**SQL Assignment**

Use the following model to answer the queries given below:

**Port\_Details**

|  |
| --- |
|  |
| Port\_id (PK ) |
| name |
| state |
| Country |
| Zip |

**Ship\_details**

|  |
| --- |
| Ship\_id (PK) |
| Name |
| Type |
| Max weight |
| Port1 (FK) |
| Port2 (FK) |

**Customer\_details**

|  |
| --- |
| Customer\_id (PK) |
| Fname |
| LastName |
| Address |
| Phno |

**Item\_details**

|  |
| --- |
| Item\_id (PK) |
| Name |
| Type |
| Base price |
| weight |

**Trip\_Details**

|  |
| --- |
| Trip\_Id |
| Ship\_id(FK) |
| Item\_id(FK) |
| Customer\_id(FK) |
| Booking\_date |
| To\_place(FK) |
| From\_place(FK) |
| Departing\_date |
| Arrival\_date |
| Receiver\_id |
| No. of Units |
| Net wt. |
| Net Amt. |
| Booking Status |
| Ship Trip status |
|  |

1. Which is the busiest port (includes both arrival and departure)?

SELECT p.Name AS BusiestPort, COUNT(\*) AS TotalTrips

FROM Port\_Details AS p

JOIN (SELECT Port1 AS Port\_id FROM Ship\_details UNION ALL SELECT Port2 AS Port\_id FROM Ship\_details) AS s ON p.Port\_id = s.Port\_id

JOIN Trip\_Details AS t ON s.Port\_id IN (t.From\_place, t.To\_place)

GROUP BY p.Name

ORDER BY TotalTrips DESC

LIMIT 1;

1. Find the list of trips and their respective details where the ship was overloaded

SELECT \* FROM Trip\_Details

WHERE Net\_weight > (SELECT Max\_weight FROM Ship\_details WHERE Ship\_id = Trip\_Details.Ship\_id);

1. List the ships which dint have any trip in the last 6 months

SELECT \* FROM Ship\_details AS s

WHERE s.Ship\_id NOT IN (

SELECT DISTINCT t.Ship\_id

FROM Trip\_Details AS t

WHERE t.Departing\_date > DATE\_SUB(NOW(), INTERVAL 6 MONTH)

);

1. Find the revenue generated by each ship till date

SELECT Ship\_id, SUM(Net\_Amount) as Revenue

FROM Trip\_Details

GROUP BY Ship\_id;

1. Find the list of items which are shipped more this year when compared with last year

SELECT Item\_id, SUM(No\_of\_Units) AS Total\_units, strftime('%Y', Booking\_date) AS Year

FROM Trip\_Details

GROUP BY Item\_id, Year

HAVING Year >= strftime('%Y', date('now', '-1 year'));

1. Find the cumulative revenue for all the items every month. The output should be in the following format:

|  |  |  |  |
| --- | --- | --- | --- |
| item name | Month | revenue | cumulative revenue |
| A | Jan-15 | 100 | 100 |
| A | Feb-15 | 75 | 175 |
| ………  ……… |  |  |  |
| B | Jan-15 | 55 | 55 |
| B | Feb-15 | 60 | 115 |
| …….. |  |  |  |
| …….. |  |  |  |

SELECT Item\_details.Name AS Item\_name,

FORMAT(Booking\_date, 'MMM-yy') AS Month,

SUM(Net\_Amount) AS Revenue,

SUM(SUM(Net\_Amount)) OVER (PARTITION BY Trip\_Details.Item\_id ORDER BY FORMAT(Booking\_date, 'yyyy-MM-dd')) AS Cumulative\_revenue

FROM Trip\_Details

JOIN Item\_details ON Trip\_Details.Item\_id = Item\_details.Item\_id

GROUP BY Item\_details.Name, FORMAT(Booking\_date, 'MMM-yy'), Trip\_Details.Item\_id

ORDER BY Item\_details.Name, FORMAT(Booking\_date, 'yyyy-MM-dd');

1. Find the customer who had shipped to the most number of places

SELECT Customer\_details.Fname, Customer\_details.LastName, COUNT(DISTINCT To\_place) AS Places\_shipped\_to

FROM Trip\_Details

JOIN Customer\_details ON Trip\_Details.Customer\_id = Customer\_details.Customer\_id

GROUP BY Customer\_details.Fname, Customer\_details.LastName

ORDER BY COUNT(DISTINCT To\_place) DESC

LIMIT 1;

1. Give the customer and trip details in which a sender had received any item in the subsequent trip of the same ship. The output should be in the following format

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| customer name | ship name | sent item | source | destination | sent date | received date | received item |
|  |  |  |  |  |  |  |  |

SELECT c.Fname AS Customer\_name, s.Name AS Ship\_name,

t1.Item\_id AS Sent\_item, t1.From\_place AS Source, t1.To\_place AS Destination,

t1.Departing\_date AS Sent\_date, t2.Arrival\_date AS Received\_date, t2.Item\_id AS Received\_item

FROM Trip\_Details AS t1

JOIN Trip\_Details AS t2 ON t1.Ship\_id = t2.Ship\_id

JOIN Customer\_details AS c ON t1.Customer\_id = c.Customer\_id

JOIN Ship\_details AS s ON t1.Ship\_id = s.Ship\_id

WHERE t2.Departing\_date > t1.Departing\_date

AND t2.Arrival\_date > t1.Arrival\_date

AND t2.Item\_id IS NOT NULL

ORDER BY c.Fname, t1.Departing\_date;

1. Find the list of ships name, trip date, return date where the return trip of the ship had taken more time than the original trip

SELECT s.Name AS ShipName, t1.Departing\_date AS TripDate, t2.Arrival\_date AS ReturnDate

FROM Trip\_Details AS t1

JOIN Trip\_Details AS t2 ON t1.Ship\_id = t2.Ship\_id AND t1.To\_place = t2.From\_place AND t1.From\_place = t2.To\_place

JOIN Ship\_details AS s ON t1.Ship\_id = s.Ship\_id

WHERE t2.Arrival\_date - t1.Departing\_date > t1.Arrival\_date - t1.Departing\_date;

1. For each ship provide the departing date in which it was the lightest and heaviest

SELECT Ship\_id, MIN(Departing\_date) AS Lightest\_Departure, MAX(Departing\_date) AS Heaviest\_Departure

FROM Trip\_Details

GROUP BY Ship\_id;

2) The relation Chocolates (name, Price). Find out names of the five most expensive chocolates.

|  |  |
| --- | --- |
| **Name** | **Price** |
| Dairy Milk | 200 |
| Five Star | 100 |
| Gems | 300 |
| Perk | 400 |
| Silk | 600 |
| Bourneville | 500 |
| Celebrations | 800 |

SELECT Name

FROM Chocolates

ORDER BY Price DESC

LIMIT 5;