

# CS 273 Homework

Shiyu Ji

## 1 Problem A

Your task is to design a database for Amazon that sells books. The following is the description of the application.

- Each book has a name and a publisher.
- Each book has many versions. Each version has a version number, the year and date it was first published, and the authors.
- Registered viewers can rate any version of a book. For each rate, we want to record its post date and rate.
- For each registered viewer, we want to record his/her userID and password.

If you do not have sufficient information, please state your assumptions clearly.

- (a) Draw an ER diagram for this application. Be sure to mark the multiplicity of each relationship of the diagram. Decide the key attributes and identify them on the diagram.
- (b) Convert the above ER diagram into a relational schema. Merge relations where appropriate. Your solution should have as few relations as possible, but they do not need to be normalized. Specify the key of each relation in your schema.
- (c) Write a 3NF for the derived relational schema.

## 2 Problem B

The following questions refer to the database schema below: Product(pid, price, color), Order(cid, pid, quantity), Customer(cid, name, age).

- (a) Write a query, in Relational Algebra, to return the names of customers who order at least one product with color “Yellow”.

**Answer:**

$$\pi_{\text{name}}(\text{Customer} \bowtie \text{Order} \bowtie \sigma_{\text{color}=\text{Yellow}}(\text{Product})).$$

- (b) Write a query in Tuple Relational Calculus and Domain Relational Calculus, to return the product with the highest price.

- (c) Write an SQL query, to return the total quantity of products ordered by customers with age greater than 50.

**Answer:**

```
select SUM(quantity)
from Order
where cid in (
  select cid
  from Customer
  where age > 50
);
```

(d) Write an SQL query, to return the pid(s) of the most ordered product(s) (i.e. the product(s) with the highest total ordered quantities).

**Answer:**

```
select pid
from Product
where pid in (
  select pid
  from (
    select pid, SUM(quantity) as QuantitySum
    from Order
    group by pid
  )
  where QuantitySum in (
    select MAX(QuantitySum)
    from (
      select Sum(quantity) as QuantitySum
      from Order
      group by pid
    )
  )
);
```

### 3 Problem C

We have an employee database with three tables. The Employee table stores information about employees. Every employee is identified by an **EmployeeID**. The Department table stores information about each department. The Vacations table stores information about the total number of days each employee takes for vacation.

- Employee(EmployeeID, FirstName, LastName, Office, Email, DepartmentID)
- Department(DepartmentID, DepartmentName)
- Vacations(EmployeeID, Days)

Answer the following queries using SQL and write its corresponding TRC if exists.

a) List the Office and Email of employee “John Smith”.

**Answer:**

```
select Office, Email
from Employee
where FirstName = 'John' and LastName = 'Smith';
```

b) List the number of vacation Days taken by each employee with the name “John Smith”.

**Answer:**

```
select Days
from Vacations
where EmployeeID in (
  select EmployeeID
  from Employee
  where FirstName = 'John' and LastName = 'Smith'
);
```

c) List the FirstName and LastName of all employees who never took a vacation day.

**Answer:**

```
select FirstName, LastName
from Employee
where EmployeeID in (
    select EmployeeID
    from Vacations
    where Days = 0;
);
```

d) List the DepartmentName of every department whose total number of vacation days taken by its employees is the largest among those of all the departments.

**Answer:**

```
select DepartmentName
from Department
where DepartmentID in (
    select DepartmentID
    from (
        select DepartmentID, SUM(Days) as DaySum
        from Employee inner join Vacations
        on Employee.EmployeeID = Vacations.EmployeeID
        group by DepartmentID
    )
    where DaySum in (
        select MAX(DaySum)
        from (
            select SUM(Days) as DaySum
            from Employee inner join Vacations
            on Employee.EmployeeID = Vacations.EmployeeID
            group by DepartmentID
        )
    )
);
```

## 4 Problem D

We perform decomposition to normalize an original schema to be of certain normal forms. For such a decomposition to be “equivalent” to the original schema, it is desirable to be lossless. To study this concept, let us consider an original schema  $R(A, B, C)$ . Suppose we decompose  $R$  into  $R_1(A, B)$  and  $R_2(B, C)$ .

(a) Is this decomposition always lossless? Answer yes or no and briefly explain why.

**Answer:** No. Suppose two tuples have the same value of  $B$  but different values of  $A$  and  $C$ . Then the decomposed tuples in  $R_1$  and  $R_2$  give 4 possible combinations, implying it is impossible to recover the original schema.

(b) Give an example instance of  $R$  (i.e. an example table with several tuples) and demonstrate its decomposition, to support your answer in (a).

**Answer:** A “lossy” relation  $R$  can be given as follows:

$A$	$B$	$C$
3	0	0
4	0	1

After decomposition  $R_1$  contains  $(3, 0)$  and  $(4, 0)$ , and  $R_2$  contains  $(0, 0)$  and  $(0, 1)$ . Clearly the possible schema before the decomposition has another possibility:  $(3, 0, 1)$  and  $(4, 0, 0)$ , implying it is impossible to recover the original schema.

## 5 Problem E

Given below is the set  $\mathbf{F}$  of functional dependencies for the relational schema:

$$R = \{A, B, C, D, E, F\}$$

$$A \rightarrow BC$$

$$BD \rightarrow E$$

$$E \rightarrow F$$

$$F \rightarrow D$$

$$E \rightarrow D$$

1. Is the set  $\mathbf{F}$  a minimal cover? Explain the reason. If  $\mathbf{F}$  is not minimal cover, find a minimal cover for  $\mathbf{F}$ .
2. Based on the minimal cover for  $\mathbf{F}$ , produce a lossless BCNF decomposition for this schema. Is the result dependency-preserving? Explain why or why not.
3. Based on the minimal cover for  $\mathbf{F}$ , produce a dependency preserving 3NF decomposition. Is the result redundancy free? Explain why or why not.