

Queries for Table Creation

```
CREATE TABLE Customer (  
    Customer_ID int IDENTITY(1,1) PRIMARY KEY,  
    Name VARCHAR(255),  
);
```

```
CREATE TABLE Bookstore(  
    Bookstore_ID int IDENTITY(1,1) PRIMARY KEY,  
);
```

```
CREATE TABLE Orders (  
    Order_ID int IDENTITY(1,1) PRIMARY KEY,  
    Date_time datetime,  
    Shipping_address VARCHAR(255),  
    Customer_ID int,  
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)  
);
```

```
CREATE TABLE Publication (  
    Pub_ID int IDENTITY(1,1),  
    Publisher VARCHAR(255),  
    Pub_Year int,  
    PRIMARY KEY (Pub_ID)  
);
```

```
CREATE TABLE Stocks_in_Bookstore (  
    Stock_ID int IDENTITY(1,1),  
    Stock_Price smallmoney,  
    Stock_Qty int,  
    Pub_ID int,  
    Bookstore_ID int,  
    PRIMARY KEY (Stock_ID),  
    FOREIGN KEY (Pub_ID) REFERENCES Publication(Pub_ID),  
    FOREIGN KEY (Bookstore_ID) REFERENCES Bookstore(Bookstore_ID)  
);
```

```
CREATE TABLE Items_In_Orders (  

```

```

    Item_ID int IDENTITY(1,1),
    Order_ID int,
    Stock_ID int,
    Item_Price smallmoney,
    Item_Quantity int,
    Delivery_Date date,
    Customer_ID int,
    Date_Time datetime,
    Comment VARCHAR(255),
    Rating int,
    PRIMARY KEY (Item_Id),
    FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID),
    FOREIGN KEY (Stock_ID) REFERENCES Stocks_in_Bookstore(Stock_ID),
    CHECK (Rating > 0 AND Rating < 6)
);

CREATE TABLE Item_Status (
    Date date,
    State varchar(20) DEFAULT 'being processed',
    Item_ID int,
    PRIMARY KEY (Date, Item_ID),
    FOREIGN KEY (Item_ID) REFERENCES Items_In_Orders(Item_ID),
    CONSTRAINT Check_Item_State CHECK (State IN ('being processed', 'shipped',
'delivered', 'returned'))
);

CREATE TABLE Employees (
    Employee_ID int IDENTITY(1,1) PRIMARY KEY,
    Name VARCHAR(255),
    Salary smallmoney
);

CREATE TABLE Complaints (
    Complaint_ID int IDENTITY(1,1) PRIMARY KEY,
    Employee_ID int,
    Customer_ID int,

```

```
Text VARCHAR(255),
Filed_Date_Time datetime,
Handled_Date_Time datetime,
FOREIGN KEY (Employee_ID) REFERENCES Employees(Employee_ID),
FOREIGN KEY (Customer_ID) REFERENCES Customer(Customer_ID)
);
```

```
CREATE TABLE Complaints_On_Order (
    Complaint_On_Order_ID int,
    Order_ID int,
    PRIMARY KEY (Complaint_On_Order_ID, Order_ID),
    FOREIGN KEY (Complaint_On_Order_ID) REFERENCES
Complaints(Complaint_ID),
    FOREIGN KEY (Order_ID) REFERENCES Orders(Order_ID)
);
```

```
CREATE TABLE Complaints_On_Bookstore (
    Complaint_On_Order_ID int,
    Bookstore_ID int,
    PRIMARY KEY (Complaint_On_Order_ID, Bookstore_ID),
    FOREIGN KEY (Complaint_On_Order_ID) REFERENCES
Complaints(Complaint_ID),
    FOREIGN KEY (Bookstore_ID) REFERENCES Bookstore(Bookstore_ID)
);
```

```
CREATE TABLE Complaint_Status (
    Date date,
    State varchar(20) DEFAULT 'pending',
    Complaint_ID int,
    PRIMARY KEY (Date, Complaint_ID),
    FOREIGN KEY (Complaint_ID) REFERENCES Complaints(Complaint_ID),
    CONSTRAINT Check_Complaint_State CHECK (State IN ('pending', 'being
handled', 'addressed'))
);
```

```
CREATE TABLE Price_History (
    Stock_ID int,
```

```
Price smallmoney,  
Start_date date,  
End_date date,  
PRIMARY KEY (Stock_ID, Price, Start_date, End_date),  
FOREIGN KEY (Stock_ID) REFERENCES Stocks_in_Bookstore(Stock_ID)  
);
```

```
CREATE TABLE Books (  
    Pub_ID int PRIMARY KEY (Pub_ID) REFERENCES Publication(Pub_ID),  
    Title VARCHAR(255)  
);
```

```
CREATE TABLE Magazines (  
    Pub_ID int PRIMARY KEY (Pub_ID) REFERENCES Publication(Pub_ID),  
    Issue int,  
    Title VARCHAR(255)  
);
```

Required SQL Queries and their Outputs

1. Find the average price of “Harry Porter Finale” on Ahamazon from 1 August 2022 to 31 August 2022.

```
SELECT AVG(P.Price) AS Avg_Price
FROM Price_History P, Stocks_In_Bookstore SIB, Books B
WHERE P.Stock_ID = SIB.Stock_ID
AND SIB.Pub_ID = B.Pub_ID
AND B.Title = 'Harry Porter Finale'
AND P.Start_date >= '2022-08-01'
AND P.End_date <= '2022-08-31'
```

Query Output:

	123 Avg_Price	
1	30.0000	

Query Explanation: We identify the average price for the book ‘Harry Porter Finale’ from the Price_History table which is used to identify the price history for the book in August.

