CSC 343: Assignment 1

Zeeshan Qureshi (g0zee)
Jaideep Bajwa (c1bajwaj)

February 10, 2011

1 Queries

1.

$$
\begin{aligned}
BR(retailer) &:= \Pi_{ID}\ \sigma_{name=\text{“Banana Republic”}}\ (Ret)\\
BRStores(store) &:= \Pi_{ID}\ \sigma_{country=\text{“Canada”}}\ (Store \bowtie BR)\\
Members &:= \Pi_{memNum}\ \sigma_{amount>2000}\ (Trans \bowtie BRStores)\\
Result &:= \Pi_{name}\ (Mem\ \bowtie\ Members)
\end{aligned}
$$

2.

$$
\begin{aligned}
NoCards &:= (\Pi_{memNum}\ Mem) - (\Pi_{Memnum}\ Card)\\
Result &:= \Pi_{name,\,phone}\ (NoCards \bowtie Mem)
\end{aligned}
$$

3.

$$
\begin{aligned}
BC(retailer) &:= \Pi_{ID}\ \sigma_{name=\text{“Book City”}}\ (Ret)\\
BCStores(store) &:= \Pi_{ID}\ (Store \bowtie BC)\\
BCTrans &:= \Pi_{memNum,\,amount}\ (Trans \bowtie BCStores)\\
NotLargest(memNum, amount) &:= \Pi_{m_1,a_1}\ \sigma_{(m_1 \neq m_2)\wedge(a_1<a_2)}\\
&\quad ((\rho_{T1(m_1,\,a_1)}\ BCTrans) \bowtie (\rho_{T2(m_2,\,a_2)}\ BCTrans))\\
Result &:= BCTrans - NotLargest
\end{aligned}
$$

4. Using relations from query 3 . . .

$$
\begin{aligned}
NotSecond(memNum, amount) &:= \Pi_{m_1,a_1}\ \sigma_{(m_1 \neq m_2)\wedge(a_1<a_2)}\\
&\quad ((\rho_{T1(m_1,\,a_1)}\ NotLargest) \bowtie (\rho_{T2(m_2,\,a_2)}\ NotLargest))\\
Result &:= NotLargest - NotSecond
\end{aligned}
$$

5. Cannot be expressed since it requires summing over the number of tuples, which is not possible in this subset of Relational Algebra.

6. Using relations from query 3 . . .

$$
\begin{aligned}
AllPossible(memNum, store) &:= (\Pi_{memNum}\ BCTrans) \bowtie (BCStores)\\
Purchases &:= \Pi_{memNum,\,store}\ (Trans \bowtie BCStores)\\
NotAll &:= AllPossible - Purchases\\
Result &:= (\Pi_{memNum}\ AllPossible) - (\Pi_{memNum}\ NotAll)
\end{aligned}
$$

7.

$$
\begin{aligned}
TwoOrMore(retailer) :=\ & \Pi_{retailer}\ \sigma_{(ID < ID2) \wedge (retailer = retailer2)} \\
& (Store \times (\rho_{S2(ID2,\, retailer2)}\ Store)) \\
OnlyOne(ID) :=\ & (\Pi_{retailer}\ Store) - TwoOrMore \\
Result :=\ & \Pi_{name}\ (OnlyOne \bowtie Ret)
\end{aligned}
$$

8. Cannot be expressed since it requires summing over the tuples of transactions for every Roots store.

9.

$$
\begin{aligned}
NoSupp :=\ & (\Pi_{memNum}\ Card) - (\Pi_{memNum}\ Supp) \\
Result :=\ & \Pi_{name}\ (NoSupp \bowtie (\sigma_{(city = \text{“Toronto”}) \wedge (country = \text{“Canada”})}\ Mem))
\end{aligned}
$$

10.

