University of Toronto
CSC343, Winter 2017

Assignment 1:
Anthony Tam, 1002583402
Adam Capparelli 1001231473

Part 1: Queries

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.

   – All pairs of items and orders.

   $$ItemAndOrderPairs(T1.OID, T1.IID, T2.OID, T2.IID) :=$$
   $$\Pi_{T1.OID,\ T1.IID,\ T2.OID,\ T2.IID}$$
   $$(\rho_{T1}(Order \bowtie LineItem)) \times (\rho_{T2}(Order \bowtie LineItem))$$

   – Non singleton orders.

   $$NonSingleton(OID) :=$$
   $$\Pi_{(\rho_{OID}T1.OID)}$$
   $$\sigma_{T1.OID=T2.OID\ \wedge\ T1.IID \neq T2.IID}$$
   $$ItemAndOrderPairs$$

   – Order IDs of Singleton orders.

   $$Singleton(OID) :=$$
   $$\Pi_{(\rho_{OID}T1.OID)}$$
   $$ItemAndOrderPairs - NonSingleton$$

– Customer IDs of all the gold members.

$$GoldMembers(CID) :=$$
$$\Pi_{CID}$$
$$\sigma_{membership='gold'}$$
$$Customers$$

– Order information of all the Singleton Orders.

$$SingletonInfo(OID, CID, when, creditCard, number) :=$$
$$Singleton \bowtie Order$$

– Order information of all the Singleton Orders made by Gold Customers in 2016.

$$2016GoldOrders(OID, CID, when, creditCard, number) :=$$
$$\sigma_{when.year=2016}$$
$$(Orders \bowtie GoldMembers)$$

– OID of all the Singleton Orders made by Gold Customers in 2016 that were not the last order.

$$NotLastOrder(OID :=$$
$$\Pi_{OID}$$
$$\sigma_{T1.when<T2.when \wedge T1.CID=T2.CID}$$
$$((\rho_{T1}2016GoldOrders) \times (\rho_{T2}2016GoldOrders))$$

– Order information of all the Singleton Orders made by Gold Customers in 2016 that were the last order.

$$LastOrders(OID, CID, when, creditCard, number) :=$$
$$GoldOrders \bowtie ((\Pi_{OID}GoldOrders) - NotLastOrder)$$

– OID of all the Singleton Orders made by Gold Customers in 2016 that were not the first order.

$$NotFirstOrder(OID) :=$$
$$\Pi_{OID}$$
$$\sigma_{T1.when>T2.when \ \wedge \ T1.CID=T2.CID}$$
$$((\rho_{T1}2016GoldOrders) \times (\rho_{T2}2016GoldOrders))$$

– Order information of all the Singleton Orders made by Gold Customers in 2016 that were the first order.

$$FirstOrders(OID, CID, when, creditCard, number) :=$$
$$GoldOrders \bowtie ((\Pi_{OID}GoldOrders) - NotFirstOrder)$$

– CID and date/time of all that customer's first and last Singleton Order in 2016.

$$FirstAndLastSingeltonOrders(CID, FirstOrders.when, LastOrders.when) :=$$
$$\Pi_{(\rho_{CID}FirstOrders.CID)}$$
$$FirstOrders \bowtie_{FirstOrders.CID=LastOrders.CID} LastOrders$$

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.

–Pairs of Orders with at least one difference.

$$AtLeastOneDifference(T1.OID, T1.IID, T1.CID, T1.when, T1.creditCard, T1.number, T1.quantity\, T2.OID, T2.IID, T2.CID, T2.when, T2.creditCard, T2.number, T2.quantity) :=$$
$$\sigma_{T1.OID \neq T2.OID \ \wedge \ T1.IID \neq T2.IID \ \wedge \ T1.when.day = T2.when.day \ \wedge \ T1.when.year = T2.when.year} (\rho_{T1}(Order \bowtie LineItem)) \times (\rho_{T2}(Order \bowtie LineItem))$$

–Pairs of Orders with no differences.

$$IdenticalPairs(T1.OID, T1.CID, T2.OID, T2.CID) :=$$
$$\Pi_{T1.OID,\ T1.CID,\ T2.OID,\ T2.CID} (\rho_{T1}(Order \bowtie LineItem)) \times (\rho_{T2}(Order \bowtie LineItem)) - AtLeastOneDifference$$

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.

