

PRACTICAL DATABRICKS

1.SELECT Statement, Display all columns for all transactions. Expected output: All columns.

The screenshot shows the Databricks workspace interface. On the left is the 'Catalog' sidebar with a search bar and a tree view containing 'My organization', 'workspace', 'system', 'Delta Shares Received', and 'samples'. The main area is titled 'New Query 2025-06-01 7:33pm'. It features a toolbar with a play button, a dropdown for '(1000)' rows, a workspace selector set to 'workspace.default', and buttons for 'Serverless Starter...', 'Serverless', 'S', and 'Save*'. The SQL editor contains the following query:

```
1 SELECT*
2 FROM workspace.default.retail_sales_dataset_practical_3;
3
4
5
6
7
8
9
```

Below the editor, the 'Raw results' tab is active, displaying a table with 8 rows and 7 columns. The columns are Transaction ID, Date, Customer ID, Gender, Age, and Product Category. The data is as follows:

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	1	2023-11-24	CUST001	Male	34	Beauty
2	2	2023-02-27	CUST002	Female	26	Clothing
3	3	2023-01-13	CUST003	Male	50	Electronics
4	4	2023-05-21	CUST004	Male	37	Clothing
5	5	2023-05-06	CUST005	Male	30	Beauty
6	6	2023-04-25	CUST006	Female	45	Beauty
7	7	2023-03-13	CUST007	Male	46	Clothing
8	8	2023-02-22	CUST008	Male	30	Electronics

2. SELECT Statement, Display only the Transaction ID, Date, and Customer ID for all records.
Expected output: Transaction ID, Date, Customer ID.

The screenshot shows the Databricks workspace interface with the same 'Catalog' sidebar. The main area is titled 'New Query 2025-06-01 7:33pm'. The SQL editor contains the following query:

```
9
10
11 SELECT `Transaction ID`,
12        DATE,
13        `Customer ID`
14 FROM workspace.default.retail_sales_dataset_practical_3;
15
16
17
```

Below the editor, the 'Raw results' tab is active, displaying a table with 6 rows and 4 columns. The columns are Transaction ID, DATE, and Customer ID. The data is as follows:

	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

3. **SELECT DISTINCT Statement.** Display all the distinct product categories in the dataset. Expected output: Product Category.

The screenshot shows a SQL query editor interface. On the left is a 'Catalog' sidebar with a search bar and a tree view containing 'My organization', 'workspace', 'system', 'Delta Shares Received', and 'samples'. The main editor area has a title bar 'New Query 2025-06-01 7:33pm' and a toolbar with a play button, '(1000)', 'workspace', 'default', and a 'Save*' button. The query text is: `SELECT DISTINCT 'PRODUCT CATEGORY'` on line 26 and `FROM workspace.default.retail_sales_dataset_practical_3;` on line 27. Below the editor, the 'Raw results' section shows a table with one column 'PRODUCT CATEGORY' and three rows: '1 Clothing', '2 Electronics', and '3 Beauty'. At the bottom, it says '945 ms | 3 rows returned' and 'Refreshed just now'.

	PRODUCT CATEGORY
1	Clothing
2	Electronics
3	Beauty

4. **SELECT DISTINCT Statement.** Display all the distinct gender values in the dataset. Expected output: Gender.

The screenshot shows the same SQL query editor interface. The query text is: `SELECT DISTINCT 'GENDER'` on line 19 and `FROM workspace.default.retail_sales_dataset_practical_3;` on line 20. The 'Raw results' section shows a table with one column 'GENDER' and two rows: '1 Male' and '2 Female'.

	GENDER
1	Male
2	Female

5. **WHERE Clause.** Display all transactions where the Age is greater than 40. Expected output: All columns.

The screenshot shows a data query interface. On the left is a 'Catalog' sidebar with a search bar and a tree view containing 'My organization', 'workspace', 'system', 'Delta Shares Received', and 'samples'. The main area is titled 'New Query 2025-06-01 7:33pm'. It features a SQL editor with the following query:

```
SELECT *  
FROM workspace.default.retail_sales_dataset_practical_3  
WHERE AGE > 40;
```

Below the editor, the 'Raw results' section displays a table with 8 rows and 7 columns: Transaction ID, Date, Customer ID, Gender, Age, and Product Category. The table is filtered to show only transactions where the age is greater than 40. At the bottom, it indicates '915 ms | 534 rows returned' and 'Refreshed just now'.

