COMP3278 Introduction to Database Management Systems

1. **[50 marks]** A library provides a book booking system for its members. The system has a database that consists of the following relational tables:
`Member(mID, name, dob)` - stores the information of members, where `mID` is the unique identifier of a member, `dob` is the date of birth of the member.

`books(bID, name, category, location, PublishedIn)` - Stores the information of facilities, where `bID` is the unique identifier of a book; `category` is the classification of the book; `location` is the location where the book is placed; `PublishedIn` is the name of country where it was published;

`Reservation(mID, bID, startTime, endTime)` - Stores the reservation records of the book, and the time intervals `[startTime, endTime]` of the reservations (which member reserves which record in which period of time, inclusive).

Express the following queries in **relational algebra expression.**
a. **[5 marks]** Find the `name` of the members who reserved a book that is published in "Austrilia".

$$\pi_{M.name}(\sigma_{F.PublishedIn='Austrilia'}(\rho_b(Books) \bowtie \rho_R(Reservation) \bowtie \rho_M(Member)))$$

b. **[15 marks]** Find the names of all comic books (`category` equals "comic") in the main library (`location` equals "main library") and the corresponding reservation count (number of time the book has been reserved, comic books with 0 counts also need to be listed).

$$_{Books.name}\mathcal{G}_{count()}\left(\sigma_{Books.category='comic' \wedge Books.location='Main\ Library'}(Books \bowtie Reservation)\right)$$

c. **[10 marks]** Find the name of the members who have reserved all books in the main library(not necessary in the same period of time).

$$\pi_{Member.name}((Member \bowtie Reservation) \div \pi_{Books.BID}(\sigma_{Books.location='Main\ Library'}(Books)))$$

d. **[20 marks]** Find the ID(s) of the members who reserved more than one books in the same period of time (you need to consider overlapping range but not just ranges with the same startTime and endTime).

$\pi_{R1.mID}(\sigma_{R1.mID=R2.mID \wedge R1.startTime<=R2.endTime \wedge R2.startTime<=R1.endTime}(\rho_{R1}(Reservation) \times \rho_{R2}(Reservation)))$

2. [50 marks] Express the following queries in **SQL code**, please submit your SQL to Moodle Virtual Programming Lab (VPL).

Consider the following three tables in a book shop.

- `books (bookID, name, PublishedIn, Price)`
- `Customers (cID, date)`
- `PurchaseDetail (cID, bookID, quantity)`

·cID references Customers (cID)
·bookID references book (bookID)

Since a Purchase can contain multiple books, the relation PurchaseDetail matches up Customers (identified by cID) with books (identified by bookID) and stores the quantity of the book purchased by each customer

Table books

bookID	name	PublishedIn	Price
1	Harry Potter	UK	143.5
2	Intro to DB	USA	250.5
3	Quotations from Chairman Mao Tse-tung	China	542
4	Grimms' Fairy Tales	Germany	374.8
5	Deep Learning	USA	NULL

Table Customers

cID	Nationality
1	USA
2	PRC
3	DPRK
4	UK
5	UK

Table PurchaseDetail

cID	bookID	quantity
1	1	6
1	3	1
1	4	3
3	2	1
3	3	1
3	5	2
4	5	1
5	1	4
5	4	4

1. List the bookID, name, Price of the book(s) with Price in between 200 and 300 (inclusive), in descending order of bookID.

bookID	name	Price
2	Intro to DB	250.5

```
Select bookID, name, Price
from books
where books.Price>200 and books.Price<300
```

2. List the bookID, name of the book(s) with 'Ha' as the prefix in name, in descending order of bookID.

bookID	name

1	Harry Potter

```
SELECT bookID, name
from books
where books.name like 'Ha%'
order by bookID desc
```

3. List the bookID, name of the book(s) in the Customers with cID equals to 1, in descending order of bookID.

bookID	name
4	Grimms' Fairy Tales
3	Quotations from Chairman Mao Tse-tung
1	Harry Potter

```
select DISTINCT B.bookID, B.name
from books B, Customers C, PurchaseDetail P
where B.bookID = P.bookID and P.cID = 1
order by B.bookID desc
```

