DDL

```
-- Drop tables

DROP TABLE ticket_info;
DROP TABLE flight_details;
DROP TABLE credit_card_details;
DROP TABLE flight;
DROP TABLE passenger;

-- Create tables

CREATE TABLE flight (
flight_id NUMBER PRIMARY KEY,
airline_id NUMBER NOT NULL,
airline_name VARCHAR(50) NOT NULL,
from_location VARCHAR(50) NOT NULL,
to_location VARCHAR(50) NOT NULL,
departure_time VARCHAR(50) NOT NULL,
arrival_time VARCHAR(50) NOT NULL,
total_seats NUMBER CHECK (total_seats >= 100 AND total_seats <= 1000)
);

CREATE TABLE passenger (
profile_id NUMBER PRIMARY KEY,
password VARCHAR(50) NOT NULL,
f_name VARCHAR(50) NOT NULL,
l_name VARCHAR(50) NOT NULL,
m_name VARCHAR(20),
address VARCHAR(50),
mobile_no VARCHAR(50) NOT NULL UNIQUE,
email VARCHAR(50) NOT NULL UNIQUE
);

CREATE TABLE credit_card_details (
profile_id NUMBER NOT NULL,
card_number VARCHAR(50) PRIMARY KEY,
card_type VARCHAR(50) NOT NULL,
expiration_month VARCHAR(50) NOT NULL,
expiration_year NUMBER NOT NULL,
FOREIGN KEY (profile_id) REFERENCES passenger(profile_id)
);

CREATE TABLE flight_details (
flight_id NUMBER NOT NULL,
flight_departure_date DATE,
price NUMBER NOT NULL CHECK (price >= 1000 AND price <= 100000),
PRIMARY KEY (flight_id, flight_departure_date),
FOREIGN KEY (flight_id) REFERENCES flight(flight_id)
);

CREATE TABLE ticket_info (
ticket_id NUMBER PRIMARY KEY,
profile_id NUMBER NOT NULL,
flight_id NUMBER NOT NULL,
flight_d_date DATE NOT NULL,
status VARCHAR(50),
FOREIGN KEY (flight_id) REFERENCES flight(flight_id),
FOREIGN KEY (profile_id) REFERENCES passenger(profile_id),
FOREIGN KEY (flight_id, flight_d_date) REFERENCES flight_details(flight_id, flight_departure_date)
);

-- Add column in the table

ALTER TABLE flight ADD duration VARCHAR(50);

-- Modify column definition in the table

ALTER TABLE flight MODIFY duration VARCHAR(20);

-- Rename the column name

ALTER TABLE flight RENAME COLUMN total_seats TO seats;

-- Drop the column from table

ALTER TABLE passenger DROP COLUMN m_name;
ALTER TABLE flight DROP COLUMN duration;
```

DML

```
--insert values into tables

insert into flight values(1,11,'Novoair','Dhaka','Jessore','10:00 am','11:30 am',200);
insert into flight values(2,11,'Novoair','Dhaka','Saiadpur','10:00 pm','11:45 pm',200);
insert into flight values(3,11,'Novoair','Dhaka','Chottogram','1:00 pm','2:30 pm',250);
insert into flight values(4,12,'US-Bangla','Dhaka','Jessore','11:00 am','12:30 pm',300);
insert into flight values(5,12,'US-Bangla','Dhaka','Saiadpur','3:00 pm','4:30 pm',300);
insert into flight values(6,12,'US-Bangla','Dhaka','Chottogram','8:00 pm','9:30 pm',400);
insert into flight values(7,13,'Biman Bangladesh','Dhaka','Jessore','10:00 am','11:30 am',250);
insert into flight values(8,13,'Biman Bangladesh','Dhaka','Saiadpur','10:00 am','11:30 am',250);
insert into flight values(9,13,'Biman Bangladesh','Dhaka','Chottogram','4:00 pm','5:38 pm',250);
insert into flight values(10,13,'Biman Bangladesh','Dhaka','Sylhet','9:00 pm','11:00 pm',300);

insert into passenger values(111,'ld','lamisa','deya','Jamalpur','01707494817','deya@gmail.com');
insert into passenger values(112,'ac','anika','chhoa','Jamalpur','01707494818','chhoa@gmail.com');
insert into passenger values(113,'ak','arpita','kundu','Khulna','01707494819','kundu@gmail.com');
insert into passenger values(114,'ms','mizan','swapon','Jamalpur','01707494820','swapon@gmail.com');
insert into passenger values(115,'ss','sadia','sornaly','Mymensingh','01707494821','sornaly@gmail.com');
insert into passenger values(116,'rr','rubaiya','raha','Rajshahi','01707494822','raha@gmail.com');
insert into passenger values(117,'jb','jytosna','begum','jamalpur','01707494823','begum@gmail.com');
insert into passenger
values(118,'tz','taslima','zannat','Mymensingh','01707494824','zannat@gmail.com');