2. Find publications that received at least 10 ratings of “5” in August 2022, and rank them by their average ratings.

```
SELECT S.Pub_ID, AVG(I.Rating) AS Avg_Rating
FROM Stocks_in_Bookstore AS S, Items_In_Orders AS I
WHERE S.Stock_ID = I.Stock_ID
AND I.Rating IS NOT NULL
AND S.Pub_ID IN (
    SELECT S.Pub_ID
    FROM Stocks_in_Bookstore AS S, Items_In_Orders AS I
    WHERE S.Stock_ID = I.Stock_ID
    AND I.Rating = 5
    AND MONTH(I.Date_Time) = 08
    AND YEAR(I.Date_Time) = 2022
    GROUP BY S.Pub_ID
    HAVING COUNT(*) >= 10
)
GROUP BY S.Pub_ID
ORDER BY Avg_Rating DESC
```

Query Output:

	123 Pub_ID ▾	123 Avg_Rating ▾	
1	3 ↗	5	
2	4 ↗	4	
3	1 ↗	4	

Query Explanation: We first extract the pub_id of all those publications that have at least 10 5-star ratings. We use these pub_id's to find the average rating of all time from the Items_In_Orders table.

3. For all publications purchased in June 2022 that have been delivered, find the average time from the ordering date to the delivery date.

```
SELECT S.Pub_ID, Result.days
FROM Stocks_in_Bookstore AS S, Items_In_Orders AS I,(
    SELECT II0.Item_ID, AVG(DATEDIFF(day, O.Date_time, II0.Delivery_Date))
AS days
    FROM Orders O, Items_In_Orders II0, Item_Status status
    WHERE O.Order_ID=II0.Order_ID
    AND status.Item_ID=II0.Item_ID
    AND MONTH(O.Date_time)=06
    AND YEAR(O.Date_time)=2022
    AND status.State = 'Delivered'
    GROUP BY II0.Item_ID) AS Result
WHERE I.Item_ID = Result.Item_ID AND I.Stock_ID = S.Stock_ID
```

Query Output:

	123 Pub_ID ▾	123 days ▾	
1	4 ↗	123	
2	5 ↗	92	
3	1 ↗	124	
4	3 ↗	76	

Query Explanation: First, we find the item_id that is purchased in June 2022, and for these item_ids, we find the average time from the ordering date to the delivery date. Then for each of the item_id, we find their publication_id, and combine them with the days.

4. Let us define the “latency” of an employee by the average that he/she takes to process a complaint. Find the employee with the smallest latency.

```
SELECT E.Employee_ID, E.Name
FROM Employees AS E
WHERE E.Employee_ID IN (
    SELECT L.Employee_ID
    FROM (
        SELECT Employee_ID, AVG(DATEDIFF(day, Filed_Date_Time,
Handled_Date_Time)) AS Avg_Latency
        FROM Complaints
        GROUP BY Employee_ID
    ) AS L
    WHERE L.Avg_Latency IN (
        SELECT MIN(Avg_Latency)
        FROM (
            SELECT Employee_ID, AVG(DATEDIFF(day, Filed_Date_Time,
Handled_Date_Time)) AS Avg_Latency
            FROM Complaints
            GROUP BY Employee_ID
        ) AS Latency
    )
)
```

Query Output:

	Employee_ID	Name
1	1	John

Query Explanation: We create the table latency, which identifies the average date difference between the filled date time and the handled date time for each employee. This is used to identify the employee with the lowest latency

5. Produce a list that contains (i) all publications published by Nanyang Publisher Company, and (ii) for each of them, the number of bookstores on Ahamazon that sell them.

```
SELECT S.Pub_ID, COUNT(DISTINCT S.Bookstore_ID) AS Bookstore_Count
FROM Stocks_in_Bookstore S
WHERE S.Pub_ID IN (
    SELECT B.Pub_ID
    FROM Books B
    WHERE B.Publisher = 'Nanyang Publisher Company'
```

```

WHERE B.Pub_ID IN (
    SELECT P.Pub_ID
    FROM Publication P
    WHERE Publisher = 'Nanyang Publisher Company'
)
UNION
SELECT M.Pub_ID
FROM Magazines M
WHERE M.Pub_ID IN (
    SELECT P.Pub_ID
    FROM Publication P
    WHERE Publisher = 'Nanyang Publisher Company'
)
)
GROUP BY S.Pub_ID

```

Query Output:

	123 Pub_ID ▼	123 Bookstore_Count ▼	
1	5 ↗	3	
2	6 ↗	3	
3	7 ↗	3	
4	8 ↗	3	

Query Explanation: First, we choose the publications that are published by Nanyang Publisher Company from the tables Books and Magazines, then for each of the publications, we check how many bookstores are selling them.

