

The various attributes and their constraints that can be identified are:

Airportid: Id assigned to each airport. Each airport has a unique and not null id.

Airportname: name of the airport.

Airportlocation: location of the airport.

Flightid: id assigned to each aircraft. Each id is unique, serial and not null.

Flightname: name of the airline. Each flight has only one airline.

Routeid: id of the route.

EconomySeats: Number of economy class seats in the flight

BusinessSeats: Number of business class seats in the flight

RouteID: ID of the route. It is unique, not null.

DepartureCity: Departure city of the flight

LandingCity: Landing city of the flight.

Username: Username of each user. Each user is unique and not null.

Password: Password of the user.

NORMALIZATION:

A – Airport ID

B – Airport Name

C – location

D-flightid

E-flight name

F - economyseats

G – business seats

H – routeid

I-departure

J-landing

K-username

L-password

Universal table:

air	Airport	locati	flighti	flightna	econo	Buisne	routeid	depa	landing	usern	pass
por	Name	on	d	me	myseat	SS		rture	C	ame	wor
tid					s	seats					d
1	cial	kochi	Fe220	indigo	100	10	23	koch	banglore	Vinay	430t
								i		ak	g
2	tvmairp	trivan	Pa560	airindia	110	20	27	triva	delhi	Alex	Eg4t
	ort	drum						ndru			
								m			
3	mumba	Mum	Oi780	Spicejet	95	30	45	mu	amritsar	akash	Srge
	i	bai						mbai			9
	airport							1			1

Before Normalisation:

Functional dependencies are:

ADHK → ABCDEFGHIJKL

Primary key: ADHK

➤ <u>1NF Form:</u>

In the 1NF Form, we split the rows that had the multivalued attributes, so that each column can have unique values.

air	airportN	Loc	flighti	flightna	econo	Buisne	Route	depart	landing	userna	passwo
por	ame	atio	d	me	myseat	ss seat	id	ure		me	rd
tid		n									
1	cial	koc	Fe220	indigo	100	10	23	kochi	banglore	vinayak	430tg
		hi									
2	tvmairpo	triva	Pa560	airindia	110	20	27	trivand	delhi	alex	Eg4T
	rt	ndru						rum			
		m									
3	Mumbai	Mu	Oi780	spicejet	95	30	45	mumb	amritsar	akash	Srge9
	airport	mba		- •				ai			-
	_	i									

> 2NF Form:

In the 2NF Form, the partial dependencies are removed, and the tables can be split as follows:

Airport (AirportID, AirportName, Airportlocation)
Flight(FlightID, FlightName, RouteID, EconomySeats, BusinessSeats)
Route(RouteID, DepartureCity, LandingCity)
Passenger(Username, password)

Functional Dependencies are:

A->BC D->EFG H->IJ K->L

➤ <u>3NF Form:</u>

In the 3NF Form, there should be no transitive dependency in table

Airport (AirportID, AirportName, Airportlocation)
Flight(FlightID, FlightName, RouteID, EconomySeats, BusinessSeats)
Route(RouteID, DepartureCity, LandingCity)
Passenger(Username, password)

In the 3NF form, we remove the transitive dependencies. The transitive dependency.

• Functional Dependencies are:

Functional Dependencies are:

A->BC

D->EFG

H->IJ

K->L

QUERIES

DDL STATEMENTS

CREATE TABLE Airport (

AirportID VARCHAR PRIMARY KEY,

AirportName VARCHAR(255),

Airportlocation VARCHAR(255))

CREATE TABLE Flight(

FlightID VARCHAR PRIMARY KEY,

FlightName VARCHAR(255),

RouteID VARCHAR,

EconomySeats INT,

BusinessSeats INT,

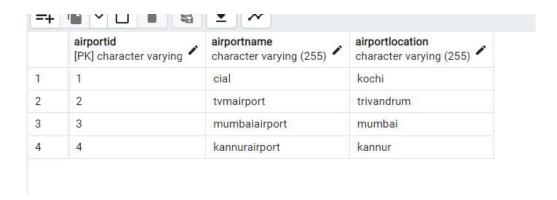
FOREIGN KEY (RouteID) REFERENCES Route

ON DELETE CASCADE ON UPDATE CASCADE);