4. List the cID of the Customer (s) with no Purchase details, in descending order of cID.

cID
2

```
select DISTINCT C.cID
from Customers C, PurchaseDetail P
where C.cID NOT IN (
select cID
from PurchaseDetail
)
order by C.cID desc
```

5. List the bookID, name of the book(s) with NULL Price, in descending order of bookID.

bookID	name
5	Deep Learning

```
select bookID, name
from books
where books.Price is NULL
order by bookID desc
```

6. List the cID, Nationality of the Customers that bought the books published in “China”, in descending order of cID, there should be no duplicate cID in the output.

cID	Nationality
3	DPRK
1	USA

```
.select DISTINCT C.cID, C.Nationality
from Customers C, PurchaseDetail P, books B
where C.cID = P.cID and P.bookID = B.bookID and B.PublishedIn = 'China'
order by C.cID desc
```

7. List the cID of the Customer(s) that has bought both the following books

book with bookID equal to 1

book with bookID equal to 3

In descending order of cID.

cID

1

```
select C.cID
from Customers C, PurchaseDetail P
where C.cID = P.cID and P.bookID IN (1,3)
group by C.cID
having count(*) = 2
order by C.cID desc
```

8. List the `cID`, `distinctBookCount` of the Customers that has bought more than 1 distinct books, in descending order of `cID`.
`distinctBookCount` is the number of distinct books bought by that Customer.

cID	distinctBookCount
5	2
3	3
1	3

```
select cID, count(bookID)
from PurchaseDetail
group by cID
having count(bookID)>1
order by cID desc
```

9. List the `cID`, `distinctBookCount` of **all** Customers in the Customers table, in descending order of `cID`.
`distinctBookCount` is the number of distinct books bought by that Customer.

cID	distinctBookCount
5	2
4	1
3	3
2	0
1	3

```
select C.cID, count(P.bookID)
from Customers C left outer join PurchaseDetail P on C.cID = P.cID
group by C.cID
order by C.cID desc
```

10. List the `bookID`, `name, Published in "USA"` that are not bought by the Customer with `cID` equal to `1`, in descending order of `bookID`.

bookID	name
2	Intro to DB
5	Deep Learning

```
select DISTINCT B.bookID, B.name
from books B left outer join PurchaseDetail P on  P.bookID = B.bookID
where B.PublishedIn = 'USA' and (P.cID != 1 or P.cID is NULL)
order by B.bookID desc
```

11. List the `cID` and `PurchaseCount` of Customer who bought the most number of books published in "China", `PurchaseCount` is the number of books (published in China) that customer bought, in descending order of `cID`

cID	PurchaseCount
3	1
1	1

```
select P2.cID, COUNT(*)
 from PurchaseDetail P2, books B2
 where B2.PublishedIn = "China"
 AND P2.bookID = B2.bookID
   GROUP BY P2.cID
  HAVING COUNT(*) =
    (
     select COUNT(*)
     from PurchaseDetail P, books B
    where B.PublishedIn = "China"
    AND P.bookID = B.bookID
    GROUP BY P.cID
    order by COUNT(*) desc
    limit 1
    )
   order by P2.cID desc;
```

12. List the most popular book's `bookID` and `name`. (the most quantity of purchases)

bookID	name
1	Harry Potter

```
SELECT P1.bookID, B.name
FROM books B, PurchaseDetail P1
WHERE P1.bookID = B.bookID
GROUP BY P1.bookID
HAVING SUM(P1.quantity) = (
  SELECT MAX(M.purchases)
  FROM (
    SELECT SUM(P2.quantity) AS purchases
    FROM PurchaseDetail P2
    GROUP BY P2.bookID
  ) AS M
);
```