$$
\begin{aligned}
PriWithSupp :=\ & \Pi_{memNum,\, cardNum} \\
& ((\rho_{Sup(cardNum,\, memSup,\, pLimit)}\ Supp) \bowtie Card) \\
AllMem :=\ & (\Pi_{memNum,\, cardNum}\ Supp) \cup PrimaryWithSupp \\
MInfo :=\ & \Pi_{country,\, cardNum,\, memNum}\ (Mem \bowtie AllMem) \\
MInfo2 :=\ & \rho_{MI2(country,\, cardNum,\, memNum2)}\ MInfo \\
SameCountry :=\ & \Pi_{cardNum}\ \sigma_{memNum < memNum2}\ (MInfo \bowtie MInfo2) \\
Result :=\ & (\Pi_{cardNum}\ MInfo) - SameCountry
\end{aligned}
$$

11.

$$
\begin{aligned}
PriPmt :=\ & \Pi_{cardNum,\, memNum,\, amount}\ ((\Pi_{cardNum,\, memNum}\ Card) \bowtie Pmt) \\
NotMax :=\ & \Pi_{cardNum,\, memNum,\, amount}\ \sigma_{amount < amount2} \\
& (PriPmt \bowtie (\rho_{P(cardNum,\, memNum,\, amount2)}\ PriPmt)) \\
MaxPriPmt :=\ & \rho_{P(cardNum,\, memPri,\, amtPri)}\ (PriPmt - NotMax) \\
SuppPmt :=\ & \rho_{S(cardNum,\, memSupp,\, amtSupp)}\ \Pi_{cardNum,\, memNum,\, amount} \\
& ((\Pi_{cardNum,\, memNum}\ Supp) \bowtie Pmt) \\
Result :=\ & \Pi_{cardNum,\, memSupp}\ \sigma_{amtSupp > amtPri}\ (SuppPmt \bowtie MaxPriPmt)
\end{aligned}
$$

12.

$$
Result :=\ \Pi_{cardNum}\ ((\Pi_{memNum}\ Card) \bowtie Supp)
$$

13.

$$
\begin{aligned}
SuppMem &:= \Pi_{cardNum,\, memNum}\,(Supp) \\
MoreThanOne &:= \Pi_{cardNum}\ \sigma_{memNum < memNum2} \\
&\quad (SuppMem \bowtie (\rho_{S(cardNum,\, memNum2)}\ SuppMem)) \\
Result &:= (\Pi_{cardNum}\ \sigma_{limit > 10000}\ Card) - MoreThanOne
\end{aligned}
$$

14.

$$
\begin{aligned}
Purchases &:= (\Pi_{cardNum}\ Card) \bowtie (\Pi_{cardNum}\ Trans) \\
Payments &:= (\Pi_{cardNum}\ Card) \bowtie (\Pi_{cardNum}\ Pmt) \\
Result &:= (\Pi_{cardNum}\ Card) - Purchases - Payments
\end{aligned}
$$

15.

$$
\begin{aligned}
SuppCard &:= \Pi_{memNum,\, cardNum}\,(Supp) \\
AtLeastTwo &:= \sigma_{cardNum < cardNum2} \\
&\quad (SuppCard \bowtie (\rho_{S(memNum,\, cardNum2)}\ SuppCard)) \\
AtLeastThree &:= \sigma_{(cardnum < cardNum3) \wedge (cardNum2 < cardNum3)} \\
&\quad (AtLeastTwo \bowtie (\rho_{S(memNum,\, cardNum3)}\ SuppCard)) \\
Result &:= \Pi_{name}\,((\Pi_{memNum}\ AtLeastThree) \bowtie Mem)
\end{aligned}
$$

2 Integrity Constraints

1.

$$\sigma_{(country \neq \text{'Canada'})\ AND\ (country \neq \text{'US'})\ AND\ (country \neq \text{'Mexico'})}(Ret) = \emptyset$$

2.

$$\sigma_{(cardNum < 555555555)\ AND\ (amount > 10000)}\ (Trans) = \emptyset$$

3.

$$\Pi_{cardNum}(Supp) \subseteq \Pi_{cardNum}(Card)$$

4.

$$\Pi_{memNum}(Card) \bowtie \Pi_{memNum}(Supp) = \emptyset$$

5.

Cannot be expressed!