**Homework Assignment 3**

Due: 11:59PM, Saturday, April 22, 2023

**1. 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: Each entity set has no relationship.

- The graph has a cycle: the entity sets are mapping with each other. (Has relationships)

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

- A weak entity set is an entity set that is dependent on another entity set (strong entity set) without a own primary key. It has attribute that provide additional information about the stronger entity set, called discriminator and its PK is consist of PK of strong entity set and discriminators of it. A strong entity set is an entity set can has weak entity set and has own PK.

(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?

- by using weak entity set, we can reduce the redundancy and sure that the relationship between the entities can be presented more accurately and get more referential integrity.

(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?

- If we ignore that, we might loss some data. When the left-hand side of a functional dependency contains null values, it might cause the missing data after the decomposition because when we do natural join after decomposition, will not every produce all the tuples from the original relation.

(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.

- both might be harmful to RDBMS. first, repetition of information cause data redundancy and inconsistency problem. It makes us to spent large storage and bring lots of uncomfortable when we try insert, update, and delete data. And if there is inability of represent information, we cannot store and get data we exactly want when we need it.

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

- Certain functional dependencies are called trivial functional dependencies because they are just part of origin, so is always true and cannot be useful information about the data.

**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 *own*ed by only one customer.

- *Customer ID* and *Account number* are unique to each *customer* and *account*, respectively.

도표이(가) 표시된 사진

자동 생성된 설명

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

도표이(가) 표시된 사진

자동 생성된 설명

(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.

도표이(가) 표시된 사진

자동 생성된 설명

(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.

도표이(가) 표시된 사진

자동 생성된 설명

(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 INT,

name VARCHAR (20),

street VARCHAR (50),

city VARCHAR (50),

province VARCHAR (50),

PRIMARY KEY (customer\_id)

);

CREATE TABLE accounts (

account\_number INT,

customer\_id INT,

balance INT,

PRIMARY KEY (account\_number)

);

CREATE TABLE own (

customer\_id INT,

account\_number INT,

opening\_date DATE,

PRIMARY KEY (customer\_id, account\_number)

);

CREATE TABLE transactions (

account\_number INT,

transaction\_datetime DATETIME,

amount INT,

PRIMARY KEY (account\_number, transaction\_datetime, amount)

);

CREATE TABLE phone (

customer\_id INT,

phone\_number VARCHAR (20),

PRIMARY KEY (customer\_id, phone\_number)

);

**3. Normalization**

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

- Not every relation in 3NF is also in BCNF. There can be cases where a relation is in 3NF but violates the requirements of BCNF. Because BCNF(3.5NF) is a stricter form of normalization than 3NF. For example, If the normal column determines the candidate key, the table will satisfy 3NF but not BCNF.

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

- yes, it is. BCNF is 3.5NF and 4NF is higher level. Every relationship in the 4NF is also in the BCNF. because in the 4NF, every relationship already removed all trivial multi-value dependencies, so it also removes all trivial functional dependencies that violate the BCNF. So, if a relationship satisfies the 4NF, we can say that it also satisfy the BCNF.

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

테이블이(가) 표시된 사진

자동 생성된 설명

* Violates 2NF, 3NF, 4NF, BCNF
* *Employee\_id* and *name* are violating 2NF.

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

테이블이(가) 표시된 사진

자동 생성된 설명

- Violates 4NF, BCNF

- *Branch* and *branch\_address* violating BCNF

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

테이블이(가) 표시된 사진

자동 생성된 설명

- Violates 3NF, 4NF, BCNF

- *dept\_id* and *dept\_name* violating 3NF.

**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: 2

-Query to find the answer: SELECT COUNT(\*) FROM store ;

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

-Answer to the question: 66

-Query to find the answer:

WITH unique\_last\_name(last\_name) AS(

SELECT last\_name FROM actor

GROUP BY last\_name

HAVING COUNT(last\_name) = 1)

SELECT COUNT(\*) FROM unique\_last\_name;

(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: 183

-Query to find the answer: SELECT COUNT(inventory\_id) 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: 599

-Query to find the answer: SELECT COUNT(DISTINCT customer\_id) FROM rental WHERE staff\_id = 1;