Name: Natalie Poche
Assignment: Homework 5
Due: April 10, 2025

Homework 5

Problem Statement: Consider the relation BookingDetails with the following schema and functional dependencies:

Relation: BookingDetails(agent, traveler_ssn, trip_id, start_location, end_location, years_expereince, passport_number, expiration_date)

Functional Dependencies:

1. agent → years_experience
2. traveler_ssn → passport_number
3. passport_number → expiration_date
4. trip_id → start_location, end_location

Instructions:

1. Compute the Primary Key:
    a. Based on the schema and functional dependencies provided, compute the primary key for the BookingDetails relation.

        The primary key is {agent, traveler_ssn, trip_id} The reason for this is because years_experienced has a functional dependency with agent, traveler_ssn has a functional dependency with passport_number, and trip_id has a functional dependency with start_location and end_location. The reason why passport_number isn't included in the primary key is because it is functionally determined by traveler_ssn, passport_number is dependent on something and therefore is not considered a primary key.
2. Decompose into BCNF:
    a. Determine if any of the functional dependencies violate the BCNF. For each violation, decompose the relation accordingly
    b. Continue the decomposition until all resulting relations are in BCNF.

        In the first functional dependency, the attribute agent volates the BCNF because the left hand side is not a superkey as it doesn't include other keys, so based on the closure test, we can use the formula $R_2 = R-(X^+ - X)$ to break the BD into :
        - BD1: {agent, years_experience}
        - BD2: {agent, traveler_ssn, trip_id, start_location, end_location, passport_number, expiration_date}
        - Removing the key isolates dependency and eliminates redundancy

In the second functional dependency traveler_ssn, also violate the BCNF beause it is not a superkey, so based on the closure test BD2 can be droken down into:

- B3 : {traveler_ssn, passport_number, expiration_date}
- B4 : {traveler_ssn, agent, trip_id, start_location, end_location}
- Removing it allows a table of traveler-specific information to remove redundancy.

In the third functional dependency, passport_number violate the BDNF in BD3 because the left is not a proper key, let alone super key, so in the clusre test BD3 is broken down into

- BD5: {passport_number, expiration_date}
- BD6: {traveler_ssn, passport_number}
- Removing it removes redundancy

In the fourth functional dependency, trip_id violates the BCNF since it's not a superkey, so the closure test breaks it down into:

- BD7: {trip_id, start_location, end_location}
- BD8: {trip_id, traveler_ssn, agent}
- Removing the key removes duplications

3. List the Final Relations:
   a. After decomposing the relation, list all final relations in BCNF.
   b. Clearly indicate the attributes in each relation and specify the primary key for each decomposed relation.

   BD1: {agent, years_experience}
   BD5: {passport_number, expiration_date}
   BD6: {traveler_ssn, passport_number}
   BD7: {trip_id, start_location, end_location}
   BD8: {trip_id, traveler_ssn, agent}
4. Justify Each Decomposition Step:
   a. For each decomposition, write a brief justification, including why the specific dependency caused a violation of the BCNF and how the decomposition resolves the violation.

   In the first functional dependency, the attribute agent volates the BCNF because the left hand side is not a superkey as it doesn't include other keys, so based on the closure test, we can use the formula $R_2 = R-(X^+ - X)$ to break the BD into :

   - BD1: {agent, years_experience}

Name: Natalie Poche
Assignment: Homework 5
Due: April 10, 2025

- BD2: {agent, traveler_ssn, trip_id, start_location, end_location, passport_number, expiration_date}
- Removing the key isolates dependency and eliminates redundancy

In the second functional dependency traveler_ssn, also violate the BCNF beause it is not a superkey, so based on the closure test BD2 can be droken down into:

- B3 : {traveler_ssn, passport_number, expiration_date}
- B4 : {traveler_ssn, agent, trip_id, start_location, end_location}
- Removing it allows a table of traveler-specific information to remove redundancy.

In the third functional dependency, passport_number violate the BDNF in BD3 because the left is not a proper key, let alone super key, so in the clusre test BD3 is broken down into

- BD5: {passport_number, expiration_date}
- BD6: {traveler_ssn, passport_number}
- Removing it removes redundancy

In the fourth functional dependency, trip_id violates the BCNF since it's not a superkey, so the closure test breaks it down into:

- BD7: {trip_id, start_location, end_location}
- BD8: {trip_id, traveler_ssn, agent}
- Removing the key removes duplications