****

**NAME: Laiba Batool**

**ROLL NO: 21i1781**

**SECTION: BS DS M**

**DATAWARE HOUSE PROJECT**

**STAR SCHEMA:**

**Dimension Tables**

1. **ProductDimension**: This table stores information about products sold by the company. It includes fields for ProductID (primary key), productName, productPrice, and supplierID.
2. **CustomerDimension**: This table contains customer information. It includes CustomerID (primary key), CustomerName, and Gender.
3. **DateDimension**: Date-related information is stored in this table. It includes DateID (primary key), d\_Year (year), d\_Month (month), and d\_Day (day).
4. **StoreDimension**: Store-related data is stored in this table, including StoreID (primary key) and storeName.
5. **SupplierDimension**: This table stores information about product suppliers. It includes SupplierID (primary key) and supplierName.

**Fact Table**

**FactSales**: This table records sales transactions and is the central fact table in the data warehouse. It includes fields for OrderID (primary key), ProductID (foreign key referencing ProductDimension), CustomerID (foreign key referencing CustomerDimension), StoreID (foreign key referencing StoreDimension), DateID (foreign key referencing DateDimension), QuantityOrdered, and TotalSale.

**(You need to print the output of the first 50 tuples at the console and copy this to your report as well.)**

Transaction{orderId=178407, orderDate=2019-04-01, productId=17, customerId=10, customerName='Mia Nelson', gender='Female', quantityOrdered=1, totalSale=399.99}

0

Transaction{orderId=189813, orderDate=2019-04-01, productId=81, customerId=3, customerName='Olivia Davis', gender='Female', quantityOrdered=1, totalSale=999.99}

1

Transaction{orderId=186650, orderDate=2019-04-01, productId=2, customerId=23, customerName='Ethan Young', gender='Male', quantityOrdered=1, totalSale=1299.99}

2

Transaction{orderId=177831, orderDate=2019-04-01, productId=66, customerId=12, customerName='Amelia Thomas', gender='Female', quantityOrdered=1, totalSale=549.99}

3

Transaction{orderId=192285, orderDate=2019-04-01, productId=86, customerId=13, customerName='Oliver Garcia', gender='Male', quantityOrdered=1, totalSale=1499.99}

4

Transaction{orderId=191585, orderDate=2019-04-01, productId=38, customerId=10, customerName='Mia Nelson', gender='Female', quantityOrdered=1, totalSale=1299.99}

5

Transaction{orderId=186754, orderDate=2019-04-01, productId=12, customerId=5, customerName='Ava Martinez', gender='Female', quantityOrdered=1, totalSale=149.99}

6

Transaction{orderId=177687, orderDate=2019-04-01, productId=46, customerId=30, customerName='Aria Perez', gender='Female', quantityOrdered=1, totalSale=1999.99}

7

Transaction{orderId=184067, orderDate=2019-04-01, productId=95, customerId=36, customerName='Aubrey Murphy', gender='Female', quantityOrdered=1, totalSale=1899.99}

8

Transaction{orderId=185103, orderDate=2019-04-01, productId=47, customerId=46, customerName='Savannah Allen', gender='Female', quantityOrdered=1, totalSale=1499.99}

9

Total processed count in this batch: 10

processAndJoin method called in DataProcessor

Transaction{orderId=181561, orderDate=2019-04-01, productId=80, customerId=17, customerName='Grayson Lewis', gender='Male', quantityOrdered=1, totalSale=99.99}

10

Transaction{orderId=181552, orderDate=2019-04-01, productId=66, customerId=3, customerName='Olivia Davis', gender='Female', quantityOrdered=1, totalSale=549.99}

11

Transaction{orderId=189001, orderDate=2019-04-01, productId=34, customerId=42, customerName='Hazel Turner', gender='Female', quantityOrdered=1, totalSale=1399.99}

12

Transaction{orderId=182666, orderDate=2019-04-01, productId=19, customerId=40, customerName='Layla Ward', gender='Female', quantityOrdered=1, totalSale=2999.99}

13

Transaction{orderId=186576, orderDate=2019-04-01, productId=52, customerId=8, customerName='Isabella Brown', gender='Female', quantityOrdered=1, totalSale=199.99}

14

Transaction{orderId=189270, orderDate=2019-04-01, productId=4, customerId=19, customerName='Logan Adams', gender='Male', quantityOrdered=1, totalSale=349.99}

15

Transaction{orderId=186297, orderDate=2019-04-01, productId=85, customerId=13, customerName='Oliver Garcia', gender='Male', quantityOrdered=1, totalSale=249.99}

