```sql
CREATE DATABASE AIR_CARGO ;
USE AIR_CARGO ;


-- QUERY 2 : CREATING A ROUTES_DETAILS TABLE WITH
THE SPECIFIED CONSTRAINT -

CREATE TABLE ROUTES_DETAILS
(ROUTE_ID INT NOT NULL, UNIQUE
(ROUTE_ID),
 FLIGHT_NUM INT NOT NULL, CHECK (FLIGHT_NUM > 0),
 ORIGIN_AIRPORT TEXT,
DESTINATION_AIRPORT TEXT, AIRCRAFT_ID VARCHAR(30),
 DISTANCE_MILES INT , CHECK(DISTANCE_MILES
> 0)) ;

 -- INSERTING VALUES INTO THE TABLE -

INSERT INTO ROUTES_DETAILS
VALUES ('1',
'1111', 'EWR', 'HNL', '767-301ER', '4962'),('2', '1112', 'HNL', 'EWR', '767-301ER',
'4962'),('3', '1113', 'EWR', 'LHR', 'A321', '3466'),('4', '1114', 'JFK', 'LAX', '767-301ER',
'2475'),
('5', '1115', 'LAX', 'JFK', '767-301ER', '2475'),('6', '1116', 'HNL', 'LAX',
'767-301ER', '2556'),('7', '1117', 'LAX', 'ORD', 'A321', '1745'),('8', '1118', 'ORD', 'EWR',
'A321', '719'),
('9', '1119', 'DEN', 'LAX', 'ERJ142', '862'),('10', '1120', 'HNL', 'DEN',
'A321', '3365'),('12', '1122', 'ABI', 'ADK', '767-301ER', '4300'),('13', '1123', 'ADK', 'BQN',
'A321', '2232'),
('14', '1124', 'BQN', 'CAK', 'A321', '2445'),('15', '1125', 'CAK', 'ANI',
'767-301ER', '2000'),('16', '1126', 'ALB', 'APN', 'A321', '1700'),('17', '1127', 'APN', 'BLV',
'767-301ER', '1900'),
('18', '1128', 'ANI', 'BGR', 'ERJ142', '2450'),('19', '1129', 'ATW',
'AVL', 'A321', '2222'),('20', '1130', 'AVL', 'BOI', '767-301ER', '3134'),('21', '1131', 'BFL',
'BET', 'A321', '2425'),
('22', '1132', 'BGR', 'BJI', 'ERJ142', '1242'),('23', '1133', 'BLV',
'BFL', '767-301ER', '2354'),('24', '1134', 'BJI', 'BQN', 'A321', '1575'),('25', '1135', 'RDM',
'BJI', 'A321', '2425'),
('26', '1136', 'BET', 'BTM', 'ERJ142', '1311'),('27', '1137', 'BOI',
'CLD', 'A321', '578'),('28', '1138', 'BOS', 'CDC', '767-301ER', '246'),('29', '1139', 'BKG',
'CRW', '767-301ER', '909'),
('30', '1140', 'BUR', 'STT', 'CRJ900', '780'),('31', '1141', 'BTM',
'CHA', 'ERJ142', '660'),('32', '1142', 'CLD', 'CHI', '767-301ER', '246'),('33', '1143', 'CDC',
'CST', 'CRJ900', '1345'),
('34', '1144', 'CRW', 'COD', 'A321', '2452'),('35', '1145', 'STT',
'CDB', 'ERJ142', '2121'),('36', '1146', 'CHA', 'COU', 'CRJ900', '1212'),('37', '1147', 'CHI',
'CST', '767-301ER', '999'),
('38', '1148', 'CST', 'DAL', 'A321', '1111'),('39', '1149', 'COD',
'SCC', 'CRJ900', '1579'),('40', '1150', 'CDB', 'DEC', 'A321', '909'),('41', '1151', 'CAE',
'DRT', 'ERJ142', '898'),
('42', '1152', 'CSG', 'BOS', '767-301ER', '890'),('43', '1153', 'CBM',
'BOI', 'A321', '8989'),('44', '1154', 'COU', 'CAK', '767-301ER', '7676'),('45', '1155', 'CCR',
'EWR', 'CRJ900', '676'),
('46', '1156', 'CDV', 'HNL', '767-301ER', '8668'),('47', '1157',