insert into passenger values(119,'tt','tafannum','tareq','chottogram','01707494825','tareq@gmai;.com');

insert into passenger values(120,'mm','mahila','mohiuddin','Coxs
bazar','01707494826','mahila@gmail.com');
```

```
insert into credit_card_details values(111,'3141 5926 5643 7890','visa','jnuary',2024);

insert into credit_card_details values(112,'4141 5926 5643 7890','business','December',2025);

insert into credit_card_details values(113,'5141 5926 5643 7890','mastercard','june',2026);

insert into credit_card_details values(114,'6141 5926 5643 7890','visa','december',2024);

insert into credit_card_details values(115,'7141 5926 5643 7890','business','january',2025);

insert into credit_card_details values(116,'9141 5926 5643 7890','mastercard','june',2026);

insert into credit_card_details values(117,'3141 8926 5643 7890','visa','december',2024);

insert into credit_card_details values(118,'3141 6926 5643 7890','business','january',2025);

insert into credit_card_details values(119,'3141 1926 5643 7890','mastercard','june',2026);

insert into credit_card_details values(120,'3141 9926 5643 7890','visa','december',2024);
```

```
insert into flight_details values(1,date '2023-04-12',4000);

insert into flight_details values(1,date '2023-04-13',4400);

insert into flight_details values(1,date '2023-04-14',4500);

insert into flight_details values(2,date '2023-04-12',5000);

insert into flight_details values(2,date '2023-04-13',5400);
insert into flight_details values(2,date '2023-04-14',5500);
insert into flight_details values(3,date '2023-04-12',6000);
insert into flight_details values(3,date '2023-04-13',6400);
insert into flight_details values(3,date '2023-04-14',6500);
insert into flight_details values(4,date '2023-04-12',4000);
insert into flight_details values(4,date '2023-04-13',4400);
insert into flight_details values(4,date '2023-04-14',4500);
insert into flight_details values(5,date '2023-04-12',5000);
insert into flight_details values(5,date '2023-04-13',5400);
insert into flight_details values(5,date '2023-04-14',5500);
insert into flight_details values(6,date '2023-04-12',6000);
insert into flight_details values(6,date '2023-04-13',6400);
insert into flight_details values(6,date '2023-04-14',6500);
insert into flight_details values(7,date '2023-04-12',4000);
insert into flight_details values(7,date '2023-04-13',4400);
insert into flight_details values(7,date '2023-04-14',4500);
insert into flight_details values(8,date '2023-04-12',5000);
insert into flight_details values(8,date '2023-04-13',5400);
insert into flight_details values(8,date '2023-04-14',5500);
insert into flight_details values(9,date '2023-04-12',6000);
insert into flight_details values(9,date '2023-04-13',6400);
insert into flight_details values(9,date '2023-04-14',6500);
insert into flight_details values(10,date '2023-04-12',4000);
insert into flight_details values(10,date '2023-04-13',4400);
insert into flight_details values(10,date '2023-04-14',4500);



insert into ticket_info values(001,111,1,date '2023-04-12','paid');
insert into ticket_info values(002,111,2,date '2023-04-12','unpaid');
insert into ticket_info values(003,111,3,date '2023-04-12','paid');
insert into ticket_info values(004,112,3,date '2023-04-12','paid');
insert into ticket_info values(005,112,4,date '2023-04-13','paid');
insert into ticket_info values(006,112,7,date '2023-04-13','paid');
insert into ticket_info values(007,113,5,date '2023-04-14','unpaid');
insert into ticket_info values(008,113,6,date '2023-04-13','paid');
insert into ticket_info values(009,114,8,date '2023-04-12','paid');
insert into ticket_info values(010,115,3,date '2023-04-12','unpaid');
insert into ticket_info values(011,116,5,date '2023-04-14','unpaid');
insert into ticket_info values(012,117,4,date '2023-04-13','paid');
