Homework Assignment 3

Maximum earnable: 77 pt.
Due: 11:59PM, April 24, 2025

- Read the assignment carefully. *You will need to* ***write and execute several SQL queries****; and* ***submit the results of your queries****.*
- You are **allowed to re-use any of the queries from the lecture slides** while developing solutions to the problems.
- This is an individual work; Please be clear with HGU CSEE Standard:
  - Submitting assignments or program codes written by others or acquired from the internet without explicit approval of the professor is regarded as cheating.
  - Showing or lending one's own homework to other student is also considered cheating that disturbs fair evaluation and hinders the academic achievement of the other student.
  - It is regarded as cheating if two or more students conduct their homework together and submit it individually when the homework is not a group assignment.
- **Use of ChatGPT or similar AI tools**: Students are prohibited from using ChatGPT or similar AI platforms to directly obtain solutions for this assignment. The intent of the assignment is to exercise your understanding and application of the course material. Leveraging AI tools to bypass this learning process is considered a breach of academic integrity. Any evidence of such behavior will result in penalties.
- When finished, submit your work to *LMS.*

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.
- The graph has a cycle.

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

(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?

(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?

(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.

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

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.

(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 *transaction*s 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.

3. Normalization.

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

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

(c) (3 pt.) The following relation violates {1NF, 2NF, 3NF, 4NF, BCNF}? Justify your answer.

employee_id	**name**	**position**	**previous_branch**
1001	Brown	Sales representative	Pohang
1001	Brown	Sales representative	Busan
1005	Hopkins	Software engineer	Seoul
2001	Kim	Software engineer	Busan
3004	Kim	Product manager	Seoul
3004	Kim	Product manager	Wonju
3005	Clermont	Project administrator	Seoul

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

employee_id	**name**	**branch**	**branch_address**
1001	Brown	Seoul	Garosu-gil 233
1004	Green	Seoul	Garosu-gil 233
1005	Hopkins	Pohang	Handong-ro 501
2001	Kim	Seoul	Garosu-gil 233
3002	Walker	Seoul	Garosu-gil 233
3004	Kim	Pohang	Handong-ro 501
3005	Clermont	Pohang	Handong-ro 501

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

employee_id	**name**	**branch**	**dept_id**	**dept_name**
1001	Brown	Seoul	202	Sales
1004	Green	Seoul	201	Operation
1005	Hopkins	Pohang	303	Software development
2001	Kim	Seoul	303	Software development
3002	Walker	Seoul	303	Software development
3004	Kim	Pohang	308	User experience
3005	Clermont	Pohang	201	Operation

4. 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:

Query to find the answer:

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

Query to find the answer:

(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:

Query to find the answer:

(d) (2 pt.) How many distinct *customers* have rented a movie title(s) from *staff_id*=1?
Answer to the question:

Query to find the answer:

(e) (3 pt.) How many distinct films rated 'PG' are available?
Answer to the question:

Query to find the answer:

(f) (3 pt.) How many active customers are living in the district of England?
Answer to the question:

Query to find the answer:

(g) (4 pt.) Considering the rental history (*rental*) and payment history (*payment*), who has paid the largest amount of money for renting movies? List the *first* and *last name* of the *customer*, the *total number* of movie rentals, and *total amount* of money s/he has paid.
Answer to the question:

Query to find the answer:

(h) (4 pt.) List three most common *categories* of film available at *store_id*=2 (if a store has multiple copies of the same film, consider each copy as an individual inventory).
Tip: Use **LIMIT** 3 *at the end of your query to limit the number of output tuples.*
Answer to the question:

Query to find the answer:

(i) (3 pt.) What is the *title* of the movie that has the longest description (*film_text.description*) among the rental store with *store_id*=2 has?
Answer to the question:

Query to find the answer:

(j) (4 pt.) Which of the films starred by "FRED COSTNER" rented the most? Write the title of the film.
Answer to the question:

Query to find the answer:

(k) (2 pt.) Using the *'customer_list'* view, list all names of people whose address is in the city of 'London'.
Answer to the question:

Query to find the answer:

(l) (3 pt.) Write a query *that uses only tables (does not use any views)* and returns the same information as in the previous problem (Problem (k)).
Answer to the question:

Query to find the answer: