2017
FEB

CSC343 Database

ASSIGNMENT 1

BRAMEYAA RAGUPATHY (1001441816)
NAYEONG PARK (1000796739)

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.

-OID of singleton orders

$$Singletons(OID) \coloneqq \Pi_{OID}\ LineItem - \Pi_{O1.OID}\sigma_{O1.OID=O2.OID \wedge O1.IID!=O2.IID}\ (\mathrm{P}_{O1}LineItem \times \mathrm{P}_{O2}LineItem)$$

-Information of gold customers who have placed singleton orders at least once during 2016

$$AtLeastOnce(CID, d) \coloneqq \Pi_{CID,d}\sigma_{membership='gold' \ \wedge d.year=2016}Customer \bowtie Singletons$$

-Dates that are not the first day of singleton order

$$NotMin(CID, d) \coloneqq \Pi_{A2.CID,A2.d}\sigma_{A1.CID=A2.CID \ \wedge A1.d.month<A2.d.month \ \wedge A1.d.day<A2.d.day}(\rho_{A1}AtLeastOnce \times \rho_{A2}AtLeastOnce)$$

-Date when the first singleton order was placed

$$FirstOrder(CID, d1) \coloneqq AtLeastOnce - NotMin$$

-Dates that are not the last day of singleton order

$$NotMax(CID, d) \coloneqq \Pi_{A1.CID,A1.d}\sigma_{A1.CID=A2.CID \ \wedge A1.d.month<A2.d.month \wedge A1.d.day<A2.d.day}(\rho_{A1}AtLeastOnce \times \rho_{A2}AtLeastOnce)$$

-Date when the last singleton order was placed

$$LastOrder(CID, d2) \coloneqq AtLeastOnce - NotMax$$

$$Answer(CID, d1, d2) \coloneqq FirstOrder \bowtie LastOrder$$

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.

-Combined information on OID, IID and date ordered

$$OrderInfo(OID, IID, d) \coloneqq \Pi_{OID,IID,d}Order \bowtie LneItem$$

-All pair of two orders placed on the same day

$AllPairsSameDay(OID1, OID2)$
$\coloneqq \Pi_{O1.OID,O2.OID}\sigma_{O1.OID>O2.OID \wedge O1.d.year=O2.d.year \wedge O1.d.month=O2.d.month \wedge O1.d.day=O2.d.day}$
$(\rho_{O1}OrderInfo \times \rho_{O2}OrderInfo)$

-Non-identical two orders where at least one item is different

$NonIdentical(OID1, OID2)$
$\coloneqq \Pi_{O1.OID,O2.OID}\sigma_{O1.OID>O2.OID \wedge O1.IID>O2.IID \wedge O1.d.year=O2.d.year \wedge O1.d.month=O2.d.month \wedge O1.d.day=O2.d.day}$
$(\rho_{O1}OrderInfo \times \rho_{O2}OrderInfo)$

-Identical pairs of orders

$$Identical(OID1, OID2) \coloneqq AllPairs - NonIdentical$$

-Intermediate relation containing CID1 and OID’s

$$SameOrder1(CID1, OID1, OID2) \coloneqq \Pi_{CID,OID,OID2}\sigma_{OID1=OID}(Identical \times Order)$$

$$Answer(CID1, OID1, CID2, OID2) \coloneqq \Pi_{CID1,OID1,CID,OID}\sigma_{OID2=OID}(SameOrder1 \times Order)$$

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.

-CID of silver customers

$$Silver(CID) \coloneqq \Pi_{CID}\sigma_{membership='silver'}Customer$$

-CID of silver customers who ordered at least twice in 2014

$AtLeastTwo2014(CID)$
$\coloneqq \Pi_{O1.CID}\sigma_{O1.OID<O2.OID \wedge O1.CID=O2.CID \wedge O1.d.year=2014 \wedge O2.d.year=2014}(\rho_{O1}Order$
$\times \rho_{O2}Order)$

-CID of silver customers who ordered at least twice in 2015

$AtLeastTwo2015(CID)$
$\coloneqq \Pi_{O1.CID}\sigma_{O1.OID<O2.OID \wedge O1.CID=O2.CID \wedge O1.d.year=2015 \wedge O2.d.year=2015}(\rho_{O1}Order$
$\times \rho_{O2}Order)$

-CID of silver customers who ordered at most once in 2015

$$AtMostOne2015(CID) \coloneqq Silver - AtLeastTwo2015$$

-CID of silver customers who did not place any order in 2016

$$Zero2016(CID) \coloneqq Silver - \Pi_{CID}\sigma_{d.year=2016}Order$$

$$Answer(CID) \coloneqq AtLeastTwo2014 \cap AtMostOne2015 \cap Zero2016$$

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.