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	3	2023-01-13	CUST003	Male	50	Electronics
2	6	2023-04-25	CUST006	Female	45	Beauty
3	7	2023-03-13	CUST007	Male	46	Clothing
4	9	2023-12-13	CUST009	Male	63	Electronics
5	10	2023-10-07	CUST010	Female	52	Clothing
6	14	2023-01-17	CUST014	Male	64	Clothing
7	15	2023-01-16	CUST015	Female	42	Electronics
8	18	2023-04-30	CUST018	Female	47	Electronics

6. **WHERE Clause.** Display all transactions where the Price per Unit is between 100 and 500. Expected output: All columns.

The screenshot shows a data query interface. On the left is a 'Catalog' sidebar with a search bar and a tree view containing 'My organization', 'workspace', 'system', 'Delta Shares Received', and 'samples'. The main area is titled 'New Query 2025-06-01 7:33pm'. It features a SQL editor with the following query:

```
SELECT *  
FROM workspace.default.retail_sales_dataset_practical_3  
WHERE 'PRICE PER UNIT' BETWEEN 100 AND 500;
```

Below the editor, the 'Raw results' section displays a table with 8 rows and 7 columns: Transaction ID, Date, Customer ID, Gender, Age, and Product Category. The table is filtered to show only transactions where the price per unit is between 100 and 500. At the bottom, it indicates '809 ms | 396 rows returned' and 'Refreshed just now'.

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	2	2023-02-27	CUST002	Female	26	Clothing
2	4	2023-05-21	CUST004	Male	37	Clothing
3	9	2023-12-13	CUST009	Male	63	Electronics
4	13	2023-08-05	CUST013	Male	22	Electronics
5	15	2023-01-16	CUST015	Female	42	Electronics
6	16	2023-02-17	CUST016	Male	19	Clothing
7	20	2023-11-05	CUST020	Male	22	Clothing
8	21	2023-01-14	CUST021	Female	50	Beautv

7. **WHERE Clause.** Display all transactions where the Product Category is either 'Beauty' or 'Electronics'. Expected output: All columns.

Catalog

Type to search...

For you

All

My organization

workspace

system

Delta Shares Received

samples

New Query 2025-06-01 7:33pm

(1000)

workspace.default

Serverless Starter... Serverless S Save*

```
43
44
45
46 SELECT *
47 FROM workspace.default.retail_sales_dataset_practical_3
48 WHERE `PRODUCT CATEGORY` = 'Beauty' OR `PRODUCT CATEGORY` = 'Electronics';
49
50
51
```

Raw results

+

894 ms | 649 rows returned

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	1	2023-11-24	CUST001	Male	34	Beauty
2	3	2023-01-13	CUST003	Male	50	Electronics
3	5	2023-05-06	CUST005	Male	30	Beauty
4	6	2023-04-25	CUST006	Female	45	Beauty
5	8	2023-02-22	CUST008	Male	30	Electronics
6	9	2023-12-13	CUST009	Male	63	Electronics
7	12	2023-10-30	CUST012	Male	35	Beauty
8	13	2023-08-05	CUST013	Male	22	Electronics

Refreshed 30 seconds ago

8. **WHERE Clause.** Display all transactions where the Product Category is not 'Clothing'. Expected output: All columns.

Catalog

Type to search...

For you

All

My organization

workspace

system

Delta Shares Received

samples

New Query 2025-06-01 7:33pm

(1000)

workspace.default

Serverless Starter... Serverless S Save*

```
50
51
52
53
54 SELECT *
55 FROM workspace.default.retail_sales_dataset_practical_3
56 WHERE `PRODUCT CATEGORY` != 'Clothing';
57
58
59
```

Raw results

+

894 ms | 649 rows returned

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	1	2023-11-24	CUST001	Male	34	Beauty
2	3	2023-01-13	CUST003	Male	50	Electronics
3	5	2023-05-06	CUST005	Male	30	Beauty
4	6	2023-04-25	CUST006	Female	45	Beauty
5	8	2023-02-22	CUST008	Male	30	Electronics
6	9	2023-12-13	CUST009	Male	63	Electronics
7	12	2023-10-30	CUST012	Male	35	Beauty
8	13	2023-08-05	CUST013	Male	22	Electronics

9. **WHERE Clause.** Display all transactions where the Quantity is greater than or equal to 3.
Expected output: All columns.

