Task: Use SQL queries to extract and analyze data from a database.

I worked with a structured dataset imported into MySQL after converting categorical variables into a numerical format to ensure compatibility with SQL operations.

I performed a series of essential SQL queries for data extraction and analysis:

- **SELECT and WHERE** clauses were used to retrieve specific columns and filter records based on conditions such as rating and category.

- **GROUP BY and HAVING** clauses enabled aggregation of data (e.g., average discounted price by category) and filtering of grouped results.

- **ORDER BY and LIMIT** helped in sorting the data and retrieving top entries based on criteria like price or rating.

- **JOINS** were implemented to combine data from related tables (e.g., `products` and reviews) using foreign keys, enabling multi-table analysis.

This end-to-end process helped in understanding the structure of the dataset, deriving insights, and preparing it for further visualization or reporting tasks.

- **Database selection**

  use as1;
  Database changed

- **Table created**

```
mysql> CREATE TABLE cleaned_data (
    ->     product_id INT,
    ->     product_name VARCHAR(255),
    ->     category INT,
    ->     discounted_price INT,
    ->     actual_price INT,
    ->     discount_percentage INT,
    ->     rating INT,
    ->     rating_count INT,
    ->     about_product INT,
    ->     user_id INT,
    ->     user_name VARCHAR(255),
    ->     review_id INT,
    ->     review_title VARCHAR(255),
    ->     review_content VARCHAR(255),
    ->     img_link VARCHAR(255),
```

```
    ->     product_link VARCHAR(255)
    -> );
Query OK, 0 rows affected (0.13 sec)
```

- **Data Loading**

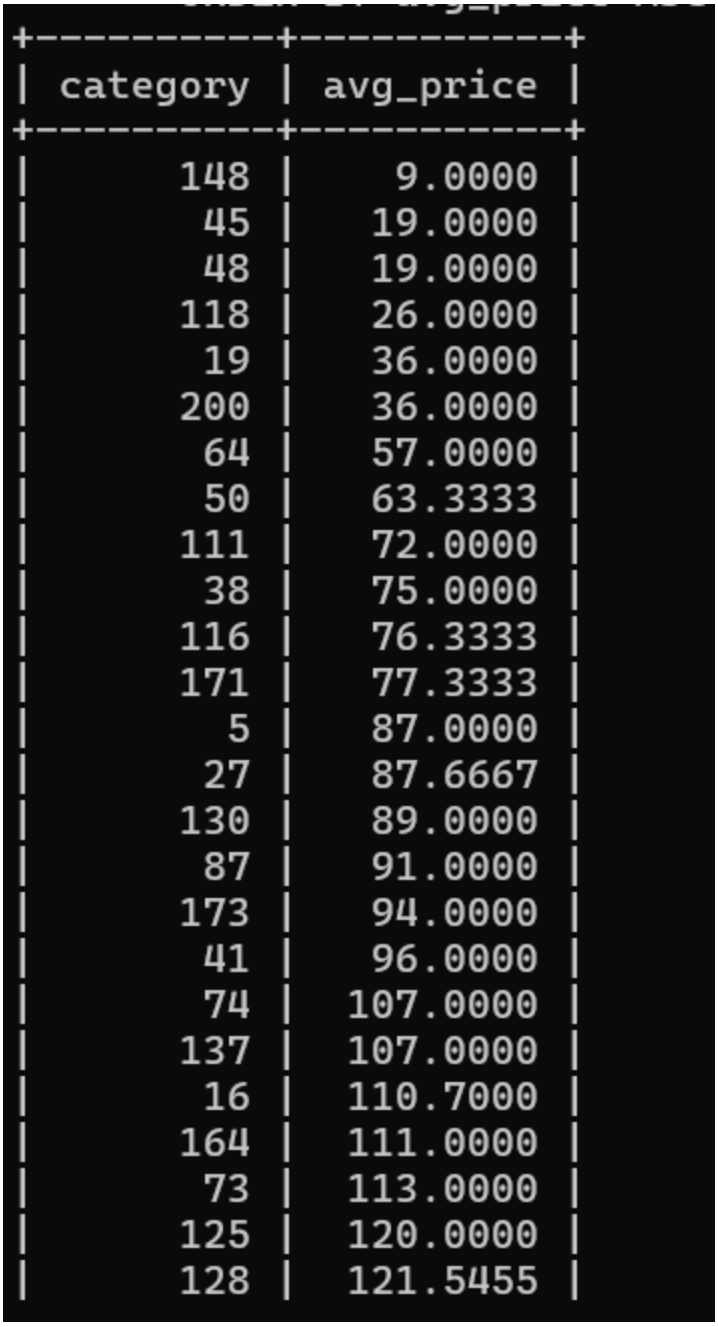
```
 SHOW VARIABLES LIKE 'secure_file_priv';
+------------------+-----------------------------------------------+
| Variable_name    | Value                                         |
+------------------+-----------------------------------------------+
| secure_file_priv | C:\ProgramData\MySQL\MySQL Server 8.0\Uploads\ |
+------------------+-----------------------------------------------+
1 row in set (0.04 sec)

mysql> LOAD DATA INFILE "C:\\ProgramData\\MySQL\\MySQL Server
8.0\\Uploads\\cleaned_data (1).csv"
    -> INTO TABLE cleaned_data
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY '"'
    -> LINES TERMINATED BY '\n'
    -> IGNORE 1 ROWS;
Query OK, 1463 rows affected (0.64 sec)
```

