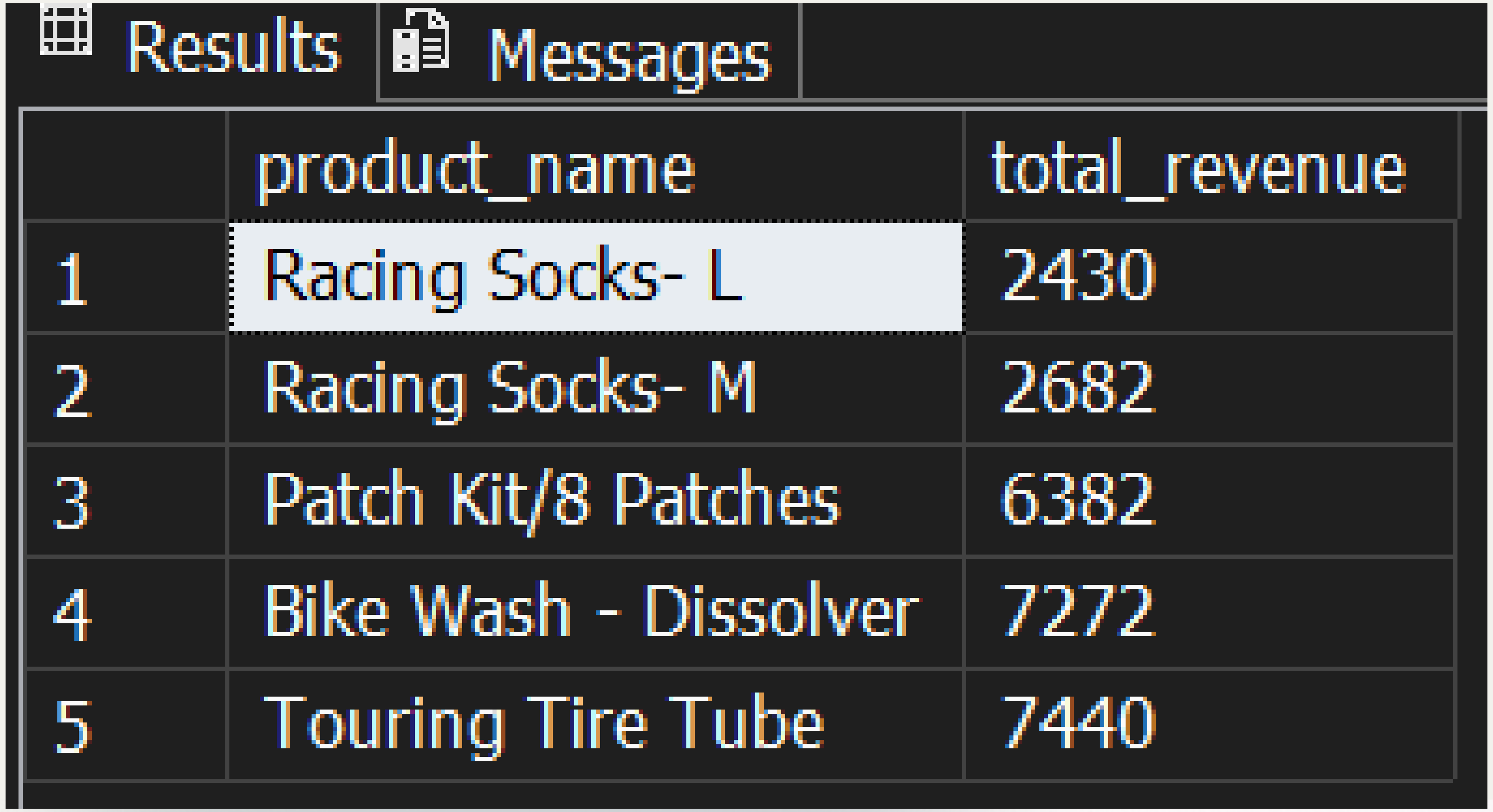
	subcategory	total_revenue
1	Road Bikes	14519438
2	Mountain Bikes	9952254
3	Touring Bikes	3844580
4	Tires and Tubes	244634
5	Helmets	225435