16

Transaction{orderId=193992, orderDate=2019-04-01, productId=75, customerId=7, customerName='Jackson Taylor', gender='Male', quantityOrdered=1, totalSale=249.99}

17

Transaction{orderId=182960, orderDate=2019-04-01, productId=45, customerId=4, customerName='Liam Wilson', gender='Male', quantityOrdered=1, totalSale=99.99}

18

Transaction{orderId=191788, orderDate=2019-04-01, productId=15, customerId=5, customerName='Ava Martinez', gender='Female', quantityOrdered=1, totalSale=1299.99}

19

Total processed count in this batch: 10

processAndJoin method called in DataProcessor

Transaction{orderId=181016, orderDate=2019-04-01, productId=100, customerId=16, customerName='Evelyn Walker', gender='Female', quantityOrdered=1, totalSale=799.99}

20

Transaction{orderId=193621, orderDate=2019-04-01, productId=4, customerId=37, customerName='Jackson Reed', gender='Male', quantityOrdered=1, totalSale=349.99}

21

Transaction{orderId=176831, orderDate=2019-04-01, productId=53, customerId=14, customerName='Harper Hall', gender='Female', quantityOrdered=1, totalSale=149.99}

22

Transaction{orderId=193823, orderDate=2019-04-01, productId=22, customerId=45, customerName='Owen Phillips', gender='Male', quantityOrdered=1, totalSale=199.99}

23

Transaction{orderId=189152, orderDate=2019-04-01, productId=6, customerId=44, customerName='Aurora Foster', gender='Female', quantityOrdered=1, totalSale=499.99}

24

Transaction{orderId=187999, orderDate=2019-04-01, productId=88, customerId=4, customerName='Liam Wilson', gender='Male', quantityOrdered=1, totalSale=799.99}

25

Transaction{orderId=183822, orderDate=2019-04-01, productId=59, customerId=13, customerName='Oliver Garcia', gender='Male', quantityOrdered=1, totalSale=699.99}

26

Transaction{orderId=176603, orderDate=2019-04-01, productId=27, customerId=29, customerName='Samuel Mitchell', gender='Male', quantityOrdered=1, totalSale=499.99}

27

Transaction{orderId=189256, orderDate=2019-04-01, productId=76, customerId=5, customerName='Ava Martinez', gender='Female', quantityOrdered=1, totalSale=349.99}

28

Transaction{orderId=180520, orderDate=2019-04-01, productId=30, customerId=48, customerName='Skylar Johnson', gender='Female', quantityOrdered=1, totalSale=199.99}

29

Total processed count in this batch: 10

processAndJoin method called in DataProcessor

Transaction{orderId=180221, orderDate=2019-04-01, productId=80, customerId=15, customerName='Elijah Robinson', gender='Male', quantityOrdered=1, totalSale=99.99}

30

Transaction{orderId=177553, orderDate=2019-04-01, productId=32, customerId=36, customerName='Aubrey Murphy', gender='Female', quantityOrdered=1, totalSale=329.99}

31

Transaction{orderId=179069, orderDate=2019-04-01, productId=66, customerId=11, customerName='Aiden White', gender='Male', quantityOrdered=1, totalSale=549.99}

32

Transaction{orderId=193851, orderDate=2019-04-01, productId=83, customerId=38, customerName='Penelope Reed', gender='Female', quantityOrdered=1, totalSale=69.99}

33

Transaction{orderId=177219, orderDate=2019-04-01, productId=19, customerId=38, customerName='Penelope Reed', gender='Female', quantityOrdered=1, totalSale=2999.99}

34

Transaction{orderId=180003, orderDate=2019-04-01, productId=70, customerId=5, customerName='Ava Martinez', gender='Female', quantityOrdered=1, totalSale=499.99}

35

Transaction{orderId=178720, orderDate=2019-04-01, productId=25, customerId=40, customerName='Layla Ward', gender='Female', quantityOrdered=2, totalSale=199.98}

36

Transaction{orderId=188535, orderDate=2019-04-01, productId=27, customerId=17, customerName='Grayson Lewis', gender='Male', quantityOrdered=1, totalSale=499.99}

37

Transaction{orderId=184548, orderDate=2019-04-01, productId=13, customerId=21, customerName='Carter Rodriguez', gender='Male', quantityOrdered=1, totalSale=649.99}

38

Transaction{orderId=184775, orderDate=2019-04-01, productId=78, customerId=42, customerName='Hazel Turner', gender='Female', quantityOrdered=1, totalSale=1199.99}

