Grading Information

- Warmup problems will be graded as one of (✓+, ✓, ✓-)
- All other problems will be graded in detail and will be given a score.

Solutions for the warmup problems will be provided along with your graded work.

Collaboration Policy

Please submit your signed collaboration policy to the CS127 handin bin before the first assignment is due. The collaboration policy can be found under the DOCS section in the course website.

Warmup #1  (Textbook Problem 2.7)

Solution:

2.7 Consider the relational database of Figure ?? . Give an expression in the relational algebra to express each of the following queries:

a. Find the names of all employees who live in city “Miami”.
b. Find the names of all employees whose salary is greater than $100,000.
c. Find the names of all employees who live in ”Miami” and whose salary is greater than $100,000.

Answer:

a. \( \Pi_{\text{name}} (\sigma_{\text{city} = "Miami"} (\text{employee})) \)
b. \( \Pi_{\text{name}} (\sigma_{\text{salary} > 100000} (\text{employee})) \)
c. \( \Pi_{\text{name}} (\sigma_{\text{city} = "Miami" \land \text{salary} > 100000} (\text{employee})) \)

Figure 1: Warmup 1

Warmup #2  (Textbook Problem 2.10)

Solution:
2.10 Consider the advisor relation shown in Figure 2.8, with $s.id$ as the primary key of advisor. Suppose a student can have more than one advisor. Then, would $s.id$ still be a primary key of the advisor relation? If not, what should the primary key of advisor be?

**Answer:** No, $s.id$ would not be a primary key, since there may be two (or more) tuples for a single student, corresponding to two (or more) advisors. The primary key should then be $s.id$, $i.id$.

Figure 2: Warmup 2

**Warmup #3 (Textbook Problem 2.13)**

Solution:

2.13 Consider the bank database of Figure 2.15. Give an expression in the relational algebra for each of the following queries:

a. Find all loan numbers with a loan value greater than $10,000$.

b. Find the names of all depositors who have an account with a value greater than $6,000$.

c. Find the names of all depositors who have an account with a value greater than $6,000$ at the “Uptown” branch.

**Answer:**

a. $\Pi_{loan\_number} (\sigma_{amount > 10000}(loan))$

b. $\Pi_{customer\_name} (\sigma_{balance > 6000} (depositor \bowtie account))$

c. $\Pi_{customer\_name} (\sigma_{balance > 6000 \land branch\_name = "Uptown"} (depositor \bowtie account))$

Figure 3: Warmup 3
Warmup #4  (Textbook Problem 6.2)

Solution:

6.2 Consider the relational database of Figure 6.22, where the primary keys are underlined. Give an expression in the relational algebra to express each of the following queries:

   a. Find the names of all employees who live in the same city and on the same street as do their managers.

   b. Find the names of all employees in this database who do not work for “First Bank Corporation”.

   c. Find the names of all employees who earn more than every employee of “Small Bank Corporation”.

Answer:

   a. $\Pi_{person\_name} ((employee \bowtie manages)
      \bowtie (manager\_name = employee2\_person\_name \land employee\_street = employee2\_street
      \land employee\_city = employee2\_city) (employee2 (employee)))$

   b. The following solutions assume that all people work for exactly one company. If one allows people to appear in the database (e.g. in employee) but not appear in works, the problem is more complicated. We give solutions for this more realistic case later.

      $\Pi_{person\_name} (\sigma_{company\_name \neq “First Bank Corporation”} (works))$

      If people may not work for any company:

      $\Pi_{person\_name} (employee) - \Pi_{person\_name} (\sigma_{company\_name = “First Bank Corporation”} (works))$

   c. $\Pi_{person\_name} (works) - (\Pi_{works.persont_name} (works
      \bowtie (works.salary \leq works2.salary \land works2.company\_name = “Small Bank Corporation”)
      \rho_{works2}(works)))$

Figure 4: Warmup 4

Problem 5 (To Be Graded)

You work as a Data Analyst for a retailer. The retailer stores all information about orders, ordered items, items in stock, and customers in a relational database. As part of your report you have to query the database for some data. Below is a sample of the database.
ORDERS

