Module 6 Assignment

The module 6 assignment provides experience writing SQL statements to generate input for data mining algorithms used for association rules and classification. All problems use the Inventory data warehouse tables. Each problem has an equal value of 10 points.

For each problem, you should adapt templates and examples given in the notes. Each problem relates to a similar problem in the lesson notes. The textbook does not cover this material as it was developed after the last edition of the textbook. I doubt that you will find other sources as it is largely original material.

Your SELECT statements should reference the tables of the Inventory Data Warehouse, described in a document referenced in the Course Software Requirements lesson of module 1. The INSERT statements are provided in another document in the same lesson. The Inventory Data Warehouse design and rows are identical from module 5 in course 2. If you added rows through the data integration assignment in module 5 of course 2, you should remove those rows or just recreate and repopulate the tables.

Problem 1: Baskets containing item sets of two items

Write a SELECT statement to generate baskets identified by the combination of customer vendor key, date key, and branch plant key for shipments (TransTypeKey = 5). Each row should contain the basket identifying columns and a combination of two items containing item master key values. Eliminate permutations (orderings) of basket items across rows. Order the result by customer vendor key, date key, branch plant key, and the first item.

Problem 2: Baskets containing item sets of three items

Write a SELECT statement to generate baskets identified by the combination of customer vendor key, date key, and branch plant key for shipments (TransTypeKey = 5). Each row should contain the basket identifying columns and a combination of three items containing item master key values. Eliminate permutations (ordering) of basket items across rows. Order the result by customer vendor key, date key, and branch plant key.

Problem 3: Association rules of size 2 with evaluation measures (support, confidence, and lift)

Write an SQL statement with three CTEs and a SELECT statement using the CTEs to generate association rules of size 2 along with evaluation measures of support, confidence, and lift. An association rule indicates the LHS and RHS of a rule. A rule in the result contains a permutation of a combination to generate the LHS and RHS of the rule. Each row should contain the rule text (concatenation of LHS -> RHS) and evaluation measures (support, confidence, and lift) for the rule. Create three CTEs for the pairs (like the statement for problem 1), rules, and counts followed by a SELECT statement using the CTEs. As in problem 1, baskets are identified by a combination of customer vendor key, date key, and branch plant key. Only consider shipments (TransTypeKey = 5). Sort the result by the rule text.

Problem 4: Association rule input as a cross product of baskets and items

Write a SELECT statement to generate baskets identified by customer vendor key, date key, and branch plant key. Generate baskets containing two or more items for customers residing in CA, calendar year of 2022, and company key = 1. The result should contain customer vendor key, date key, branch plant key, item master key, and a basket indicator (1 if item is in the basket, 0 otherwise). Order the result by customer vendor key, date key, and branch plant key in ascending order.

Problem 5: Association rule input as a nested list of items in each basket

Write a SELECT statement to generate shipment baskets (TransType = 5) with baskets identified by a combination of customer vendor key date key, and branch plant key. Only generate baskets with two or more items. The result should contain customer vendor key, date key, branch plant key, item master key, and an array of item master keys. Order the result by customer vendor key, date key, and branch plant key in ascending order.

Problem 6: CTE using the ROW_NUMBER analytic function for event history ordering

Write a CTE with a SELECT statement to generate shipments for a combination of customer vendor key and branch plant key. The entity in the input for a classification algorithm is the combination of customer vendor key and branch plant key. ItemMasterKey identifies items. Unit cost is the weight in descending order. Only generate rows for shipments (TransTypeKey = 5), CompanyKey = 5, and first quarter of 2022. The result should contain customer vendor key, branch plant key, customer state, customer zip, item master key, item unit cost, and the row number of the item. Partition the analytic function by customer vendor key and branch plant key. Order the analytic function by descending item unit cost. After the CTE, write a simple SELECT statement to retrieve all rows and columns of the CTE. Sort by customer vendor key and branch plant key.

Problem 7: Classification algorithm input using a CTE and a query for entities with only one event

Write a SELECT statement to generate rows with only one shipment (TransType = 5) for company key 5 in first quarter 2022. Eliminate rows not having exactly 1 shipment. Essentially, this query flattens the result of query 6 to entities having one event with default values for events 2 and 3. Use the CTE from problem 6 to order the event history. The result should contain the customer vendor key, branch plant key, customer state, customer zip, item master key, unit cost, and default values (0) for items 2 and 3 (both item master key and unit cost). The combination of the customer vendor key and branch plant key represent the entity in a row. The item number and unit cost in a row should be the values of the item with maximum unit cost related to the entity (combination of customer vendor key and branch plant key). Order the result by customer vendor key and branch plant key.

Problem 8: Classification algorithm input using a CTE and query for entities with exactly two events

Write a SELECT statement to generate rows with exactly two shipments (TransType = 5) for company key 5 in first quarter 2022. Eliminate rows not having exactly 2 shipments. Essentially, this query flattens the result of query 6 to entities having exactly two shipments. Use the CTE from problem 6 to order the event history. The result should contain the customer vendor key, branch plant key, customer state, customer zip, item master key for item 1, unit cost for item 1, item master key for item 2, unit cost for item 2, and default values (0) for item 3 (both item master key and unit cost). The combination of the customer vendor key and branch plant key represent the entity in a row. The item number and unit cost values in a row should be the values of the items with row numbers 1 and 2. Order the result by customer vendor key and branch plant key.

Problem 9: Classification algorithm input using a CTE and query for entities with exactly three events

Write a SELECT statement to generate rows with 3 or more shipments (TransType = 5) for company key 5 in first quarter 2022. The result should contain only the largest 3 values for unit cost for these shipments. Use the CTE from problem 6 to order the event history. The result should contain the customer vendor key, branch plant key, customer state, customer zip, item master key for item 1, unit cost for item 1, item master key for item 2, unit cost for item 2, and item master key for item 3, unit cost for item 3. The combination of the customer vendor key and branch plant key represent the entity in a row. The item number and unit cost values in a row should be the values of the shipments with row numbers 1 to 3. Order the result by customer vendor key and branch plant key.

Problem 10: Classification algorithm input using a CTE and union of queries for entities with a range of event sizes (1 to 3)

The result should contain rows with one to three shipments. The result should contain columns for the customer vendor key, branch plant key, customer state, customer zip, item master key for item 1, unit cost for item 1, item master key for item 2, unit cost for item 2, and item master key

for item 3, and unit cost for item 3. Use the CTE from problem 6 to order the event history along with a UNION of SELECT statements from problems 7 to 9. Order the result by customer vendor key and branch plant key. Note that a SELECT statement can only contain a single ORDER BY clause at the end of the statement. Thus, your statement should remove the ORDER BY clauses in statements for problems 7 to 9.

Grading

Upon completion of this assignment, you should read the self-evaluation rubric and document your self-evaluation using the Reflective Quiz for the Module 6 Assignment. Before evaluation using the self-evaluation rubric, you should create a document with a SELECT statement and snapshot of partial results for each problem. You should not perform evaluation until each statement executes without syntax errors. After completing the assignment, you should apply the self-evaluation rubric for a detailed review of each problem. You should use the reflective quiz to document your self-evaluation.