insert into ticket_info values(013,118,5,date '2023-04-14','unpaid');
insert into ticket_info values(014,118,6,date '2023-04-13','paid');
insert into ticket_info values(015,119,3,date '2023-04-12','paid');
insert into ticket_info values(016,119,7,date '2023-04-13','paid');
insert into ticket_info values(017,119,2,date '2023-04-12','unpaid');
insert into ticket_info values(018,120,8,date '2023-04-12','paid');
insert into ticket_info values(019,120,8,date '2023-04-12',null);
```

```
--show all the entries
```

```
select * from flight;
select * from passenger;
select * from credit_card_details;
select * from flight_details;
select * from ticket_info;


--select multiple row using where condition


select * from flight where flight_id=2;
select * from passenger where profile_id=2;
select * from credit_card_details where card_number='3141 5926 5643 7890';
select * from flight_details where flight_id=2 and flight_departure_date=date '2023-04-13';
select * from ticket_info where ticket_id=017;
select * from flight where seats>100;


--select one row using where condition


select * from flight_details where flight_id=2;


--basic select commands


select distinct f_name from passenger;
select all f_name from passenger;
select flight_id, seats/2 as halfSeats,seats from flight;
```

```
---cartesian product

select * from flight a,flight_details b where a.flight_id=b.flight_id;

--range search

select * from flight_details where price>3000 and price<5000;

--delete all tuples

--delete from passenger;

--delete a single row

delete flight_details where flight_id=1 and flight_departure_date=date'2023-04-12';
```

```
--string operations


--(first name starts with l or t)
select f_name from passenger where f_name like 'l%' or f_name like 't%';
--(airline name has the character a)
select * from flight where airline_name like '%a%';




--ordering the display of tuple


select * from credit_card_details order by expiration_year;
select * from passenger order by f_name;
select * from passenger order by f_name desc;



--set operations


select * from credit_card_details where expiration_year=2024
union
select * from credit_card_details where expiration_year=2023;

select * from credit_card_details where expiration_year=2024
intersect
select * from credit_card_details where expiration_month='jnuary';
select * from credit_card_details where expiration_year=2024
minus
select * from credit_card_details where expiration_month='jnuary';



-null values


select * from ticket_info where status is null;


--aggregate functions


select avg(price) as average_price from flight_details;
select max(price) as maximum_price from flight_details;
select min(price) as minimum_price from flight_details;
select sum(seats) as all_seats from flight;
select count(*) from ticket_info;


--agregate Functions – haing cause andgroup by


select avg(price),flight_id from flight_details group by flight_id;
select avg(price),flight_id from flight_details group by flight_id having avg(price)>5000;
```

```sql
--nested subqueries

select * from ticket_info where (flight_d_date,flight_id )in
(select flight_departure_date,flight_id from flight_details where flight_id in
(select flight_id from flight where airline_id=11));

--set membership

select f_name from passenger where f_name not in ('lamisa','anika');
select *  from passenger where f_name in ('lamisa','anika');

--update

update flight_details set price=1.03*price where price<=5000;
update flight set seats=10+seats where seats=200;

--join
```

```sql
--(natural join)
select * from passenger natural join credit_card_details;
select * from flight natural join flight_details where flight_id in(1,2,3);

--(inner join)
select * from flight join flight_details using(flight_id);
