

SC2207 Introduction to Databases: Lab 5 Report



Lab Group A54

Group No.: 3

Lab Supervisor: Prof Zhao Jun

Teaching Assistant: Sun You Chen

Group Members	
Name	Matriculation No.
Jadhav Chaitanya Dhananjay	U2121503D
Palaniswamy Tarun Kumar	U2121932F
Muhammad Irfan bin Ameer Hamzah	U2122409L
Rishabh Alexander John	U2121475H
Zhang Xiaoyang	U2121950J

Table of Contents

SQL DDL Commands for Table Creation:	2
SQL Commands for Queries	11
1. Query 1	11
2. Query 2	12
3. Query 3	13
4. Query 4	14
5. Query 5	15
6. Query 6	16
7. Query 7	17
8. Query 8	19
9. Query 9	21
Printout of Table Records	24

SQL DDL Commands for Table Creation:

1. Publication

```
create table dbo.publication
(
    pubid      varchar(50) not null
    primary key,
    publisher  varchar(100) not null,
    year       int          not null
)
go
```

The Publication table stores the Publication ID, Publisher and Year of Publishing for all Amazon Publications.

2. Books

```
create table dbo.books
(
    pubid varchar(50) not null
    primary key
    references dbo.publication,
    title varchar(50) not null
)
go
```

The Books table stores the publication ID, linking each Book to a Publication in the Publication Table. It also contains the title of the book.

3. Magazines

```
create table dbo.magazines
(
    pubid varchar(50) not null
    primary key
    references dbo.publication,
    title varchar(100) not null,
    issue int not null
)
go
```

The Magazines table stores the publication ID, linking each Magazine to a Publication in the Publication table. It also contains the title and issue number of the magazine.

4. Bookstore

```
create table dbo.bookstore
(
    bookstoreid varchar(50) not null
    primary key
)
go
```

The Bookstore table stores the bookstoreid of all bookstores. Each bookstore has a unique bookstoreid.

5. Stocks in Bookstores

```
create table dbo.stocks_in_bookstores
(
    stockid    varchar(50) not null
    primary key,
    stock_price float      not null,
    stock_qty   int         not null,
    bookstoreid varchar(50) not null
    references dbo.bookstore,
    pubid       varchar(50) not null
    references dbo.publication
)
go
```

The stocks in bookstore table stores the stock id, stock price, stock quantity, bookstore id (foreign key from bookstore table), and pubid (foreign key from the the publication table).

6. Price History

```
create table dbo.price_history
(
    stockid    varchar(50) not null
    references dbo.stocks_in_bookstores,
    price       float      not null,
    start_date  date        not null,
    end_date    date        not null,
    primary key (stockid, start_date, end_date)
)
go
```

The price history table stores the stockid (foreign key from stocks in bookstores), price, start date and end date. It keeps track of the price of that particular stock within the stipulated time frame

7. Customers

```
create table dbo.customers
(
    cid varchar(50) not null
    primary key,
    name varchar(100) not null
)
go
```

The customer table stores the customer id and customer name of all Amazon customers.

8. Orders

```
create table dbo.orders
(
    orderid          varchar(50) not null
    primary key,
    date_time        datetime not null,
    shipping_address varchar(500) not null,
    cid              varchar(50) not null
    references dbo.customers
)
go
```

The orders table stores the unique order id, the date_time of when the order was placed, the shipping address, and id of the customer who placed the order (foreign key from customers).

9. Items in Orders 1

```
create table dbo.items_in_orders_1
(
    itemid      varchar(50) not null
    primary key,
    item_price  float,
    item_qty    int          not null,
    orderid     varchar(50) not null
    references  dbo.orders,
    delivery_date date,
    stockid     varchar(50) not null
    references  dbo.stocks_in_bookstores
)
go

CREATE TRIGGER update_item_price
ON dbo.items_in_orders_1
AFTER INSERT, UPDATE
AS
BEGIN
    UPDATE iio1
    SET iio1.item_price = ph.price
    FROM dbo.items_in_orders_1 AS iio1
    JOIN dbo.price_history AS ph ON iio1.stockid = ph.stockid
    WHERE iio1.delivery_date BETWEEN ph.start_date AND ph.end_date;
END;
go
```

The items_in_orders_1 stores the primary details of the item such as its id, price, quantity and matches it to the respective order and stock id. The delivery date is also included.

