**Tables and Tuples:**

Canc(pnr (PK), rfare);

Class(cname (PK));

Classseats(trainno (PK), sp (PK), dp (PK), doj (PK), class (PK), fare, seatleft);

Pd(pnr (PK), pname (PK), page (PK), pgender (PK));

Resv(pnr (PK), id, trainno, sp, dp, doj, tfare, class, nos, status);

Schedule(id (PK), trainno, sname, arrival\_time, departure\_time, distance);

Station(id, sname (PK));

User(id (PK), emailid, password, mobileno, dob);

Train(trainno (PK), 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)

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

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)

**Resulting Normalized Relations:**

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