- **Queries**

1. **Basic SELECT query**

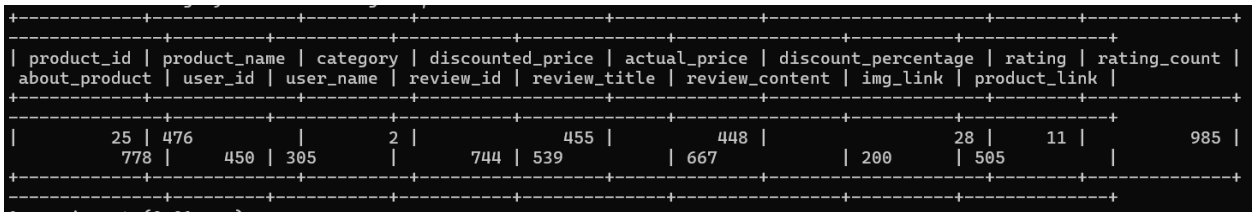
```
SELECT category, AVG(discounted_price) AS avg_price
    -> FROM cleaned_data
    -> WHERE rating > 10
    -> GROUP BY category
    -> ORDER BY avg_price ASC;
```



category	avg_price
148	9.0000
45	19.0000
48	19.0000
118	26.0000
19	36.0000
200	36.0000
64	57.0000
50	63.3333
111	72.0000
38	75.0000
116	76.3333
171	77.3333
5	87.0000
27	87.6667
130	89.0000
87	91.0000
173	94.0000
41	96.0000
74	107.0000
137	107.0000
16	110.7000
164	111.0000
73	113.0000
125	120.0000
128	121.5455

2. Filtering data with WHERE

```
SELECT *
  -> FROM cleaned_data
  -> WHERE category = 2 AND rating > 3;
```



product_id	product_name	category	discounted_price	actual_price	discount_percentage	rating	rating_count	about_product	user_id	user_name	review_id	review_title	review_content	img_link	product_link
25	476		2	455	448	28	11	985	778	450	305		744	539	667	200	505	

3. Aggregation with GROUP BY and HAVING

```
SELECT category, COUNT(*) AS total_products, AVG(discounted_price)
AS avg_price
    -> FROM cleaned_data
    -> GROUP BY category
    -> HAVING AVG(discounted_price) > 100;
```

category	total_products	avg_price
10	231	286.8961
49	18	267.2778
89	24	317.6667
97	63	256.3651
93	49	263.0816
98	6	453.5000
94	6	317.8333
91	2	381.0000
80	1	327.0000
90	3	253.0000
95	3	359.3333
79	1	217.0000
96	3	195.6667
7	1	393.0000
92	1	350.0000
81	1	355.0000
86	1	182.0000
88	1	200.0000
119	76	152.5395
103	12	204.5000
117	68	246.5441
58	13	303.7692
76	52	259.0769
102	5	382.8000
99	3	382.6667
104	16	327.9375
100	2	123.0000
113	1	414.0000
112	5	256.4000
114	10	348.2000
6	3	546.0000
105	5	357.8000
107	7	371.1429
115	8	234.6250
108	2	319.0000
101	4	217.5000
109	1	467.0000
77	8	199.2500

4. **Sorting with ORDERING BY**

```
SELECT product_name, discounted_price
  -> FROM cleaned_data
  -> WHERE category = 3
  -> ORDER BY discounted_price DESC;
```

```
+--------------+------------------+
| product_name | discounted_price |
+--------------+------------------+
| 514          |              538 |
| 444          |              163 |
+--------------+------------------+
2 rows in set (0.00 sec)
```

5.Limiting Results with LIMIT

```
SELECT product_name, discounted_price
FROM cleaned_data
ORDER BY discounted_price DESC
LIMIT 10;
```

```
+--------------+------------------+
| product_name | discounted_price |
+--------------+------------------+
| 1026         |              549 |
| 1070         |              549 |
| 119          |              549 |
| 1025         |              549 |
| 107          |              549 |
| 650          |              549 |
| 1027         |              549 |
| 3            |              549 |
| 118          |              549 |
| 180          |              549 |
+--------------+------------------+
10 rows in set (0.01 sec)
```

