Comp 5311 Database Management Systems

4b. Structured Query Language Exercises
SQL-Algebra Review

• Suppose a bookstore has the following five relational tables:
  BOOK (BID, TITLE, AID, SUBJECT, QUANTITY-IN-STOCK)
  AUTHOR (AID, FIRST-NAME, LAST-NAME)
  CUSTOMER (CID, FIRST-NAME, LAST-NAME)
  ORDER-DETAILS (OID, BID, QUANTITY)
  ORDER (OID, CID, ORDER-YEAR)

• ASSUMPTIONS:
  Keys are underlined and foreign keys are in italics. Each author has authored at least one book in the store. Each book has exactly one author. Each order is made by exactly one customer and has one or more associated record in ORDER-DETAILS (e.g., an order may contain different books).
• Write the following queries in relational algebra and SQL.

**Query 1:** Find all distinct book titles of the author whose last name is “Luo”.

**algebra:**

$$\pi_{\text{TITLE}}(\sigma_{\text{LAST-NAME} = 'Luo'}(\text{BOOK JOIN AUTHOR.AID = BOOK.AID AUTHOR}))$$

**SQL:**

```sql
SELECT DISTINCT B.TITLE
FROM BOOK AS B, AUTHOR AS A
WHERE A.LAST-NAME = 'Luo' AND A.AID = B.AID
```
• **Query 2:** Find the last name and first name of all authors who wrote books in both the subjects of “ART” and “BUSINESS”.

• **algebra:**

\[
\pi_{\text{LAST-NAME}, \text{FIRST-NAME}} \\
( ( \pi_{\text{AID}, \text{LAST-NAME}, \text{FIRST-NAME}} (\sigma_{\text{BOOK.SUBJECT} = 'ART'} (\text{BOOK JOIN} \ \text{AUTHOR.AID} = \text{BOOK.AID} \ \text{AUTHOR}))) \cap \\
( \pi_{\text{AID}, \text{LAST-NAME}, \text{FIRST-NAME}} (\sigma_{\text{BOOK.SUBJECT} = 'BUSINESS'} (\text{BOOK JOIN} \ \text{AUTHOR.AID} = \text{BOOK.AID} \ \text{AUTHOR}))) )
\]

• **SQL:**

```
SELECT A1.LASTNAME, A1.FIRSTNAME
FROM AUTHOR AS A1, BOOK AS B1
( SELECT A2.AID
  FROM AUTHOR AS A2, BOOK AS B2
  WHERE A2.AID = B2.AID AND B2.SUBJECT = 'BUSINESS'
)
```
**Query 3:** Find the IDs of customers that have ordered **all** books in the subject of "ART".

* Algebra: *

\[
\left( \pi_{\text{CID}, \text{BID}} (\text{ORDERDETAILS} \Join_{\text{ORDER.OID} = \text{ORDERDETAILS.OID}} \text{ORDER}) \right) / \left( \pi_{\text{BID}} (\sigma_{\text{BOOK.SUBJECT} = 'ART'} \text{BOOK}) \right)
\]

* SQL: *

```
SELECT CID
FROM CUSTOMER
WHERE NOT EXISTS
  ((SELECT BID
   FROM BOOK
   WHERE SUBJECT = 'ART')
  EXCEPT
  (SELECT ORDERDETAILS.BID
   FROM ORDER, ORDERDETAILS
   WHERE ORDER.CID = CUSTOMER.CID AND ORDER.OID = ORDERDETAILS.OID))
```
• **Query 4:** Find the last name and first name of all authors who wrote books in at least two subjects.

• **algebra:**

\[ \pi_{\text{LAST-NAME,FIRST-NAME}}(\sigma_{\text{B1.SUBJECT} \neq \text{B2.SUBJECT}} (\rho(\text{B1,BOOK}) \Join_{\text{AID}} \text{AUTHOR} \Join_{\text{AID}} \rho(\text{B2,BOOK}))) \]

• **SQL:**

```
SELECT A.LASTNAME, A.FIRSTNAME
FROM AUTHOR A, BOOK B1, BOOK B2
WHERE B1.AID=A.AID AND B2.AID=A.AID AND
B1.SUBJECT ≠ B2.SUBJECT
```
• For the following queries only use SQL
• **Query 5:** Find the last name and first name of all authors who wrote books in exactly two subjects.

• **SQL:**

```sql
SELECT A.LASTNAME, A.FIRSTNAME
FROM AUTHOR A, BOOK B
WHERE B.AID=A.AID
GROUP-BY A.AID, A.LASTNAME, A.FIRSTNAME
HAVING COUNT(DISTINCT SUBJECT)=2
```
• **Query 6**: For each customer, display the CID, last-name, and the number of orders in 2010 - include only customers that made more than 100 orders in 2010.

• **SQL:**

```
SELECT CUSTOMER.CID, LAST-NAME, COUNT(*)
FROM ORDER, CUSTOMER
WHERE ORDER.CID=CUSTOMER.CID AND ORDER-YEAR=2010
GROUP-BY CUSTOMER.CID, LAST-NAME
HAVING COUNT(*)>100
```
• **Query 7**: Display the CID and last-name of customer(s) who have ordered the **largest total** quantity of books.

• **SQL**:

```sql
SELECT TEMP.CID, TEMP.LAST-NAME
FROM
(SELECT C.CID, LAST-NAME, SUM(QUANTITY) AS TOTALQ
FROM CUSTOMER C, ORDER O, ORDER-DETAILS OD
WHERE C.CID=O.CID AND O.OID = OD.OID
GROUP-BY C.CID, C.LAST-NAME) AS TEMP
WHERE TEMP.TOTALQ =
(SELECT MAX(TOTALQ) FROM TEMP)
```
• Give SQL queries for the following algebra expressions
  
  \[ \pi_{Ename}(\sigma_{Eid=5} \text{Employee}) \]
  
  SELECT Ename
  FROM Employee
  WHERE Eid=5
  
  \[ \pi_{Eid} \text{Employee} - \pi_{Eid} \text{Works} \]
  
  SELECT Eid FROM Employee
  EXCEPT
  SELECT Eid FROM Works
  
  \[ \pi_{E,Ename}(p(E, \text{Employee}) \join_{Eid} (\pi_{Eid} \text{Employee} - \pi_{Eid} \text{Works})) \]
  
  SELECT Ename FROM Employee
  WHERE NOT EXISTS (SELECT *
    FROM Works
    WHERE Employee.Eid=Works.Eid)
• $\pi_{\text{Employee}.\text{Ename}, \text{Department}.\text{Eid}} (\text{Employee \ JOIN}_{\text{Eid}} \text{Works \ JOIN}_{\text{Did}} \text{Department})$

  SELECT Employee.\text{Ename}, Department.\text{Eid}
  FROM Employee, Works, Department
  WHERE Employee.\text{Eid}=\text{Works}.\text{Eid} \ AND \ Works.\text{Did}=\text{Department}.\text{Did}$

• $p(\text{E1, Employee}), p(\text{E2, Employee})$

• $\pi_{\text{Employee}.\text{Ename}} (\text{Employee \ JOIN}_{\text{Eid}} ((\pi_{\text{E1}.\text{Eid}} \text{E1})-
  (\pi_{\text{E1}.\text{Eid}} (\sigma_{\text{E1}.\text{Salary}<\text{E2}.\text{Salary}} \text{E1XE2}))))$

  SELECT E1.\text{name}
  FROM Employee as E1
  WHERE E1.\text{salary} \geq \text{ALL} (\text{SELECT E2}.\text{salary}
    \ FROM Employee \ as \ E2)$