Practical 1: Databricks SQL Queries

SELECT STATEMENT

1. Display all columns for all transactions.

```
select * from practical.practicals.retail_dataset;
```

See performance (1)

Table

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	2	2023-02-27	CUST002	Female	26	Clothing	2	500
3	3	2023-01-13	CUST003	Male	50	Electronics	1	30
4	4	2023-05-21	CUST004	Male	37	Clothing	1	500
5	5	2023-05-06	CUST005	Male	30	Beauty	2	50
6	6	2023-04-25	CUST006	Female	45	Beauty	1	30
7	7	2023-03-13	CUST007	Male	46	Clothing	2	25
8	8	2023-02-22	CUST008	Male	30	Electronics	4	25
9	9	2023-12-13	CUST009	Male	63	Electronics	2	300
10	10	2023-10-07	CUST010	Female	52	Clothing	4	50
11	11	2023-02-14	CUST011	Male	23	Clothing	2	50
12	12	2023-10-30	CUST012	Male	35	Beauty	3	25
13	13	2023-08-05	CUST013	Male	22	Electronics	3	500
14	14	2023-01-17	CUST014	Male	64	Clothing	4	30

2. Display only the Transaction ID, Date, and Customer ID for all records

```
select `Transaction ID`, `Date`, `Customer ID`
from practical.practicals.retail_dataset;
```

See performance (1)

Table

	Transaction ID	Date	Customer ID
1	1	2023-11-24	CUST001
2	2	2023-02-27	CUST002
3	3	2023-01-13	CUST003
4	4	2023-05-21	CUST004
5	5	2023-05-06	CUST005
6	6	2023-04-25	CUST006

SELECT DISTINCT

3. Display all the distinct product categories in the dataset.

```
select distinct `Product Category` from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	Product Category
1	Clothing
2	Beauty
3	Electronics

4. Display all the distinct gender values in the dataset.

```
select distinct gender from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	gender
1	Female
2	Male

WHERE CLAUSE

5. Display all transactions where the Age is greater than 40.

```
select * from practical.practicals.retail_dataset where age>40;
```

See performance (1) Optimize

Table +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	3	2023-01-13	CUST003	Male	50	Electronics	1	30
2	6	2023-04-25	CUST006	Female	45	Beauty	1	30
3	7	2023-03-13	CUST007	Male	46	Clothing	2	25
4	9	2023-12-13	CUST009	Male	63	Electronics	2	300
5	10	2023-10-07	CUST010	Female	52	Clothing	4	50
6	14	2023-01-17	CUST014	Male	64	Clothing	4	30
7	15	2023-01-16	CUST015	Female	42	Electronics	4	500
8	18	2023-04-30	CUST018	Female	47	Electronics	2	25
9	19	2023-09-16	CUST019	Female	62	Clothing	2	25
10	21	2023-01-14	CUST021	Female	50	Beauty	1	500
11	24	2023-11-29	CUST024	Female	49	Clothing	1	300
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50
13	28	2023-04-23	CUST028	Female	43	Beauty	1	500

6. Display all transactions where the Price per Unit is between 100 and 500.

```
select * from practical.practicals.retail_dataset where `Price per Unit` between 100 AND 500;
```

See performance (1) Optimize

Table +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	2	2023-02-27	CUST002	Female	26	Clothing	2	500
2	4	2023-05-21	CUST004	Male	37	Clothing	1	500
3	9	2023-12-13	CUST009	Male	63	Electronics	2	300
4	13	2023-08-05	CUST013	Male	22	Electronics	3	500
5	15	2023-01-16	CUST015	Female	42	Electronics	4	500
6	16	2023-02-17	CUST016	Male	19	Clothing	3	500
7	20	2023-11-05	CUST020	Male	22	Clothing	3	300
8	21	2023-01-14	CUST021	Female	50	Beauty	1	500
9	24	2023-11-29	CUST024	Female	49	Clothing	1	300
10	26	2023-10-07	CUST026	Female	28	Electronics	2	500
11	28	2023-04-23	CUST028	Female	43	Beauty	1	500
12	30	2023-10-29	CUST030	Female	39	Beauty	3	300
13	31	2023-05-23	CUST031	Male	44	Electronics	4	300
14	35	2023-08-05	CUST035	Female	58	Beauty	3	300

7. Display all transactions where the Product Category is either 'Beauty' or 'Electronics'.

```
select *
from practical.practicals.retail_dataset
whereE `Product Category` = 'Beauty'
    OR `Product Category` = 'Electronics';
```

See performance (1) Optimize

Table +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500
10	18	2023-04-30	CUST018	Female	47	Electronics	2	25
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50

8. Display all transactions where the Product Category is **not** 'Clothing'.

```
select * from practical.practicals.retail_dataset
where `Product Category` != 'Clothing';
```

See performance (1) Optimize

Table +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
1	1	2023-11-24	CUST001	Male	34	Beauty	3	50
2	3	2023-01-13	CUST003	Male	50	Electronics	1	30
3	5	2023-05-06	CUST005	Male	30	Beauty	2	50
4	6	2023-04-25	CUST006	Female	45	Beauty	1	30
5	8	2023-02-22	CUST008	Male	30	Electronics	4	25
6	9	2023-12-13	CUST009	Male	63	Electronics	2	300
7	12	2023-10-30	CUST012	Male	35	Beauty	3	25
8	13	2023-08-05	CUST013	Male	22	Electronics	3	500
9	15	2023-01-16	CUST015	Female	42	Electronics	4	500
10	18	2023-04-30	CUST018	Female	47	Electronics	2	25
11	21	2023-01-14	CUST021	Female	50	Beauty	1	500
12	25	2023-12-26	CUST025	Female	64	Beauty	1	50
13	26	2023-10-07	CUST026	Female	28	Electronics	2	500
14	27	2023-08-03	CUST027	Female	38	Beauty	2	25