6. Subquery in WHERE Clause

```
SELECT product_name, discounted_price
FROM cleaned_data
WHERE discounted_price > (
   SELECT AVG(discounted_price)
   FROM cleaned_data
);
```

product_name	discounted_price
1156	350
1262	315
1078	393
1279	273
827	329
654	327
324	543
393	427
130	273
323	543
1245	273
1235	344
983	430
1323	546
117	523
734	310
321	543
1153	350
1070	549
369	427
1176	318
298	408
76	304
13	350
106	500
1140	443
1063	455
1177	326
954	309
114	315
75	304
47	480
1319	327
225	327
1285	350
1167	350
930	473
1244	273

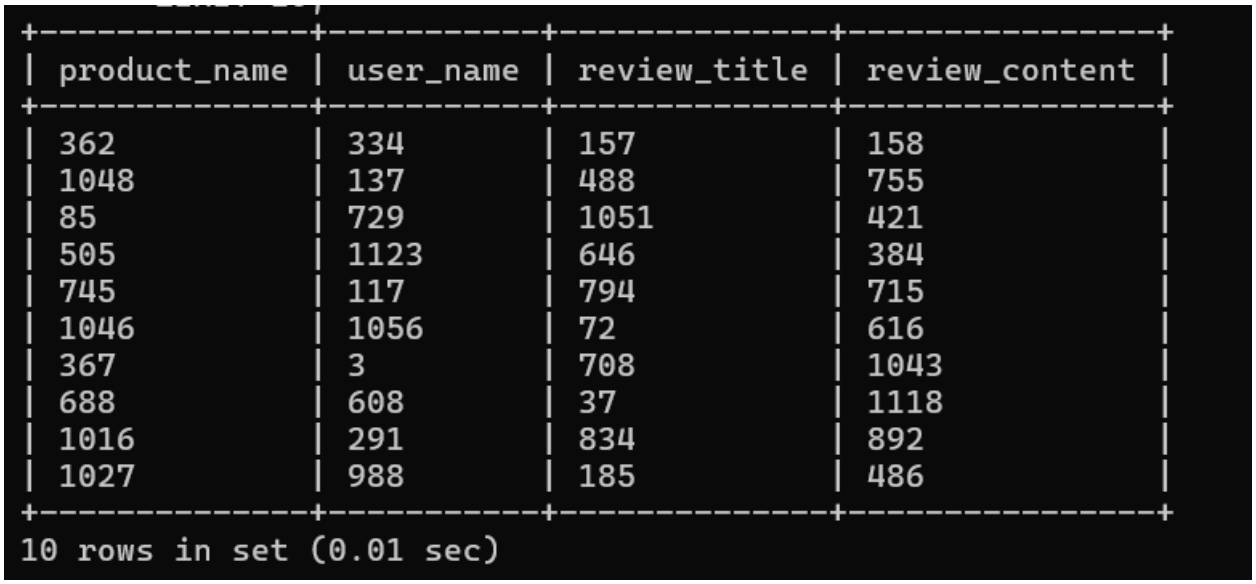
7. JOINS

```
 CREATE TABLE products AS
    -> SELECT DISTINCT
    ->   product_id,
    ->   product_name,
    ->   category,
    ->   discounted_price,
    ->   actual_price,
    ->   discount_percentage,
    ->   rating,
    ->   rating_count,
    ->   about_product,
    ->   product_link,
    ->   img_link
    -> FROM cleaned_data;
Query OK, 1463 rows affected (0.15 sec)
Records: 1463  Duplicates: 0  Warnings: 0

mysql> CREATE TABLE reviews AS
    -> SELECT
    ->   review_id,
    ->   product_id,
    ->   user_id,
    ->   user_name,
    ->   review_title,
    ->   review_content
    -> FROM cleaned_data;
Query OK, 1463 rows affected (0.10 sec)
Records: 1463  Duplicates: 0  Warnings: 0

mysql> SELECT
    ->   p.product_name,
    ->   r.user_name,
    ->   r.review_title,
    ->   r.review_content
    -> FROM products p
    -> INNER JOIN reviews r
```

```
  ->   ON p.product_id = r.product_id
  -> WHERE p.rating > 20
  -> ORDER BY p.rating DESC
  -> LIMIT 10;
```



product_name	user_name	review_title	review_content
362	334	157	158
1048	137	488	755
85	729	1051	421
505	1123	646	384
745	117	794	715
1046	1056	72	616
367	3	708	1043
688	608	37	1118
1016	291	834	892
1027	988	185	486

10 rows in set (0.01 sec)