Project name: Cohort Analysis is using MySQL

Step 1: Data Preparation

Ensure you have the necessary data in a suitable format. Your data should include transaction details like `InvoiceNo`, `CustomerID`, `InvoiceDate`, `Quantity`, `UnitPrice`, ` Country `,` Description` and `StockCode`.

Assuming your table is named `online_retail_data`, it should have the following structure:
Load your data into this table.

Step 2: Data Validation and Cleanup

Check for any records with null values or invalid data and clean them up.

```sql
-- Count total records
SELECT COUNT(*) FROM `online_retail_data`;
-- Check for records with null CustomerID
SELECT * FROM `online_retail_data` WHERE CustomerID IS NULL;
-- Check for records with zero or negative Quantity
SELECT * FROM `online_retail_data` WHERE Quantity <= 0;
```

Remove or handle any invalid records based on your findings.

Step 3: Create Cohort Analysis for Customer Retention

Use Common Table Expressions (CTEs) to preprocess your data and calculate necessary fields for cohort analysis.

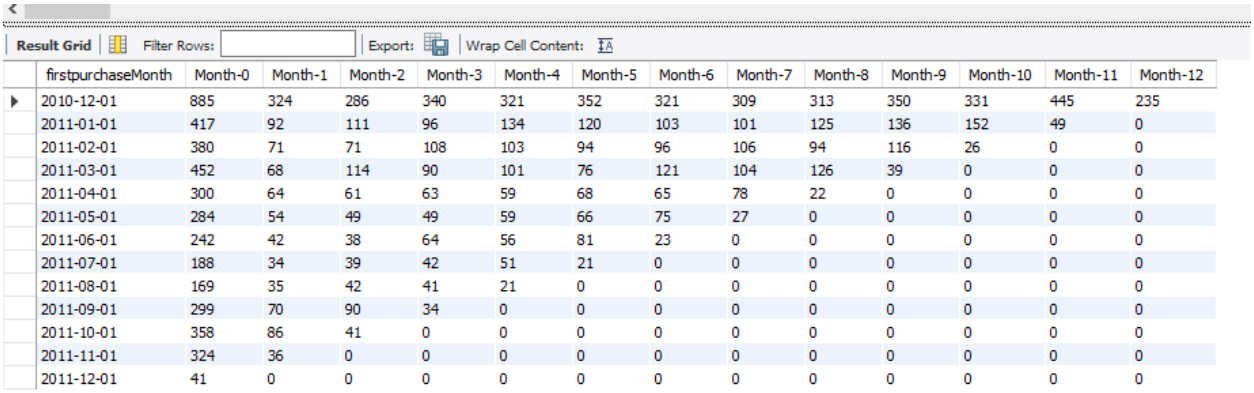
```sql
WITH CTE1 AS
(
   SELECT
      InvoiceNo,
      CustomerID,
      STR_TO_DATE(InvoiceDate, '%d/%m/%y') AS InvoiceDate
   FROM `online_retail_data`
   WHERE CustomerID IS NOT NULL AND InvoiceNo IS NOT NULL
),
CTE2 AS
(
   SELECT
      CustomerID,
```

```
      InvoiceNo,
      InvoiceDate,
      DATE_SUB(InvoiceDate, INTERVAL DAY(InvoiceDate) - 1 DAY) AS purchaseMonth,

      DATE_SUB(MIN(InvoiceDate) OVER (PARTITION BY CustomerID), INTERVAL
DAY(MIN(InvoiceDate) OVER (PARTITION BY CustomerID)) - 1 DAY) AS
firstpurchaseMonth
   FROM CTE1
),
CTE3 AS
(
   SELECT
      CustomerID,
      purchaseMonth,
      firstpurchaseMonth,
      CONCAT('Month-', TIMESTAMPDIFF(MONTH,
         DATE_SUB(firstpurchaseMonth, INTERVAL DAY(firstpurchaseMonth) - 1 DAY),
         DATE_SUB(purchaseMonth, INTERVAL DAY(purchaseMonth) - 1 DAY)
      )) AS CohortMonth
   FROM CTE2
)
SELECT
   firstpurchaseMonth,
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-0' THEN CustomerID END) AS
'Month-0',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-1' THEN CustomerID END) AS
'Month-1',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-2' THEN CustomerID END) AS
'Month-2',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-3' THEN CustomerID END) AS
'Month-3',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-4' THEN CustomerID END) AS
'Month-4',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-5' THEN CustomerID END) AS
'Month-5',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-6' THEN CustomerID END) AS
'Month-6',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-7' THEN CustomerID END) AS
'Month-7',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-8' THEN CustomerID END) AS
'Month-8',
   COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-9' THEN CustomerID END) AS
'Month-9',
```

```
    COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-10' THEN CustomerID END) AS 'Month-10',
    COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-11' THEN CustomerID END) AS 'Month-11',
    COUNT(DISTINCT CASE WHEN CohortMonth = 'Month-12' THEN CustomerID END) AS 'Month-12'
FROM CTE3
GROUP BY firstpurchaseMonth
ORDER BY firstpurchaseMonth;
```

