```sql
-- DATABASE EXPLORATION


-- 1. Explore All Objects In The Database
SELECT * FROM INFORMATION_SCHEMA.TABLES
```

133 %

Results | Messages

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	DataWarehouseAnalytics	gold	dim_customers	BASE TABLE
2	DataWarehouseAnalytics	gold	dim_products	BASE TABLE
3	DataWarehouseAnalytics	gold	fact_sales	BASE TABLE

```sql
-- 2. Explore All Columns In The Database
SELECT * FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME = 'dim_customers';
```

133 %

Results | Messages

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH
1	DataWarehouseAnalytics	gold	dim_customers	customer_key	1	NULL	YES	int	NULL
2	DataWarehouseAnalytics	gold	dim_customers	customer_id	2	NULL	YES	int	NULL
3	DataWarehouseAnalytics	gold	dim_customers	customer_number	3	NULL	YES	nvarchar	50
4	DataWarehouseAnalytics	gold	dim_customers	first_name	4	NULL	YES	nvarchar	50
5	DataWarehouseAnalytics	gold	dim_customers	last_name	5	NULL	YES	nvarchar	50
6	DataWarehouseAnalytics	gold	dim_customers	country	6	NULL	YES	nvarchar	50
7	DataWarehouseAnalytics	gold	dim_customers	marital_status	7	NULL	YES	nvarchar	50
8	DataWarehouseAnalytics	gold	dim_customers	gender	8	NULL	YES	nvarchar	50
9	DataWarehouseAnalytics	gold	dim_customers	birthdate	9	NULL	YES	date	NULL
10	DataWarehouseAnalytics	gold	dim_customers	create_date	10	NULL	YES	date	NULL

```sql
-- DIMENSIONS EXPLORATION

-- 1. Explore All Countries Our Customers Come From
SELECT DISTINCT country FROM gold.dim_customers;
```

133 %

Results | Messages

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

```sql
-- 2. Explore All Product Categories "The Major Divisions"
SELECT DISTINCT category, subcategory, product_name FROM gold.dim_products
ORDER BY 1, 2, 3;
```

133 %

Results | Messages

	category	subcategory	product_name
1	NULL	NULL	HL Mountain Pedal
2	NULL	NULL	HL Road Pedal
3	NULL	NULL	LL Mountain Pedal
4	NULL	NULL	LL Road Pedal
5	NULL	NULL	ML Mountain Pedal
6	NULL	NULL	ML Road Pedal
7	NULL	NULL	Touring Pedal
8	Accessories	Bike Racks	Hitch Rack - 4-Bike
9	Accessories	Bike Stands	All-Purpose Bike Stand
10	Accessories	Bottles and Cages	Mountain Bottle Cage
11	Accessories	Bottles and Cages	Road Bottle Cage
12	Accessories	Bottles and Cages	Water Bottle - 30 oz.
13	Accessories	Cleaners	Bike Wash - Dissolver
14	Accessories	Fenders	Fender Set - Mountain
15	Accessories	Helmets	Sport-100 Helmet- Black
16	Accessories	Helmets	Sport-100 Helmet- Blue
17	Accessories	Helmets	Sport-100 Helmet- Red
18	Accessories	Hydration Packs	Hydration Pack - 70 oz.
19	Accessories	Lights	Headlights - Dual-Beam
20	Accessories	Lights	Headlights - Weatherproof
21	Accessories	Lights	Taillights - Battery-Powered
22	Accessories	Locks	Cable Lock
23	Accessories	Panniers	Touring-Panniers- Large
24	Accessories	Pumps	Minipump
25	Accessories	Pumps	Mountain Pump

```sql
-- DATE EXPLORATION

-- 1. Find The Date Of The First And Last Order
SELECT
MIN (order_date) AS first_order_date,
MAX (order_date) AS last_order_date
FROM gold.fact_sales;
```

133 %

Results | Messages

	first_order_date	last_order_date
1	2010-12-29	2014-01-28

```sql
-- 2. How Many Years Of Sales Are Available?
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
FROM gold.fact_sales;
```

133 %

Results | Messages

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

```sql
-- 3. Find The Youngest And Oldest Customer
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;
```

