Peiyuan Sun 1001144745
David Hong Yang 1001409647

# CSC343 – Assignment 1

## Part 1

1, Find the manufacturers who make an item whose type is a descendant of "apparel" in the subcategory hierarchy/ies. Report the manufacturer ID, name, address, and phone number.

Cannot be expressed

2, Let's say a "singleton order" is one that includes exactly one item. Find all gold customers who have made at least one singleton order in 2016. Report their CID, and the date and time when they made their first and their last singleton order that year.

- Gold customers

$\mathit{GoldCustomer(CID)} := \Pi_{\mathit{CID}}\, \sigma_{\mathit{membership} = \text{“}\mathit{gold}\text{”}}\, \mathit{Customer}$

- Get all customers who have made at least one singleton order in 2016

$\mathit{AtLeastOneItem(OID)} := \Pi_{\mathit{OID}}\, \mathit{LineItem}$

$\mathit{AtLeastTwoItem(OID)} :=$
$\Pi_{\mathit{L1.OID}}\, \sigma_{\mathit{L1.OID} \neq \mathit{L2.OID}\ \wedge\ \mathit{L1.IID} = \mathit{L2.IID}}\, [(\rho_{\mathit{L1}}\mathit{LineItem}) \times (\rho_{\mathit{L2}}\mathit{LineItem})]$

$\mathit{SingletonOrder(OID)} := \mathit{AtLeastTwoItem} - \mathit{AtLeastOneItem}$

$\mathit{AllSingletonOrder(CID, When)} := \Pi_{\mathit{CID, When}}\, (\mathit{SingletonOrder} \bowtie \mathit{Order})$

- Filter all gold customers

$\mathit{GoldSingletonOrder(CID, When)} := \Pi_{\mathit{CID, When}}\, (\mathit{AllSingletonOrder} \bowtie \mathit{GoldCustomer})$

- Get the all customers' first and last order time. First order is all singleton orders - not first singleton orders. Last order is all singleton orders - not last singleton orders.

$\mathit{First(CID, first)} :=$
$\mathit{GoldSingletonOrder(CID, When)} -$
$(\Pi_{\mathit{G1.CID, G1.When}}\, \sigma_{\mathit{G1.CID} = \mathit{G2.CID}\ \wedge\ \mathit{G1.when} > \mathit{G2.When}}\, [(\rho_{\mathit{G1}}\mathit{GoldSingletonOrder}) \times (\rho_{\mathit{G2}}\mathit{GoldSingletonOrder})])$

$\mathit{Last(CID, last)} :=$
$\mathit{GoldSingletonOrder(CID, When)} -$
$(\Pi_{\mathit{G1.CID, G1.When}}\, \sigma_{\mathit{G1.CID} = \mathit{G2.CID}\ \wedge\ \mathit{G1.when} < \mathit{G2.When}}\, [(\rho_{\mathit{G1}}\mathit{GoldSingletonOrder}) \times (\rho_{\mathit{G2}}\mathit{GoldSingletonOrder})])$

- Retrieve the answer

$\mathit{Answer(CID, first, last)} := \Pi_{\mathit{CID, first, last}}\, (\mathit{First} \bowtie \mathit{Last})$

3, Suppose we consider two orders to be "identical" if they contain exactly the same items (ignoring quantity). Find all pairs of customers who have made identical orders on the same day. Report each customer's CID and OID for the order that was identical. A pair could have multiple identical orders on the same day. If so, report them all.

- Get items with only IID attribute

$ItemList(IID) := \Pi_{IID}\, LineItem$

- Get all orders with only OID attribute

$OrderList(OID) := \Pi_{OID}\, LineItem$

- Get all orders with their respective item ids

$AllTransaction(OID, IID) := \Pi_{OID,\, IID}\, LineItem$

- Get all possible combinations of orders and items

$AllPossibility(OID, IID) := ItemList \times OrderList$

- Get all the items that do not exist in the orders (but does exist in other orders)

$NotExist(OID, IID) := AllPossibility - AllTransaction$

- Get all order pairs that are not identical

$NotIdentical(OID\_1, OID\_2) :=$
$\Pi_{A.OID,\, N.OID}\, \sigma_{A.IID\, =\, N.IID}\, [(\rho_{A} AllTransaction) \times (\rho_{N} NotExist)]$

- Remove duplicates from the order pairs

$NoDuplicateNotIdentical(OID\_1, OID\_2) := \Pi_{OID\_1,\, OID\_2}\, \sigma_{OID\_1\, >\, OID\_2}\, NotIdentical$