Customer Retention Cohort Analysis Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

firstpurchaseMonth	Month-0	Month-1	Month-2	Month-3	Month-4	Month-5	Month-6	Month-7	Month-8	Month-9	Month-10	Month-11	Month-12
2010-12-01	885	324	286	340	321	352	321	309	313	350	331	445	235
2011-01-01	417	92	111	96	134	120	103	101	125	136	152	49	0
2011-02-01	380	71	71	108	103	94	96	106	94	116	26	0	0
2011-03-01	452	68	114	90	101	76	121	104	126	39	0	0	0
2011-04-01	300	64	61	63	59	68	65	78	22	0	0	0	0
2011-05-01	284	54	49	49	59	66	75	27	0	0	0	0	0
2011-06-01	242	42	38	64	56	81	23	0	0	0	0	0	0
2011-07-01	188	34	39	42	51	21	0	0	0	0	0	0	0
2011-08-01	169	35	42	41	21	0	0	0	0	0	0	0	0
2011-09-01	299	70	90	34	0	0	0	0	0	0	0	0	0
2011-10-01	358	86	41	0	0	0	0	0	0	0	0	0	0
2011-11-01	324	36	0	0	0	0	0	0	0	0	0	0	0
2011-12-01	41	0	0	0	0	0	0	0	0	0	0	0	0

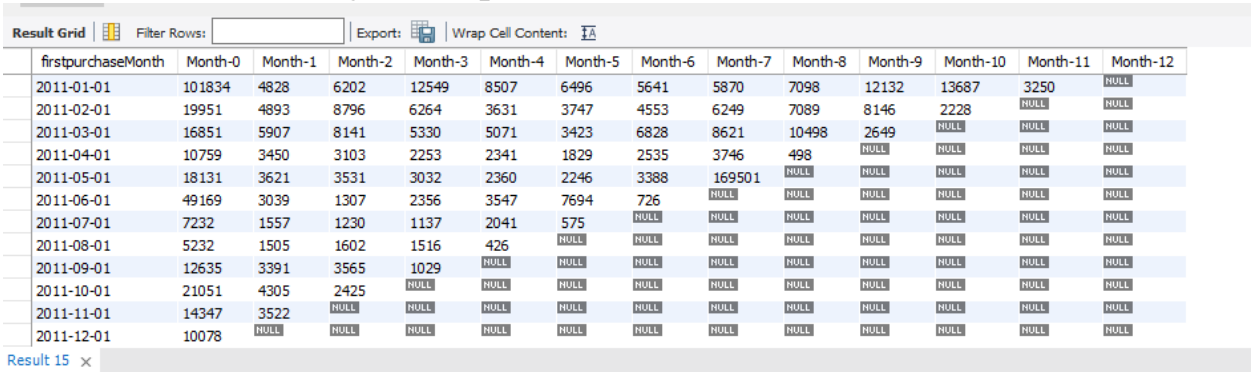
Step 4: Create Cohort Analysis for Revenue