The item price column is updated automatically via an SQL update trigger that looks up price_history and finds the price of the item at the time of delivery. For this reason, we let the item_price remain null when initialized.

10. Items in Orders 2

```
create table dbo.items_in_orders_2
(
    itemid          varchar(50) not null
    references dbo.items_in_orders_1,
    comment         varchar(100),
    rating          float,
    feedback_date_time datetime
)
go
```

The items_in_orders_2 table stores the item ID (foreign key from items_in_orders_1), comments, rating and date and time of the feedback.

11. Items In Orders 3

```
create table dbo.items_in_orders_3
(
    orderid varchar(50) not null
    references dbo.orders,
    stockid varchar(50) not null
    references dbo.stocks_in_bookstores,
    itemid  varchar(50) not null
    references dbo.items_in_orders_1,
    primary key (orderid, stockid)
)
go
```

The items_in_orders_3 table stores the orderid (foreign key from orders), stockid (foreign key from stocks_in_bookstores), and itemid (foreign key from items_in_orders_1). It matches an item ID to its stockid and orderid.

12. Items in Orders 4

```
create table dbo.items_in_orders_4
(
   orderid varchar(50) not null
    primary key
    references dbo.orders,
    cid varchar(50) not null
    references dbo.customers
)
go
```

The items_in_orders_4 table stores the orderid (foreign key from orders) and customerid (foreign key from customers).

13. Order Status

```
create table dbo.orderstatus
(
    itemid      varchar(50) not null
    references dbo.items_in_orders_1,
    status_date date      not null,
    state       char(50)   not null
    constraint CK_orderstatus_state
        check ([state] = 'returned' OR [state] = 'delivered' OR
[state] = 'shipped' OR [state] = 'being processed'),
    primary key (itemid, status_date)
)
go
```

The order status table contains the itemid (foreign key from items_in_orders_1), date of status, and state. We use a constraint to ensure that the state of an order is only within certain permissible values.

14. Complaints

```
create table dbo.complaints
(
    complaintsid    varchar(50) not null
    primary key,
    cid             varchar(50) not null
    references dbo.customers,
    eid             varchar(50)
    references dbo.employees,
    handled_date_time datetime,
    complain_text   text        not null,
    filled_date_time datetime not null
)
go
```

The complaints table contains the complaintsid, customerid (foreign key from customers), employeeid (foreign key from employees), date the complaint was handled by the employee, complaint text and date of filing complaint.

15. Employees

```
create table dbo.employees
(
    eid varchar(50) not null
    primary key,
    name varchar(100) not null,
    salary int not null
    constraint CK_employees_salary
        check ([salary] > 0)
)
go
```

Employees table that stores details of the name and salary of the employee as well as a unique ID for identification during complaint handling.

16. Complaints on Bookstore

```
create table dbo.complaints_on_bookstore
(
    complaintsid varchar(50) not null
    primary key
    references dbo.complaints,
    bookstoreid  varchar(50)
    references dbo.bookstore
)
go
```

The complaints_on_bookstore table contains the complaintsid (foreign key from complaints) and bookstoreid (foreign key from bookstore).