select * from passenger join credit_card_details using(profile_id);

--(outer join)
select * from passenger left outer  join credit_card_details using(profile_id);
select * from passenger right  outer  join credit_card_details using(profile_id);
select * from passenger full  outer  join credit_card_details using(profile_id);
select * from flight left outer  join flight_details using(flight_id);
select * from flight right  outer  join flight_details using(flight_id);
select * from flight full  outer  join flight_details using(flight_id);



--view



create view personal_info as select f_name,l_name,address,mobile_no,email from passenger;
create view visa_card_details as select * from credit_card_details where card_type='visa';
select * from personal_info;
select * from visa_card_details;

--(view from another view)
create view passenger_details as select f_name,l_name,mobile_no from personal_info;
```

```sql
--(view drop)
drop view personal_info;
drop view passenger_details;
drop view visa_card_details;


--with clause


with max_price(p) as (select max(price) from flight_details)
select * from flight_details a,max_price b where a.price=b.p;


with avg_price(p) as (select avg(price) from flight_details)
select * from flight_details a,avg_price b where a.price>b.p;


--"some" and "all" clause


select * from flight where seats> some(select seats from flight where seats<250);
select * from flight where seats> all(select seats from flight where seats<250);
```

```sql
--"exists" and "not exists" clause

select * from passenger where profile_id>=115 and exists (select * from credit_card_details where card_type='visa');

select * from passenger where profile_id>=115 and not  exists (select * from credit_card_details where card_type='Master');

-- show user name

show user;

-- show the list of tables

select table_name from user_tables;

-- show all tables and views from the tab

select * from tab;

-- descriptions of table

desc passenger;

desc credit_card_details;

desc flight;

desc flight_details;

desc ticket_info;
```

PL/SQL

```
-- PL/SQL variable declaration and print value
SET serveroutput ON

DECLARE
  id flight_details.flight_id%type;
  dates flight_details.flight_departure_date%type;
  price flight_details.price%type;
BEGIN
  SELECT flight_id, flight_departure_date, price INTO id, dates, price
  FROM flight_details
  WHERE flight_id = 1 AND flight_departure_date = DATE '2023-04-12';

  dbms_output.put_line('id = ' || id || ', date = ' || dates || ', price = ' || price);
END;
/

-- Insert and set default value
SET serveroutput ON

DECLARE
  id flight_details.flight_id%type := 1;
  dates flight_details.flight_departure_date%type := DATE '2023-04-15';
  price flight_details.price%type := 6000;
BEGIN
  INSERT INTO flight_details VALUES (id, dates, price);
```

```
END;
/

-- %ROWTYPE
SET serveroutput ON

DECLARE
  flight flight_details%rowtype;
BEGIN
  SELECT flight_id, flight_departure_date, price INTO flight.flight_id, flight.flight_departure_date, flight.price
  FROM flight_details
  WHERE flight_id = 1 AND flight_departure_date = DATE '2023-04-12';

  dbms_output.put_line('id = ' || flight.flight_id || ', date = ' || flight.flight_departure_date || ', price = ' || flight.price);
END;
/

-- Insert and set default value using %ROWTYPE
SET serveroutput ON

DECLARE
  flight flight_details%rowtype;
BEGIN
  flight.flight_id := 2;
  flight.flight_departure_date := DATE '2023-04-15';
  flight.price := 5800;
  INSERT INTO flight_details VALUES flight;
```

```
END;
/

-- Creating a cursor
SET serveroutput ON

DECLARE
  CURSOR pass_cur IS
    SELECT profile_id, f_name, l_name FROM passenger;
  p_id passenger.profile_id%type;
  f_name passenger.f_name%type;
  l_name passenger.l_name%type;
BEGIN
  OPEN pass_cur;
  FETCH pass_cur INTO p_id, f_name, l_name;

