

109- NOTE: INNER JOIN output is rows that are common in the both the tables.

110- Display customer first name, last name and payments they made using tables (customer, payment).

```
SELECT
first_name
,last_name
,payment.*
FROM customer
INNER JOIN payment
ON customer.customer_id=payment.customer_id
ORDER BY payment.customer_id
```

111- Display comparison between sale of economy, business and comfort seats from table (boarding_passes, seats).

```
SELECT
fare_conditions,
COUNT(fare_conditions)
FROM boarding_passes
INNER JOIN seats
ON boarding_passes.seat_no=seats.seat_no
GROUP BY fare_conditions
ORDER BY COUNT(fare_conditions) DESC
```

112- NOTE: FULL OUTER JOIN output is all rows from both the tables

113- Display data of tickets that have not been allotted boarding passes yet using table (boarding_passes, tickets).

```
SELECT t.*
FROM tickets as t
FULL OUTER JOIN boarding_passes as b
ON t.ticket_no=b.ticket_no
WHERE seat_no IS NULL
```

114- NOTE: LEFT OUTER JOIN output is all rows from left table.

115- Display the number of times each seat has been booked by passengers from table (seats, boarding_passes).

```
SELECT s.seat_no,COUNT(*)
FROM seats AS s
LEFT JOIN boarding_passes AS b
ON s.seat_no = b.seat_no
GROUP BY s.seat_no
ORDER BY count(*) DESC
```

116- Display the number of times each row (A,B,C,...) has been booked by passengers from table (seats, boarding_passes).

```
SELECT RIGHT(s.seat_no,1) AS seat_row
,COUNT(*)
FROM seats AS s
LEFT JOIN boarding_passes AS b
ON s.seat_no = b.seat_no
GROUP BY seat_row
ORDER BY COUNT(*) DESC
```

117- NOTE: RIGHT OUTER JOIN output is all rows from left table.

118- NOTE: TABLE_A LEFT JOIN TABLE_B == TABLE_B RIGHT JOIN TABLE_A.

119- Display contact details of customer from 'Texas' from table (customer, address).

```
SELECT
first_name
,last_name
,phone
,district
FROM customer AS c
INNER JOIN address AS a
ON c.address_id = a.address_id
WHERE district = 'Texas'
```

118- Display addresses that belong to no customer from table (customer, address).

```
SELECT
a.*
FROM customer AS c
RIGHT JOIN address AS a
ON c.address_id = a.address_id
WHERE c.customer_id IS NULL
```

119- JOIN conditions can also be used as filters like WHERE clause

120- Display the average prices of seats from table (seats, ticket_flights).

```
SELECT
seat_no
,ROUND(AVG(amount),2)
FROM boarding_passes as b
INNER JOIN ticket_flights as t
ON b.ticket_no = t.ticket_no AND b.flight_id = t.flight_id
GROUP BY seat_no
ORDER BY avg(amount) DESC
```

121: NOTE: In this case we require two conditions for join because the pair of ticket_no & flight_id together makes the primary key.

122- NOTE: You can also join multiple tables as per your requirements.

123- NOTE: While joining more than two tables via 'INNER JOIN' joining order doesn't matter.

124- NOTE: The order of joining multiple does matter in case of LEFT/RIGHT JOIN.

125- Get passenger ticket number, passenger name and their scheduled departure time from tables(ticket_flights,tickets,flights).

```
SELECT
t.ticket_no,t.passenger_name,f.scheduled_departure
FROM tickets AS t
INNER JOIN ticket_flights AS tf
ON t.ticket_no = tf.ticket_no
INNER JOIN flights AS f
ON tf.flight_id = f.flight_id
```

126. Get all customer's firstName,lastName and email address those who are from Brazil using tables(customer, address, city, country).

```
SELECT
```

```

first_name, last_name, email, country
FROM customer
INNER JOIN address
ON customer.address_id = address.address_id
INNER JOIN city
ON address.city_id=city.city_id
INNER JOIN country
ON city.country_id=country.country_id AND country.country='Brazil'
ORDER BY first_name, last_name

```

127. Get all customer's firstName, lastName and email address those who are from Brazil using tables(customer, address, city, country). (another method)

```

SELECT
first_name, last_name, email, country
FROM customer
LEFT JOIN address
ON customer.address_id = address.address_id
LEFT JOIN city
ON address.city_id=city.city_id
LEFT JOIN country
ON city.country_id=country.country_id
WHERE country='Brazil'
ORDER BY first_name, last_name

```

128. Which passenger (passenger_name) has spent most amount in their bookings (total_amount)?

```

SELECT
passenger_name, sum(amount) as total_amount
FROM tickets
INNER JOIN ticket_flights
ON tickets.ticket_no=ticket_flights.ticket_no
GROUP BY passenger_name
ORDER BY sum(amount) DESC
LIMIT 1

```

129. Which fare_condition has ALEKSANDR IVANOV used the most?

```

SELECT
fare_conditions, count(*)
FROM tickets
LEFT JOIN ticket_flights
ON tickets.ticket_no=ticket_flights.ticket_no
WHERE passenger_name = 'ALEKSANDR IVANOV'
GROUP BY fare_conditions
ORDER BY count(*) DESC
LIMIT 1

```

130. Which title has GEORGE LINTON rented the most often?

```

SELECT
title, count(*)
FROM rental
INNER JOIN inventory
ON rental.inventory_id = inventory.inventory_id
INNER JOIN customer
ON rental.customer_id = customer.customer_id
AND customer.first_name = 'GEORGE'
AND customer.last_name = 'LINTON'
INNER JOIN film
ON inventory.film_id = film.film_id
GROUP BY title
ORDER BY count(*) DESC

```

LIMIT 1