Catalog

Type to search...

For you All

My organization

- workspace
- system

Delta Shares Received

- samples

New Query 2025-06-01 7:33pm

(1000) workspace.default

Serverless Starter... Serverless S Save*

```
58
59
60 SELECT *
61 FROM workspace.default.retail_sales_dataset_practical_3
62 WHERE `Quantity` >= 3;
63
64
65
66
```

Raw results +

	Transaction ID	Date	Customer ID	Gender	Age	Product Category
1	8	2023-02-22	CUST008	Male	30	Electronics
2	10	2023-10-07	CUST010	Female	52	Clothing
3	14	2023-01-17	CUST014	Male	64	Clothing
4	15	2023-01-16	CUST015	Female	42	Electronics
5	17	2023-04-22	CUST017	Female	27	Clothing
6	23	2023-04-12	CUST023	Female	35	Clothing
7	31	2023-05-23	CUST031	Male	44	Electronics
8	38	2023-03-21	CUST038	Male	38	Beauty

855 ms | 263 rows returned

Refreshed just now

10. **Aggregate Functions.** Count the total number of transactions. Expected output: Total Transactions.

Catalog

Type to search...

For you All

My organization

- workspace
- system

Delta Shares Received

- samples

New Query 2025-06-01 7:33pm

(1000) workspace.default

Serverless Starter... Serverless S Save*

```
63
64
65
66
67 SELECT COUNT(`TRANSACTION ID`) AS `Total Transaction`
68 FROM workspace.default.retail_sales_dataset_practical_3;
69
70
71
```

Raw results +

	Total Transaction
1	1000

585 ms | 1 row returned

Refreshed just now

11. Aggregate Functions. Find the average Age of customers. Expected output: Average_Age.

The screenshot shows the Databricks SQL Editor interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, and Query History. The main area displays a new query titled 'New Query 2025-06-01 3:33pm'. The query text is:

```
SELECT AVG('AGE') AS 'AVERAGE AGE'
FROM workspace.default.retail_sales_dataset_practical_3;
```

Below the query editor, the 'Raw results' section shows a table with one row and one column:

1.2 AVERAGE AGE
41.392

12. Aggregate Functions. Find the total quantity of products sold. Expected output: Total_Quantity.

The screenshot shows the Databricks SQL Editor interface. On the left is a sidebar with navigation options: Catalog, Type to search..., For you, All, My organization, workspace, system, Delta Shares Received, samples. The main area displays a new query titled 'New Query 2025-06-01 7:33pm'. The query text is:

```
SELECT COUNT('QUANTITY') AS 'TOTAL QUANTITY'
FROM workspace.default.retail_sales_dataset_practical_3;
```

Below the query editor, the 'Raw results' section shows a table with one row and one column:

1.3 TOTAL QUANTITY
1000

At the bottom of the interface, there is a status bar showing '775 ms | 1 row returned' and 'Refreshed 2 minutes ago'.

13. Aggregate Functions. Find the maximum Total Amount spent in a single transaction. Expected output: Max_Total_Amount.

The screenshot shows the Databricks SQL Editor interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, and Data Ingestion. The main panel displays a new query titled 'New Query 2025-06-01 3:33pm'. The query text is:

```
80  
81  
82  
83  
84 SELECT MAX('TOTAL AMOUNT') AS 'MAX TOTAL AMOUNT'  
85 FROM workspace.default.retail_sales_dataset_practical_3;  
86  
87  
88  
89  
90
```

The query is executed, and the results are displayed in a table with the following structure:

	MAX TOTAL AMOUNT
1	2000

14. Aggregate Functions. Find the minimum Price per Unit in the dataset. Expected output: Min_Price_per_Unit.

The screenshot shows the Databricks SQL Editor interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, and Data Ingestion. The main panel displays a new query titled 'New Query 2025-06-01 3:33pm'. The query text is:

```
87  
88  
89  
90  
91  
92 SELECT MIN('PRICE PER UNIT') AS 'Min Price Per Unit'  
93 FROM workspace.default.retail_sales_dataset_practical_3;  
94  
95  
96  
97
```

The query is executed, and the results are displayed in a table with the following structure:

	Min Price Per Unit
1	25

15. **GROUP BY Statement.** Find the number of transactions per Product Category. Expected output: Product Category, Transaction_Count.