3. the 5 worst-perforaming product on terms of sales

input

```
select top 5 p.product_name ,
sum(f.sales_amount)total_revenue
from [gold.fact_sales] f
left join [gold.dim_products] p
on p.product_key =
f.product_key
group by p.product_name
order by total_revenue
```

output



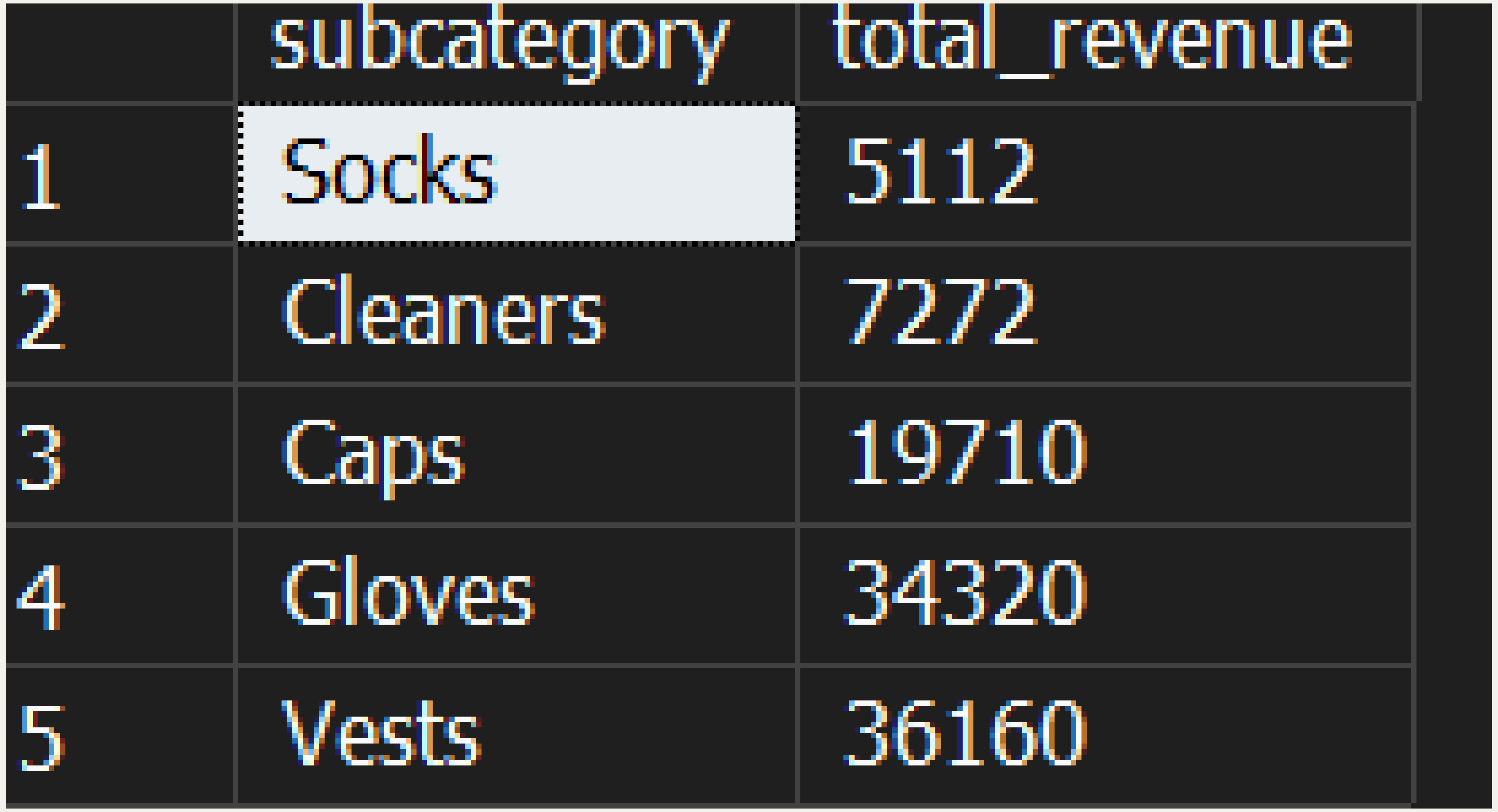
	product_name	total_revenue
1	Racing Socks- L	2430
2	Racing Socks- M	2682
3	Patch Kit/8 Patches	6382
4	Bike Wash - Dissolver	7272
5	Touring Tire Tube	7440