6. Find bookstores that made the most revenue in August 2022.

```

SELECT Bookstore_ID, Sales
FROM (
    SELECT Bookstore_ID, SUM(II0.Item_Price * II0.Item_Quantity) AS Sales
    FROM Stocks_in_Bookstore SIB, Items_In_Orders II0, Orders O
    WHERE SIB.Stock_ID = II0.Stock_ID
    AND O.Order_ID = II0.Order_ID
    AND YEAR(O.Date_time) = 2022
    AND MONTH(O.Date_time) = 08
    GROUP BY SIB.Bookstore_ID
) AS B
WHERE B.Sales IN (
    SELECT MAX(Sales) as Max_Sales
    FROM (

```



```

Sales
        SELECT Bookstore_ID, SUM(II0.Item_Price * II0.Item_Quantity) AS
        FROM Stocks_in_Bookstore SIB, Items_In_Orders II0, Orders O
        WHERE SIB.Stock_ID = II0.Stock_ID
        AND O.Order_ID = II0.Order_ID
        AND YEAR(O.Date_time) = 2022
        AND MONTH(O.Date_time) = 08
        GROUP BY SIB.Bookstore_ID
    ) AS Bookstore_Sales
)

```

Query Output:

	123 Bookstore_ID	123 Sales	
1	2	1268.0000	

Query Explanation: Bookstore_Sales is used to contain the total sales for each bookstore in the month of August. We then identify the max of sales in this table. We identify the corresponding bookstore with the maximum sales.

7. For customers that made the most number of complaints, find the most expensive publication he/she has ever purchased.

```

SELECT M.Customer_ID, S.Pub_ID
FROM (
    SELECT I.Customer_ID, MAX(I.Item_Price) AS Max_Cost
    FROM Items_In_Orders AS I, Stocks_In_Bookstore AS S
    WHERE I.Customer_ID IN (
        SELECT CC.Customer_ID
        FROM (
            SELECT C.Customer_ID, COUNT(C.Complaint_ID) AS Total
            FROM Complaints C
            GROUP BY C.Customer_ID
        ) AS CC
    )
    WHERE CC.Total IN (
        SELECT MAX(Total) AS Max_Complaints
        FROM (
            SELECT C.Customer_ID, COUNT(C.Complaint_ID) AS Total
            FROM Complaints C
            GROUP BY C.Customer_ID
        ) AS Complaint_Count
    )
)

```

```

        AND I.Stock_ID = S.Stock_ID
        GROUP BY I.Customer_ID
    ) AS M, Items_In_Orders I, Stocks_In_Bookstore S
WHERE I.Stock_ID = S.Stock_ID
AND I.Customer_ID = M.Customer_ID
AND M.Max_Cost = I.Item_Price

```

Query Output:

	123 Customer_ID ▼	123 Pub_ID ▼	
1	11	5 ↗	

Query Explanation: For this query, the first step is to identify the customer who makes the most complaints. This is done from the Complaint table. We then find the purchase made by the customer, whose item price is equal to the item price.

8. Find publications that have never been purchased by any customer in July 2022, but are the top 3 most purchased publications in August 2022.

```

SELECT TOP 3 p.Pub_ID, COUNT(*) as num_purchased
FROM Publication p
INNER JOIN Stocks_in_Bookstore s ON p.Pub_ID = s.Pub_ID
INNER JOIN Items_In_Orders i ON s.Stock_ID = i.Stock_ID
INNER JOIN Orders o ON i.Order_ID = o.Order_ID
WHERE o.Date_time >= '2022-08-01' AND o.Date_time <= '2022-08-31'
AND p.Pub_ID NOT IN (
    SELECT DISTINCT p2.Pub_ID
    FROM Publication p2
    INNER JOIN Stocks_in_Bookstore s2 ON p2.Pub_ID = s2.Pub_ID
    INNER JOIN Items_In_Orders i2 ON s2.Stock_ID = i2.Stock_ID
    INNER JOIN Orders o2 ON i2.Order_ID = o2.Order_ID
    WHERE o2.Date_time >= '2022-07-01' AND o2.Date_time <= '2022-07-31'
)
GROUP BY p.Pub_ID
ORDER BY num_purchased DESC;

```

Query Output:

	123 Pub_ID ▼	123 num_purchased ▼
1	2	5
2	7	4
3	5	3

Query Explanation: For this query, we first identify those publications that were purchased in July 2022. We then identify those publications that were purchased in August but not in July. We then rank them based on the number of items that were purchased for that particular publication in August.

9. Find publications that are increasingly being purchased over at least 3 months.

```
WITH sales_by_month AS (
SELECT SIB.Pub_ID, DATEFROMPARTS(YEAR(O.Date_time), MONTH(O.DATE_time), 1) AS
Purchased_Year_Month, SUM(IIO.Item_Quantity) AS Total_Sales
FROM Stocks_in_Bookstore SIB, Items_In_Orders IIO, orders O
WHERE O.Order_ID = IIO.Order_ID
AND IIO.Stock_ID = SIB.Stock_ID
GROUP BY SIB.Pub_ID, DATEFROMPARTS(YEAR(O.Date_time), MONTH(O.DATE_time), 1)
)
SELECT DISTINCT S1.Pub_ID
FROM sales_by_month S1, sales_by_month S2, sales_by_month S3
WHERE S1.Total_Sales < S2.Total_Sales
AND S2.Total_Sales < S3.Total_Sales
AND S1.Pub_ID = S2.Pub_ID
AND S1.Pub_ID = S3.Pub_ID
AND DATEADD(month, 1, S1.Purchased_Year_Month) = S2.Purchased_Year_Month
AND DATEADD(month, 2, S1.Purchased_Year_Month) = S3.Purchased_Year_Month
```

Query Output:

	123 Pub_ID ▼
1	1
2	4
3	8

Query Explanation: sales_by_month is first created, which contains the sales for each publication in each month. We then compare three different copies of the same sales_by_month table to identify those publications that show an increasing trend in their sales over three consecutive months.

