CSC 472A
Introduction to Database Systems
Project Database Design and Implementation

1. Project description and purpose. Choose an application that you think is useful that has relevance in database system design. The database must have unnormalized relation so that you can come out with 1NF and 2NF .The database must have minimum of three entity sets in the final or normalized form. Each entity set must have at least 2 attributes.

   1. This is a database for department store operations. The database tracks inventory, sales, transactions, employees, deliveries, customers, etc. The purpose of this project is for the database to act like an enterprise system for a department store.

2. Unnormalized Form. Show the unnormalized relations. Specify the name, attributes separated by commas and place in parenthesis for each relation.

   1. inventory(product_id, product_name, supplier_name, supplier_location, price, quantity, transaction_id, sale_DateTime, discard_DateTime, discard_reason, discard_quantity)
   2. time sheet(cashier_id, name timeInOut, date)
   3. delivery_items(route_id, start_location, arrival_DateTime, product_id)
   4. customer(customer_id, name, location)

3. Normalized Form. Perform normalization or decomposition. Eliminate any partial functional dependencies, transitive functional dependences, repetitive repetition, and anomalies. Show or explain how you perform the normalization or decomposition process.

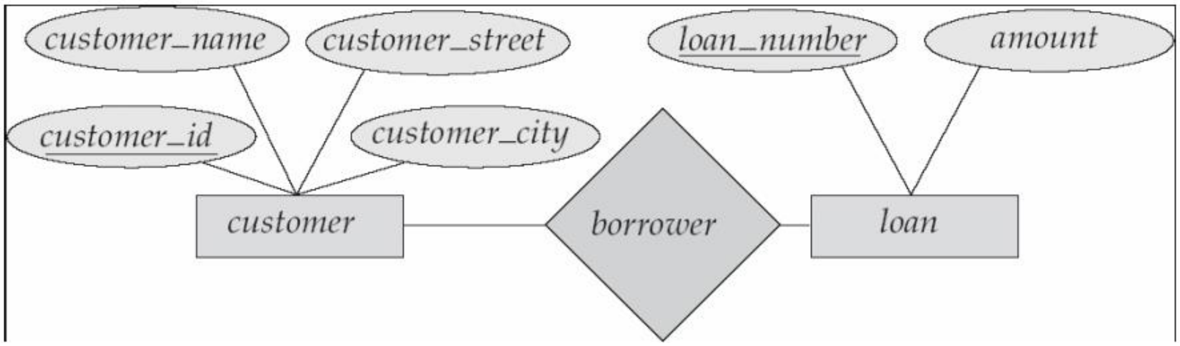
   1. 1NF
      1. inventory(product_id, product_name, price, quantity, supplier_name, city, state, transaction_id, sale_date, sale_time, cashier_id, cashier_first_name, cashier_last_name, discard_date, discard_time, discard_reason, discard_quantity)
      2. time sheet(cashier_id, first_name, last_name, time_in, time_out, date)
      3. delivery_items(route_id, city, state, date, time, product_id)
      4. customer(customer_id, first_name, last_name, city, state)
   2. 2NF
      1. inventory(product_id, product_name, price, quantity, supplier_name, city, state)
      2. sales(transaction_id, product_id, date, time, cashier_id, cashier_first_name, cashier_last_name)
      3. discarded_items(product_id, date, time, reason, quantity)
      4. time sheet(cashier_id, first_name, last_name, time_in, time_out, date)

    5. delivery_items(route_id, city, state, date, time, product_id)
    6. customer(customer_id, first_name, last_name, city, state)
3. 3NF
    1. See problem 4

4. Show all the schemas of the resulting or normalized relations in your database. Specify the name, attributes separated by commas and place in parenthesis for each relation. Underline the primary key(s).