- Get all possible pairs of orders with no duplicates

$AllPair(OID\_1, OID\_2) := \Pi_{L1.OID,\, L2.OID}\, \sigma_{L1.OID\, >\, L2.OID}\, [(\rho_{L1} LineItem) \times (\rho_{L2} LineItem)]$

- Get identical order pairs

$Identical(OID\_1, OID\_2) := AllPair - NoDuplicateNotIdentical$

- Get CID corresponding to OID_1

$Answer\_0(OID\_1, CID\_1, OID\_2) := \Pi_{I.OID\_1,\, O.CID,\, I.OID\_2}\, \sigma_{I.OID\_1\, =\, O.OIDof}\, [(\rho_{I} Identical) \times (\rho_{O} Order)]$

- Get final answer with required attributes

*Answer(OID_1, CID_1, OID_2, CID_2) :=*
$\Pi_{A.OID\_1,\ A.CID\_1,\ A.OID\_2,\ O.CID}\ \sigma_{A.OID\_2 = O.OID}\ [(\rho_A Answer\_0) \times (\rho_O Order)]$

4, Find all customers who have a silver membership, have placed at least two orders in 2014, fewer than 2 orders in 2015, and no orders at all in 2016. Report the CID.

- Silver customers

*SilverCustomer(CID) :=* $\Pi_{CID}\ \sigma_{membership = \text{“silver”}}\ Customer$

- Get all customers with at least two orders in 2014

*AtLeast2OrderIn2014(CID) :=*
$\Pi_{O1.CID}\ \sigma_{O1.OID \neq O2.OID \wedge O1.CID = O2.CID \wedge O1.When = 2014 \wedge O2.When = 2014}\ [(\rho_{O1}Order) \times (\rho_{O2}Order)]$

- Get all customers with at least two orders in 2015

*AtLeast2OrderIn2015(CID) :=*
$\Pi_{O1.CID}\ \sigma_{O1.OID \neq O2.OID \wedge O1.CID = O2.CID \wedge O1.When = 2015 \wedge O2.When = 2015}\ [(\rho_{O1}Order) \times (\rho_{O2}Order)]$

- Get all customers with fewer than two orders in 2015

*FewerThan2OrderIn2015(CID) :=* $\Pi_{CID}\ Customer - AtLeast2OrderIn2015$

- Get all customers with at least one order in 2016

*AtLeast1OrderIn2016(CID) :=* $\Pi_{CID}\ \sigma_{When = 2016}\ Order$

- Get all customers with no orders in 2016

*NoOrderIn2016(CID) :=* $\Pi_{CID}\ Customer - AtLeast1OrderIn2016$

- Answer is the intersection between all contraints

*Answer(CID) :=* $SilverCustomer \cap AtLeast2OrderIn2014 \cap FewerThan2OrderIn2015 \cap NoOrderIn2016$

5, Let’s say the “top cost” on any order is the cost of the most expensive item. (There could be several items tied for that top cost.) Among all the orders a customer places in a year, let’s say their “skimpiest” order is the one whose top cost is the lowest. (There could be several orders tied for skimpiest.) For each customer who has ever placed an order, find their skimpiest order. If several orders for that customer are tied for skimpiest, report them all. Report the customer ID, order ID, and the order’s top cost.

- Get items with more details (price in this case)

*ItemDetail(OID, price) :=* $\Pi_{OID,\ price}\ Item \bowtie LineItem$

- Get all orders that do not have their top cost price

*NotTopCost(OID, price) :=*
$\Pi_{I1.OID,\ I1.price}\,\sigma_{I1.OID\ =\ I2.OID\ \wedge\ I1.price\ <\ I2.price}\ [(\rho_{I1}\mathit{ItemDetail}) \times (\rho_{I2}\mathit{ItemDetail})]$

- Get all orders with the corresponding price of their top cost item

$\mathit{TopCost(OID,\ price)} := \Pi_{OID,\ price}\,\mathit{ItemDetail} - \mathit{NotTopCost}$

- Natural join to get CID information

$\mathit{TopCostOrder(CID,\ OID,\ price)} := \Pi_{CID,\ OID,\ price}\,\mathit{TopCost} \bowtie \mathit{Order}$

- Get not skimpiest order

*NotSkimpiestOrder(CID, OID, price) :=*
$\Pi_{T1.CID,\ T1.OID,\ T1.price}\,\sigma_{T1.CID\ =\ T2.CID\ \wedge\ T1.price\ >\ T2.price}\ [(\rho_{T1}\mathit{TopCostOrder}) \times (\rho_{T2}\mathit{TopCostOrder})]$

- Answer is obtained by applying set difference between all top cost orders and not skimpiest orders

$\mathit{Answer(CID,\ OID,\ price)} := \Pi_{CID,\ OID,\ price}\,\mathit{TopCostOrder} - \mathit{NotSkimpiestOrder}$

6, Find every order that includes at least one item for which reviewers unanimously gave it a rating of 0 1 and at least one item for which reviewers unanimously gave it a rating of 52 . Report the customer ID, customer's last name and first name, order ID, and when the order was placed.