<table>
<thead>
<tr>
<th>Order</th>
<th>Customer</th>
<th>Status</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Alicia</td>
<td>Delivered</td>
<td>300</td>
</tr>
<tr>
<td>2</td>
<td>Ben</td>
<td>Delivered</td>
<td>1700</td>
</tr>
<tr>
<td>3</td>
<td>Alicia</td>
<td>Processing</td>
<td>1600</td>
</tr>
<tr>
<td>4</td>
<td>Caitlin</td>
<td>Processing</td>
<td>700</td>
</tr>
</tbody>
</table>

CUSTOMERS

<table>
<thead>
<tr>
<th>Customer</th>
<th>City</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alicia</td>
<td>Providence</td>
</tr>
<tr>
<td>Ben</td>
<td>Boston</td>
</tr>
<tr>
<td>Caitlin</td>
<td>Boston</td>
</tr>
</tbody>
</table>

ORDERITEMS

<table>
<thead>
<tr>
<th>Order</th>
<th>Item</th>
<th>Quantity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Mandolin</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Guitar</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>Fiddle</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>String Bass</td>
<td>2</td>
</tr>
<tr>
<td>4</td>
<td>Mandolin</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>Guitar</td>
<td>1</td>
</tr>
</tbody>
</table>

STOCKITEMS

<table>
<thead>
<tr>
<th>Item</th>
<th>Quantity</th>
<th>Price</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mandolin</td>
<td>17</td>
<td>300</td>
</tr>
<tr>
<td>Guitar</td>
<td>53</td>
<td>400</td>
</tr>
<tr>
<td>String Bass</td>
<td>0</td>
<td>800</td>
</tr>
<tr>
<td>Fiddle</td>
<td>34</td>
<td>650</td>
</tr>
</tbody>
</table>

Give expressions in relational algebra to answer the following questions:

1. Which orders have already been delivered to Boston?
   \[ \Pi_{\text{Order}}(\sigma_{\text{Status} = "Delivered" \land \text{City} = "Boston"}(\text{ORDERS} \bowtie \text{CUSTOMERS})) \]

2. Which orders that are still “Processing” include items that are currently out of stock?
   \[ \Pi_{\text{Order}}(\sigma_{\text{Status} = "Processing" \land \text{StockQuantity} = 0}(\text{ORDERS} \bowtie \text{ORDERITEMS} \bowtie (\rho_{SI}(\text{Item, StockQuantity, Price})\text{STOCKITEMS}))) \]

3. Which customers have not ordered a guitar or a string bass?
   \[ \Pi_{\text{Customer}}(\text{CUSTOMERS}) - \Pi_{\text{Customer}}(\sigma_{\text{Item} = "Guitar" \lor \text{Item} = "String Bass"}(\text{ORDERS} \bowtie \text{ORDERITEMS})) \]

4. Which customer has spent the most money at the store?
   \[ \text{TotalSpent} \leftarrow \rho_{\text{Totals}(Customer, Total)}(\text{Customer} \bowtie \text{SUM(Total})\text{ORDERS}) \]
   \[ \text{TotalSpent2} \leftarrow \text{TotalSpent} \]
   \[ \Pi_{\text{TotalSpent, Customer}}(\text{TotalSpent} \times \text{TotalSpent2} - \text{TotalSpent} \bowtie \text{TotalSpent2} \bowtie \text{Total Spent2}) \]

5. Which customers have ordered the best-selling item?
   \[ O1 \leftarrow \sigma_{\text{SUM(Quantity)}}\text{ORDERITEMS} \]
   \[ O2 \leftarrow O1 \]
   \[ \text{MaxItem} \leftarrow \rho_{\text{MaxItem(Item)}}O1.\text{Item}(O1 \times O2 - O1 \bowtie \text{O1.SUM(Quantity)}<O2.SUM(Quantity) \bowtie O2) \]
   \[ \Pi_{\text{Customer}}(\text{ORDERS} \bowtie \text{ORDERITEMS} \bowtie \text{MaxItem}) \]

6. Which orders contain the most expensive item?
   \[ S2 \leftarrow \text{STOCKITEMS} \]
   \[ \text{MaxPriceItem} \leftarrow \rho_{\text{MaxPriceItem(Item)}}(\Pi_{\text{STOCKITEMS.Item}}(\text{STOCKITEMS} \times S2 - \text{STOCKITEMS} \bowtie \text{STOCKITEMS.Price}<S2.Price \bowtie S2)) \]
   \[ \Pi_{\text{Order}}(\text{ORDERITEMS} \bowtie \text{MaxPriceItem}) \]