Printout of all the table records

1. Customer

	Customer_ID	Name
1	1	Merissa
2	2	Skippie
3	3	Nealy
4	4	Ferdinanda
5	5	Kylila
6	6	Nessy
7	7	Hanni
8	8	Hanson
9	9	Lin
10	10	Viki
11	11	Mendel
12	12	Agustin
13	13	Jessica
14	14	Myrah
15	15	Kendrick

2. Bookstore

	Bookstore_ID
1	1
2	2
3	3

3. Orders

	¹²³ Order_ID	¹ Date_time	^{asc} Shipping_address	¹²³ Customer_ID
1	1	2022-05-13 00:00:00.000	16 Ruskin Center	11 🔗
2	2	2022-06-06 00:00:00.000	6 Garrison Road	2 🔗
3	3	2022-01-09 00:00:00.000	426 Muir Center	5 🔗
4	4	2022-06-12 00:00:00.000	3 Lakewood Alley	1 🔗
5	5	2022-07-21 00:00:00.000	06 Scott Junction	11 🔗
6	6	2022-02-08 00:00:00.000	50264 Vera Terrace	4 🔗
7	7	2022-06-05 00:00:00.000	635 Grim Crossing	9 🔗
8	8	2022-02-19 00:00:00.000	4406 Muir Plaza	15 🔗
9	9	2022-07-19 00:00:00.000	8 Fairfield Avenue	9 🔗
10	10	2022-03-21 00:00:00.000	35350 Corry Plaza	12 🔗
11	11	2022-04-01 00:00:00.000	2285 Sycamore Point	14 🔗
12	12	2022-06-08 00:00:00.000	40 Bobwhite Parkway	13 🔗
13	13	2022-03-18 00:00:00.000	78506 Lake View Circle	11 🔗
14	14	2022-02-19 00:00:00.000	892 Forest Dale Center	12 🔗
15	15	2022-06-26 00:00:00.000	5100 Eagan Court	14 🔗
16	16	2022-02-24 00:00:00.000	5982 Oneill Plaza	3 🔗
17	17	2022-05-13 00:00:00.000	33435 Maple Wood Point	4 🔗
18	18	2022-06-12 00:00:00.000	2 Sunbrook Court	13 🔗
19	19	2022-08-13 00:00:00.000	8 Magdeline Terrace	3 🔗
20	20	2022-08-03 00:00:00.000	7 Park Meadow Point	11 🔗
21	21	2022-08-04 00:00:00.000	887 Del Mar Point	9 🔗
22	22	2022-08-03 00:00:00.000	941 Meadow Vale Circle	11 🔗
23	23	2022-08-13 00:00:00.000	03 Bowman Crossing	7 🔗
24	24	2022-08-09 00:00:00.000	07201 Artisan Park	4 🔗
25	25	2022-08-15 00:00:00.000	28006 Holy Cross Avenue	5 🔗
26	26	2022-08-10 00:00:00.000	3949 North Drive	9 🔗
27	27	2022-08-03 00:00:00.000	6 Manufacturers Junction	3 🔗
28	28	2022-08-13 00:00:00.000	167 Straubel Junction	8 🔗
29	29	2022-08-14 00:00:00.000	36882 Hoard Plaza	11 🔗
30	30	2022-08-08 00:00:00.000	3 Summit Drive	3 🔗
31	31	2022-08-14 00:00:00.000	57 Bayside Park	12 🔗
32	32	2022-08-13 00:00:00.000	05846 Fieldstone Plaza	7 🔗
33	33	2022-08-10 00:00:00.000	3 Truax Plaza	5 🔗
34	34	2022-08-08 00:00:00.000	898 Waywood Center	10 🔗
35	35	2022-08-10 00:00:00.000	50562 Garrison Center	4 🔗
36	36	2022-08-15 00:00:00.000	857 Nevada Junction	4 🔗
37	37	2022-08-01 00:00:00.000	4892 Utah Park	2 🔗
38	38	2022-08-10 00:00:00.000	9 Westport Terrace	8 🔗
39	39	2022-08-01 00:00:00.000	3 Crest Line Alley	12 🔗
40	40	2022-08-07 00:00:00.000	5 Lakewood Gardens Drive	10 🔗
41	41	2022-08-08 00:00:00.000	01 Doe Crossing Junction	9 🔗
42	42	2022-08-12 00:00:00.000	7 Macpherson Lane	7 🔗
43	43	2022-08-12 00:00:00.000	51658 Debs Court	2 🔗
44	44	2022-08-03 00:00:00.000	90284 Stoughton Place	8 🔗
45	45	2022-08-13 00:00:00.000	55625 Gulseth Park	13 🔗
46	46	2022-08-16 00:00:00.000	2 Westend Parkway	2 🔗