'DAL', 'LAX', 'CRJ900', '675'),('48', '1158', 'SCC', 'DEN', 'A321', '5645'),('49', '1159',
'DEC', 'ABI', 'A321', '4533'),('50', '1160', 'DRT', 'ORD', 'A321', '2445');


-- QUERY 3 :
PASSENGER DETAILS WHO HAVE TRAVELLED IN ROUTE_ID 1 TO 25 -
SELECT * FROM
PASSENGERS_ON_FLIGHTS
WHERE ROUTE_ID BETWEEN 1 AND 25
ORDER BY CUSTOMER_ID;


--  QUERY 4 :
TOTAL NUMBER OF PASSENGERS AND TOTAL REVENUE FROM BUSINESS CLASS -
SELECT COUNT(DISTINCT
CUSTOMER_ID) AS 'TOTAL PASSENGERS' ,
SUM(PRICE_PER_TICKET * NO_OF_TICKETS) AS 'TOTAL REVENUE
FROM BUSSINESS CLASS'
FROM TICKET_DETAILS
WHERE CLASS_ID = 'BUSSINESS' ;
```

```sql
-- QUERY 5 :
GENERATING FULL NAME OF THE CUSTOMER -
SELECT UCASE(CONCAT(FIRST_NAME , ' ' , LAST_NAME)) AS
FULL_NAME
FROM CUSTOMER_DETAILS ;


-- QUERY 6 : CUSTOMER DETAILS WHO HAVE BOOKED A TICKET
-
SELECT * FROM CUSTOMER
WHERE CUSTOMER_ID IN
(SELECT DISTINCT CUSTOMER_ID FROM
TICKET_DETAILS)
ORDER BY CUSTOMER_ID ;


-- QUERY 7 : CUSTOMER DETAILS BASED ON CUSTOMER_ID AND
BRAND(EMIRATES) -
SELECT CUSTOMER_ID, FIRST_NAME , LAST_NAME FROM CUSTOMER
WHERE CUSTOMER_ID IN

(SELECT CUSTOMER_ID FROM TICKET_DETAILS WHERE BRAND = 'EMIRATES' )
ORDER BY CUSTOMER_ID ;



--
QUERY 8 : CUSTOMERS WHO HAVE TRAVELLED IN ECONOMY CLASS -
SELECT * FROM PASSENGERS_ON_FLIGHTS

GROUP BY SEAT_NUM
HAVING CLASS_ID = 'ECONOMY'
ORDER BY CUSTOMER_ID ;

-- QUERY 9 : DISPLAY IF
TOTAL REVENUE CROSSED 10000 -
SELECT IF(SUM(PRICE_PER_TICKET) >10000 , "YES",
"NO") AS 'DID REVENUE CROSSED 10000?'
FROM TICKET_DETAILS;


-- QUERY 11 : MAXIMUM
TICKET PRICE OF EACH CLASS -
SELECT DISTINCT CLASS_ID ,
MAX(PRICE_PER_TICKET) OVER (PARTITION
BY CLASS_ID) AS 'MAXIMUM TICKET PRICE'
FROM TICKET_DETAILS ;


-- QUERY 12 : EXTRACTING
PASSENGER DETAILS ON ROUTE 4 BY CREATING AN INDEX TO IMPROVE SPEED AND PERFORMANCE OF THE TABLE
-
CREATE INDEX INDEX1 ON PASSENGERS_ON_FLIGHTS (ROUTE_ID);
SELECT * FROM
PASSENGERS_ON_FLIGHTS
WHERE ROUTE_ID = 4 ;





-- QUERY 13 : DISPLAYING EXECUTION PLAN FOR THE
QUERY BELOW -
 SELECT * FROM ROUTES WHERE ROUTE_ID = 4 ;





-- QUERY 14 :  DISPLAYING TOTAL
PRICE OF ALL TICKETS BOOKED BY A CUSTOMER ACROSS DIFFERENT AIRCRAFT IDs -
SELECT  CUSTOMER_ID,
AIRCRAFT_ID , SUM(PRICE_PER_TICKET * NO_OF_TICKETS) AS 'TOTAL PRICE' FROM TICKET_DETAILS
GROUP
BY CUSTOMER_ID, AIRCRAFT_ID
ORDER BY CUSTOMER_ID, AIRCRAFT_ID ;
```

```sql
-- QUERY 15 : CREATING A VIEW
TO SHOW PASSENGER DETAILS IN BUSINESS CLASS -

CREATE VIEW AIRCARGO_VIEW AS
SELECT * FROM
TICKET_DETAILS
WHERE CLASS_ID = 'BUSSINESS' ;

SELECT * FROM AIRCARGO_VIEW ;




-- QUERY 16 :
