1. Subcategories\_of\_apparel (a) := project\_{a} (select\_{b=’apparel’} Subcategory)

Manufacturers\_of\_type (manufacturer) := project\_{manufacturer} (Subcategories\_of\_apparel natural\_join {a=type} Item)

Answer (MID, name, address, phone) := project\_{MID, name, address, phone} (Manufacturers\_of\_type natural\_join{manufacturer=MID} Manufacturer)

1. Some\_Table(stuff) := Rename{T1}(Order natural\_join Line\_Item) X Rename{T2}(Order natural\_join Line\_Item)

Non-singleton ():= Project\_{T1.OID} select{T1.OID = T2.OID and T1.IID != T2.IID} Some\_Table

Singleton (OID):= Project\_{OID} (Some\_Table) – Non-singleton

Gold\_members (CID) := Project\_{CID} (select\_{membership=’gold’} Customers)

Orders () := Singleton natural\_join Order

Gold\_Orders() := Orders natural\_join Gold\_members

Not\_Last\_Orders() := project\_{OID} select\_{T1.when < T2.when} (rename{T1}(Gold\_Orders) X rename{T2}(Gold\_Orders))

Last\_Orders() := Gold\_Orders natural\_join ((project\_{OID} Gold\_orders) – Not\_Last\_Orders)

Not\_First\_Orders() := project\_{OID} select\_{T1.when > T2.when} (rename{T1}(Gold\_Orders) X rename{T2}(Gold\_Orders))

First\_Orders() := Gold\_Orders natural\_join ((project\_{OID} Gold\_orders) – Not\_First\_Orders)

Not finished yet…

3.Pairs of orders with at least one difference (everything) := (select\_{T1.OID != T2.OID and T1.IID != T2.IID and T1.when.day = T2.when.day} ((rename T1 Order natural\_join Line\_item) X (rename T2 Order natural\_join Line\_item)))

Pairs of orders with no differences (T1.OID, T1.CID, T2.OID, T2.CID) := project\_{T1.OID, T1.CID, T2.OID, T2.CID}(((rename T1 Order natural\_join Line\_item) X (rename T2 Order natural\_join Line\_item)) – Pairs\_of\_orders\_with-at\_least\_one\_difference)

4. At\_least\_2\_orders\_2014 (CID) := (project\_{T1.CID} select\_{T1.OID != T2.OID and T1.CID = T2.CID and T1.when.year = T2.when.year = 2014 (rename T1 Order) X (rename T2 Order))

At\_least\_2\_orders\_2015 (CID) := (project\_{T1.CID} select\_{T1.OID != T2.OID and T1.CID = T2.CID and T1.when.year = T2.when.year = 2015 (rename T1 Order) X (rename T2 Order))

Less\_than\_2\_orders\_2015 (CID) := (project\_{CID} select\_{when.year = 2015} Order) – At\_least\_2\_orders\_2015

No\_orders\_2016 (CID) := (project\_{CID} Customer) - (project\_{CID} select\_{when.year = 2016} Order)

Silver\_customers(CID) := project\_{CID} select\_{membership=’silver’} Customer

Answer (CID) := Silver\_customers intersect No\_orders\_2016 intersect Less\_than\_2\_orders\_2015 intersect At\_least\_2\_orders\_2014

5. All\_items\_with\_cost () := Item natural\_join Line\_Item natural\_join Order

Not\_Top\_Cost () := select\_{(T1.OID = T2.OID and T1.price<=T2.price) or T1.OID != T2.OID} (rename T1 All\_items\_with\_cost) x (rename T2 All\_items\_with\_cost)

Top\_Cost () := project\_{T1.CID, T1.OID, T1.price} (rename T1 All\_items\_with\_cost) x (rename T2 All\_items\_with\_cost) – Not\_Top\_Cost

Not\_skimpiest () := select\_{(T1.CID = T2.CID and T1.price < T2.price) or T1.CID != T2.CID} Top\_Cost X Top\_Cost

Skimpiest() := project\_{T1.CID, T1.OID, T1.price} (Top\_Cost X Top\_Cost – Not\_skimpiest)

6. Not\_highest\_rating (everything) := (select\_{T1.IID = T2.IID and T1.when != T2.when and T1.review < T2.review} (rename T1 Review X rename T2 Review))

Items\_with\_all0 (IID) := (project\_{T1.IID} select\_{T1.review=0} (rename T1 Review X rename T2 Review – Not\_highest\_rating))

Not\_lowest\_rating (everything) := (select\_{T1.IID = T2.IID and T1.when != T2.when and T1.review > T2.review} (rename T1 Review X rename T2 Review))

Items\_with\_all5 (IID) := (project\_{T1.IID} select\_{T1.review=5} (rename T1 Review X rename T2 Review – Not\_lowest\_rating))

Orders\_with\_a0 (OID) := (project\_{OID} (Line\_item natural\_join Items\_with\_all0))

Orders\_with\_a5 (OID) := (project\_{OID} (Line\_item natural\_join Items\_with\_all5))

Orders\_with\_a0and5 (OID) := Orders\_with\_a0 union Orders\_with\_a5

Answer (CID, firstname, lastname, OID, when) := (project\_{CID, firstname, lastname, OID, when} ((Customer natural\_join Order) natural\_join Orders\_with\_a0and5))

7.

8. Not\_last\_order () := (select\_{(T1.IID = T2.IID and T1.when < T2.when) or T1.IID != T2.IID } (rename T1 (Order natural\_join Line\_item) X rename T2 (Order natural\_join Line\_item)))

Last\_order () := project\_{T1.IID, T1.CID} (rename T1 (Order natural\_join Line\_item) X rename T2 (Order natural\_join Line\_item))) – Not\_last\_order

9. Helpful\_atleastonce () := project\_{reviewer, item, reader} select\_{helpful=’yes’} Helpfulness

Helpful\_atleasttwice ():= project\_{T1.reviewer, T1.item, T1.reader} select\_{T1.reviewer = T2.reviewer and T1.item = T2.item and T1.reader != T2.reader} (Helpful\_atleastonce X Helpful\_atleastonce)

Helpful\_exactlyonce () := Helpful\_atleastonce – Helpful\_atleasttwice

At\_least\_one\_no () := project\_{reviewer, item} select\_{helpful=’no’} Helpfulness

Helpful\_more\_than\_not () := (project\_{reviewer, item, price} ((project\_{reviewer, item} Helpful\_exactlyonce) - At\_least\_one\_no) natural\_join {item=IID} Item))

10.

11. Get all top category items

Natural join with review

Project that down to customers -> item type

Make checklist of customer to every item type Cartesian product

Set subtraction

12. Get all orders and types in those orders (Line Item, Order, Item)

Get all types that are book or subcategory of book

Product all order ID from step 1 with all types from step2

Subtract Step 1 from step 3 -> what’s left are orders we don’t want

Get compliment

13.

1. (project\_{CID, IID} Review) - (project\_{CID, IID} (Line\_Item natural\_join Order)) = null
2. Non\_members () := select\_{membership=’none’ and price < 50} (Item natural\_join Line\_Item natural\_join Order natural\_join Customer)

Silver\_members\_over\_50 () := project\_{OID} select\_{membership=’silver’ and price > 50} (Item natural\_join Line\_Item natural\_join Order natural\_join Customer)

Silver\_members\_under\_50 () := (project\_{CID} select\_{membership=’silver’} Order natural\_join Customer natural\_join ((project\_{OID} Order) - Silver\_members\_over\_50))

Silver\_members\_under\_50 union Non\_members = null