-Find the top cost item(s) for each order and put it in a table
-a list of items

$$OrderItems(OID, CID, IID, price) \coloneqq \Pi_{OID,CID,IID,price}[(\Pi_{OID,CID,IID}(Order \bowtie LineItem)) \bowtie Item]$$

-From that list, ones that are not top cost of that order

$NotTop(OID, CID, IID, price) \coloneqq \Pi_{T1.OID,T1.CID,T1.IID,T1.price}\sigma_{T1.OID=T2.OID \wedge T1.price<T2.price}(\rho_{T1}OrderItems \times \rho_{T2}OrderItems)$

-Get the top cost item(s) in each order

$$TC(OID, CID, IID, price) \coloneqq OrderItems - NotTop$$

->find the skimpiest order for each customer
-Take TC and find the items with the price that is not the lowest

$NotLowest(OID, CID, IID, price) \coloneqq \Pi_{T1.OID,T1.CID,T1.IID,T1.price}\sigma_{T1.CID=T2.CID \wedge T1.price>T2.price}(\rho_{T1}TC \times \rho_{T2}TC)$

-Remove them from the top cost list

$$Answer(CID, OID, price) \coloneqq \Pi_{CID,OID,price}(TC - NotLowest)$$

6. Find every order that includes at least one item for which reviewers unanimously gave it a rating of 0 and at least one item for which reviewers unanimously gave it a rating of 5. Report the customer ID, customer's last name and first name, order ID, and when the order was placed.

-Items with unanimous 0
-Review list ordered by Item IID, sum(ratings) == 0 (in SQL)

$$ItemsNotZero(IID) \coloneqq \Pi_{IID}\sigma_{rating>0 \vee rating<0}Review$$
$$ItemsZero(IID) \coloneqq \Pi_{IID}Review - ItemsNotZero$$

-Items with unanimous 5
-Review list ordered by Item IID, sum(ratings) == count(ratings)*5

$$ItemsNotFive(IID) \coloneqq \Pi_{IID}\sigma_{rating>5 \vee rating<5}Review$$
$$ItemsFive(IID) \coloneqq \Pi_{IID}Review - ItemsNotFive$$

-Orders with at least one from each list above

$$LineItems0(OID, IID) \coloneqq \Pi_{OID,IID}\sigma_{LineItem.IID=ItemsZero.IID}(LineItem \times ItemsZero)$$
$$LineItems5(OID, IID) \coloneqq \Pi_{OID,IID}\sigma_{LineItem.IID=ItemsFive.IID}(LineItem \times ItemsFive)$$
$$AnsOandI(OID, IID) \coloneqq LineItems0 \cap LineItems5$$

-Customer and order information for each order in list above

$$CustomerAndOrder(OID, CID, when, creditCard, number, email, lastName, firstName) \coloneqq \sigma_{Order.OID=AnsOandI.OID}((Order \bowtie_{Order.CID=Customer.CID} Customer) \times AnsOandI)$$

$$Answer \coloneqq \Pi_{CID,lastName,firstName,OID,when} CustomerAndOrder$$

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.

-All pairs of readers and reviewers

$$AllPairs(reader, reviewer) \coloneqq \Pi_{reader,reviewer} Helpfulness$$

-Pairs of reader and reviewer where a reader always found reviews helpful

$$PositiveReader(reader, reviewer) \coloneqq AllPairs - \Pi_{reader,reviewer}\sigma_{helpful='no'} Helpfulness$$

-Ideal situation where a reader has assessed all reviews of its reviewer and found them helpful.

$$ShouldHaveBeen(reviewer, item, reader) \coloneqq \Pi_{reviewer,item} Helpfulness \bowtie PositiveReader$$

-Pairs of reader and reviewer where a reader has not fully assessed every review of its reviewer

$$MissingAssessment(reader, reviewer) \coloneqq \Pi_{reader,reviewer}(ShouldHaveBeen - \Pi_{reviewer,item,reader} Helpfulness)$$

$$Answer(CID1, CID2) \coloneqq PositiveReader - MissingAssessme$$

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.

-Natural join the orders with the line items, project the IID, CID and when

$$OrdersList(OID, CID, IID, when) \coloneqq \Pi_{OID,CID,IID,when} Order \bowtie LineItem$$

-Find the not last time each of these items were ordered

$$NotLast(OID, CID, IID, when) \coloneqq \Pi_{T1.OID,T1.CID,T1.IID,T1.when}\sigma_{T1.when<T2.when}(\rho_{T1} OrdersList \times \rho_{T2} OrdersList)$$

-Remove them from the total list

$$LastOrdered(CID, IID) \coloneqq \Pi_{CID,IID}(OrdersList - NotLast)$$

-Output who ordered it and the item

$$Answer(IID, CID) \coloneqq LastOrdered$$

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.