- Get all items that have only more than one unique rating (e.g. a '1' and a '4')

*NotUnanimouslyItem(IID, rating) :=*
$\Pi_{R1.CID,\ R1.IID,\ R1.rating}\,\sigma_{R1.IID\ =\ R2.IID\ \wedge\ R1.rating\ \neq\ R2.rating}\ [(\rho_{R1}\mathit{Review}) \times (\rho_{R2}\mathit{Review})]$

- Get all items that have only one rating (e.g. all 1's, all 5's, etc)

$\mathit{UnanimouslyItem(IID,\ rating)} := \Pi_{IID,\ rating}\,\mathit{Review} - \mathit{NotUnanimouslyItem}$

- Get all items with unanimous 0's

$\mathit{Rating0Item(IID)} := \Pi_{IID}\,\sigma_{Rating\ =\ 0}\ \mathit{UnanimouslyItem}$

- Get all items with unanimous 5's

$\mathit{Rating5Item(IID)} := \Pi_{IID}\,\sigma_{Rating\ =\ 5}\ \mathit{UnanimouslyItem}$

- Get all orders with unanimous 0's

$\mathit{Rating0Order(OID)} := \Pi_{OID}\,\mathit{Rating0Item} \bowtie \mathit{LineItem}$

- Get all orders with unanimous 5's

$Rating5Order(OID) := \Pi_{OID} Rating5Item \bowtie LineItem$

- Get all orders with unanimous 0's and 5's

$RequiredOrder(OID) := Rating0Order \cap Rating5Order$

- Get answer which has additional attributes

$Answer(CID, lastName, firstName, OID, When) :=$
$\Pi_{CID,\ lastName,\ firstName,\ OID,\ When} Customer \bowtie Order \bowtie RequiredOrder$

7, Find all pairs of customers c1 and c2 such that: c2 has reviewed at least one item, and c1 assessed every review of c2 as helpful.

- Reviews that readers assessed as not helpful

$NotHelpful(reader, reviewer) := \Pi_{reader,\ reviewer} \sigma_{helpful\ =\ \text{“}no\text{”}} Helpfulness$

- Set difference ensures that reader has marked all reviews made by reviewer helpful. Also, if a reviewer is in 'Helpfulness' relation, he must have made at least one review

$Answer(reader, reviewer) := \Pi_{reader,\ reviewer} Helpfulness - NotHelpful$

8, For every item that has been ordered, find the last customer to order it. Report the item ID and the customer ID of the customer who ordered it last. If several customers are tied to be last to order a particular item, report a tuple for each of these customers.

- Get all items with desired attributes

$OrderItemDetail(IID, CID, when) := \Pi_{IID,\ CID,\ when} (Order \bowtie LineItem)$

- Get all customers that are not the last one to order corresponding item

$NotLast(IID, CID) :=$
$\Pi_{O1.IID,\ O1.CID} \sigma_{O1.IID\ =\ O2.IID\ \wedge\ O1.When\ <\ O2.When\ \wedge\ O1.CID\ \neq\ O2.CID} [(\rho_{O1} OrderItemDetail) \times (\rho_{O2} OrderItemDetail)]$

- Get all customers that are the last ones to order the item

$Answer(IID, CID) := \Pi_{IID,\ CID} OrderItemDetail - NotLast$

9, Find all the customers who have given a review that at most one reader assessed as helpful. For each of these customers, find every review that had more “yes” (helpful) assessments than “no” assessments. Report the customer ID, item ID, and item price. (A customer will appear multiple times if they have more than one qualifying review.)

Cannot be expressed

10, Find all customers who have given at least three reviews, and for whom the rating they give has always gone down over time from review to review. (This customer has grown increasingly dissatisfied, so maybe we should reach out to him or her.) Report the customer ID, last name, and email address, and the item ID for the last item they reviewed.