133 %

Results | Messages

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

```sql
-- MEASURES EXPLORATION

-- Generate Report That Shows All Key Metrics Of The Business
SELECT 'Total Sales' AS measure_name, SUM(sales_amount) AS measure_value FROM gold.fact_sales
UNION ALL
SELECT 'Total Quantity', SUM(quantity) FROM gold.fact_sales
UNION ALL
SELECT 'Average Price', AVG(price) FROM gold.fact_sales
UNION ALL
SELECT 'Total Nr. Orders', COUNT(DISTINCT order_number) FROM gold.fact_sales
UNION ALL
SELECT 'Total Nr. Products', COUNT(product_name) FROM gold.dim_products
UNION ALL
SELECT 'Total Nr. Customers', COUNT(customer_key) FROM gold.dim_customers
```

133 %

Results | Messages

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

```sql
-- MAGNITUDE ANALYSIS

-- 1. Find The Total Customers By Countries
SELECT
country, COUNT(customer_key) AS total_customers
FROM gold.dim_customers
GROUP BY country
ORDER BY total_customers DESC;
```

133 %

Results | Messages

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

```sql
-- 2. Find The Total Customers By Gender
SELECT
gender, COUNT(customer_key) AS total_customers
FROM gold.dim_customers
GROUP BY gender
ORDER BY total_customers DESC;
```

133 %

Results | Messages

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

```sql
-- 3. Find The Total Products By Category
SELECT
category, COUNT(product_key) AS total_products
FROM gold.dim_products
GROUP BY category
ORDER BY total_products DESC;
```

133 %

Results | Messages

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

```sql
-- 4. What Is The Average Costs In Each Category?
SELECT
category, AVG(cost) AS avg_costs
FROM gold.dim_products
GROUP BY category
ORDER BY avg_costs DESC;
```

133 %

Results | Messages

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

```sql
-- 5. What Is The Total Revenue Generated For Each Category?
SELECT
p.category, SUM(f.sales_amount) AS total_revenue
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_products AS p
ON p.product_key = f.product_key
GROUP BY p.category
ORDER BY total_revenue DESC;
```

133 %

Results | Messages

	category	total_revenue
1	Bikes	28316272
2	Accessories	700262
3	Clothing	339716

```sql
-- 6. What Is The Total Revenue Generated By Each Customer?
SELECT
c.customer_key, c.first_name, c.last_name,
SUM(f.sales_amount) AS total_revenue
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_customers AS c
ON c.customer_key = f.customer_key
GROUP BY c.customer_key, c.first_name, c.last_name
ORDER BY total_revenue DESC;
```

133 %

Results | Messages

	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
15	243	Larry	Munoz	11067

```sql
-- 7. What Is The Distribution Of Items Sold Across Countries?
SELECT
c.country, SUM(f.sales_amount) AS total_sold_items
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_customers AS c
ON c.customer_key = f.customer_key
GROUP BY c.country
ORDER BY total_sold_items DESC;
```

133 %

Results | Messages

	country	total_sold_items
1	United States	9162327
2	Australia	9060172
3	United Kingdom	3391376
4	Germany	2894066
5	France	2643751
6	Canada	1977738
7	n/a	226820

```sql
-- RANKING ANALYSIS

-- 1. Which 5 Products Generate The Highest Revenue?
SELECT TOP 5
p.product_name, SUM(f.sales_amount) AS total_revenue
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_products AS p
ON p.product_key = f.product_key
GROUP BY p.product_name
ORDER BY total_revenue DESC;
```

133 %

Results | Messages

	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

```sql
-- 2. What Are The 3 Worst Performing Products In Terms Of Sales?
SELECT TOP 5
p.product_name, SUM(f.sales_amount) AS total_revenue
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_products AS p
ON p.product_key = f.product_key
GROUP BY p.product_name
ORDER BY total_revenue;
```

133 %

Results | Messages

	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

```sql
-- 3. Find The Top 10 Customers Who Have Generated The Highest Revenue?
SELECT
c.customer_key, c.first_name, c.last_name,
SUM(f.sales_amount) AS total_revenue
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_customers AS c
ON c.customer_key = f.customer_key
GROUP BY c.customer_key, c.first_name, c.last_name
ORDER BY total_revenue DESC;
```

133 %

Results | Messages

	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
15	243	Larry	Munoz	11067

```sql
-- 4. Find The 3 Customers With The Lowest Orders Placed?
SELECT TOP 3
c.customer_key, c.first_name, c.last_name,
COUNT(DISTINCT order_number) AS total_orders
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_customers AS c
ON c.customer_key = f.customer_key
GROUP BY c.customer_key, c.first_name, c.last_name
ORDER BY total_orders;
```

133 %

Results | Messages

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