17. Complaints on Orders

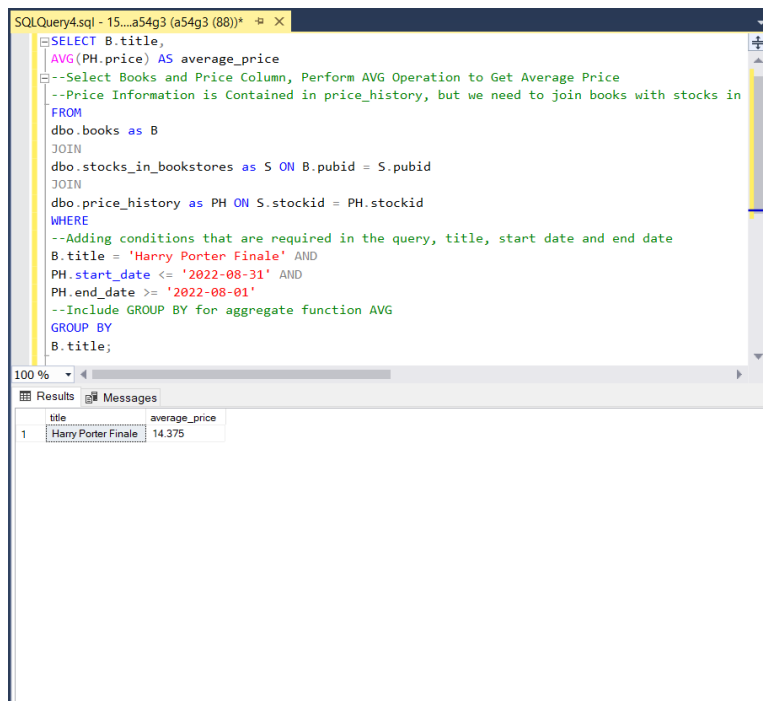
```
create table dbo.complaints_on_orders
(
    complaintsid varchar(50) not null
    primary key
    references dbo.complaints,
    orderid      varchar(50)
    references dbo.orders
)
go
```

The complaints_on_bookstore table contains the complaintsid (foreign key from complaints) and orderid (foreign key from orders).

SQL Commands for Queries

1. Query 1

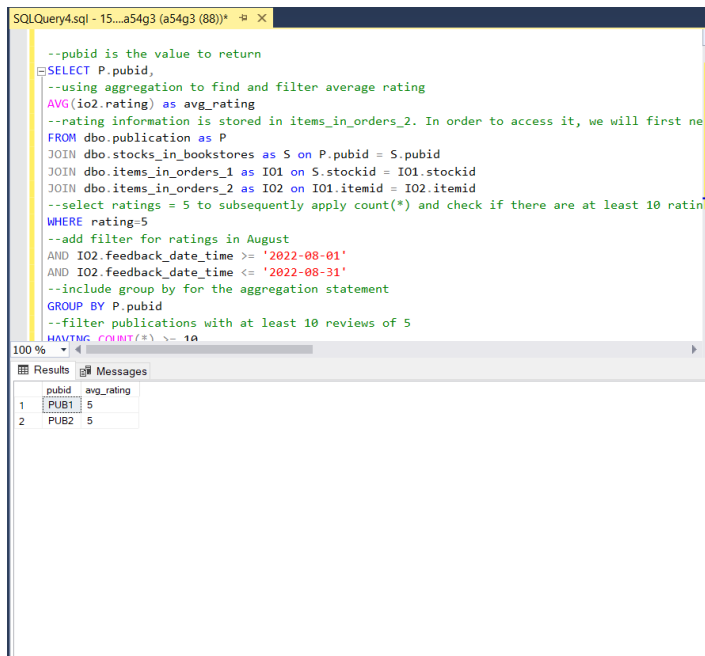
```
SELECT B.title,
AVG(PH.price) AS average_price
FROM
dbo.books as B
JOIN
dbo.stocks_in_bookstores as S ON B.pubid = S.pubid
JOIN
dbo.price_history as PH ON S.stockid = PH.stockid
WHERE
B.title = 'Harry Porter Finale' AND
PH.start_date <= '2022-08-31' AND
PH.end_date >= '2022-08-01'
GROUP BY
B.title;
```



The query joins the books, stocks_in_bookstores, and price_history tables to retrieve the relevant price information. The AVG function is used to calculate the average price of the book based on the prices in the price_history table that fall within the specified date range. The results are grouped by the title column from the books table.

