ITP30010 21900136 Kim, Shinhoo

Homework Assignment 1
Due: 11:59PM March 24, 2023

1. (3 pt. each) Please answer to the following questions in your own words.
(a) (Exercise 6.5) An E-R diagram can be viewed as a graph. What do the following mean in terms of the structure of an enterprise schema?
- The graph is disconnected.

Answer: If you think entity sets of E-R diagram as nodes in a graph and the relationship sets are edges, then you can understand that there are some pairs of entity sets that are not in the relationship sets because “The graph is disconnected” means a connection between nodes is disconnected.

- The graph has a cycle.

Answer: Like the above understanding between graph and E-R diagram, “The graph has a cycle” can imply there are pairs of entity sets that are in the relationship.

(b) (Exercise 6.17) Explain the difference between a weak and a strong entity set.

Answer: For Entity set to be called as Strong Entity set, it should be independent of any other entities in a schema which means it should have a primary key so that it can be uniquely identified. On the other hand, weak entity cannot be independent by itself. It is created in a relation between entity sets in order to identify the relationship of them with a discriminator or an union of the primary keys of the identifying entity sets. So, it would not have any meaning without those entity sets and by itself. So, the big difference between a week and a strong entity set is whether it is independent or not.

(c) (Exercise 6.19) We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?

Answer: There are several reasons why we still need or keep them as “week entity sets”. Firstly, if you give the key of strong entity to week entity sets, then the data will be duplicated and consequent possible inconsistencies. Secondly, weak entities will provide better understanding of the logical structure since it reflects the logical structure that has a relation of other entities. Lastly, since weak entities does not have any meaning without relying entities, the weak entities will be removed automatically when its relying entities are deleted.

(d) (Exercise 7.10) Our discussion of lossless decomposition implicitly assumed that attributes on the left-hand side of a functional dependency cannot take on null values. What could go wrong on decomposition, if this property is violated?

Answer: In Lossless decomposition, there will be no information that will be lost in the decomposition process but it there are the attributes on the left-hand side of a functional dependency were to take on null values, then it could possibly affect the determination of the values on the right-hand side and could lead to inconsistencies in the data which means there could be information lose in decomposition process of a lossy decomposition.

(e) (Exercise 7.21) Repetition of information and inability to represent information can be defined as next:
- Repetition of information: a condition in a relational database where the values of one attribute are determined by the values of another attribute in the same relation, and both values are repeated throughout the relation.
- Inability to represent information: a condition where there is a relation- ship that exists among only a proper subset of the attributes in a relation.

Explain why each of these properties may indicate a bad relational-database design.

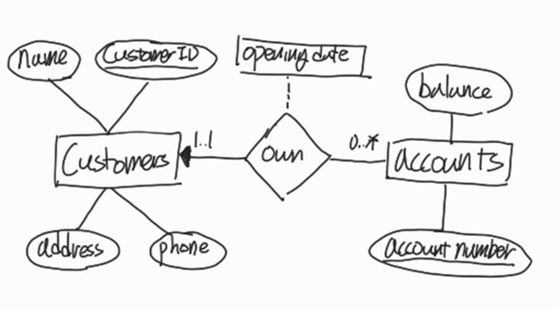
Asnwer: Repetition of information can cause inconsistencies because if the value of one attribute changes, all the corresponding values in the relation must also be updated. If this is not done correctly, it can obviously cause inconsistent data. Repetition of information can also lead to inefficiencies because it increases the amount of storage space required and can slow down database operations. Inability to represent information can lead to difficulties in maintaining the data because it may require the use of null values or complex workarounds to represent relationships among a proper subset of the attributes in a relation. This can make it difficult to ensure data integrity and can make it more difficult to query and update the data. Those are the potential bad relational- database design cause by the two situations.

(f) (Exercise 7.22) Why are certain functional dependencies called trivial functional dependencies?

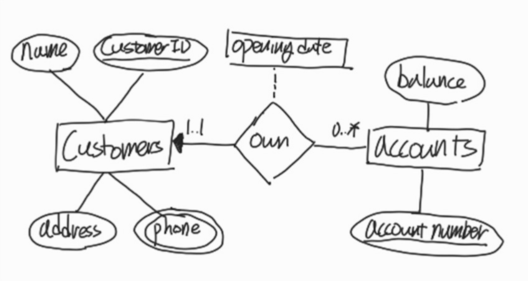
Answer: Trivial functional dependencies can hold for any relation, regardless of the data it contains. That is why they are called “trivial”. This is because the values of the attributes on the right-hand side are determined by the values of the attributes on the left-hand side, and since the right-hand side is a subset of the left-hand side, it means that the values on the right-hand side are already included in the values on the left-hand side.