–CID of customers who have placed at least two orders in 2014.

$$AtLeast2Orders2014(CID) :=$$
$$\Pi_{(\rho_{CID} T1.CID)} \sigma_{T1.OID \neq T2.OID \ \wedge \ T1.CID = T2.CID \ \wedge \ T1.when.year = T2.when.year = 2014} (\rho_{T1} Order) \times (\rho_{T2} Order)$$

–CID of customers who have placed at least two orders in 2015.

$$AtLeast2Orders2015(CID) :=$$
$$\Pi_{(\rho_{CID} T1.CID)} \sigma_{T1.OID \neq T2.OID \wedge T1.CID = T2.CID \wedge T1.when.year = T2.when.year = 2015} [(\rho_{T1} Order) \times (\rho_{T2} Order)]$$

–CID of customers who have placed less than two orders in 2015.

$$LessThan2Orders2015(CID) :=$$
$$(\Pi_{CID}\ \sigma_{when.year=2015}\ Order) - AtLeast2Orders2015$$

–CID of customers who have placed no orders in 2016.

$$NoOrders2016(CID) :=$$
$$(\Pi_{CID}\ Customer) - (\Pi_{CID}\ \sigma_{when.year=2016}\ Order)$$

–CID of customers who are silver members.

$$SilverCustomers(CID) :=$$
$$\Pi_{CID}\ \sigma_{membership='silver'}\ Customer$$

– CID of customers who are silver members, have placed at least 2 orders in 2014, less than 2 in 2015 and none in 2016.

$$SilverCustomersAtLeast22014LessThan22015None2016(CID) :=$$
$$SilverCustomers \cap NoOrders2016 \cap LessThan2Orders2015 \cap AtLeast2Orders2014$$

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.

–The cost of every item in every order.

$$AllItemsCost(OID, CID, price) :=$$
$$\Pi_{OID,CID,price}\ Item \bowtie LineItem \bowtie Order$$

–CID, OID, and price of every item that is not the top cost item for that order.

$$NotTopCost(OID, CID, price) :=$$
$$\Pi_{\rho_{OID}T1.OID,\ \rho_{CID}T1.CID,\ \rho_{price}T1.price}$$
$$\sigma_{T1.OID=T2.OID\ \wedge\ T1.price\leq T2.price}$$
$$(\rho_{T1}AllItemsCost) \times (\rho_{T2}AllItemsCost)$$

–CID, OID, and price of every item that is the top cost item for that order.

$$TopCost(OID, CID, price) :=$$
$$\Pi_{OID,CID,price}\ AllItemsCost - NotTopCost$$

–CID, OID, and price of the top cost item for every order that is not the skimpiest.

$$NotSkimpiest(OID, CID, price) :=$$
$$\Pi_{\rho_{OID}T1.OID,\ \rho_{CID}T1.CID,\ \rho_{price}T1.price}$$
$$\sigma_{T1.CID=T2.CID\ \wedge\ T1.price>T2.price}$$
$$(\rho_{T1}TopCost) \times (\rho_{T2}TopCost)$$

–CID, OID, and price of the top cost item for the skimpiest order for that customer.

$$Skimpiest(OID, CID, price) :=$$
$$\Pi_{OID,CID,price}\ TopCost - NotSkimpiest$$

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 5[2]. Report the customer ID, customer's last name and first name, order ID, and when the order was placed.

–The IID and review for every item where the review was not the highest rating.

$$NotHighestRating(IID, review) :=$$
$$\Pi_{\rho_{IID}T1.IID,\ \rho_{review}T1.review}$$
$$\sigma_{T1.IID=T2.IID\ \wedge\ T1.when\neq T2.when\ \wedge\ T1.review<T2.review}$$
$$(\rho_{T1}Review) \times (\rho_{T2}Review)$$

–The IID and review for every item where the review was not the lowest rating.

$$NotLowestRating(IID, review) :=$$
$$\Pi_{\rho_{IID}T1.IID,\rho_{review}T1.review}$$
$$\sigma_{T1.IID=T2.IID\ \wedge\ T1.when\neq T2.when\ \wedge\ T1.review>T2.review}$$
$$(\rho_{T1}Review) \times (\rho_{T2}Review)$$

[1] An item must have been reviewed at least once in order to pass this condition.
[2] Ditto!