9. Display all transactions where the Quantity is greater than or equal to 3.

```
select * from practical.practicals.retail_dataset
where `Quantity`>= '3';
```

See performance (1) Optimize

Table +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category	Quantity	Price per Unit
37	71	2023-07-14	CUST071	Female	51	Beauty	4	25
38	72	2023-05-23	CUST072	Female	20	Electronics	4	500
39	73	2023-08-21	CUST073	Male	29	Electronics	3	30
40	74	2023-11-22	CUST074	Female	18	Beauty	4	500
41	75	2023-07-06	CUST075	Male	61	Beauty	4	50
42	78	2023-07-01	CUST078	Female	47	Clothing	3	500
43	82	2023-12-26	CUST082	Female	32	Beauty	4	50
44	84	2023-11-28	CUST084	Female	38	Electronics	3	30
45	85	2023-02-06	CUST085	Male	31	Clothing	3	50
46	86	2023-11-08	CUST086	Male	19	Beauty	3	30
47	89	2023-10-01	CUST089	Female	55	Electronics	4	500
48	92	2023-08-25	CUST092	Female	51	Electronics	4	30

AGGREGATE FUNCTIONS

10. Count the total number of transactions.

```
select count(`Transaction ID`) AS Total_Transactions
from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	Total_Transactions
1	1000

11. Find the average Age of customers.

```
select AVG(Age) AS Averega_age from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	1.2 Averega_age
1	41.392

12. Find the total quantity of products sold.

```
select sum(`Quantity`) AS Total_Quantity
from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	123 Total_Quantity
1	2514

13. Find the maximum Total Amount spent in a single transaction.

```
select max(`total Amount`) AS Max_Total_Amount
from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	123 Max_Total_Amount
1	2000

14. Find the minimum Price per Unit in the dataset.

```
select min(`price per unit`) AS Min_Price_per_unit
from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	Min_Price_per_unit
1	25

GROUP BY Statement

15. Find the number of transactions per Product Category.

```
select `Product Category`,
count(`Transaction ID`) AS Transaction_Count
from practical.practicals.retail_dataset
group by `Product Category`;
```

See performance (1)

Table +

	Product Category	Transaction_Count
1	Clothing	351
2	Beauty	307
3	Electronics	342

16. Find the total revenue (Total Amount) per gender.

```
select `Gender`,
sum(`Total Amount`) AS Total_Revenue
from practical.practicals.retail_dataset
group by `Gender`;
```

See performance (1)

Table +

	Gender	Total_Revenue
1	Female	232840
2	Male	223160

17. Find the average Price per Unit per product category.

```
select `Product Category`,
avg(`Price per Unit`) AS Average_Price
from practical.practicals.retail_dataset
group by `Product Category`;
```

See performance (1)

Table +

	Product Category	Average_Price
1	Clothing	174.28774928774928
2	Beauty	184.05537459283389
3	Electronics	181.90058479532163

HAVING CLAUSE

18. Find the total revenue per product category where total revenue is greater than 10,000.

```
select `Product Category`,
sum(`Total Amount`) AS Total_Revenue
from practical.practicals.retail_dataset
having sum(`Total Amount`) >1000;
```

Table +

	Product Category	Total_Revenue
1	Clothing	155580
2	Beauty	143515
3	Electronics	156905

19. Find the average quantity per product category where the average is more than 2.

```
select `Product Category`,
avg(`Quantity`) AS Average_Quantity
from practical.practicals.retail_dataset
group by `Product Category`
having avg(`Quantity`) > 2;
```

See performance (1)

Table

	Product Category	Average_Quantity
1	Clothing	2.547008547008547
2	Beauty	2.511400651465798
3	Electronics	2.482456140350877

CASE Statement

20. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'.

```
select `Transaction ID`,
`Total Amount`,
case
    when `Total Amount`> 1000 then 'High'
    else 'Low'
end as Spending_Level
from practical.practicals.retail_dataset;
```

See performance (1)

Table

	Transaction ID	Total Amount	Spending_Level
13	13	1500	High
14	14	120	Low
15	15	2000	High
16	16	1500	High
17	17	100	Low
18	18	50	Low
19	19	50	Low
20	20	900	Low
21	21	500	Low

21. Display a new column called Age_Group that labels customers as:
    - 'Youth' if Age < 30

- 'Adult' if Age is between 30 and 59
- 'Senior' if Age >= 60 *Expected output:* Customer ID, Age, Age_Group

```sql
select `Customer ID`,
 `Age`,
case
    when Age < 30 then 'Youth'
    when Age Between 30 AND 59 then 'Adult'
    else 'Senior'
end as Age_Group
from practical.practicals.retail_dataset;
```

See performance (1)

Table +

	Customer ID	Age	Age_Group
9	CUST009	63	Senior
10	CUST010	52	Adult
11	CUST011	23	Youth
12	CUST012	35	Adult
13	CUST013	22	Youth
14	CUST014	64	Senior
15	CUST015	42	Adult
16	CUST016	19	Youth
17	CUST017	27	Youth