  WHILE pass_cur%FOUND LOOP
    dbms_output.put_line(p_id || ' ' || f_name || ' ' || l_name);
    FETCH pass_cur INTO p_id, f_name, l_name;
  END LOOP;

  CLOSE pass_cur;
END;
/

-- %ISOPEN, %FOUND, %ROWCOUNT, %NOTFOUND
SET serveroutput ON

DECLARE
```

```
  CURSOR pass_cur IS
    SELECT profile_id, f_name, l_name FROM passenger;
  p_id passenger.profile_id%type;
  f_name passenger.f_name%type;
  l_name passenger.l_name%type;
BEGIN
  OPEN pass_cur;
  FETCH pass_cur INTO p_id, f_name, l_name;

  WHILE pass_cur%FOUND LOOP
    dbms_output.put_line(p_id || ' ' || f_name || ' ' || l_name);
    FETCH pass_cur INTO p_id, f_name, l_name;
  END LOOP;

  IF (pass_cur%NOTFOUND) THEN
    dbms_output.put_line('end');
  ELSE
    dbms_output.put_line('wrong');
  END IF;

  dbms_output.put_line('row ' || pass_cur%ROWCOUNT);
  CLOSE pass_cur;

  IF (pass_cur%ISOPEN) THEN
    dbms_output.put_line('Cursor is open');
  ELSE
    dbms_output.put_line('Cursor is closed');
  END IF;
END;
```

```
/

-- Procedure with parameter type IN
CREATE OR REPLACE PROCEDURE proc(id IN NUMBER) IS
  name passenger.f_name%TYPE;
BEGIN
  SELECT f_name INTO name FROM passenger WHERE passenger.profile_id = id;
  dbms_output.put_line(name);
END;
/

SET serveroutput ON

DECLARE
  var1 passenger.profile_id%TYPE := 111;
BEGIN
  proc(var1);
END;
/

-- Procedure with parameter type OUT
CREATE OR REPLACE PROCEDURE proc(name OUT passenger.f_name%TYPE) IS
BEGIN
  SELECT f_name INTO name FROM passenger WHERE passenger.profile_id = 111;
END;
/

SET serveroutput ON
```

```
DECLARE
  name passenger.f_name%TYPE;
BEGIN
  proc(name);
  dbms_output.put_line(name);
END;
/

-- Procedure with parameter type IN and OUT
CREATE OR REPLACE PROCEDURE proc(id IN NUMBER, name OUT passenger.f_name%TYPE) IS
BEGIN
  SELECT f_name INTO name FROM passenger WHERE passenger.profile_id = id;
END;
/

SET serveroutput ON

DECLARE
  name passenger.f_name%TYPE;
  id passenger.profile_id%TYPE;
BEGIN
  id := 111;
  proc(id, name);
  dbms_output.put_line(name);
END;
/

-- Procedure with no parameters
CREATE OR REPLACE PROCEDURE proc IS
```

```
  name passenger.f_name%TYPE;
BEGIN
  SELECT f_name INTO name FROM passenger WHERE passenger.profile_id = 111;
  dbms_output.put_line(name);
END;
/

SET serveroutput ON

DECLARE
BEGIN
  proc();
END;
/

-- Procedure with parameter type IN OUT
CREATE OR REPLACE PROCEDURE proc(name IN passenger.f_name%TYPE, id IN OUT NUMBER) IS
  aa NUMBER;
  b NUMBER;
BEGIN
  SELECT id INTO aa FROM passenger WHERE passenger.f_name = name;
  dbms_output.put_line(aa);
  b := aa + 10;
  dbms_output.put_line(b);
  UPDATE passenger SET profile_id = b WHERE profile_id = aa;
END;
/

SET serveroutput ON
```

```
DECLARE
  name passenger.f_name%TYPE := 'lamisa';
  id passenger.profile_id%TYPE;
BEGIN
  proc(name, id);
END;
/

-- Error showing
SHOW ERRORS PROCEDURE proc;

-- Dropping a procedure
DROP PROCEDURE proc;

-- Function
CREATE OR REPLACE FUNCTION fun(idd IN NUMBER)
RETURN passenger.f_name%TYPE IS
  f_name passenger.f_name%TYPE;
BEGIN
  SELECT f_name INTO f_name FROM passenger WHERE passenger.profile_id = idd;
  RETURN f_name;
END;
/

SET serveroutput ON

DECLARE
  id passenger.profile_id%TYPE := 111;
```

```
BEGIN
  dbms_output.put_line(fun(id));
END;
/

-- Error showing
SHOW ERRORS FUNCTION fun;

-- Dropping a function
DROP FUNCTION fun;

-- If/Else/Loop
SET serveroutput ON

DECLARE
  CURSOR c IS
    SELECT * FROM flight_details;
  avrg NUMBER;
  rec flight_details%ROWTYPE;
BEGIN
  OPEN c;
  FETCH c INTO rec;
  SELECT AVG(price) INTO avrg FROM flight_details;

  WHILE c%FOUND LOOP
    IF rec.price < avrg THEN
      dbms_output.put_line('Price is low');
    ELSE
      dbms_output.put_line('Price is high');
    END IF;

    FETCH c INTO rec;
  END LOOP;

  CLOSE c;
END;
/
```