The screenshot shows the Databricks SQL Editor interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses. The main area is divided into a Catalog pane on the left and a query editor on the right. The Catalog pane shows a search bar and a tree view with 'My organization' containing 'workspace', 'system', and 'Delta Shares Received' containing 'samples'. The query editor shows a new query titled 'New Query 2025-06-01 3:46pm'. The SQL code is:

```
SELECT `PRODUCT CATEGORY`,  
       COUNT(`TRANSACTION ID`) AS `TRANSACTION COUNT`  
FROM workspace.default.retail_sales_dataset_practical_3  
GROUP BY ALL;
```

 Below the code, the 'Raw results' table is displayed with 3 rows and 2 columns: 'PRODUCT CATEGORY' and 'TRANSACTION COUNT'. The results are: 1 Clothing (351), 2 Electronics (342), and 3 Beauty (307).

	PRODUCT CATEGORY	TRANSACTION COUNT
1	Clothing	351
2	Electronics	342
3	Beauty	307

16. **GROUP BY Statement.** Find the total revenue (Total Amount) per gender. Expected output: Gender, Total_Revenue.


The screenshot shows the Databricks SQL Editor interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses. The main area is divided into a Catalog pane on the left and a query editor on the right. The Catalog pane shows a search bar and a tree view with 'My organization' containing 'workspace', 'system', and 'Delta Shares Received' containing 'samples'. The query editor shows a new query titled 'New Query 2025-06-01 3:46pm'. The SQL code is:

```
SELECT `GENDER`,  
       SUM(`TOTAL AMOUNT`) AS `TOTAL REVENUE`  
FROM workspace.default.retail_sales_dataset_practical_3  
GROUP BY ALL;
```

 Below the code, the 'Raw results' table is displayed with 2 rows and 2 columns: 'GENDER' and 'TOTAL REVENUE'. The results are: 1 Male (223160) and 2 Female (232840).

	GENDER	TOTAL REVENUE
1	Male	223160
2	Female	232840

17. GROUP BY Statement. Find the average Price per Unit per product category. Expected output: Product Category, Average_Price.

 Search data, notebooks, recents, and more... CTRL + P

New

- Workspace
- Recents
- Catalog
- Workflows
- Compute
- Marketplace

SQL

- SQL Editor**
- Queries
- Dashboards
- Genie
- Alerts
- Query History
- SQL Warehouses

Catalog

Type to search...

For you All

- My organization
 - workspace
 - system
- Delta Shares Received
 - samples

New Query 2025-06-01 3:33pm

Run all (1000) workspace default New SQL editor: OFF


Send feedback

```
115
116
117
118
119 SELECT `PRODUCT CATEGORY`,
120        AVG(`PRICE PER UNIT`) AS `AVERAGE PRICE`
121 FROM workspace.default.retail_sales_dataset_practical_3
122 Group BY ALL;
123
124
125
```

Raw results

	PRODUCT CATEGORY	1.2 AVERAGE PRICE
1	Clothing	174.28774928774928
2	Electronics	181.90058479532163
3	Beauty	184.05537459283389

18. HAVING Clause. Find the total revenue per product category where total revenue is greater than 10,000. Expected output: Product Category, Total_Revenue.

 Search data, notebooks, recents, and more... CTRL + P

New

- Workspace
- Recents
- Catalog
- Workflows
- Compute
- Marketplace

SQL

- SQL Editor**
- Queries
- Dashboards
- Genie
- Alerts
- Query History
- SQL Warehouses

Catalog

Type to search...