47	47	2022-08-02 00:00:00.000	7884 Bashford Park	11	↗
48	48	2022-08-13 00:00:00.000	0185 International Crossing	5	↗
49	49	2022-08-13 00:00:00.000	2 Annamark Court	8	↗
50	50	2022-08-12 00:00:00.000	70 Dottie Trail	15	↗
51	51	2022-08-01 00:00:00.000	6 Ronald Regan Point	15	↗
52	52	2022-08-04 00:00:00.000	0524 8th Way	1	↗
53	53	2022-08-10 00:00:00.000	2926 Helena Plaza	12	↗
54	54	2022-08-12 00:00:00.000	783 Butterfield Way	7	↗
55	55	2022-08-13 00:00:00.000	6079 Elmside Alley	9	↗
56	56	2022-08-07 00:00:00.000	54 Farragut Plaza	6	↗
57	57	2022-08-08 00:00:00.000	615 High Crossing Road	9	↗
58	58	2022-08-03 00:00:00.000	282 Carioca Crossing	2	↗
59	59	2022-08-15 00:00:00.000	55 Mariners Cove Center	12	↗
60	60	2022-08-09 00:00:00.000	48 Fieldstone Trail	2	↗
61	61	2022-08-05 00:00:00.000	40 Moland Alley	4	↗
62	62	2022-08-06 00:00:00.000	816 Mockingbird Trail	11	↗
63	63	2022-08-09 00:00:00.000	3 Tomscot Court	8	↗
64	64	2022-08-05 00:00:00.000	6970 Iowa Parkway	12	↗
65	65	2022-08-02 00:00:00.000	42273 Cottonwood Road	1	↗
66	66	2022-08-03 00:00:00.000	33 Forster Circle	2	↗
67	67	2022-08-06 00:00:00.000	41 Birchwood Junction	3	↗
68	68	2022-08-11 00:00:00.000	62516 Jenna Street	3	↗
69	69	2022-08-05 00:00:00.000	99 Hallows Parkway	11	↗
70	70	2022-08-04 00:00:00.000	2 International Street	9	↗
71	71	2022-08-15 00:00:00.000	5 Annamark Way	13	↗
72	72	2022-08-03 00:00:00.000	311 Lerdahl Trail	12	↗
73	73	2022-08-06 00:00:00.000	5 Coolidge Plaza	15	↗
74	74	2022-08-09 00:00:00.000	7921 American Court	13	↗
75	75	2022-08-15 00:00:00.000	118 Birchwood Plaza	5	↗

4. Stocks_in_Bookstore

	Stock_ID	Stock_Price	Stock_Qty	Pub_ID	Bookstore_ID
1	1	37.00	100	1	1
2	2	24.00	120	2	1
3	3	30.00	110	3	1
4	4	32.00	95	4	1
5	5	40.00	70	5	1
6	6	40.00	70	6	1
7	7	10.00	50	7	1
8	8	12.00	50	8	1
9	9	30.00	90	1	2
10	10	26.00	85	2	2
11	11	27.00	100	3	2
12	12	28.00	80	4	2
13	13	35.00	60	5	2
14	14	38.00	40	6	2
15	15	10.00	50	7	2
16	16	12.00	50	8	2
17	17	35.00	130	1	3
18	18	25.00	100	2	3
19	19	34.00	110	3	3
20	20	35.00	100	4	3
21	21	42.00	70	5	3
22	22	36.00	50	6	3
23	23	9.00	50	7	3
24	24	10.00	50	8	3

