**Answer 2:**

* SELECT \* FROM customer

WHERE customer\_id IN (SELECT DISTINCT customer\_id FROM passengers\_on\_flights WHERE route\_id between 1 and 25)

ORDER BY customer\_id;

**Answer 3:**

* SELECT COUNT(DISTINCT customer\_id) AS num\_passenger, SUM(no\_of\_tickets \* price\_per\_ticket) AS total\_revenue

FROM ticket\_details

WHERE class\_id = 'Bussiness';

**Answer 4:**

* SELECT CONCAT(first\_name, " ", last\_name) AS Full\_name

FROM customer;

**Answer 5:** There are two ways to solve:-

1. SELECT DISTINCT c.customer\_id, c.first\_name, c.last\_name

FROM customer c

JOIN ticket\_details td

ON c.customer\_id = td.customer\_id

WHERE no\_of\_tickets > 0

ORDER BY customer\_id ASC;

1. SELECT first\_name, last\_name FROM customer

WHERE customer\_id

IN (SELECT DISTINCT tb.customer\_id FROM customer c,ticket\_details tb);

**Answer 6:** There are two ways to solve:-

* SELECT first\_name, last\_name FROM customer

WHERE customer\_id IN (SELECT DISTINCT customer\_id FROM ticket\_details WHERE brand = "Emirates");

* SELECT DISTINCT c.customer\_id, c.first\_name, c.last\_name FROM customer c

JOIN ticket\_details td

ON c.customer\_id = td.customer\_id

WHERE brand = "Emirates"

ORDER BY 1;

**Answer 7:** There are two ways to solve:-

* SELECT \* FROM customer c

INNER JOIN (SELECT DISTINCT customer\_id FROM passengers\_on\_flights pof WHERE class\_id = "economy plus") pof

ON c.customer\_id = pof.customer\_id;

* SELECT c.customer\_id, c.first\_name, c.last\_name FROM customer c

INNER JOIN passengers\_on\_flights pof

ON c.customer\_id = pof.customer\_id

WHERE pof.class\_id = 'Economy Plus'

GROUP BY c.customer\_id, c.first\_name, c.last\_name

HAVING COUNT(\*) >= 1;

**Answer 8:**

* SELECT IF(SUM(no\_of\_tickets \* price\_per\_ticket) >= 10000, 'Crossed 10K', 'Not Crossed 10K') AS revenue\_check

FROM ticket\_details;

**Answer 9:** -- create a new user

* CREATE USER if not exists 'Anubhav'@'127.0.0.1' IDENTIFIED BY 'password';

-- grant selected privileges to the new user

* GRANT SELECT, INSERT, UPDATE, DELETE ON database\_name.\* TO new\_user;

-- grant all privileges to the new user

* GRANT all privileges ON airlines TO 'Anubhav'@'127.0.0.1';

**Answer 10:**

* SELECT DISTINCT class\_id, MAX(price\_per\_ticket) OVER (PARTITION BY class\_id) AS max\_ticket\_price

FROM ticket\_details

ORDER BY max\_ticket\_price;

**Answer 11:**

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

**Answer 12:**

* CREATE INDEX idx\_route\_id ON passengers\_on\_flights (route\_id);
* EXPLAIN SELECT \* FROM passengers\_on\_flights WHERE route\_id = 4;

**Answer 13:** WITH ROLLUP: The WITH ROLLUP option adds extra rows to the result set that provide subtotal and grand total values for each customer\_id and for all customers, respectively.

* SELECT customer\_id, aircraft\_id, SUM(price\_per\_ticket \* no\_of\_tickets) AS total\_price

FROM ticket\_details

GROUP BY customer\_id, aircraft\_id WITH ROLLUP;

WITHOUT ROLLUP:

* 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;

**Answer 14:** CREATE VIEW

* CREATE VIEW Bussiness\_Class\_Customers As

SELECT c.\*, td.brand FROM customer c

INNER JOIN (SELECT DISTINCT customer\_id, brand FROM ticket\_details WHERE class\_id = 'Bussiness' ORDER BY customer\_id) td

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

DISPLAY VIEW

* SELECT \* FROM Bussiness\_Class\_Customers**;**

**Answer 15:** CREATE PROCEDURE

* DELIMITER //

CREATE PROCEDURE Check\_distance()

Begin

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

End //

DELIMITER **;**

CALL PROCEDURE

* CALL check\_distance**;**

**Answer 16:**

* DELIMITER //

CREATE FUNCTION group\_dist(dist INT)

returns VARCHAR(10)

deterministic

Begin

DECLARE dist\_cat CHAR(3);

IF dist BETWEEN 0 AND 2000 THEN

SET dist\_cat = 'SDT';

ELSEIF dist BETWEEN 2001 AND 6500 THEN

SET dist\_cat = 'IDT';

ELSEIF dist > 6500 THEN

SET dist\_cat = 'LDT';

End IF;

return(dist\_cat);

END //

* CREATE PROCEDURE group\_dist\_proc()

Begin

SELECT flight\_num, distance\_miles, group\_dist(distance\_miles) AS distance\_category FROM routes;

End //

DELIMITER ;

* CALL group\_dist\_proc;

**Answer 17:** Without Stored procedure:

* SELECT p\_date, customer\_id, class\_id,

CASE

WHEN class\_id in ('Bussiness', 'Economy Plus') THEN 'Yes'

Else 'No'

End as complimentary\_service FROM ticket\_details;

With store Procedure:

* DELIMITER //

CREATE FUNCTION check\_comp\_serv (cls varchar (15))

returns char (3)

deterministic

Begin

declare comp\_ser char (3);

If cls in ('Bussiness', 'Economy Plus') then

set comp\_ser = 'Yes';

Else

set comp\_ser = 'NO';

End if;

return(comp\_ser);

End //

* CREATE PROCEDURE check\_comp\_serv\_proc()

Begin

SELECT p\_date, customer\_id, class\_id, check\_comp\_serv(class\_id) AS complimentary\_service FROM ticket\_details;

End //

DELIMITER ;

* CALL check\_comp\_serv\_proc;

**Answer 18:**

* SELECT \* FROM customer

WHERE last\_name = 'Scott'

LIMIT 1;