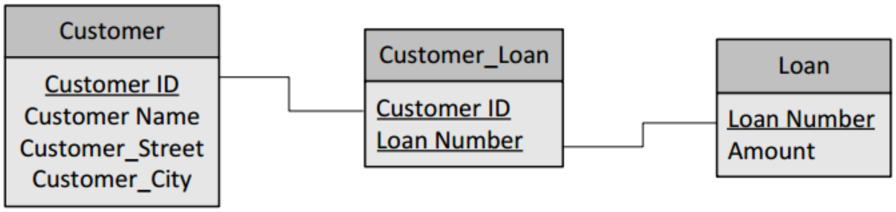
1. inventory(product_id, product_name, supplier_name, price, quantity)
2. sales(transaction_id, product_id, quantity)
3. transaction(transaction_id, cashier_id, customer_id, date, time)
4. time_sheet(cashier_id, date, time_in, time_out)
5. cashiers(cashier_id, first_name, last_name)
6. customer(customer_id, first_name, last_name, city, state)
7. supplier(supplier_name, city, state)
8. delivery_route(route_id, city, state, arrival_time, arrival_date)
9. delivery_items(product_id, route_id, quantity)
10. discarded_items(product_id, date, time, reason, quantity)

5. Basic ER diagram. Produce a basic ER diagram of your database which reflects entity sets, attributes, and relationships among them. Underline the primary key on the ER diagram. An example is shown in below.



Example of ER Diagram

6. Relational Database Schema. Convert the ER diagram into a relational database schema. Underline primary keys.



Customer Customer ID Customer Name Customer_Street Customer_City Loan Loan Number Amount Customer_Loan Customer ID Loan Number Example of Relational Database Schema Note: You can replace the relation name of Customer_Loan with borrower.

7. Algebraic Statements. Formulate any two algebraic queries that are relevant to the application you have chosen. For example, loan.number (⍰amount > 1200 (loan))

1. product_id(product_name == ‘iPhone’ (inventory))
2. cashier_id(first_name == ‘Justin’ (cashiers))

8. SQL Statements. Formulate any two queries in SQL statements that are relevant to the application you have chosen.

1. SELECT * FROM sales WHERE transaction_ID = 1776;
2. SELECT * FROM inventory WHERE supplier_name = ‘Nestle’;

9. XPath Expression. Convert three algebraic queries into XPath expressions.

1. /data/inventory/product_name/text()
2. /data/inventory[price > 500]
3. /data/supplier/supplier_name

10. Create Relations. Create two relations using SQL statements including constraints, i.e. show the SQL commands how to create all the relations.

1. CREATE TABLE cashiers (cashier_id NUMBER(4), first_name VARCHAR2(20), last_name VARCHAR2(20), CONSTRAINT cashiers_cashier_id_pk PRIMARY KEY (cashier_id));
2. CREATE TABLE supplier (supplier_name VARCHAR(20), city VARCHAR2(20), state VARCHAR2(2), CONSTRAINT supplier_supplier_name_pk PRIMARY KEY (supplier_name));

11. Populate Relations. Insert three records in at least two relations, i.e. show SQL commands on how to populate the relations.

1. INSERT INTO cashiers (cashier_id, first_name, last_name) VALUES (1234, ‘John’, ‘Smith’);
2. INSERT INTO supplier (supplier_name, city, state) VALUES (‘Great Value’, ‘Bentonville’, ‘AK’);

3. INSERT INTO supplier (supplier_name, city, state) VALUES (‘Anheuser Busch’, ‘St. Louis’, ‘MO’);

12. Briefly describe what were the least and most challenging parts of the above and explain why. Give any recommendation about the project.

    1. The most challenging part of the project was the normalization process. It’s difficult to distinguish between 1NF and 2NF. I think I have a solid grasp of 3NF and UNF. I watched some videos on YouTube that I thought may be helpful but I wasn’t sure if they were accurate and if what I was doing applied to my project. A suggestion to improve this course would be to include video lectures with the course materials. I’m a visual learner and it would help greatly if I could visualize the complex concepts while listening to an explanation.

Grading Areas:

1. Overall Presentation - 20 %

2. Process of normalization - 25 %

3. Effort - 25 %

4. Completeness - 30 %