5. Items_In_Orders

	¹²¹ Item_ID	¹²² Order_ID	¹²³ Stock	¹²² Item_Price	¹²² Item_Quant	¹²³ Delivery_Date	¹²² Custom	¹²³ Date_Time	¹²⁴ Comment	¹²² Rating
1		1	5	30.0000	3	2022-10-27	11	2022-05-13 00:00:00.000	visualize strategic technologies	5
2		2	12	35.0000	2	2022-10-07	2	2022-06-06 00:00:00.000	incubate dot-com mindshare	5
3		3	4	32.0000	4	2022-09-12	5	2022-01-09 00:00:00.000	cultivate cross-platform systems	1
4		4	6	26.0000	1	2022-09-09	1	2022-06-12 00:00:00.000	evolve back-end methodologies	5
5		5	9	35.0000	1	2022-09-04	11	2022-07-21 00:00:00.000	maximize plug-and-play schemas	4
6		6	10	25.0000	4	2022-11-01	4	2022-02-08 00:00:00.000	visualize customized interfaces	5
7		7	5	30.0000	4	2022-09-05	9	2022-06-05 00:00:00.000	generate clicks-and-mortar bandwidth	4
8		8	4	32.0000	3	2022-09-27	15	2022-02-19 00:00:00.000	reintermediate 24/365 action-items	4
9		9	8	28.0000	5	2022-10-14	9	2022-07-19 00:00:00.000	matrix 24/7 e-commerce	2
10		10	8	28.0000	3	2022-10-11	12	2022-03-21 00:00:00.000	enhance next-generation partnerships	2
11		11	6	26.0000	1	2022-10-24	14	2022-04-01 00:00:00.000	facilitate virtual mindshare	4
12		12	8	28.0000	1	2022-10-05	13	2022-06-08 00:00:00.000	redefine bricks-and-clicks e-services	3
13		13	5	30.0000	4	2022-10-18	11	2022-03-18 00:00:00.000	aggregate frictionless ROI	5
14		14	1	37.0000	2	2022-10-23	12	2022-02-19 00:00:00.000	incubate back-end schemas	2
15		15	4	32.0000	1	2022-09-01	14	2022-06-26 00:00:00.000	enhance compelling relationships	5
16		16	6	26.0000	2	2022-09-18	3	2022-02-24 00:00:00.000	e-enable collaborative solutions	5
17		17	7	27.0000	4	2022-09-18	4	2022-05-13 00:00:00.000	facilitate B2B eyeballs	2
18		18	8	28.0000	2	2022-10-19	13	2022-06-12 00:00:00.000	strategize turn-key convergence	5
19		19	12	25.0000	1	2022-09-28	3	2022-08-13 00:00:00.000	User-friendly multimedia database	5
20		20	3	30.0000	1	2022-10-02	11	2022-08-03 00:00:00.000	Down-sized fresh-thinking open system	5
21		21	11	35.0000	2	2022-10-08	9	2022-08-04 00:00:00.000	Multi-layered context-sensitive application	5
22		22	5	40.0000	4	2022-09-14	11	2022-08-03 00:00:00.000	Function-based 5th generation parallelism	5
23		23	3	30.0000	3	2022-09-05	7	2022-08-13 00:00:00.000	Profit-focused dedicated local area network	5
24		24	3	30.0000	4	2022-09-11	4	2022-08-09 00:00:00.000	Decentralized system-worthy local area network	5
25		25	11	35.0000	2	2022-10-01	5	2022-08-15 00:00:00.000	Universal secondary time-frame	5
26		26	10	38.0000	2	2022-09-10	9	2022-08-10 00:00:00.000	Configurable tertiary task-force	5
27		27	8	27.0000	10	2022-09-01	3	2022-08-03 00:00:00.000	Visionary zero tolerance Graphic Interface	5
28		28	5	40.0000	1	2022-09-05	8	2022-08-13 00:00:00.000	Enterprise-wide full-range matrix	5
29		29	7	26.0000	1	2022-10-09	11	2022-08-14 00:00:00.000	Adaptive user-facing benchmark	5
30		30	12	25.0000	4	2022-10-11	3	2022-08-08 00:00:00.000	Polarised bi-directional interface	5
31		31	15	42.0000	3	2022-09-08	12	2022-08-14 00:00:00.000	Distributed human-resource functionalities	5
32		32	14	35.0000	1	2022-10-04	7	2022-08-13 00:00:00.000	Intuitive optimizing hardware	5
33		33	2	24.0000	2	2022-09-29	5	2022-08-10 00:00:00.000	Persistent explicit neural-net	5
34		34	2	24.0000	2	2022-10-07	10	2022-08-08 00:00:00.000	Upgradable bi-directional collaboration	5
35		35	1	37.0000	2	2022-09-11	4	2022-08-10 00:00:00.000	Organized leading edge monitoring	5
36		36	7	26.0000	2	2022-10-04	4	2022-08-15 00:00:00.000	Right-sized mobile instruction set	5
37		37	1	37.0000	1	2022-10-04	2	2022-08-01 00:00:00.000	Advanced hybrid moderator	5
38		38	12	25.0000	2	2022-10-07	8	2022-08-10 00:00:00.000	Quality-focused eco-centric local area network	5
39		39	2	24.0000	4	2022-10-11	12	2022-08-01 00:00:00.000	Centralized bifurcated service-desk	5
40		40	14	35.0000	4	2022-09-22	10	2022-08-07 00:00:00.000	Customer-focused dedicated software	5
41		41	2	24.0000	3	2022-09-29	9	2022-08-08 00:00:00.000	Universal zero defect initiative	5
42		42	15	42.0000	1	2022-09-09	7	2022-08-12 00:00:00.000	Profit-focused web-enabled website	5
43		43	13	34.0000	4	2022-09-26	2	2022-08-12 00:00:00.000	Networked upward-trending architecture	5
44		44	9	28.0000	3	2022-09-05	8	2022-08-03 00:00:00.000	Compatible secondary emulation	5
45		45	12	25.0000	3	2022-09-15	13	2022-08-13 00:00:00.000	Managed intermediate algorithm	5
46		46	9	12.0000	4	2022-10-07	3	2022-08-16 00:00:00.000	Horizontal client-driven standardization	5
47		47	9	12.0000	3	2022-10-04	3	2022-08-02 00:00:00.000	Optional homogeneous neural-net	5
48		48	9	12.0000	1	2022-10-02	1	2022-08-13 00:00:00.000	Enterprise-wide contextually-based policy	5
49		49	9	12.0000	1	2022-09-11	1	2022-08-13 00:00:00.000	Balanced context-sensitive interface	5
50		50	17	12.0000	1	2022-09-29	1	2022-08-12 00:00:00.000	Polarised contextually-based focus group	5
51		51	17	12.0000	1	2022-09-10	1	2022-08-01 00:00:00.000	Devolved maximized core	5
52		52	17	12.0000	3	2022-10-08	1	2022-08-04 00:00:00.000	Organic interactive infrastructure	5
53		53	17	12.0000	1	2022-09-18	3	2022-08-10 00:00:00.000	Focused even-keeled hub	5
54		54	17	12.0000	1	2022-09-04	1	2022-08-12 00:00:00.000	Visionary heuristic customer loyalty	5
55		55	9	12.0000	3	2022-09-23	1	2022-08-13 00:00:00.000	Business-focused secondary hardware	5
56		56	12	27.0000	2	2022-09-05	2	2022-08-07 00:00:00.000	Cross-platform fresh-thinking structure	5
57		57	12	27.0000	1	2022-09-19	1	2022-08-08 00:00:00.000	Multi-channelled static conglomeration	5
58		58	12	27.0000	4	2022-10-01	2	2022-08-03 00:00:00.000	Robust logistical approach	5
59		59	12	27.0000	1	2022-09-11	2	2022-08-15 00:00:00.000	Vision-oriented bi-directional flexibility	5
60		60	20	34.0000	4	2022-09-01	3	2022-08-09 00:00:00.000	Ameliorated holistic workforce	5
61		61	20	34.0000	2	2022-09-06	1	2022-08-05 00:00:00.000	Synergistic background task-force	5
62		62	20	34.0000	1	2022-09-18	3	2022-08-06 00:00:00.000	Persistent eco-centric knowledge base	5
63		63	20	34.0000	4	2022-10-03	3	2022-08-09 00:00:00.000	Enhanced multi-tasking capability	5
64		64	12	27.0000	3	2022-09-05	2	2022-08-05 00:00:00.000	Operative bi-directional paradigm	5
65		65	20	34.0000	1	2022-09-02	2	2022-08-02 00:00:00.000	Programmable multi-state portal	5
66		66	19	25.0000	1	2022-09-07	3	2022-08-03 00:00:00.000	Polarised needs-based protocol	5
67		67	19	25.0000	2	2022-09-16	1	2022-08-06 00:00:00.000	Visionary client-server local area network	5
68		68	11	26.0000	4	2022-09-05	2	2022-08-11 00:00:00.000	Ameliorated clear-thinking toolset	5
69		69	11	26.0000	1	2022-09-26	3	2022-08-05 00:00:00.000	Re-contextualized mobile collaboration	5
70		70	19	25.0000	4	2022-09-26	3	2022-08-04 00:00:00.000	Right-sized empowering groupware	5
71		71	11	26.0000	1	2022-10-05	3	2022-08-15 00:00:00.000	User-centric multimedia encryption	5
72		72	11	26.0000	1	2022-09-09	1	2022-08-03 00:00:00.000	Customizable global function	5
73		73	19	25.0000	4	2022-10-10	3	2022-08-06 00:00:00.000	Pre-emptive homogeneous matrices	5
74		74	11	26.0000	3	2022-10-02	3	2022-08-09 00:00:00.000	Optimized demand-driven infrastructure	5
75		75	19	25.0000	4	2022-09-09	2	2022-08-15 00:00:00.000	Public-key systemic functionalities	5