Use another set of CTEs to preprocess your data and calculate necessary fields for revenue cohort analysis.

```sql
WITH CTE1 AS
(
   SELECT
      InvoiceNo,
      CustomerID,
      STR_TO_DATE(InvoiceDate, '%d/%m/%y') AS InvoiceDate,
      ROUND(Quantity * UnitPrice) AS Revenue
   FROM `online_retail_data`
   WHERE CustomerID IS NOT NULL AND InvoiceNo IS NOT NULL
),
CTE2 AS
(
   SELECT
      CustomerID,
      InvoiceNo,
      Revenue,
      InvoiceDate,
      DATE_SUB(InvoiceDate, INTERVAL DAY(InvoiceDate) - 1 DAY) AS purchaseMonth,
      DATE_SUB(MIN(InvoiceDate) OVER (PARTITION BY CustomerID), INTERVAL
DAY(MIN(InvoiceDate) OVER (PARTITION BY CustomerID)) - 1 DAY) AS
firstpurchaseMonth
   FROM CTE1
),
CTE3 AS
(
   SELECT
      CustomerID,
      Revenue,
      purchaseMonth,
      firstpurchaseMonth,
      CONCAT('Month-', TIMESTAMPDIFF(MONTH,
         DATE_SUB(firstpurchaseMonth, INTERVAL DAY(firstpurchaseMonth) - 1 DAY),
         DATE_SUB(purchaseMonth, INTERVAL DAY(purchaseMonth) - 1 DAY)
      )) AS CohortMonth
   FROM CTE2
)
SELECT
   firstpurchaseMonth,
   SUM(CASE WHEN CohortMonth = 'Month-0' THEN Revenue END) AS 'Month-0',
   SUM(CASE WHEN CohortMonth = 'Month-1' THEN Revenue END) AS 'Month-1',
   SUM(CASE WHEN CohortMonth = 'Month-2' THEN Revenue END) AS 'Month-2',
   SUM(CASE WHEN CohortMonth = 'Month-3' THEN Revenue END) AS 'Month-3',
```

```
    SUM(CASE WHEN CohortMonth = 'Month-4' THEN Revenue END) AS 'Month-4',
    SUM(CASE WHEN CohortMonth = 'Month-5' THEN Revenue END) AS 'Month-5',
    SUM(CASE WHEN CohortMonth = 'Month-6' THEN Revenue END) AS 'Month-6',
    SUM(CASE WHEN CohortMonth = 'Month-7' THEN Revenue END) AS 'Month-7',
    SUM(CASE WHEN CohortMonth = 'Month-8' THEN Revenue END) AS 'Month-8',
    SUM(CASE WHEN CohortMonth = 'Month-9' THEN Revenue END) AS 'Month-9',
    SUM(CASE WHEN CohortMonth = 'Month-10' THEN Revenue END) AS 'Month-10',
    SUM(CASE WHEN CohortMonth = 'Month-11' THEN Revenue END) AS 'Month-11',
    SUM(CASE WHEN CohortMonth = 'Month-12' THEN Revenue END) AS 'Month-12'
FROM CTE3
GROUP BY firstpurchaseMonth
ORDER BY firstpurchaseMonth;
```

Revenue Cohort Analysis Output:



Result Grid | Filter Rows: | Export: | Wrap Cell Content:

firstpurchaseMonth	Month-0	Month-1	Month-2	Month-3	Month-4	Month-5	Month-6	Month-7	Month-8	Month-9	Month-10	Month-11	Month-12
2011-01-01	101834	4828	6202	12549	8507	6496	5641	5870	7098	12132	13687	3250	NULL
2011-02-01	19951	4893	8796	6264	3631	3747	4553	6249	7089	8146	2228	NULL	NULL
2011-03-01	16851	5907	8141	5330	5071	3423	6828	8621	10498	2649	NULL	NULL	NULL
2011-04-01	10759	3450	3103	2253	2341	1829	2535	3746	498	NULL	NULL	NULL	NULL
2011-05-01	18131	3621	3531	3032	2360	2246	3388	169501	NULL	NULL	NULL	NULL	NULL
2011-06-01	49169	3039	1307	2356	3547	7694	726	NULL	NULL	NULL	NULL	NULL	NULL
2011-07-01	7232	1557	1230	1137	2041	575	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2011-08-01	5232	1505	1602	1516	426	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2011-09-01	12635	3391	3565	1029	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2011-10-01	21051	4305	2425	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2011-11-01	14347	3522	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL
2011-12-01	10078	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Result 15

Conclusion:

By leveraging MySQL for cohort analysis, we gained a deeper understanding of customer retention and revenue patterns. The insights derived from this analysis are crucial for making informed business decisions, optimizing marketing strategies, and improving customer engagement and retention efforts. This project demonstrated the power of cohort analysis in uncovering valuable trends and patterns that can drive business growth and enhance overall performance.