2. Draw the E-R diagrams for the following databases. Be sure to indicate the cardinalities of the relationship.
(a) (3 pt.) Design a database for a bank, including information about customers, their accounts, and the own relationship between them. Information about a customer includes their name, address, phone, and customer ID. An Account has an account number and balance. Also, the own relationship keeps opening date of each account. Note that:

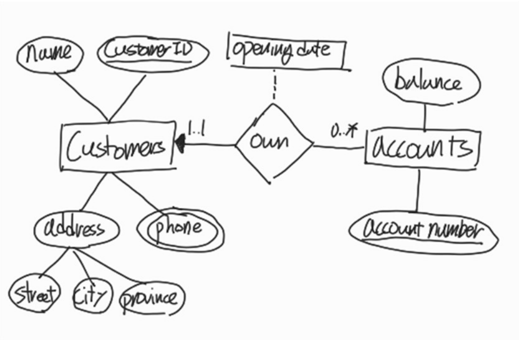
- A customer can own multiple accounts.
- An account is owned by only one customer.
- Customer ID and Account number are unique to each customer and account, respectively.



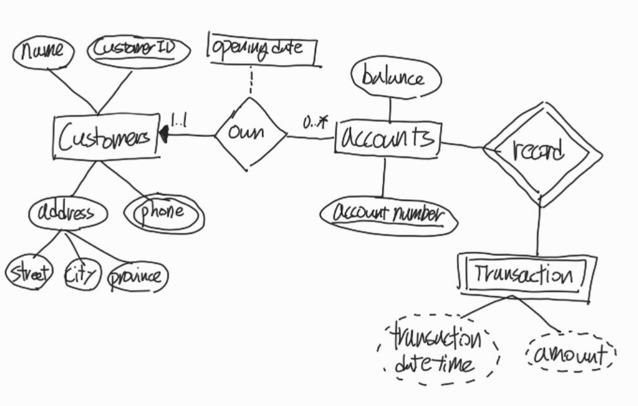
(b) (1 pt.) Modify your original diagram of Problem 2(a) such that a customer can have multiple phone numbers.



(c) (1 pt.) Change your diagram of Problem 2(b) such that a customer has an address represented by composite attributes (which are street-city-province triplets). Note that multiple customers may live at a single address.



(d) (2 pt.) Add a weak entity set of transactions next to account (connected via a relationship record). This entity set contains transaction datetime and amount as its attributes.



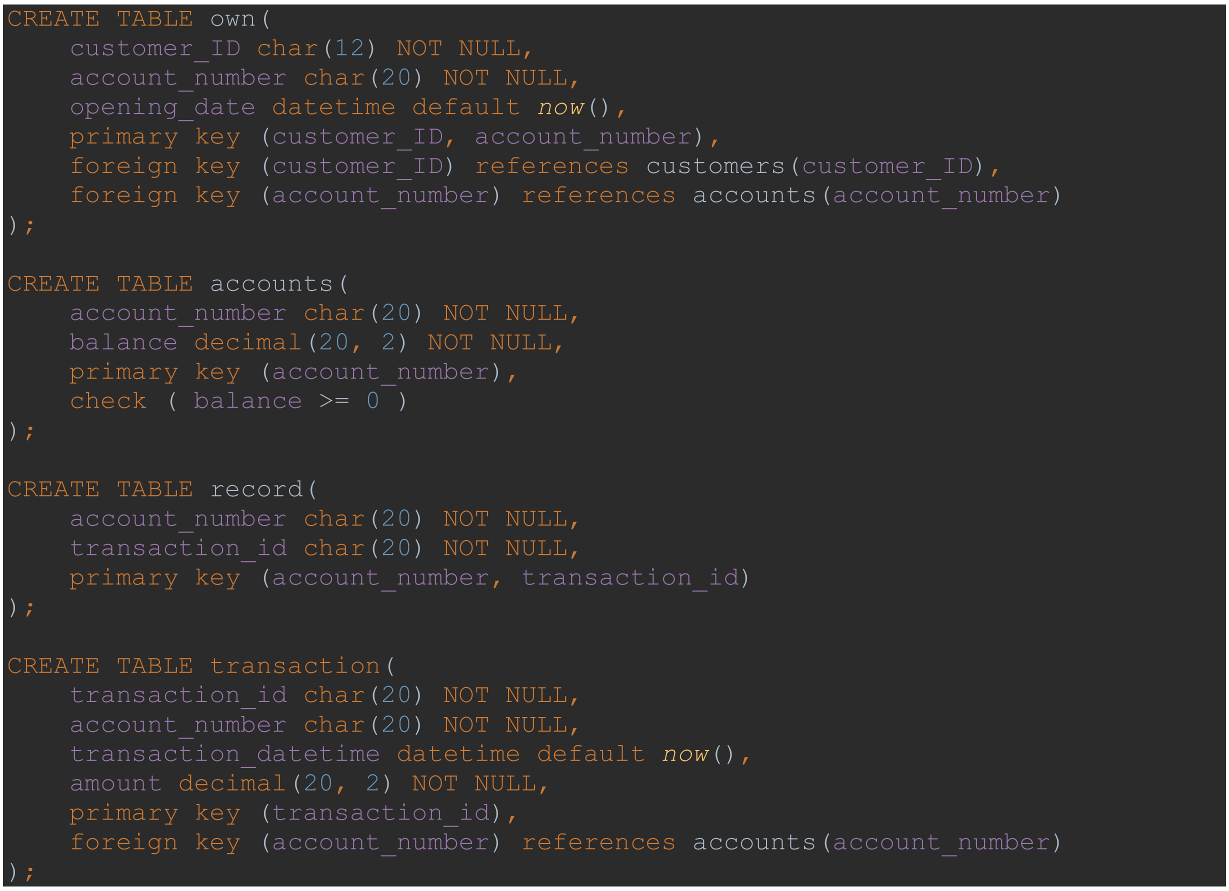
(e) (3 pt.) Convert the following E-R diagram into SQL DDL (CREATE TABLE statements).