CREATE TABLE Route(

RouteID VARCHAR PRIMARY KEY,

```
DepartureCity VARCHAR(255),
 LandingCity VARCHAR(255));
CREATE TABLE Login(Username VARCHAR PRIMARY KEY,
  Password VARCHAR(255))
INSERT Values
INSERT into Airport values(1, 'cial', 'kochi'),
     (2,'tvmairport','trivandrum'),
     (3,'mumbaiairport','mumbai'),
     (4,'kannurairport','kannur');
     INSERT into Flight values('Fe220','indigo',23,100,10),
     ('Pa560', 'airindia', 27, 110, 20),
     ('Oi780', 'Spicejet', 45, 95, 30),
     ('Ba580','indigo',24,100,9);
     INSERT into Route values(23,'kochi','banglore'),
     (27,'trivandrum','delhi'),
     (45, 'mumbai', 'amritsar'),
     (24,'kannur','mumbai');
     INSERT into login values('vinayak','430tg'),
     ('alex', 'Eg4t'),
     ('akash', 'Srge9')
select * from airport
```



select * from flight

	flightid [PK] character varying	flightname character varying (255)	routeid character varying	economyseats integer	businessseats integer
1	Fe220	indigo	23	100	10
2	Pa560	airindia	27	110	20
3	Oi780	Spicejet	45	95	30
1	Ba580	indigo	24	100	g

select * from route

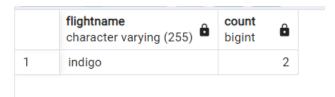
	routeid [PK] character varying	departurecity character varying (255)	landingcity character varying (255)
1	23	kochi	banglore
2	27	trivandrum	delhi
3	45	mumbai	amritsar
1	24	kannur	mumbai

select * from login

	username [PK] character varying (255)	password character varying (255)
1	vinayak	430tg
2	alex	Eg4t
3	akash	Srge9

1. Aggregate function, group by, having

select Flightname, count(FlightID) from Flight group by Flightname having count(FlightID)>1;



2. Order by

select * from Flight order by Economyseats;

	flightid [PK] character varying	flightname character varying (255)	routeid character varying	economyseats integer	businessseats integer
1	Oi780	Spicejet	45	95	30
2	Fe220	indigo	23	100	10
3	Ba580	indigo	24	100	9
4	Pa560	airindia	27	110	20

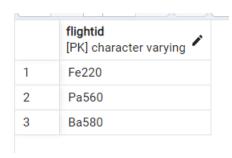
3. Join, outer join

SELECT * FROM Route RIGHT JOIN Flight ON Route.RouteID = Flight.RouteID;

	routeid character varying	departurecity character varying (255)	landingcity character varying (255)	flightid character varying	flightname character varying (255)	routeid character varying	economyseats integer	businessseats integer
1	23	kochi	banglore	Fe220	indigo	23	100	10
2	27	trivandrum	delhi	Pa560	airindia	27	110	20
3	45	mumbai	amritsar	0i780	Spicejet	45	95	30
4	24	kannur	mumbai	Ba580	indigo	24	100	9

4. Query having Boolean operators

select FlightID from Flight where Economyseats>99;



5. Query having arithmetic operators

select FlightID, Economyseats-50 as Updatedseats from Flight where Economyseats>50;

	flightid [PK] character varying	updatedseats integer
1	Fe220	50
2	Pa560	60
3	Oi780	45
4	Ba580	50

6. Search query using string operators

select * from Flight where Flightname like 'ind%'

	flightid [PK] character varying	flightname character varying (255)	routeid character varying	economyseats integer	businessseats integer
1	Fe220	indigo	23	100	10
2	Ba580	indigo	24	100	9

7. Usage of to_char, extract

SELECT DepatureTime, TO_CHAR(DepatureTime, 'HH') as Time FROM Time;

	depaturetime time without time zone	time text
1	12:00:00	12
2	11:00:00	11

select * from Time where extract(Hour from DepatureTime)=12;

	depaturetime time without time zone	landingtime time without time zone
1	12:00:00	03:00:00

8. Between, in, not between, not in

SELECT FlightID, Flightname FROM Flight WHERE Businessseats between 10 AND 20;

	flightid [PK] character varying	flightname character varying (255)
1	Fe220	indigo
2	Pa560	airindia

SELECT FlightID, Flightname FROM Flight WHERE Businessseats Not between 10 AND 20;

	flightid [PK] character varying	flightname character varying (255)
1	Oi780	Spicejet
2	Ba580	indigo

Select password from login where username in ('alex','akash'):



9. Set operations

select Airportname from Airport union select Flightname from Flight;

	airportname character varying (255)
1	mumbaiairport
2	tvmairport
3	airindia
4	cial
5	kannurairport
6	indigo
7	Spicejet

10. Subquery using exists/not exists, any, all

SELECT Flightname FROM Flight f WHERE EXISTS (

SELECT 1 FROM Route r WHERE r.RouteID = f.RouteID);

	flightname character varying (255)
1	indigo
2	airindia
3	Spicejet
4	indigo