-CID of customers who reviewed at least three times

$$3R(CID, IID, when, rating) \coloneqq \Pi_{C1.CID, C1.IID, C1.when, C1.rating}$$

$$\sigma_{C1.CID=C2.CID \wedge C2.CID=C3.CID \wedge C1.CID=C3.CID \wedge C1.IID!=C2.IID \wedge C2.IID!=C3.IID \wedge C1.IID!=C3.IID}(\rho_{C1}Review \times \rho_{C2}Review \times \rho_{C2}Review)$$

-CID of those who are not satisfied with their items over time

$$NotHappy(CID) \coloneqq$$

$$\Pi_{R1.CID}\sigma_{R1.CID=R2.CID \wedge R2.CID=R3.CID \wedge R3.CID=R1.CID \wedge R1.when>R2.when>R3.when \wedge R1.rating>R2.raiting>R3.raiting}(\rho_{R1}3R \times \rho_{R2}3R \times \rho_{R3}3R)$$

-Items reviewed except the last one from customers who reviewed at least three times

$$OldReviews(CID, IID, when, rating) \coloneqq \Pi_{R1.CID, R1.IID, R1.when, R1.raing}\sigma_{R1.when<R2.when}(\rho_{R1}3R \times \rho_{R2}3R)$$

$$Answer(CID, lastName, email, IID) \coloneqq \Pi_{CID, lastName, email, IID}(Customer \bowtie \left(\Pi_{CID, IID}(3R - OldReviews)\right))$$

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.

-List of items that are part of the Subcategory list as a subcategory

$$SubList(IID, type) \coloneqq \Pi_{IID, type}\sigma_{type=a}(Items \times Subcategory)$$

-Subtract above from total list of items, now have top-cat types

$$Top(type) \coloneqq \Pi_{type}((\Pi_{IID, type}Item) - SubList)$$

-Go through list of reviews and keep CIDs that have reviewed items with these types

$$ReviewList(CID, IID, type) \coloneqq \Pi_{CID, IID, type}(Review \bowtie Item)$$

$$Answer(CID) \coloneqq \Pi_{CID}\sigma_{Top.type=ReviewList.type}(Top \times ReviewList)$$

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.

-List of items (order with at least one line item)

$$ListOfItems(OID, IID, type) \coloneqq \Pi_{OID,IID,type}(ListItem \bowtie Item)$$

-List of items that are not of type “book”

$$NoBooks(OID, IID, type) \coloneqq \Pi_{OID,IID,type}\sigma_{type!='book'}ListOfItems$$

-List of Items in the subcategory of another type

$$ItemswSub(OID, IID, type, b) \coloneqq \Pi_{OID,IID,type,b}(ListOfItems \bowtie_{type=a} Subcategory)$$

-List of items who’s type is a subcategory of book

$$SubBooks(OID, IID, type) \coloneqq \Pi_{OID,IID,type}\sigma_{b='book'}ItemswSub$$

-Remove above list from the list with items not of type “book”

$$NoSubBooks(OID) \coloneqq \Pi_{OID}(NoBooks - SubBooks)$$

-Remove all orders that had a non-book item

$$Answer(OID) \coloneqq (\Pi_{OID}ListOfItems) - NoSubBooks$$

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.

-Pairs of customer and an item reviewed by the customer
$ItemReviewed(CID, IID) \coloneqq \Pi_{CID,IID} Review$

-Pairs of customer and an item they ordered
$ItemOrdered(CID, IID) \coloneqq \Pi_{CID,IID} Order \bowtie LineItem$

$$ItemReviewed - ItemOrdered = \emptyset$$

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.

-Orders made by silver customers
$Silvers(OID) \coloneqq \Pi_{OID} \sigma_{membership='silver'} Customer \bowtie Order$

-Orders that include at least one item that costs over 50
$AtLeastOne50(OID) \coloneqq \Pi_{OID} \sigma_{price \geq 50} LineItem \bowtie Item$

$$Silvers - AtLeastOne50 = \emptyset$$

-Orders that included any items under $50
$ItemUnder50(OID) \coloneqq \Pi_{OID} \sigma_{price<50} LineItem \bowtie Item$

-Orders made by non-member customers
$Peasants(OID) \coloneqq \Pi_{OID} \sigma_{membership='none'} Customer \bowtie Order$

$$Peasants \cap ItemUnder50 = \emptyset$$