6. Item_Status

	Date	State	Item_ID
1	2022-08-17	being processed	1
2	2022-08-17	being processed	2
3	2022-08-17	being processed	8
4	2022-08-17	being processed	13
5	2022-08-17	being processed	14
6	2022-08-17	being processed	17
7	2022-08-17	being processed	19
8	2022-08-17	being processed	24
9	2022-08-17	being processed	26
10	2022-08-17	being processed	27
11	2022-08-17	being processed	41
12	2022-08-17	being processed	42
13	2022-08-17	being processed	43
14	2022-08-17	being processed	44
15	2022-08-17	being processed	50
16	2022-08-17	being processed	51
17	2022-08-17	being processed	53
18	2022-08-17	being processed	60
19	2022-08-17	being processed	62
20	2022-08-17	being processed	64
21	2022-08-17	being processed	66
22	2022-08-17	being processed	68
23	2022-08-17	being processed	69
24	2022-08-17	being processed	71
25	2022-08-17	being processed	72
26	2022-08-17	being processed	75
27	2022-08-18	being processed	7
28	2022-08-18	being processed	10
29	2022-08-18	being processed	16
30	2022-08-18	being processed	20
31	2022-08-18	being processed	21
32	2022-08-18	being processed	22
33	2022-08-18	being processed	37
34	2022-08-18	being processed	38
35	2022-08-18	being processed	39
36	2022-08-18	being processed	40
37	2022-08-18	being processed	46
38	2022-08-18	being processed	49
39	2022-08-18	being processed	57
40	2022-08-18	being processed	58
41	2022-08-18	being processed	59
42	2022-08-18	being processed	67
43	2022-08-19	being processed	9
44	2022-08-19	being processed	12
45	2022-08-19	being processed	15
46	2022-08-19	being processed	23

47	2022-08-19	being processed	25 🔗
48	2022-08-19	being processed	28 🔗
49	2022-08-19	being processed	31 🔗
50	2022-08-19	being processed	35 🔗
51	2022-08-19	being processed	36 🔗
52	2022-08-19	being processed	47 🔗
53	2022-08-19	being processed	48 🔗
54	2022-08-19	being processed	54 🔗
55	2022-08-19	being processed	61 🔗
56	2022-08-19	being processed	65 🔗
57	2022-08-19	being processed	70 🔗
58	2022-08-20	being processed	3 🔗
59	2022-08-20	being processed	4 🔗
60	2022-08-20	being processed	5 🔗
61	2022-08-20	being processed	6 🔗
62	2022-08-20	being processed	11 🔗
63	2022-08-20	being processed	18 🔗
64	2022-08-20	being processed	29 🔗
65	2022-08-20	being processed	30 🔗
66	2022-08-20	being processed	32 🔗
67	2022-08-20	being processed	33 🔗
68	2022-08-20	being processed	34 🔗
69	2022-08-20	being processed	45 🔗
70	2022-08-20	being processed	52 🔗
71	2022-08-20	being processed	55 🔗
72	2022-08-20	being processed	56 🔗
73	2022-08-20	being processed	63 🔗
74	2022-08-20	being processed	73 🔗
75	2022-08-20	being processed	74 🔗
76	2022-08-22	shipped	8 🔗
77	2022-08-22	shipped	22 🔗
78	2022-08-22	shipped	39 🔗
79	2022-08-22	shipped	50 🔗
80	2022-08-22	shipped	52 🔗
81	2022-08-22	shipped	75 🔗
82	2022-08-23	shipped	2 🔗
83	2022-08-23	shipped	10 🔗
84	2022-08-23	shipped	11 🔗
85	2022-08-23	shipped	17 🔗
86	2022-08-23	shipped	23 🔗
87	2022-08-23	shipped	25 🔗
88	2022-08-23	shipped	26 🔗
89	2022-08-23	shipped	39 🔗
90	2022-08-23	shipped	42 🔗
91	2022-08-23	shipped	55 🔗
92	2022-08-23	shipped	70 🔗