For you All

- My organization
 - workspace
 - system
- Delta Shares Received
 - samples

New Query 2025-06-01 3:33pm

Run all (1000) workspace default New SQL editor: OFF

Send feedback

```
125
126
127
128 SELECT `PRODUCT CATEGORY`,
129        SUM(`TOTAL AMOUNT`) AS `TOTAL REVENUE`
130 FROM workspace.default.retail_sales_dataset_practical_3
131 GROUP BY ALL
132 HAVING SUM(`TOTAL AMOUNT`) >10000;
133
134
135
```

Raw results

	PRODUCT CATEGORY	TOTAL REVENUE
1	Clothing	155580
2	Electronics	156905
3	Beauty	143515

19. HAVING Clause. Find the average quantity per product category where the average is more than 2. Expected output: Product Category, Average_Quantity.

The screenshot shows the Databricks SQL Editor interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses. The main panel displays a SQL query in the editor:

```
136  
137  
138 SELECT 'PRODUCT CATEGORY',  
139        AVG('QUANTITY') AS 'AVERAGE QUANTITY'  
140 FROM workspace.default.retail_sales_dataset_practical_3  
141 GROUP BY ALL  
142 HAVING AVG('QUANTITY') >2;  
143  
144  
145  
146
```

Below the query editor, the 'Raw results' section shows a table with 3 rows and 2 columns: PRODUCT CATEGORY and AVERAGE QUANTITY.

	PRODUCT CATEGORY	1.2 AVERAGE QUANTITY
1	Clothing	2.547008547008547
2	Electronics	2.482456140350877
3	Beauty	2.511400651465798

20. CASE Statement. Display a column called Spending_Level that shows 'High' if Total Amount > 1000, otherwise 'Low'. Expected output: Transaction ID, Total Amount, Spending_Level.

The screenshot shows the Databricks SQL Editor interface. The left sidebar contains navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, and SQL Warehouses. The main panel displays a SQL query in the editor:

```
146  
147  
148 SELECT 'Transaction ID',  
149        'Total Amount',  
150        CASE  
151          WHEN 'Total Amount' >1000 THEN 'High'  
152          ELSE 'Low'  
153          END AS 'Spending Level'  
154 FROM workspace.default.retail_sales_dataset_practical_3;  
155  
156
```

Below the query editor, the 'Raw results' section shows a table with 9 rows and 4 columns: Transaction ID, Total Amount, and Spending Level.

	Transaction ID	Total Amount	Spending Level
1	1	150	Low
2	2	1000	Low
3	3	30	Low
4	4	500	Low
5	5	100	Low
6	6	30	Low
7	7	50	Low
8	8	100	Low
9	9	600	Low

21. **CASE Statement.** 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.

The screenshot shows the Databricks SQL Editor interface. On the left is a sidebar with navigation options: New, Workspace, Recents, Catalog, Workflows, Compute, Marketplace, SQL, SQL Editor (selected), Queries, Dashboards, Genie, Alerts, Query History, SQL Warehouses, Data Engineering, Job Runs, Data Ingestion, and Pipelines. The main area is divided into a Catalog pane on the left and a query editor on the right. The Catalog pane shows a search bar and a tree view with 'My organization' containing 'workspace', 'system', and 'Delta Shares Received' containing 'samples'. The query editor shows a SQL query that uses a CASE statement to categorize customers by age. Below the query, the 'Raw results' are displayed as a table with 10 rows and 3 columns: Customer ID, Age, and Age Group.

```

157
158
159 SELECT 'Customer ID',
160        'Age',
161        CASE
162          WHEN 'Age' < 30 THEN 'Young'
163          WHEN 'Age' BETWEEN 30 AND 59 THEN 'Adult'
164          WHEN 'Age' >= 60 THEN 'Senior'
165        END AS 'Age Group'
166 FROM workspace.default.retail_sales_dataset_practical_3
167

```

	Customer ID	Age	Age Group
1	CUST001	34	Adult
2	CUST002	26	Young
3	CUST003	50	Adult
4	CUST004	37	Adult
5	CUST005	30	Adult
6	CUST006	45	Adult
7	CUST007	46	Adult
8	CUST008	30	Adult
9	CUST009	63	Senior
10	CUST010	57	Adult