CREATE DATABASE Air\_Cargo\_Analysis;

USE Air\_Cargo\_Analysis;

SET sql\_mode=(SELECT REPLACE(@@sql\_mode,'ONLY\_FULL\_GROUP\_BY',''));

CREATE TABLE route\_details (route\_id int NOT NULL,

flight\_num int NOT NULL,

origin\_airport varchar(20),

destination\_airport varchar(20),

aircraft\_id varchar(10),

distance int NOT NULL,

UNIQUE(route\_id), CHECK (distance>0));

SELECT\*FROM passengers\_on\_flights

WHERE route\_id BETWEEN 1 AND 25;

SELECT COUNT(class\_id = 'Business') AS BUSINESS\_CLASS\_COUNT, SUM(no\_of\_tickets\*price\_per\_ticket) AS REVENUE\_TOTAL FROM ticket\_details

WHERE class\_id = 'Bussiness';

SELECT CONCAT(first\_name, " ", last\_name) AS FULL\_NAME FROM customer;

SELECT customer\_id, CONCAT(first\_name, " " , last\_name) AS NAME, COUNT(no\_of\_tickets) AS NO\_OF\_TICKETS

FROM customer

JOIN ticket\_details USING (customer\_id)

GROUP BY customer\_id, NAME

ORDER BY NO\_OF\_TICKETS;

SELECT customer\_id, first\_name, last\_name FROM customer

JOIN ticket\_details USING(customer\_id)

WHERE brand = 'Emirates';

SELECT customer.customer\_id, customer.first\_name, customer.last\_name, passengers\_on\_flights.class\_id

FROM customer

JOIN passengers\_on\_flights ON customer.customer\_id = passengers\_on\_flights.customer\_id

WHERE passengers\_on\_flights.class\_id = 'Economy Plus';

SELECT customer.customer\_id, customer.first\_name, customer.last\_name, passengers\_on\_flights.class\_id

FROM customer

JOIN passengers\_on\_flights ON customer.customer\_id = passengers\_on\_flights.customer\_id

GROUP BY customer\_id

HAVING class\_id = 'Economy Plus'

ORDER BY customer\_id;

SELECT IF(SUM(no\_of\_tickets \* Price\_per\_ticket)>10000,'Revenue Crossed 10000','Revenue Less Than 10000')

AS REVENUE\_STATUS FROM ticket\_details;

CREATE USER 'new\_user'@'localhost' IDENTIFIED BY 'new\_password';

GRANT ALL ON Air\_Cargo\_Analysis.\* TO 'new\_user'@'localhost';

SELECT customer\_id,class\_id,brand , MAX(Price\_per\_ticket) OVER (PARTITION BY class\_id) AS max\_price FROM ticket\_details;

SELECT \* FROM passengers\_on\_flights WHERE route\_id = 4 ;

SELECT \* FROM passengers\_on\_flights HAVING route\_id = 4 ;

SELECT customer\_id, aircraft\_id , class\_id , sum(no\_of\_tickets \* Price\_per\_ticket)

AS total\_price

FROM ticket\_details

GROUP BY aircraft\_id WITH ROLLUP ;

DROP VIEW IF EXISTS business\_class ;

CREATE VIEW business\_class AS

SELECT customer\_id, class\_id, brand

FROM ticket\_details

WHERE class\_id = 'bussiness' ;

SELECT \* FROM business\_class ;

DROP PROCEDURE IF EXISTS passenger\_details;

DELIMITER &&

CREATE PROCEDURE passenger\_details(route int)

BEGIN

SELECT \* FROM route\_details

WHERE route\_id=route;

END &&;

CALL passenger\_details(2);

DROP PROCEDURE IF EXISTS travelled\_distance()

DELIMITER //

CREATE PROCEDURE travelled\_distance()

BEGIN

SELECT \* FROM routes WHERE distance\_miles>2000;

END //

CALL travelled\_distance();

DROP PROCEDURE IF EXISTS group\_of\_distance

DELIMITER //

CREATE PROCEDURE group\_of\_distance (IN distance int,OUT CATEGORY VARCHAR(40))

BEGIN

DECLARE DIST INT DEFAULT 1;

SELECT distance\_miles INTO DIST FROM routes WHERE distance\_miles=distance;

IF DIST BETWEEN 0 AND 2000 THEN

SET CATEGORY='SHORT DISTANCE TRAVELLED';

ELSEIF DIST BETWEEN 2000 AND 6500 THEN

SET CATEGORY='INTERMEDIATE DISTANCE TRAVELLED';

ELSE SET CATEGORY='LONG DISTANCE TRAVELLED';

END IF ;

END //

CALL group\_of\_distance(1558, @CATEGORY);

SELECT @CATEGORY

DROP FUNCTION IF EXISTS Complementary\_Services

DELIMITER //

CREATE FUNCTION Complementary\_Services(class\_id VARCHAR(40))

RETURNS VARCHAR(10) DETERMINISTIC

BEGIN

DECLARE SERVICE VARCHAR(20);

IF class\_id = 'Economy Plus' OR 'Business' THEN

SET SERVICE = 'YES';

ELSE SET SERVICE = 'NO';

END IF;

RETURN SERVICE;

END //

SELECT p\_date,customer\_id,class\_id, Complementary\_Services(class\_id) AS SERVICE FROM ticket\_details;

DROP PROCEDURE IF EXISTS First\_Scott

DELIMITER //

CREATE PROCEDURE First\_Scott()

BEGIN

DECLARE a VARCHAR(50);

DECLARE b VARCHAR(50);

DECLARE cursor\_1 CURSOR FOR SELECT first\_name, last\_name FROM customer

WHERE last\_name = 'Scott';

OPEN cursor\_1;

REPEAT FETCH cursor\_1 INTO a,b;

UNTIL b=0

END REPEAT;

SELECT a AS first\_name , b as last\_name;

CLOSE cursor\_1;

END //

CALL First\_Scott()