- Consider the cardinalities of the relationship.
- Recall that we do not allow attributes to have non-primitive data types.

```
CREATE TABLE customers
(
    customer_ID char(12) NOT NULL,
    name varchar(100) NOT NULL,
    primary key (customer_ID)
);

CREATE TABLE customer_phone
(
    customer_ID char(12) NOT NULL,
    phone char(11) NOT NULL,
    primary key (customer_ID, phone),
    foreign key (customer_ID) references customers(customer_ID)
);

CREATE TABLE customer_address
(
    customer_ID char(12) NOT NULL,
    street varchar(50),
    city varchar(50),
    province varchar(50),
    primary key (customer_ID, street, city, province),
    foreign key (customer_ID) references customers(customer_ID)
);
```



```
CREATE TABLE own(
    customer_ID char(12) NOT NULL,
    account_number char(20) NOT NULL,
    opening_date datetime default now(),
    primary key (customer_ID, account_number),
    foreign key (customer_ID) references customers(customer_ID),
    foreign key (account_number) references accounts(account_number)
);

CREATE TABLE accounts(
    account_number char(20) NOT NULL,
    balance decimal(20, 2) NOT NULL,
    primary key (account_number),
    check ( balance >= 0 )
);

CREATE TABLE record(
    account_number char(20) NOT NULL,
    transaction_id char(20) NOT NULL,
    primary key (account_number, transaction_id)
);

CREATE TABLE transaction(
    transaction_id char(20) NOT NULL,
    account_number char(20) NOT NULL,
    transaction_datetime datetime default now(),
    amount decimal(20, 2) NOT NULL,
    primary key (transaction_id),
    foreign key (account_number) references accounts(account_number)
);
```

3. Normalization.

(a) (3 pt.) Is every relation in 3NF also in BCNF? If yes, explain why. If no, given a counter example.

Answer: No, not every relation in 3NF is also in BCNF. The difference between the two normal forms is subtle but important. Every relation in BCNF is also in 3NF, but the reverse is not necessarily true.

Counter example: Consider a relation R with attributes A, B, and C. Suppose that the functional dependencies are A → B and B → C. In this case, the relation is in 3NF because all non-prime attributes (C) are dependent on the primary key (A). However, it is not in BCNF because there is a non-trivial functional dependency (B → C) where B is not a superkey.

(b) (3 pt.) Is every relation in 4NF also in BCNF? If yes, explain why. If no, given a counter example.

Answer: Yes, every relation in 4NF is also in BCNF. This is because 4NF is a stronger normal form than BCNF. Since every relation in 4NF must also be in BCNF, it follows that every relation in 4NF is also in BCNF.

(d) (3 pt.) The following relation violates {1NF, 2NF, 3NF, 4NF, BCNF}?