CREATING A PROCEDURE TO DETERMINE THE ROUTES ALONG WITH AN ERROR HANDLER -
DELIMITER //
CREATE
PROCEDURE ROUTES_PROC_WITH_ERROR_HANDLER()
BEGIN
DECLARE CONTINUE HANDLER FOR SQLSTATE
'42S02'
SELECT 'SQLSTATE Handler - Table Not Found' AS msg;
DECLARE CONTINUE HANDLER FOR
SQLEXCEPTION
BEGIN
GET DIAGNOSTICS CONDITION 1 @sqlstate =
RETURNED_SQLSTATE, @errno =
MYSQL_ERRNO,
@text = MESSAGE_TEXT;
SET @full_error = CONCAT("SQLEXCEPTION Handler - ERROR
", @errno, " (", @sqlstate, "): ", @text); SELECT
@full_error AS
msg;
END;

SELECT ROUTE_ID, FLIGHT_NUM FROM ROUTES ;
END //

CALL
ROUTES_PROC_WITH_ERROR_HANDLER() ;




-- QUERY 17 : CREATING PROCEDURE TO DISPLAY PASSENGER
DETAILS WHERE DISTANCE TRAVELLED IS > 2000 -
DELIMITER //
CREATE PROCEDURE ROUTES_PROC()

BEGIN
SELECT * FROM ROUTES WHERE
DISTANCE_MILES > 2000 ;
END //

CALL ROUTES_PROC()
;




-- QUERY 18 : CREATING A STORED PROCEDURE WHICH WILL DIVIDE DISTANCE TRAVELLED BY EACH
FLIGHT INTO THREE CATEGORIES -
DELIMITER //
CREATE PROCEDURE CATEGORIES(FLIGHT_NUMBER
INT)
BEGIN
DECLARE DIST INT DEFAULT 1;
DECLARE CATEGORY TEXT ;
SELECT DISTANCE_MILES INTO DIST
FROM ROUTES
WHERE FLIGHT_NUMBER = FLIGHT_NUM  ;
IF
```

```
DIST BETWEEN 0 AND 2000 THEN SET CATEGORY =
'SHORT DISTANCE TRAVEL';
ELSEIF DIST > 2000 AND DIST <= 6500 THEN SET CATEGORY =
'INTERMEDIATE DISTANCE TRAVEL';
ELSEIF DIST > 6500 THEN SET CATEGORY = 'LONG DISTANCE
TRAVEL';
END IF ;
SELECT CATEGORY;
END //

CALL CATEGORIES(1111) ;


-- QUERY 19 : EXTRACTING
TICKET DETAILS AND SPECIFYING IF COMPLIMENTARY SERVICES ARE PROVIDED OR NO USING A STORED
FUNCTION -

DELIMITER //
CREATE FUNCTION COMP_SERVICES_FUNC(CLASS TEXT)
RETURNS TEXT
DETERMINISTIC
BEGIN
DECLARE SERVICES TEXT ;
IF
CLASS = 'BUSSINESS' THEN SET SERVICES = 'YES'
;
ELSEIF CLASS = 'ECONOMY PLUS' THEN SET SERVICES = 'YES' ;
ELSE SET SERVICES = 'NO' ;
END
IF;
RETURN (SERVICES);
END //

-- CREATING A STORED PROCEDURE CONTAINING THE ABOVE CREATED
STORED FUNCTION AS AN INPUT -

DELIMITER //
CREATE PROCEDURE COMP_SERVICES_PROC()
BEGIN
SELECT
P_DATE, CUSTOMER_ID, CLASS_ID ,COMP_SERVICES_FUNC(CLASS_ID) AS 'COMPLIMENTARY SERVICES
PROVIDED?' FROM TICKET_DETAILS ;
END //


CALL COMP_SERVICES_PROC() ;






-- QUERY 20 :
CREATING A CURSOR TO GET THE FIRST RECORD OF CUSTOMER HAVING LAST NAME 'SCOTT' -


DELIMITER
//
CREATE PROCEDURE CURSOR_()
BEGIN
DECLARE A VARCHAR (100);
DECLARE B VARCHAR (100);
DECLARE
CURSOR_1 CURSOR FOR SELECT FIRST_NAME , LAST_NAME FROM CUSTOMER_DETAILS
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;
// DELIMITER ;
```

```
CALL CURSOR_();
```