NORMALISATION

Tables and Tuples:

Canc: This table stores information about canceled reservations.

Attributes: pnr (Primary Key), rfare

Class: Contains information about different classes offered on trains.

Attributes: cname (Primary Key)

Classseats: Stores seat availability information for different classes on trains.

Attributes: trainno (Primary Key), sp (Primary Key), dp (Primary Key), doj (Primary Key), class (Primary Key), fare, seatleft

Pd: This table holds passenger details.

Attributes: pnr (Primary Key), pname (Primary Key), page (Primary Key), pgender (Primary Key)

Resv: Stores reservation details.

Attributes: pnr (Primary Key), id, trainno, sp, dp, doj, tfare, class, nos, status

Schedule: Contains schedule information for trains.

Attributes: id (Primary Key), trainno, sname, arrival_time, departure_time, distance

Station: Holds information about stations.

Attributes: id, sname (Primary Key)

User: Stores user information.

Attributes: id (Primary Key), emailid, password, mobileno, dob

Train: Contains details about trains.

Attributes: trainno (Primary Key), tname, sp, st, dp, dt, dd, distance

Reduction with normalization

Step 1: Identify Functional Dependencies

Canc:pnr -> rfare

Class:cname -> (no other attributes provided, assuming it is a candidate key)

Classseats:(trainno, sp, dp, doj, class) -> (fare, seatleft)[1]

Pd:pnr -> (pname, page, pgender)

Resv:pnr -> (id, trainno, sp, dp, doj, tfare, class, nos, status)

Schedule:id -> (trainno, sname, arrival_time, departure_time, distance)

Station:sname -> id

[1] Prashast Arnav Akaash

User:id -> (emailid, password, mobileno, dob)

Train:trainno -> (tname, sp, st, dp, dt, dd, distance)

Step 2: Normalize the Relations

First Normal Form (1NF):

Canc (1NF):pnr, rfare

Class (1NF):cname

Classseats (1NF):trainno, sp, dp, doj, class, fare, seatleft

Pd (1NF):pnr, pname, page, pgender

Resv (1NF):pnr, id, trainno, sp, dp, doj, tfare, class, nos, status

Schedule (1NF):id, trainno, sname, arrival_time, departure_time, distance

Station (1NF):id, sname

User (1NF):id, emailid, password, mobileno, dob

Train (1NF):trainno, tname, sp, st, dp, dt, dd, distance

Second Normal Form (2NF):

Classseats (2NF):(trainno, sp, dp, doj, class) -> (fare, seatleft)

Schedule (2NF):id -> (trainno, sname, arrival_time, departure_time, distance)

Station (2NF):sname -> id

User (2NF):id -> (emailid, password, mobileno, dob)

Train (2NF):trainno -> (tname, sp, st, dp, dt, dd, distance)

Third Normal Form (3NF):

Classseats (3NF):(trainno, sp, dp, doj) -> class

class -> (fare, seatleft)

Resv (3NF):pnr -> (id, trainno, sp, dp, doj, tfare, class, nos, status)

Schedule (3NF):id -> (trainno, sname, arrival_time, departure_time)

trainno -> distance

User (3NF):id -> (emailid, password, mobileno, dob)

Train (3NF):trainno -> (tname, sp, st, dp, dt, dd, distance)

[2] Prashast Arnav Akaash

[3]

BCNF Normalized Relations:

Canc: There are no non-trivial functional dependencies, so it is already in BCNF.

Class: There are no non-trivial functional dependencies, so it is already in BCNF.

Classseats: The primary key consists of multiple attributes (trainno, sp, dp, doj, class), and the attribute 'fare' seems to depend only on 'class' (since fare likely varies based on the class). Thus, there's no non-trivial functional dependency violation here, and it's already in BCNF.

Pd: The primary key consists of multiple attributes (pnr, pname, page, pgender), and there are no non-trivial functional dependencies, so it is already in BCNF.

Resv: The primary key consists of a single attribute (pnr), and 'tfare' seems to depend only on 'class' (similar to Classseats), which violates BCNF. We need to decompose this table to ensure BCNF.

We'll create a new table, let's call it ResvDetails, with primary key (pnr, class), and move 'tfare' to this new table. This ensures that 'tfare' is dependent on the superkey (pnr, class).

Schedule: There are no non-trivial functional dependencies, so it is already in BCNF.

Station: There are no non-trivial functional dependencies, so it is already in BCNF.

User: There are no non-trivial functional dependencies, so it is already in BCNF.

Train: There are no non-trivial functional dependencies, so it is already in BCNF.

Resulting Normalized Relations:

ResvDetails: Attributes: pnr (Primary Key), class (Primary Key), tfare

Canc (1NF):pnr, rfare

Class (1NF):cname

Classseats (3NF):trainno, sp, dp, doj, class, fare, seatleft

Pd (1NF):pnr, pname, page, pgender

Resv (3NF):pnr, id, trainno, sp, dp, doj, tfare, class, nos, status

Schedule (3NF):id, trainno, sname, arrival_time, departure_time, distance

Station (2NF):id, sname

User (3NF):id, emailid, password, mobileno, dob

Train (3NF):trainno, tname, sp, st, dp, dt, dd, distance

[3] Prashast Arnav Akaash