Answer: 2NF, 3NF, 4NF, BCNF

(e) (3 pt.) The following relation violates {1NF, 2NF, 3NF, 4NF, BCNF}?

Answer: 2NF, 3NF, 4NF, BCNF

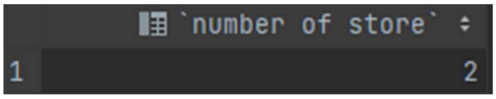
(f) (3 pt.) The following relation violates {1NF, 2NF, 3NF, 4NF, BCNF}?

Answer: 3NF, 4NF, BCNF

5. More SQL queries. Launch and access the MySQL databases distributed with the class virtual machine. Below uses the “sakila” database (DVD rental database), which consists of 16 tables regarding movie inventory, actors, customers, rental history, payment information, etc. For each of the following questions, find the answer based on the information recorded in the database and write a query that shows how you obtained the answer.

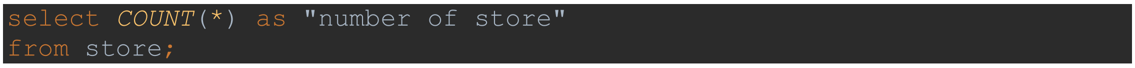
(a) (2 pt.) How many stores are found in the database?

Answer to the question:



```
    `number of store`
1                   2
```

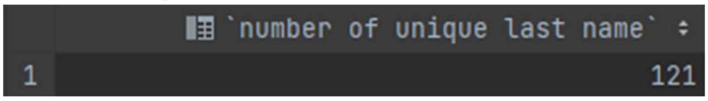
Query to find the answer:



```
select COUNT(*) as "number of store"
from store;
```

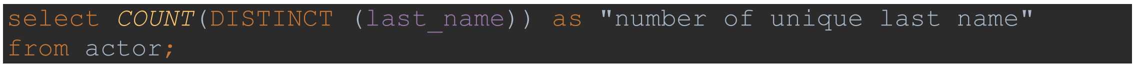
(b) (2 pt.) How many unique last names are found in the actor relation?

Answer to the question:



```
    `number of unique last name`
1                            121
```

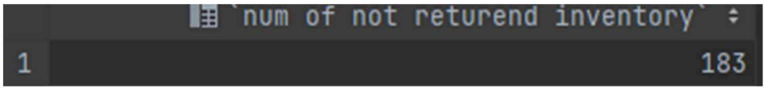
Query to find the answer:



```
select COUNT(DISTINCT (last_name)) as "number of unique last name"
from actor;
```

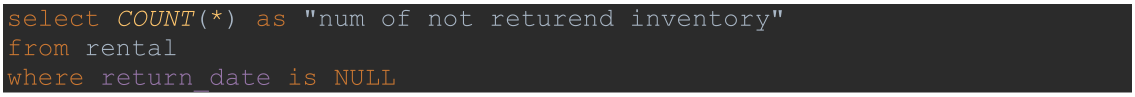
(c) (2 pt.) According to the database, how many inventories (DVDs) have not been returned (inventories that have not been returned do not have return_date)?

Answer to the question:



```
    `num of not returend inventory`
1                               183
```

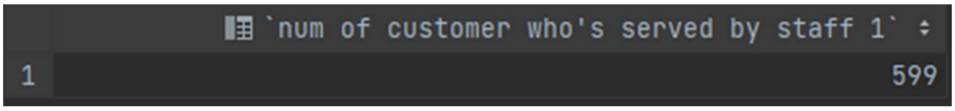
Query to find the answer:



```
select COUNT(*) as "num of not returend inventory"
from rental
where return_date is NULL
```

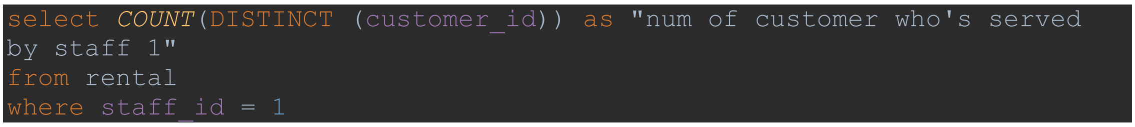
(d) (2 pt.) How many distinct customers have rented a movie title(s) from staff_id=1?

Answer to the question:



```
    `num of customer who's served by staff 1`
1                                         599
```

Query to find the answer:



```
select COUNT(DISTINCT (customer_id)) as "num of customer who's served
by staff 1"
from rental
where staff_id = 1
```