Assignment 1 - Relational Algebra

May 2020

Team information

1. Member 1: gavinsit, Gavin, Sit, gavinsit@my.yorku.ca, 215043870
2. Member 2: login, first name, last name, email, student number

Relational Algebra Symbols[1]

Symbol name	LaTex Code	symbol
assignment	:=	$:=$
select	\sigma	σ
project	\pi	π
rename	\rho	ρ
cross product	\times	$\times$
natural join	\bowtie	$\bowtie$
theta join	\bowtie	$\bowtie_\theta$
less than	<	$<$
greater than	>	$>$
less than or equal	\leq	$\leq$
greater than or equal	\geq	$\geq$
equal	=	$=$
not equal	\neq	$\neq$
and	\wedge	$\wedge$
or	\vee	$\vee$
not	\neg	$\neg$

[1]source: http://www.cs.uleth.ca/ rice/latex/worksheet.pdf

Part 1: Additional Integrity Constraints

1. $Answer := Bid \bowtie_{Bid.CID=Listing.CID \wedge Bid.LID=Listing.LID} (Listing) = \emptyset$

   // The theta-join of the two given the condition where the CID and LID are equal should yield no result.

2. $Answer := \sigma_{Price<StartingPrice \vee Price>buyoutPrice}(Bid \bowtie_{Bid.LID=Listing.LID} (Listing)) = \emptyset$

   // Given theta-join of all the bids that match the listing CID, the selection of all items outside of range should be empty.

3. $\sigma_{time<postedTime \vee time>expiryTime}(Bid \bowtie_{Bid.LID=Listing.LID} (Listing)) = \emptyset$

   // Selection of all the tuples outside of range of theta join of bid and listing where LID is equal should yield no result

4. Cannot be expressed

5. $A := Bid - \sigma_{Time<Time1 \wedge CID=CID1}(Bid \times \rho_{BID,CID,LID,Time,Price \rightarrow BID1,CID1,LID1,Time1,Price1}(Bid))$

   $Answer := Bid \bowtie_{Bid.LID=A.LID \wedge Bid.time>A.time \wedge Bid.price<A.price} (A) = \emptyset$

   //Select all times of the same listing that has a lower time and we subtract Bid from that result so we get the latest time for each listing.

   //Given the latest bid for each listing in A, we check that the price of the bids that come later on the same listing result in no results.

Part 2: Queries

1. $Answer := \pi_{name}(Customer \bowtie (\pi_{LID}(BID) \bowtie_{Bid.LID=Listing.LID} (\pi_{LID,CID}(Listing))))$

   //Select the CID which is from the listings that have been theta-joined with bid. Any listing whose LID is found in Bid means at least one person has bid on the item. Then theta join with customer and project the name of those who sold something.

2. $Max := Reputateion - \sigma_{number<number1}(Reputation \times \rho_{CID,number \rightarrow CID1,number1}(Reputation))$

   $Answer := \pi_{CID,name}(A \bowtie Customer)$

   // Max first gets the max reputation and then joining the highest reputations remaining and customers give us the customer(s) with the highest reputation which we then project to return only the CID and name of the customer.

3. $H := \pi_{price}(\sigma_{LID=14563}(Bid)) - \pi_{price}(\sigma_{price<price1 \wedge LID=14563}(Bid \times \rho_{BID,CID,LID,Time,Price \rightarrow BID1,CID1,LID1,Time1,Price1}(Bid)$

   $Answer := \pi_{price}(\sigma_{LID=14563 \wedge price<>H}(Bid)) -$

   $\pi_{price}(\sigma_{price<price1 \wedge LID=14563}(\sigma_{price<>H}(Bid) \times \sigma_{price1<>H}(\rho_{BID,CID,LID,Time,Price \rightarrow BID1,CID1,LID1,Time1,Price1}(Bid)))$

   //First get the highest bid of that listing number which is denoted by H by removing all the prices that match the listing ID that has a number bigger than itself. Then we repeat the process but this time taking out the highest price which will yield the second highest price.

4. $A := Bid \bowtie_{price<>price1} (\rho_{BID,time,price \rightarrow Bid1,time1,price1}(Bid))$

   $B := A \bowtie_{price<>price3 \wedge price1<>price3} (\rho_{BID,time,price \rightarrow BID3,time3,price3}(BID))$

   $Answer := \pi_{name}(Customer \bowtie B)$

   // A yields the relation (BID, time, price, CID, LID, BID1, time1, price1) which shows the customer has bid at least twice on the same listing (price not equal price1 makes it so that they don't have repeats).

   // Next, repeat the process to get the customers that bid at least three times on one listing and B yields the following relation: (BID, time, price, CID, LID, BID1, time1, price1, BID3, time3, price3).

   // Finally, get only the customer using projection.

5. $A := Bid - \pi_{CID,BID,LID,time,price}(\sigma_{price<price1 \wedge LID=LID1}(BID \times \rho_{CID,BID,LID,time,price \rightarrow CID1,BID1,LID1,time1,price1}(Bid)))$

   $B := (\pi_{CID,LID}(\pi_{CID,LID}(A) \bowtie_{A.LID=Listing.LID} (\pi_{LID,IID,CID}(Listing))) \bowtie Rating) \bowtie Customer$

   $Answer := (Rating \bowtie Customer) - B$

   // Take all the customers and subtract the ones that gave a trustworthy rating which are the ones that won something and this is represented with A. // After getting A, we use it to get the rating that the trustworthy customers made (if any) which we theta-join with Rating and customer and this gives us B the ratings of all the trustworthy customers which we subtract from all the ratings made so we are left with the untrustworthy ratings

6. $(\pi_{LID}(Bid) \bowtie Listing) - \pi_{LID,IID,CID,postedTime,expiryTime,startingPrice,buyoutPrice}(\sigma_{startingPrice<startingPrice1}((\pi_{LID}(Bid) \bowtie Listing) \times \pi_{startingPrice1}(\rho_{staringPrice \rightarrow startingPrice1}(\pi_{LID}(Bid) \bowtie Listing))))$

   // We theta-join Bid.LID with listing so we get all the listings that have at least one bid meaning it has been sold. Then we find the max by getting all the startingPrice that isn't the max and subtracting it from all the possibilities all the while only doing these to listings that have sold something.

7. $A := \sigma_{buyoutPrice>buyoutPrice1 \vee TODAY>expiryTime}(Listing \times \rho_{LID,IDD,CID,postedTime,expiryTime,startingPrice,buyoutPrice \rightarrow LID1,ID}$

   $B := Listing - \pi_{LID,IDD,CID,postedTime,expiryTime,startingPrice,buyoutPrice}(A)$

   $Answer := \pi_{buyoutPrice,name}(B \bowtie Item)$

   // A represents all the possible listings that has another listing with lower buyoutPrice or is expired (meaning the expiryTime is greater than 'TODAY').

   // B is all the listings subtracted by the listings A, that do not work.

   //Since B is the listing with the lowest price, we can use join to match it with the correct item and output the name.

8. // cannot be expressed

9. // cannot be expressed

10. $A := Listing \bowtie_{Listing.LID=Bid.LID \wedge price<buyoutPrice} Bid$

    $Answer := Listing - \pi_{LID,IID,CID,postedTime,expiryTime,startingPrice,buyoutPrice}(A)$

    // First get all the listings that have at least one occurrence of not selling at buyoutPrice. This does not include those listings that haven't sold at all so we also have to include that as not selling

    // Subtract all the items that don't work and you will be left with all the listings that sold for only buyout price