93	2022-08-24	shipped	6 ↗
94	2022-08-24	shipped	24 ↗
95	2022-08-24	shipped	35 ↗
96	2022-08-24	shipped	41 ↗
97	2022-08-24	shipped	52 ↗
98	2022-08-24	shipped	65 ↗
99	2022-08-24	shipped	74 ↗
100	2022-08-25	shipped	7 ↗
101	2022-08-25	shipped	28 ↗
102	2022-08-25	shipped	32 ↗
103	2022-08-25	shipped	40 ↗
104	2022-08-25	shipped	46 ↗
105	2022-08-25	shipped	47 ↗
106	2022-08-25	shipped	48 ↗
107	2022-08-25	shipped	49 ↗
108	2022-08-25	shipped	57 ↗
109	2022-08-25	shipped	58 ↗
110	2022-08-25	shipped	64 ↗
111	2022-09-02	delivered	65 ↗
112	2022-09-03	delivered	55 ↗
113	2022-09-05	delivered	7 ↗
114	2022-09-05	delivered	23 ↗
115	2022-09-05	delivered	64 ↗
116	2022-09-09	delivered	75 ↗
117	2022-09-11	delivered	24 ↗
118	2022-09-11	delivered	35 ↗
119	2022-09-12	shipped	1 ↗
120	2022-09-14	delivered	22 ↗
121	2022-09-18	delivered	17 ↗
122	2022-09-22	delivered	40 ↗
123	2022-09-26	delivered	70 ↗
124	2022-09-27	delivered	8 ↗
125	2022-10-01	delivered	25 ↗
126	2022-10-01	delivered	58 ↗
127	2022-10-02	delivered	48 ↗
128	2022-10-04	delivered	32 ↗
129	2022-10-04	delivered	47 ↗
130	2022-10-07	delivered	2 ↗
131	2022-10-07	delivered	46 ↗
132	2022-10-08	delivered	52 ↗
133	2022-10-11	delivered	10 ↗
134	2022-10-11	delivered	39 ↗
135	2022-10-27	delivered	1 ↗
136	2022-11-01	delivered	6 ↗

7. Employees

	Employee_ID	Name	Salary
1	1	John	500.00
2	2	Rick	500.00
3	3	John	500.00
4	4	Rick	500.00

8. Complaints

	Complaint_ID	Employee_ID	Customer_ID	Text	Filed_Date_Time	Handled_Date_Time
1	1	1	13	This book was not delivered to me	2022-08-01 00:00:00.000	2022-08-03 00:00:00.000
2	2	2	3	This order was not delivered to me	2022-08-10 00:00:00.000	2022-08-05 00:00:00.000
3	3	1	14	This book was not delivered to me	2022-08-04 00:00:00.000	2022-08-05 00:00:00.000
4	4	2	11	This order was not delivered to me	2022-08-04 00:00:00.000	2022-08-25 00:00:00.000
5	5	1	13	This book was not delivered to me	2022-08-01 00:00:00.000	2022-08-03 00:00:00.000
6	6	2	3	This order was not delivered to me	2022-08-10 00:00:00.000	2022-08-20 00:00:00.000
7	7	1	14	This book was not delivered to me	2022-08-04 00:00:00.000	2022-08-05 00:00:00.000
8	8	2	11	This order was not delivered to me	2022-08-21 00:00:00.000	2022-08-25 00:00:00.000
9	9	1	11	This book was not delivered to me	2022-08-01 00:00:00.000	2022-08-03 00:00:00.000

9. Complaints_On_Order

	Complaint_On_Order_ID	Order_ID
1	2	16
2	4	1

10. Complaints_On_Bookstore

	Complaint_On_Order_ID	Bookstore_ID
1	1	2
2	3	1
3	5	2

11. Complaint_Status

	Date	State	Complaint_ID
1	2022-08-01	pending	1
2	2022-08-02	being handled	1
3	2022-08-03	addressed	1

12. Price_History

	Stock_ID	Price	Start_date	End_date
1	1	30.00	2022-08-01	2022-08-03
2	1	30.00	2022-08-03	2022-08-05

13. Publication

	Pub_ID	Publisher	Pub_Year
1	1	Bloomsbury	2011
2	2	ABRAMS	2007
3	3	Little, Brown and Company	2006
4	4	Disney-Lucasfilm Press	1977
5	5	Nanyang Publisher Company	2009
6	6	Nanyang Publisher Company	2010
7	7	Nanyang Publisher Company	2021
8	8	Nanyang Publisher Company	2021

14. Books

	Pub_ID	Title
1	1	Harry Porter Finale
2	2	Diary of a Wimpy Kid
3	3	Twilight
4	4	Star Wars
5	5	Essays on Heritage
6	6	Essays on Technology

15. Magazines

	Pub_ID	Issue	Title
1	7	10	Nanyang Fishing Monthly
2	8	9	Nanyang Walker