39

Total processed count in this batch: 10

processAndJoin method called in DataProcessor

Transaction{orderId=177130, orderDate=2019-04-01, productId=98, customerId=19, customerName='Logan Adams', gender='Male', quantityOrdered=1, totalSale=499.99}

40

Transaction{orderId=187228, orderDate=2019-04-01, productId=69, customerId=38, customerName='Penelope Reed', gender='Female', quantityOrdered=1, totalSale=699.99}

41

Transaction{orderId=186362, orderDate=2019-04-01, productId=6, customerId=2, customerName='Noah Smith', gender='Male', quantityOrdered=1, totalSale=499.99}

42

Transaction{orderId=192696, orderDate=2019-04-01, productId=7, customerId=41, customerName='Dylan Nelson', gender='Male', quantityOrdered=1, totalSale=249.99}

43

Transaction{orderId=181218, orderDate=2019-04-01, productId=89, customerId=13, customerName='Oliver Garcia', gender='Male', quantityOrdered=1, totalSale=6499.99}

44

Transaction{orderId=191947, orderDate=2019-04-01, productId=73, customerId=41, customerName='Dylan Nelson', gender='Male', quantityOrdered=1, totalSale=1099.99}

45

Transaction{orderId=183517, orderDate=2019-04-01, productId=11, customerId=32, customerName='Addison Cooper', gender='Female', quantityOrdered=1, totalSale=2499.99}

46

Transaction{orderId=191334, orderDate=2019-04-01, productId=44, customerId=41, customerName='Dylan Nelson', gender='Male', quantityOrdered=1, totalSale=179.99}

47

Transaction{orderId=180711, orderDate=2019-04-01, productId=29, customerId=23, customerName='Ethan Young', gender='Male', quantityOrdered=1, totalSale=349.99}

48

Transaction{orderId=186362, orderDate=2019-04-01, productId=15, customerId=26, customerName='Grace King', gender='Female', quantityOrdered=1, totalSale=1299.99}

49

**HYBRID JOIN ALGORITHM:**

Input:

- DataProcessor: Object responsible for managing data processing.

- Connection conn: Database connection for data retrieval.

- BlockingQueue<Transaction> queue: Blocking queue for incoming transactions.

- StreamGenerator streamGenerator: Object responsible for generating data streams.

Initialization:

- joinQueue: Deque to keep track of product IDs to be joined.

- MD\_SEGMENT\_SIZE: Predefined segment size for master data.

Main Loop:

while not Thread.currentThread().isInterrupted() and not isProcessingComplete():

if not queue.isEmpty():

loadTransactions()

processedCount = loadAndProcessMD()

if processedCount > 0:

afterMDProcessing()

streamGenerator.adjustChunkSize(processedCount)

elif streamGenerator.isCompleted():

break

else:

sleep(3000 milliseconds)

Load Transactions:

loadTransactions():

transaction = queue.take()

productId = transaction.getProductId()

dataProcessor.addTransaction(transaction)

if productId not in joinQueue:

joinQueue.offer(productId)

Load and Process Master Data:

loadAndProcessMD():

processedCount = 0

if size of joinQueue >= MD\_SEGMENT\_SIZE:

segmentIds = []

for i in 1 to MD\_SEGMENT\_SIZE:

productId = joinQueue.poll()

segmentIds.add(productId)

dataProcessor.loadSegmentIntoDiskBuffer(conn, segmentIds)

processedCount = processAndJoinData(segmentIds)

return processedCount

After MD Processing:

afterMDProcessing():

try:

dataProcessor.processAndJoin(conn)

except SQLException:

Handle SQLException

Process and Join Data:

processAndJoinData(segmentIds):

processedCount = 0

for productId in segmentIds:

md = dataProcessor.getMasterData(productId)

if md is not null:

transactions = dataProcessor.getTransactions(productId)

if transactions is not null:

for transaction in transactions:

processJoinedData(transaction, md)

processedCount += 1

print("Total processed count in this batch: " + processedCount)

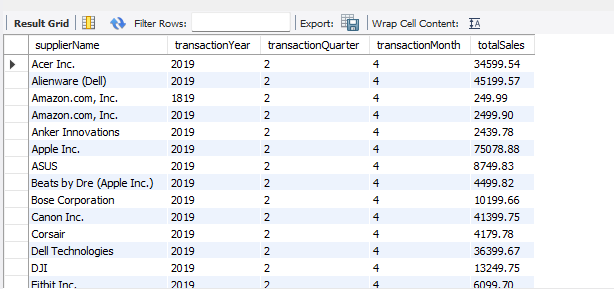
return processedCount

Check Processing Completion:

