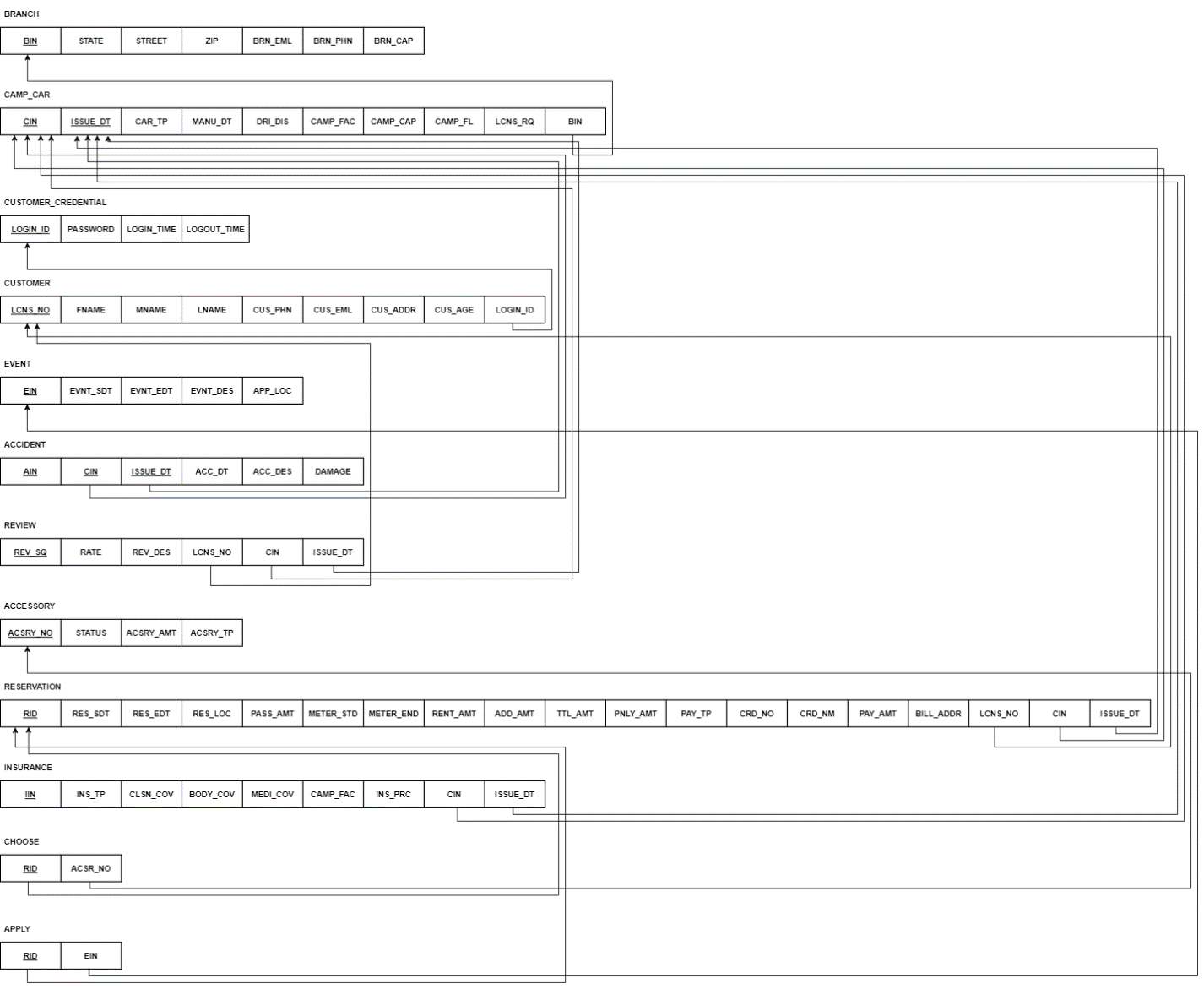
**Functional Dependencies and Anomalies**

In this document, we will dissect the functional dependencies and find anomalies from the result of our ER diagram in Figure 1 below.



Figure

1. **Functional dependencies**

From the attributes, we will identify the functional dependencies to verify the primary key in our ER diagram.

1. **Branch**

Graphical user interface, text, application, email

Description automatically generated

Branch ID -> branch state, branch street, branch zip code, branch email, branch phone, branch capacity

1. **Camping car**

Graphical user interface, text, application, email

Description automatically generated

Camping car ID -> issue date, car type, manufacture date, distance driven, facilities, passenger, availability, required license, branch ID

1. **Customer credentials**

Graphical user interface, text, application, email

Description automatically generated

Login ID -> password, login time, logout time

1. **Customer name**

Graphical user interface, text, application, email

Description automatically generated

License number -> First name, middle name, last name, phone number, email, address, age, login ID

1. **Event**

Graphical user interface, text, application, email

Description automatically generated

Event ID -> Event start date, event end date, event description, event applied location

1. **Accident**

Graphical user interface, text, application, email

Description automatically generated

Accident ID -> Camping car ID, issue date, accident date, accident description, damage

1. **Review**



Review ID -> rate of service, review description, camping car reviewed

1. **Accessory**

Graphical user interface, text, application, email

Description automatically generated

Accessory ID -> accessory status, accessory amount, accessory type

1. **Reservation**

Graphical user interface, text, application, email

Description automatically generated

Reservation ID -> reservation start date, reservation end date, reservation start location, passenger number, meter start, meter end, rental amount, additional amount, total amount, penalty amount, payment type, card number, card name, pay amount, billing address, license number, camping car ID, issue date.

1. **Insurance**

Graphical user interface, text, application, email

Description automatically generated

Insurance ID -> insurance type, collision coverage, body coverage, medical coverage, camping car facilities, insurance price, camping car ID, issue date.

1. **Choose**

Graphical user interface, text, application, email

Description automatically generated

Reservation ID -> accessory number

1. **Apply**

Graphical user interface, text, application, email

Description automatically generated

Reservation ID -> Event ID

1. **Anomalies**

From here, we can analyze and anomalies we might have in our structure. There are two types of anomalies in this structure.

1. Expected insertion anomalies
2. Camping car: distance driven.
3. Customer credential: login time and logout time.
4. Reservation: Meter end, penalty amount, pay amount.

These attributes should be updated after the service has been used. Data for these attributes may not be available, and we will keep that in mind when building our database.

1. Insertion anomalies that need to be resolved
2. In case of payment type being other than card, card number, card name, and billing address will not be available.
3. **Resolution**

To resolve the payment section in the reservation table, we will create a **Pay** table and **Pay Type** table. After separating the attributes related to the pay and pay type table, from the reservation table, reservation table will have Pay ID as foreign key, that is related to the Pay table, and then in the Pay table we have the Pay type as foreign key which is related to the Pay Type table.

Table

Description automatically generated

1. **Final form**

테이블이(가) 표시된 사진

자동 생성된 설명