4. the 5 worst-perforaming product on terms of sales subcategory

input

```
select top 5 p.subcategory ,
sum(f.sales_amount)total_revenue
from [gold.fact_sales] f
left join [gold.dim_products] p
on p.product_key =
f.product_key
group by p.subcategory
order by total_revenue
```

output



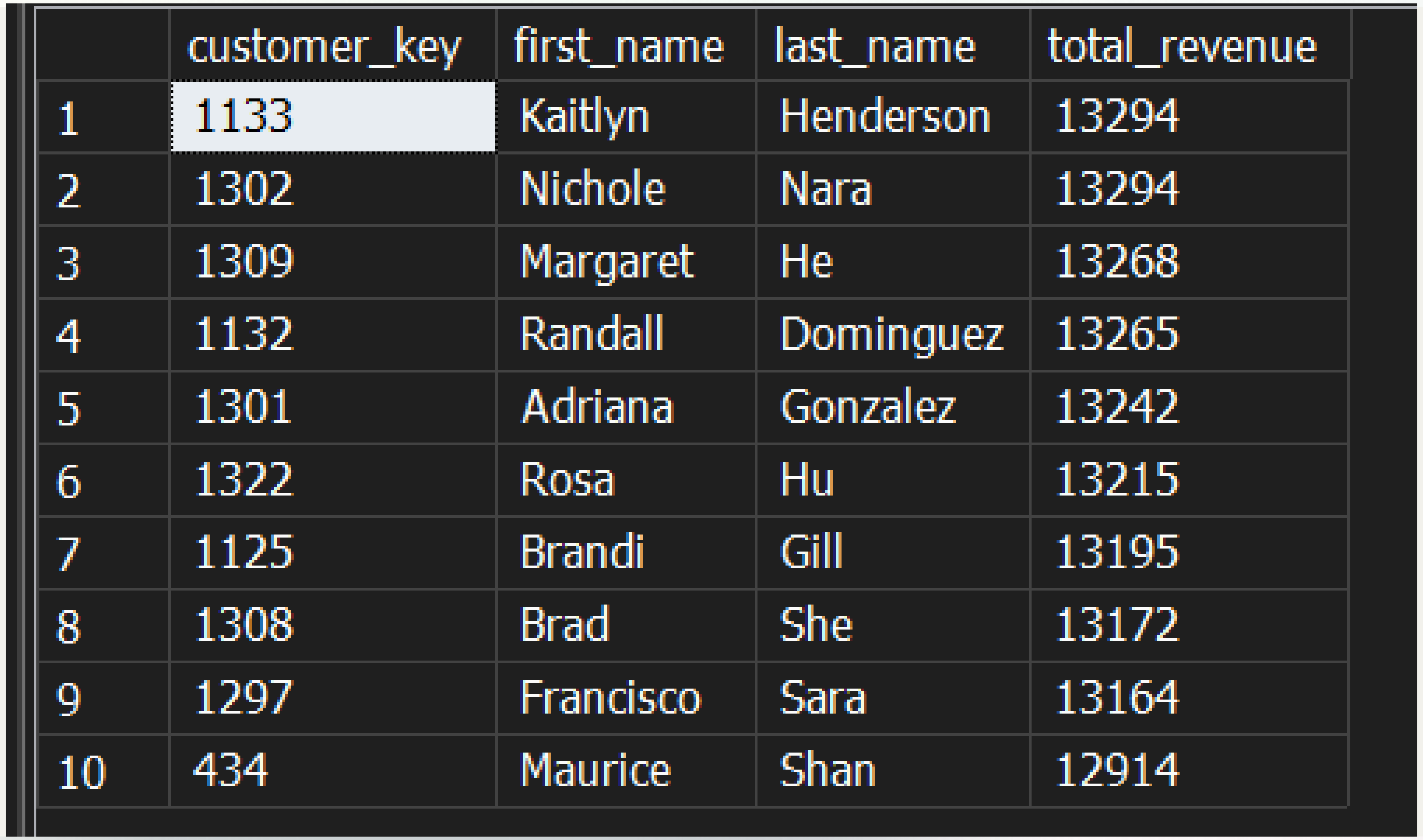
	subcategory	total_revenue
1	Socks	5112
2	Cleaners	7272
3	Caps	19710
4	Gloves	34320
5	Vests	36160