–The IID for every item where the top review was a 0.

$ItemsWithAll0(IID) :=$
Π_{IID}
$\sigma_{review=0}$
$((\Pi_{IID,review}Review) - NotHighestRating)$

–The IID for every item where the lowest review was a 5.

$ItemsWithAll5(IID) :=$
Π_{IID}
$\sigma_{review=5}$
$((\Pi_{IID,review}Review) - NotLowestRating)$

–The OID for every order containing at least one item with all 0 reviews.

$OrdersWithA0(OID) :=$
Π_{OID}
$LineItem \bowtie ItemsWithAll0$

–The OID for every order containing at least one item with all 5 reviews.

$OrdersWithA5(OID) :=$
Π_{OID}
$LineItem \bowtie ItemsWithAll5$

–The OID for every order containing at least item one with all 5 reviews and at least one item with all 0 reviews.

$OrdersWithA0And5(OID) :=$
$OrdersWithA0 \cap OrdersWithA5$

–Customer ID, first name, last name, OID and time for all orders from the table OrdersWithA0And5.

$InfoForOrdersWithA0And5(CID, firstNamelastName, OID, when) :=$
$\Pi_{CID,firstName,lastName,OID,when}$
$(Customer \bowtie Order) \bowtie OrdersWithA0And5$

7. Find all pairs of customers c_1 and c_2 such that: c_2 has reviewed at least one item, and c_1 assessed every review of c_2 as helpful.

– All reviews which were marked as helpful.

$HelpfulReviews(c_1, c_2, IID) :=$
$\Pi_{\rho_{c_1}reviewer,\rho_{c_2}reader,\rho_{IID}item}$
$\sigma_{helpful='yes'}$
$Helpfulness$

– Fake list of all customers marking all reviews as helpful.

$AllHelpfulReviews(c_1, c_2, IID) :=$
$\Pi_{c1.CID,c2.CID,IID}$
$(\rho_{c1}Customer \times \rho_{c2}Customer \times Item)$

– Find the list of customers who did not mark every review of a user a helpful.

$NonPairs(CID) :=$

$(AllHelpfulReviews - HelpfulReaders)$

– All the pairs of customers who marked every review as helpful.

$AllReviewPairs(c_1, c_2) :=$
$(\Pi_{c_1,c_2} AllHelpfulReviews) - (\Pi_{c_1,c_2} NonPairs)$

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.

– CID and OID for every order that was not the last order for that item.

$NotLastOrder(CID, OID, IID) :=$
$\Pi_{\rho_{CID}T1.CID,\rho_{OID}T1.OID,\rho_{IID}T1.IID}$
$\sigma_{T1.IID=T2.IID \wedge T1.when<T2.when}$
$(\rho_{T1}(Order \bowtie LineItem)) \times (\rho_{T2}(Order \bowtie LineItem))$

– CID and IID for every order that was the last order for that item.

$LastOrder(CID, IID) :=$
$\Pi_{CID,IID}$
$(\Pi_{CID,OID,IID}(\rho_{T1}(Order \bowtie LineItem))) - NotLastOrder$

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.

– All customers who have at least 3 reviews.

$ThreeReviews(CID) :=$
$\Pi_{\rho_{CID}c1.CID}$
$\sigma_{r1.CID=r2.CID=r3.CID \wedge r1.IID!=r2.IID \wedge r2.IID!=r3.IID \wedge r1.IID!=r3.IID}$
$(\rho_{r1}Review \times \rho_{r2}Review \times \rho_{r3}Review)$

– Pairs of customer reviews.

$ReviewPairs(r1.CID, r1.IID, r1.when, r1.rating, r2.IID, r2.when, r2.rating) :=$
$\Pi_{r1.CID,r1.IID,r1.when,r1.rating,r2.IID,r2.when,r2.rating}$
$((\rho_{r1}Review \bowtie_{r1.CID=r2.CID} \rho_{r2}Review) \bowtie ThreeReviews)$

– Find pairs of reviews where the reviews are increasing or the dates are not in decreasing order.

