**Queries**

1. **Create an SQL query to identify a list of passengers based on their method of payment. For each method of payment, Airline can plan to offer corresponding incentives/rewards to passengers.**

**Explanation:** From booked tickets and passenger lists, for each method of payment, Airline can plan to offer corresponding incentives/rewards to the passengers. We can create an SQL query to identify a list of passengers based on their method of payment. The report can then be used for predicting the lift curve for an incentive program for credit card users.

**Code-**

select p.passenger\_id, p.passenger\_first\_name, p.passenger\_last\_name, p1.payment\_method

from passenger p

join trip\_ticket t

on p.passenger\_id=t.passenger\_id

join payment p1

on t.payment\_id=p1.payment\_id

where p1.payment\_method='&enter\_payment\_method';

**Output-**

Graphical user interface

Description automatically generated

1. **Create a query to identify the longest flight and its price. Order by decreasing price.**

**Explanation:** We can create a query to identify the longest flight and its price. Different airlines might have different prices for a similar duration of the flight. With this result, we can predict a tentative price an airline can aim for a future new flight between new destinations with similar flight durations.

**Code-**

SELECT f.flight\_id, (f.arrival\_time - f.departure\_time) as duration, p.amount

FROM trip\_ticket tt JOIN flight f

ON(tt.flight\_id = f.flight\_id)

JOIN payment p

ON(tt.payment\_id = p.payment\_id)

ORDER by duration desc

;

**Output-**

Table

Description automatically generated

1. **Create a SQL query to know the most flown/preferred airline.**

**Explanation:** We can create a SQL query to know the most flown/preferred airline from the ticket booking history and passenger list. The query can help to know which airline needs improvements in attracting the passengers.

**Code-**

SELECT count(tt.passenger\_id) as "Passenger Count", A.airline\_name

FROM trip\_ticket TT JOIN flight F

ON (TT.flight\_id = f.flight\_id)

RIGHT JOIN airline A

ON(F.airline\_id = A.airline\_id)

RIGHT JOIN passenger p

ON ( p.passenger\_id = TT.passenger\_id)

GROUP BY A.airline\_name

ORDER BY count(tt.passenger\_id) desc

;

**Output-**

Table

Description automatically generated

1. **Create a SQL query to identify trip search results that yielded a successful ticket booking. Output the search id and the ticket id for the same passengers with origin and destination city.**

**Explanation:** We can create a SQL query to identify trip search results that yielded a successful ticket booking. This powerful search helps to determine the high-demand routes for airlines. Those searches which did not result in a ticket booking can be determined as low demand routes by the airlines.

**Code-**

SELECT result2.search\_id, result2.origin\_city, result2.destination\_city, result1.ticket\_id

FROM (

SELECT tt.ticket\_id as ticket\_id, f.origin\_city as origin\_city, f.destination\_city as destination\_city, f.departure\_date as departure\_date, f.airline\_id as airline\_id, tt.passenger\_id as passenger\_id

FROM trip\_ticket tt JOIN flight f

ON(tt.flight\_id = f.flight\_id)

) result1

JOIN

(

SELECT ts.search\_id as search\_id, ts.origin\_city as origin\_city, ts.destination\_city as destination\_city, ts.departure\_date as departure\_date, p.passenger\_id as passenger\_id

FROM trip\_search ts JOIN passenger p

ON(ts.passenger\_id = p.passenger\_id)

)result2

ON(result1.passenger\_id = result2.passenger\_id)

WHERE (result1.origin\_city = result2.origin\_city and result1.destination\_city = result2.destination\_city and result1.departure\_date = result2.departure\_date)

;

**Output-**

Table

Description automatically generated

1. **Create an SQL query using the decode function.**

**Explanation:** We can create an SQL query to identify which airline is used more frequently by passengers. The query can help the airlines to improve and take their business to the next level.

**Code-**

select flight\_id,

DECODE(airline\_id, 1 ,'Indigo', 2 ,'Spicejet', 3 ,'Vistara', 4 ,'Akasa', 5 ,'GoFirst', 6 ,'TruJet', 7 ,'Flybig')output

from flight;

**Output-**

Table

Description automatically generated with medium confidence

Table

Description automatically generated with low confidence

1. **--create an SQL query to display no of flights having the same origin city greater than 10.**

**Explanation:** We can create an SQL query to identify which city has more frequent flights. This can help various stakeholders involved in the process plan their business accordingly.

**Code:**

select origin\_city,count(origin\_city)

from flight

group by origin\_city

having count(origin\_city)>10 ;

**Output:**

Graphical user interface, text, application, email

Description automatically generated