Business School
UNIVERSITY OF COLORADO DENVER

Information Systems Program

# Module 6
# SQL for Data Mining Input

## Lesson 8: SQL Coding to Generate Training Data with Top Event History

# Lesson Objectives

- Use the CASE function and ROW_NUMBER analytic function in example SQL statements
- Apply the CTE statement pattern to write SELECT statements for event ordering
- Write SELECT statements for event queries
- Apply the complete statement pattern to combine an event ordering query with event queries

# Event Ordering Query as a CTE I

```
-- Example 1 for CTE using Order Entry tables
WITH CTERankedProducts AS (
SELECT Customer.CustNo, CustZip, CustBal,
       ProdNo, Qty,
       ROW_NUMBER() OVER ( PARTITION BY
       Customer.CustNo ORDER BY Qty DESC)
        AS QtyRank
 FROM Customer, OrderTbl, OrdLine
 WHERE Customer.CustNo = OrderTbl.CustNo
   AND OrdLine.OrdNo = OrderTbl.OrdNo )
-- Simple statement using the CTE
SELECT * FROM CTERankedProducts;
```



# Event Ordering Query as a CTE II

```
-- Example 2 of CTE using the Store Sales tables
WITH CTERankedSales AS (
SELECT TimeNo, SSSales.CustId, CustState,
       CustZip, ItemId, SalesDollar,
       ROW_NUMBER() OVER ( PARTITION BY TimeNo,
         SSSales.CustId ORDER BY SalesDollar DESC)
         AS SalesDollarRank
 FROM SSCustomer INNER JOIN SSSales
   ON SSCustomer.CustId = SSSales.CustId )
-- Simple statement using the CTE
SELECT * FROM CTERankedSales;
```

# Query for Top Event using a CTE

```
-- Example 3 with Order Entry tables
SELECT CustNo, CustZip, CustBal,
       ProdNo AS ProdNo1, Qty AS Qty1,
       NULL AS ProdNo2, 0 AS Qty2,
       NULL AS ProdNo3, 0 AS Qty3
 FROM CTERankedProducts
 WHERE CustNo IN
( SELECT CustNo
   FROM CTERankedProducts
   GROUP BY CustNo
   HAVING MAX(QtyRank) = 1 );
```



# Query for Second Event using a CTE

```
-- Example 4 for SELECT statement using the CTE for second event
SELECT CTE1.CustNo, CTE1.CustZip, CTE1.CustBal,
       CASE 1 WHEN CTE1.QtyRank THEN CTE1.ProdNo ELSE CTE2.ProdNo
                END AS ProdNo1,
       CASE 1 WHEN CTE1.QtyRank THEN CTE1.Qty ELSE CTE2.Qty
              END AS Qty1,
       CASE 2 WHEN CTE1.QtyRank THEN CTE1.ProdNo ELSE CTE2.ProdNo
              END AS ProdNo2,
       CASE 2 WHEN CTE1.QtyRank THEN CTE1.Qty ELSE CTE2.Qty
              END AS Qty2,
       NULL AS ProdNo3, 0 AS Qty3
 FROM CTERankedProducts CTE1, CTERankedProducts CTE2
 WHERE CTE1.CustNo IN
  (SELECT CustNo FROM CTERankedProducts
    GROUP BY CustNo HAVING MAX(QtyRank) = 2 )
   AND CTE1.CustNo = CTE2.CustNo AND CTE1.ProdNo < CTE2.ProdNo;
```



# Notes about the Complete Query

- 2 UNION operations for event queries 1 to 3
- Default value (0) for numeric event column instead of NULL value with PostgreSQL
- Optional ORDER BY after the last UNION operation
- Script document with event query 3+, complete example, and other examples
- Additional examples in the extra script document for the store sales tables

# Summary

- CTE to order events by weight using ROW_NUMBER
- Event queries using the CASE function, subqueries, and self-joins
- Apply statement pattern and complete examples
- Study complete examples in the script documents