- Get all customers who have made at least three reviews , and the ratings of the items have gotten lower over time

$DissatisfiedID(CID) :=$
$\Pi_{R1.CID}\, \sigma_{R1.CID = R2.CID = R3.CID \wedge R1.When > R2.When > R3.When \wedge R1.rating > R2.rating > R3.rating}$
$[(\rho_{R1}\ Review) \times (\rho_{R2}\ Review) \times (\rho_{R3}\ Review)]$

- Get all items which these customers have purchased (this is necessary because a customer may have more than 3 reviews

$DissatisfiedReview(CID, IID, When) := \Pi_{CID,\ IID,\ When}\, DissatisfiedID \bowtie Review$

- Get all items which are not ordered last by these customers

$NotLastItem(CID, IID) :=$
$\Pi_{D1.CID,\ D1.IID}\, \sigma_{D1.CID = D2.CID \wedge D1.When < D2.When}\, [(\rho_{D1}\ DissatisfiedReview) \times (\rho_{D2}\ DissatisfiedReview)]$

- Get all items which are ordered last

$LastItem(CID, IID) := \Pi_{CID,\ IID}\, DissatisfiedReview - NotLastItem$

- Get answer with additional attributes

$Answer := \Pi_{CID,\ lastName,\ email,\ IID}\, (LastItem \bowtie Customer)$

11, A “top-level category” is one that is not a subcategory of anything else. Find all customers who have reviewed an item in each top-level category. Report just the customer ID. Note: An item type that has no subcategories and no parent category — it is not connected to any of the hierarchies — is considered a top-level category. We have to look in the Item relation to find these.

- Get all item types that are a subcategory of another item

$LowCategory(type) := \Pi_{a}\, Subcategory$

- Convert these item types to item ID's

$LowItem(IID) := \Pi_{IID} (LowCategory \bowtie Item)$

- Get all top level items (which must not exist as a low level item).

$TopItem(IID) := \Pi_{IID} Item - LowItem$

- Get Customer ID's for customers who have made a review of a top level item

$Answer(CID) := \Pi_{CID} (Review \bowtie TopItem)$

12, Find the orders with at least one item, and for which every item on the order had a type that was either "book" or a direct a subcategory of "book". Report the order ID.

- Get all item types that are a direct subcategory of 'book'

$DirectSubcategoryOfBook(type) := \Pi_{a}\, \sigma_{b\, =\, \text{“book”}}\, Subcategory$

- Get items that are either of type book or a direct subcategory of book

$RequiredItem(IID) := (\Pi_{IID}\, \sigma_{type\, =\, \text{“book”}}\, Item) \cup (\Pi_{IID} (Item \bowtie DirectSubcategoryOfBook))$

- Get orders with at least one required item

$Answer(OID) := \Pi_{OID} (LineItem \bowtie RequiredItem)$

13, Find the orders with more than three items, and for which at least half of the items have a category that is not "book". Report the order ID, customer ID, and the credit that they used.

Cannot be expressed

Part 2

1, A customer who reviews an item must have ordered that item.

- There cannot be a customer who reviews an item and does not order it

$\Pi_{CID} Review - \Pi_{CID} Order = \emptyset$

2, Orders made by gold members have no limit on the items that can be included. However, orders made by silver members must include at least one item costing over $50, and orders made by non-members cannot include any items costing under $50.

Peiyuan Sun 1001144745
David Hong Yang 1001409647

- Split the integrity constraint into two cases. One for silver member, the other for none members

- Get all items with price over 50

$ItemsOver50(IID) := \Pi_{IID}\, \sigma_{price > 50}\, Item$

- Get all orders that have no item over 50

$NoOrdersOver50(OID) := \Pi_{OID}\, LineItem - \Pi_{OID}\, (LineItem \bowtie ItemsOver50)$

- Get all silver customers

$SilverMember(CID) := \Pi_{CID}\, \sigma_{membership = \text{“silver”}}\, Customer$

- There should be no overlap between silver member orders and orders with all items not over 50

$\Pi_{OID}\, (SilverMember \bowtie Order) \cap NoOrdersOver50 = \varnothing$

- Get all non member customers

$NoneMember(CID) := \Pi_{CID}\, \sigma_{membership = \text{“none”}}\, Customer$

- Get all items with price less than or equal to 50

$ItemsUnder50(IID) := \Pi_{IID}\, \sigma_{price <= 50}\, Item$

- There cannot exist any none member that has an order consisting of items

$\Pi_{OID}\, (Order \bowtie NoneMember) - \Pi_{OID}\, (LineItem \bowtie ItemsUnder50) = \varnothing$