AIR CARGO ANALYSIS

**create database air\_cargo ;**

**use air\_cargo;**

**show tables;**

**desc routes ;**

**select \* from routes ;**

/\* 2. Write a query to create route\_details table using suitable data types for the fields,

such as route\_id, flight\_num, origin\_airport, destination\_airport, aircraft\_id, and distance\_miles.

Implement the check constraint for the flight number and unique constraint for the route\_id fields.

Also, make sure that the distance miles field is greater than 0.

\*/

**use air\_cargo ;**

**create table route\_details (**

**Route\_id int not null UNIQUE,**

**Flight\_Number int not null,**

**CHECK(Flight\_Number>0) ,**

**Origin\_Airport varchar(25),**

**Destination\_Airport varchar(25),**

**Aircraft\_Id varchar(25),**

**Distance\_Miles int ,**

**CHECK(Distance\_Miles>0)**

**);**

select \* from route\_details;

/\* 3. Write a query to display all the passengers (customers) who have travelled in routes 01 to 25.

Take data from the passengers\_on\_flights table.

\*/

**select customer.customer\_id,First\_name,Last\_name,route\_id**

**from customer,passengers\_on\_flights**

**where route\_id between 1 and 25 ;**

/\* Q4) Write a query to identify the number of passengers and total revenue in business class

from the ticket\_details table.

\*/

**select class\_id,count(customer\_id),sum(price\_per\_ticket) from ticket\_details**

**where class\_id = "bussiness" ;**

/\* Q5.Write a query to display the full name of the customer by extracting

the first name and last name from the customer table.

\*/

**select first\_name,last\_name,concat(Firt\_name, " ", last\_name) as NAME from Customer ;**

/\* Q6) Write a query to extract the customers who have registered and booked a ticket.

Use data from the customer and ticket\_details tables. \*/

**select c.first\_name,c.last\_name,td.p\_date,td.no\_of\_tickets,Price\_per\_ticket**

**from customer c,ticket\_details td**

**where c.customer\_id=td.customer\_id ;**

/\* Q7) Write a query to identify the customer’s first name and last name based on

their customer ID and brand (Emirates) from the ticket\_details table.

\*/

**SELECT first\_name, last\_name,brand**

**FROM customer**

**INNER JOIN ticket\_details**

**USING (customer\_id) WHERE brand = "Emirates";**

select\* from passengers\_on\_flights ;

show databases ;

/\* 8. Write a query to identify the customers who have travelled by Economy Plus class

using Group By and Having clause on the passengers\_on\_flights table.

\*/

**SELECT first\_name, last\_name,class\_id**

**FROM customer**

**INNER JOIN passengers\_on\_flights**

**USING (customer\_id) group by First\_name,Last\_name,class\_id having class\_id = "Economy Plus";**

/\* Q9) Write a query to identify whether the revenue has crossed 10000

using the IF clause on the ticket\_details table.

\*/

**select Price\_per\_ticket ,**

**if(sum(Price\_per\_ticket)>10000,"Revenue has Crossed1000","Revenue has Not Crossed 10000") Revenue**

**from ticket\_details group by Price\_per\_ticket ;**

/\*

# Q10 ) Write a query to create and grant access to a new user to perform operations on a database.

\*/

**create user 'shubham'@'localhost' identified by 'mysql.123' ;**

**GRANT ALL PRIVILEGES ON air\_cargo.\* TO 'shubham'@'localhost';**

/\* Q)11. Write a query to find the maximum ticket price for each class using

window functions on the ticket\_details table.

\*/

**select class\_id ,max(Price\_per\_ticket) over(partition by class\_id) as "Maximum Ticket Price"**

**from ticket\_details ;**

# Q13-->

**select distinct c.first\_name, p.route\_id from customer c,passengers\_on\_flights p**

**where p.route\_id=4 limit 20;**

**select c.first\_name ,p.route\_id from customer c, passengers\_on\_flights p**

**where c.customer\_id=p.customer\_id and p.route\_id=4;**

/\* Q) 12 Write a query to extract the passengers whose route ID is 4 by improving the speed

and performance of the passengers\_on\_flights table.

\*/

**explain select c.first\_name ,p.route\_id from customer c,passengers\_on\_flights p**

**where c.customer\_id=p.customer\_id and p.route\_id=4 ;**

# Q.13) For the route ID 4, write a query to view the execution plan of the

# passengers\_on\_flights table.

**select \* from passengers\_on\_flights where route\_id = 4 ;**

**create index indx\_route on passengers\_on\_flights(route\_id) ;**

**drop index indx\_route on passengers\_on\_flights ;**

**select first\_name,last\_name from customer where last\_name like "%scott" order by forst;**

/\* Q14) Write a query to calculate the total price of all tickets booked by

# a customer across different aircraft IDs

using rollup function.

\*/

**select customer\_id,sum(price\_per\_ticket) from ticket\_details group by customer\_id with rollup ;**

# Q.15) Write a query to create a view with only business class customers

# along with the brand of airlines.

**select c.first\_name,td.class\_id,td.brand from customer c,ticket\_details td**

**where c.customer\_id = td.customer\_id**

**and td.class\_id = "Bussiness" ;**

/\* 16. Write a query to create a stored procedure to get the details of all passengers

flying between a range of routes defined in run time. Also, return an error

message if the table doesn't exist. \*/

**select \* from passengers\_on\_flights ;**

/\* 17. Write a query to create a stored procedure that extracts all the details from the routes table

where the travelled distance is more than 2000 miles.

\*/

**call Route ;**

/\* 18. Write a query to create a stored procedure that groups the distance travelled

by each flight into three categories.

The categories are, short distance travel (SDT) for >=0 AND <= 2000 miles,

intermediate distance travel (IDT) for >2000 AND <=6500,

and long-distance travel (LDT) for >6500.

\*/

**call Distance\_Travelled ;**

/\* 19. Write a query to extract ticket purchase date, customer ID, class ID and

specify if the complimentary services are provided for the specific class

using a stored function in stored procedure on the ticket\_details table.

\*/

**call Complimentary\_Services ;**

**select \* from customer ;**

/\* 20. Write a query to extract the first record of the customer whose last name ends with Scott

using a cursor from the customer table.

\*/

call scott\_cur ;

**create Procedure 'get\_sudent\_info' ()**

**BEGIN**

**select \* from student where student.age=31 ;**

**END**

**;**

**call get\_student\_info;**

# to give option to make store procedure parameterized

# IN, Out or IN and OUT

**CREATE PROCEDURE 'get\_student\_info'(IN age int)**

**BEGIN**

**select \* from student where student.age=age**

**END;**

**call get\_student\_info(27) ;# here we have input parameter 27 [5.00]**

**CREATE PROCEDURE 'get\_student\_info'(OUT records int)**

**BEGIN**

**select count(\*) into records from student where student.age=31**

**END;**

**call get\_student\_info(@records) ;**

**select @records as Totalrecords;**

#--> it prints count of total rcords havig age = 31

# 9.00 IN and OUT

**CREATE PROCEDURE 'get\_student\_info'(INOUT records int, IN age int)**

**BEGIN**

**select count(\*) into records from student where student.age=age**

**END;**

**call get\_student\_info(@records,27) ;**

**select @records as Totalrecords;**