5 . top 10 customer who have generated the highest revune

input

```
select TOP 10 c.customer_key ,
c.first_name , c.last_name , sum
(f.sales_amount) as
total_revenue
from [gold.fact_sales] f
left join [gold.dim_customers]c
on f.customer_key =
c.customer_key
group by  c.customer_key ,
c.first_name , c.last_name
order by total_revenue DESC
```

output



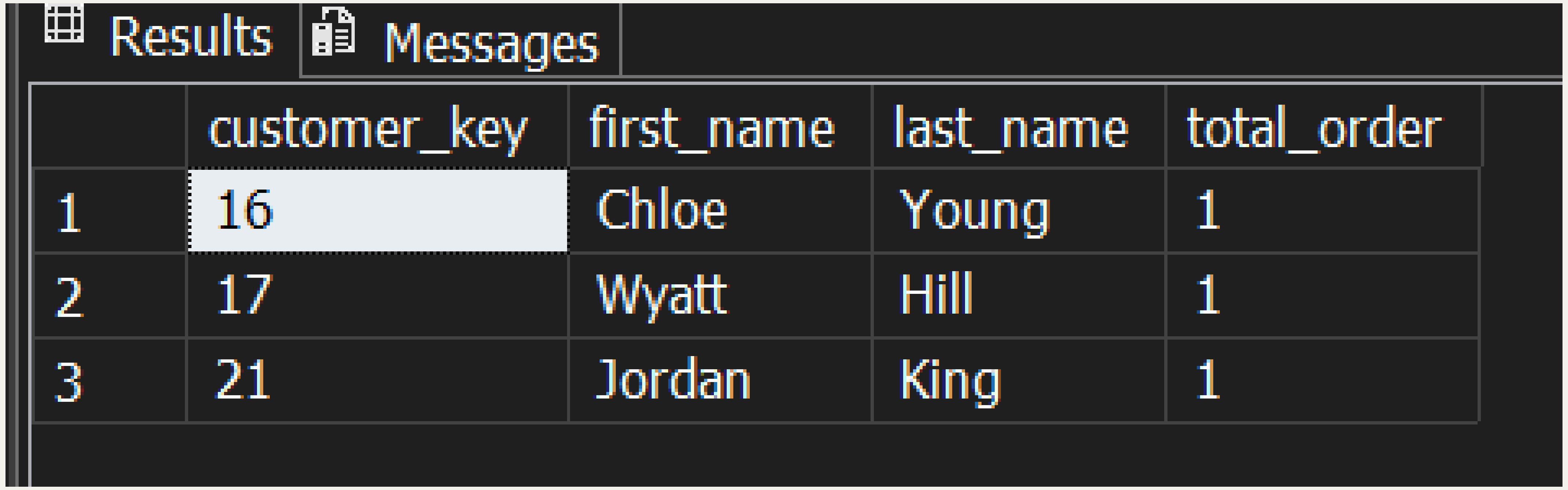
	customer_key	first_name	last_name	total_revenue
1	1133	Kaitlyn	Henderson	13294
2	1302	Nichole	Nara	13294
3	1309	Margaret	He	13268
4	1132	Randall	Dominguez	13265
5	1301	Adriana	Gonzalez	13242
6	1322	Rosa	Hu	13215
7	1125	Brandi	Gill	13195
8	1308	Brad	She	13172
9	1297	Francisco	Sara	13164
10	434	Maurice	Shan	12914

6 .the 3 customers with the fewest orders placed

input

```
select TOP 3 c.customer_key ,
c.first_name , c.last_name ,
count(DISTINCT order_number)
as total_order
from [gold.fact_sales] f
left join [gold.dim_customers]c
on f.customer_key =
c.customer_key
group by  c.customer_key ,
c.first_name , c.last_name
order by total_order
```

output



	customer_key	first_name	last_name	total_order
1	16	Chloe	Young	1
2	17	Wyatt	Hill	1
3	21	Jordan	King	1

summary

Key Findings Summary

1. **Revenue & Sales**
   - Total revenue: $29.36M (2010-2014)
   - Bikes dominate (96.5% of revenue)
   - Top product: *Mountain-200 Silver* ($1.39M)
   - Strong Q2 seasonality

2. **Customers**
   - 18,484 active buyers (40.5% from US)
   - Gender split: Male (50.5%), Female (49.4%)
   - Age range: 39–109 years (potential outliers)

3. **Products**
   - 295 SKUs across 4 categories
   - Bikes have highest avg cost ($949)
   - 7 products missing category data

Data Quality Issues

1. 337 customers with n/a country
2. NULL values in product categories
3. Age outliers (>100 years)

Conclusion

- **Priority:** Target US males aged 40–60 (highest spenders).

337

Critical: Clean 337 incomplete customer records.

96.5%

Opportunity: Bike sales yield 96.5% revenue; optimize inventory.

Thank You

by rama albidani