**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 DATE\_SUB(t2.Arrival\_date, INTERVAL (t1.Arrival\_date - t1.Departing\_date) DAY) > 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;

-- Port\_Details table

INSERT INTO Port\_Details (Port\_id, Name, State, Country, Zip)

VALUES (1, 'Port A', 'State A', 'Country A', '12345');

-- Ship\_Details table

INSERT INTO Ship\_Details (Ship\_id, Name, Type, Max\_weight, Port1, Port2)

VALUES (1, 'Ship 1', 'Type 1', 1000, 1, 2);

-- Customer\_Details table

INSERT INTO Customer\_Details (Customer\_id, Fname, LastName, Address, Phno)

VALUES (1, 'Paritosh', 'Dev', 'Address 1', '1234567890');

-- Item\_Details table

INSERT INTO Item\_Details (Item\_id, Name, Type, Base\_price, Weight)

VALUES (1, 'Item 1', 'Type 1', 50, 10);

-- Trip\_Details table

INSERT INTO Trip\_Details (Trip\_Id, Ship\_id, Item\_id, Customer\_id, Booking\_date, To\_place, From\_place, Departing\_date, Arrival\_date, Receiver\_id, No\_of\_Units, Net\_wt, Net\_Amt, Booking\_Status, Ship\_Trip\_Status)

VALUES (1, 1, 1, 1, '2023-05-01', 1, 2, '2023-05-02', '2023-05-03', 2, 5, 50, 250, 'Confirmed', 'Completed');

Q1

| BusiestPort | TotalTrips |
| --- | --- |
| Port B | 1 |

Q2

| Trip\_Id | Ship\_id | Item\_id | Customer\_id | Booking\_date | To\_place | From\_place | Departing\_date | Arrival\_date | Receiver\_id | No\_of\_Units | Net\_wt | Net\_Amt | Booking\_Status | Ship\_Trip\_Status |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 1 | 1 | 1 | 1 | 2023-05-15 | 2 | 1 | 2023-05-16 | 2023-05-20 | 1 | 10 | 1200 | 200 | Booked | In progress |

Q3

Q4

| Ship\_id | Revenue |
| --- | --- |
| 1 | 200 |

| Item\_id | Total\_units | Year |
| --- | --- | --- |
| 1 | 10 | 2023 |

Q5

Q6

| Item\_name | Month | Revenue | Cumulative\_revenue |
| --- | --- | --- | --- |
| Item 1 | 2023-05-15 | 200 | 200 |

Q7

| Fname | LastName | Places\_shipped\_to |
| --- | --- | --- |
| Paritosh | Dev | 1 |

Q8

Q9

Q10

| Ship\_id | Lightest\_Departure | Heaviest\_Departure |
| --- | --- | --- |
| 1 | 2023-05-16 | 2023-05-16 |
| 2 | 2023-06-11 | 2023-06-11 |

CREATE TABLE Port\_Details (

Port\_id INTEGER PRIMARY KEY,

Name VARCHAR(255),

State VARCHAR(255),

Country VARCHAR(255),

Zip VARCHAR(10)

);

CREATE TABLE Ship\_Details (

Ship\_id INTEGER PRIMARY KEY,

Name VARCHAR(255),

Type VARCHAR(255),

Max\_weight INTEGER,

Port1 INTEGER,

Port2 INTEGER,

FOREIGN KEY (Port1) REFERENCES Port\_Details(Port\_id),

FOREIGN KEY (Port2) REFERENCES Port\_Details(Port\_id)

);

CREATE TABLE Customer\_Details (

Customer\_id INTEGER PRIMARY KEY,

Fname VARCHAR(255),

LastName VARCHAR(255),

Address VARCHAR(255),

Phno VARCHAR(20)

);

CREATE TABLE Item\_Details (

Item\_id INTEGER PRIMARY KEY,

Name VARCHAR(255),

Type VARCHAR(255),

Base\_price INTEGER,

Weight INTEGER

);

CREATE TABLE Trip\_Details (

Trip\_Id INTEGER,

Ship\_id INTEGER,

Item\_id INTEGER,

Customer\_id INTEGER,

Booking\_date DATE,

To\_place INTEGER,

From\_place INTEGER,

Departing\_date DATE,

Arrival\_date DATE,

Receiver\_id INTEGER,

No\_of\_Units INTEGER,

Net\_wt INTEGER,

Net\_Amt INTEGER,

Booking\_Status VARCHAR(255),

Ship\_Trip\_Status VARCHAR(255),

FOREIGN KEY (Ship\_id) REFERENCES Ship\_Details(Ship\_id),

FOREIGN KEY (Item\_id) REFERENCES Item\_Details(Item\_id),

FOREIGN KEY (Customer\_id) REFERENCES Customer\_Details(Customer\_id),

FOREIGN KEY (To\_place) REFERENCES Port\_Details(Port\_id),

FOREIGN KEY (From\_place) REFERENCES Port\_Details(Port\_id)

);