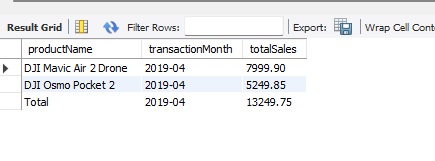
isProcessingComplete():

return streamGenerator.isCompleted() and queue.isEmpty()

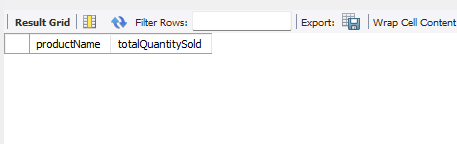
**OUTPUT OF OLAP QUEURIES:**#1



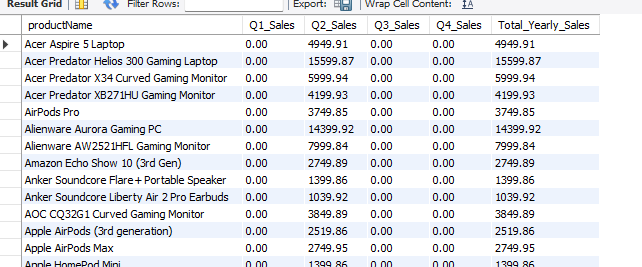
# 2



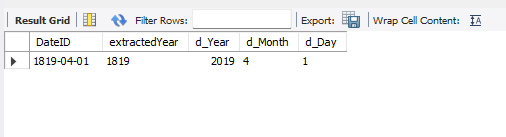
#3



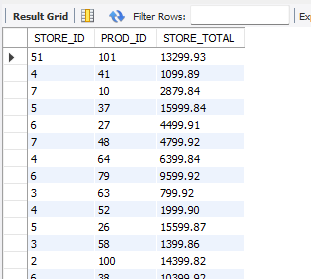
#4

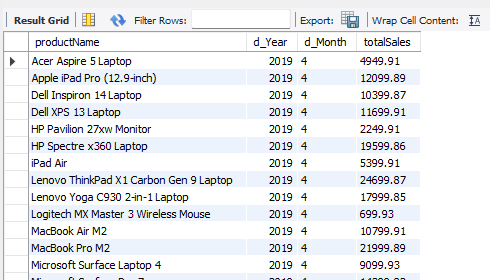


# 5

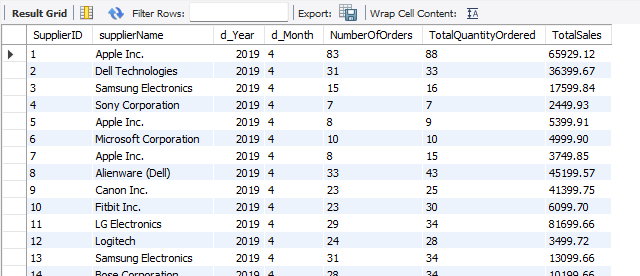


#6

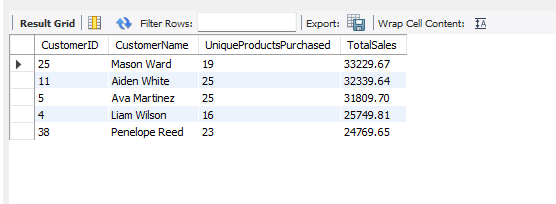


#7  
  


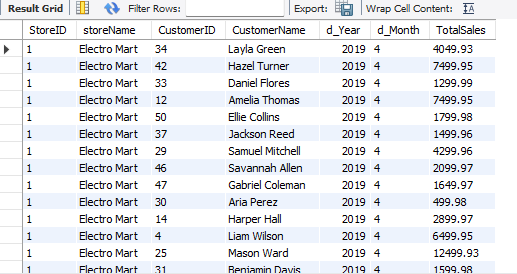
# 8



# 9



# 10



**TWO SHORTCOMINGS OF HYBRID JOIN:**

1. Complexity: HYBRIDJOIN involves managing multiple queues and segments, which adds complexity to the data processing workflow.
2. Delay: Due to the need to accumulate data before processing, there can be a delay in data integration, making it less suitable for real-time applications.

From the project, I learned about important data warehousing concepts such as data modeling and ETL (Extract, Transform, Load) processes. Additionally, I gained insights into the Hybrid Join technique, which is a powerful method for efficiently combining data from different sources, specifically transaction data and master data. Also I learned how to load data to datawarehouse and apply OLAP queries.