$NotValid(CID) :=$
$\Pi_{r1.CID}$
$\sigma_{r1.review>r2.review \vee r1.when>r2.when}$
$ReviewPairs$

– Get the pairs of customers with descending reviews.

$$DecendingReviews(CID) :=$$
$$(\pi_{r1.CID} ReviewPairs) - NotValid$$

– Find the items which are not the last item the customer has reviewed.

$$NonMinIID(CID, IID) :=$$
$$\Pi_{\rho CIDr1.CID, \rho_{IID} r1.IID}$$
$$\sigma_{r1.CID=r2.CID \ \wedge \ r1.when<r2.when}$$
$$ReviewPairs \bowtie_{r1.CID=CID} DecendingReviews$$

– Get a list of customers and items which are the last item the customer reviewed.

$$MinIID(CID, IID) :=$$
$$(\Pi_{\rho_{CID} r1.CID, \rho_{r1.IID} IID} ReviewPairs) - NonMinIID$$

– Join these items to the customer and fetch the customer info.

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

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.

    – All items whose type is a top-level category.

    $$TopLevelItems(IID, type) :=$$
    $$\Pi_{IID, type}$$
    $$Item \bowtie ((\Pi_{type} Item) - (\Pi_{\rho_{type} a} Subcategory))$$

    – All CIDs and types of all reviews of a top level item.

    $$TopLevelReviews(CID, type) :=$$
    $$\Pi_{CID, type}$$
    $$TopLevelItems \bowtie Review$$

    – Ideal list where every customer who reviewed at least one top level item type reviewed every top level item type.

    $$IdealList(CID, type) :=$$
    $$(\Pi_{CID} TopLevelReviews) \times ((\Pi_{type} Item) - (\Pi_{\rho_{type} a} Subcategory))$$

    – CID of customers that were missing a review of at least one top level type.

    $$MissingAtLeastOneType(CID) :=$$
    $$\Pi_{CID}$$
    $$(IdealList - TopLevelReviews)$$

    – CID of customers that reviewed at least one item in every top level type.

    $$ReviewForEveryType(CID) :=$$
    $$(\Pi_{CID} TopLevelReviews) - MissingAtLeastOneType$$

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.

– OID and type of items for every order.

$$OIDAndType(OID, type) :=$$
$$\Pi_{OID,type}$$
$$Order \bowtie LineItem \bowtie Item$$

– All types that are book or a subcategory of book.

$$BookAndSubcategory(type) :=$$
$$(\Pi_{\rho_{type}a}\sigma_{b='book'}Subcategory) \cup (\Pi_{type}\sigma_{type='book'}Item)$$

– Ideal list where every OID has all types of items either book or subcategory of book.

$$IdealList(OID, type) :=$$
$$(\Pi_{OID}OIDAndType) \times BookAndSubcategory$$

– OID of orders that have at least one item that is not a book or subcategory of book.

$$MissingAtLeastOneType(OID) :=$$
$$\Pi_{OID}$$
$$(IdealList - OIDAndType)$$

– OID of orders where every item is a book or subcategory of book.

$$OrdersWithAllBooksOrSubcategoriesOfBook(CID) :=$$
$$(\Pi_{OID}OIDAndType) - MissingAtLeastOneType$$

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: Additional Integrity Constraints

1. A customer who reviews an item must have ordered that item.
$$(\Pi_{CID,IID}Review) - (\Pi_{CID,IID}(LineItem \bowtie Order)) = \emptyset$$

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.

– OID of all nonmembers with an item in the order under $50.

$$NonMembersUnder50(OID) :=$$
$$\Pi_{OID}$$
$$\sigma_{membership='none' \wedge price<50}$$
$$(Item \bowtie LineItem \bowtie Order \bowtie Customer)$$

– OID of all silver members with an item in the order over $50.

$$SilverMembersOver50(OID) :=$$
$$\Pi_{OID}$$
$$\sigma_{membership='silver' \wedge price>50}$$
$$(Item \bowtie LineItem \bowtie Order \bowtie Customer)$$

– OID of all silver members with all items in the order under $50.

$$SilverMembersUnder50(OID) :=$$
$$(\Pi_{OID}$$
$$(\Pi_{CID}$$
$$\sigma_{membership='silver'}$$
$$Order) \bowtie Customer \bowtie ((\Pi_{OID}Order) - SilverMembersOver50))[10pt]$$

$$SilverMembersUnder50 \cup NonMembersUnder50 = \emptyset$$