databricks05 - Aggregate Functions

Create Tables

Run the cell below to create tables for the questions in this notebook.

```
%run ../Utilities/05-CreateTables
```

Declared the following table:
- **revenue1**

Declared the following table:
- **revenue2**

Declared the following table:
- **revenue3**

Declared the following table:
- **revenue4**

Declared the following tables:
- **products**

Declared the following table:
- **sales**

Question 1: Min Function

Summary

Compute the minimum value from the `Amount` field for each unique value in the `TrueFalse` field in the table `revenue1`.

Steps to complete

Write a SQL query that achieves the following:

- Computes the number of `true` and `false` records in the `TrueFalse` field from the table `revenue1`
- Renames the new column to `count`
- Store the records in a temporary view named `q1Results` with the following schema:

column	type
TrueFalse	boolean
MinAmount	int

A properly completed solution should produce a view similar to this sample output:

TrueFalse	count
true	4956
false	5044

```
SELECT
COUNT(TrueFalse)
FROM
revenue1
```

	count(TrueFalse)
1	10000

Showing all 1 rows.

```
CREATE OR REPLACE TEMPORARY VIEW q1Results AS
SELECT
  TrueFalse,
  CAST(count AS INT) AS count
FROM
(
  SELECT
    TrueFalse,
    COUNT(TrueFalse) AS count
  FROM
    revenue1
  GROUP BY TrueFalse
);

SELECT
*
FROM
q1Results
```

	TrueFalse	count
1	true	5017
2	false	4983

Showing all 2 rows.

```
DESCRIBE q1Results
```

	col_name	data_type	comment
1	TrueFalse	boolean	null
2	count	int	null

Showing all 2 rows.

Question 2: Max Function

Summary

Compute the maximum value from the `Amount` field for each unique value in the `TrueFalse` field in the table `revenue2`.

Steps to complete

- Computes the maximum `Amount` for `True` records and `False` records from the `TrueFalse` field from the table `revenue2`
- Renames the new column to `maxAmount`
- Store the records in a temporary view named `q2Results` with the following schema:

column	type
TrueFalse	boolean
maxAmount	double

A properly completed solution should produce a DataFrame similar to this sample output:

TrueFalse	MaxAmount
true	2243937.93
false	2559457.1799999997

```
CREATE OR REPLACE TEMPORARY VIEW q2Results AS
SELECT
TrueFalse,
max(Amount) maxAmount
FROM revenue2
GROUP BY TrueFalse;
```

```
SELECT
*
FROM q2Results
```

	TrueFalse	maxAmount
1	true	9998.93
2	false	9997.19

Showing all 2 rows.

Question 3: Avg Function

Summary

Compute the average of the `Amount` field for each unique value in the `TrueFalse` field in the table `revenue3`.

Steps to complete

- Computes the average of `Amount` for `True` records and `False` records from the `TrueFalse` field in the table `revenue3`.
- Renames the new column to `avgAmount`
- Store the records in a temporary view named `q3Results` with the following schema:

column	type
TrueFalse	boolean
avgAmount	double

A properly completed solution should produce a DataFrame similar to this sample output:

TrueFalse	AvgAmount
true	2243937.93
false	2559457.1799999997

```
CREATE OR REPLACE TEMPORARY VIEW q3Results AS
SELECT
TrueFalse,
avg(Amount) avgAmount
FROM revenue2
GROUP BY TrueFalse;


SELECT
*
FROM q3Results
```

	TrueFalse	avgAmount
1	true	5033.45083516045
2	false	5086.278091511144

Showing all 2 rows.

Question 4: Pivot

Summary

Calculate the total `Amount` for `YesNo` values of **true** and **false** in 2002 and 2003 from the table `revenue4`.

Steps to complete

- Casts the `UTCTime` field to Timestamp and names the new column `Date`
- Extracts a `Year` column from the `Date` column
- Filters for years greater than 2001 and less than or equal to 2003
- Groups by `YesNo` and creates a pivot table to get the total `Amount` for each year and each value in `YesNo`
- Represents each total amount as a float rounded to two decimal places
- Store the results into a temporary table named `q4results`

A properly completed solution should produce a view similar to this sample output:

YesNo	2002	2003

YesNo	2002	2003
true	61632.3	8108.47
false	44699.99	35062.22

```
CREATE OR REPLACE TEMPORARY VIEW q4results AS
SELECT
  *
FROM
  (
    SELECT
      YesNo,
      YEAR(CAST(UTCTime AS Timestamp)) Year
    FROM
      revenue4
      WHERE
      YEAR(CAST(UTCTime AS Timestamp)) IN (2002, 2003)
      ) PIVOT (
    COUNT(Year) FOR Year IN (2002, 2003)
  )
;

SELECT
*
FROM
q4results
```

	YesNo	2002	2003
1	true	223	237
2	false	224	238

Showing all 2 rows.

Question 5: Null Values and Aggregates

Summary

Compute sums of `amount` grouped by `aisle` after dropping null values from `products` table.

Steps to complete

- Drops any rows that contain null values in either the `itemId` or the `aisle` column
- Aggregates sums of the `amount` column grouped by `aisle`
- Store the results into a temporary view named `q5Results`

```
SELECT
aisle,
SUM(amount) amount
FROM
products
WHERE (itemId IS NOT NULL) AND (aisle IS NOT NULL)
GROUP BY aisle;
```

	aisle	amount
1	3	63
2	5	14
3	7	107
4	12	56
5	2	126
6	8	8

Showing all 6 rows.

Question 6: Generate Subtotals By Rollup

Summary

Compute averages of `income` grouped by `itemName` and `month` such that the results include averages across all months as well as a subtotal for an individual month from the `sales` table.

Steps to complete

- Coalesces null values in the `month` column generated by the `ROLLUP` clause
- Store the results into a temporary view named `q6Results`

Your results should look something like this:

itemName	month	avgRevenue
Anim	10	4794.16
Anim	7	5551.31
Anim	All months	5046.54
Aute	4	4069.51
Aute	7	3479.31
Aute	8	6339.28
Aute	All months	4489.41
...	...	...

```
CREATE OR REPLACE TEMPORARY VIEW q6Results AS
SELECT
  itemName,
  COALESCE(month(date), "All months") AS month,
  ROUND(AVG(revenue))  AS avgRevenue
FROM sales
GROUP BY ROLLUP (itemName, month)
ORDER BY itemName, month;

SELECT * FROM q6Results
```

	itemName	month	avgRevenue
1	null	null	5087
2	Ad	null	3685
3	Ad	10	103
4	Ad	3	8424
5	Ad	4	2529
6	Adipiscing	null	5414
7	Adipiscing	5	6066

Truncated results, showing first 1000 rows.