2. Query 2

```
SELECT P.pubid,  
AVG(io2.rating) as avg_rating  
FROM dbo.publication as P  
JOIN dbo.stocks_in_bookstores as S on P.pubid = S.pubid  
JOIN dbo.items_in_orders_1 as IO1 on S.stockid = IO1.stockid  
JOIN dbo.items_in_orders_2 as IO2 on IO1.itemid = IO2.itemid  
WHERE rating=5  
AND IO2.feedback_date_time >= '2022-08-01'  
AND IO2.feedback_date_time <= '2022-08-31'  
GROUP BY P.pubid  
HAVING COUNT(*) >= 10  
ORDER BY avg_rating DESC;
```



The screenshot shows a SQL Server Enterprise Manager interface. The top pane displays a SQL query with extensive comments explaining each step: selecting the publication ID, joining tables to find ratings of 5 in August 2022, grouping by publication ID, and filtering for at least 10 reviews. The bottom pane shows the results of the query in a table with two columns: 'pubid' and 'avg_rating'. Two rows are visible, both with an average rating of 5.

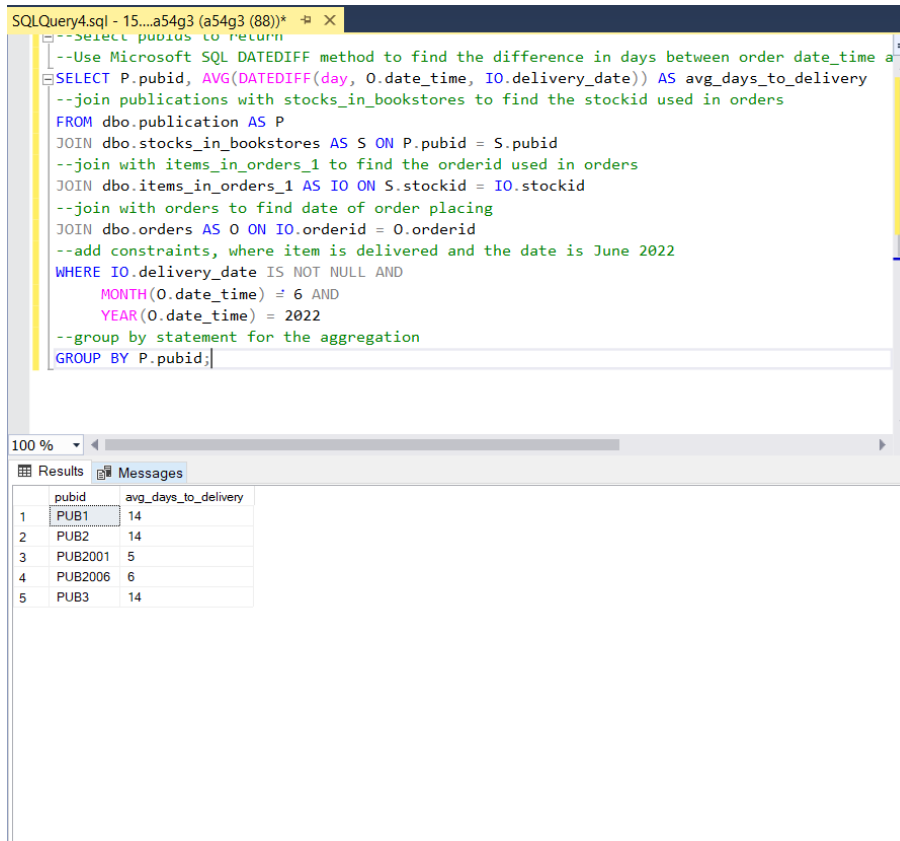
```
--pubid is the value to return  
SELECT P.pubid,  
--using aggregation to find and filter average rating  
AVG(io2.rating) as avg_rating  
--rating information is stored in items_in_orders_2. In order to access it, we will first ne  
FROM dbo.publication as P  
JOIN dbo.stocks_in_bookstores as S on P.pubid = S.pubid  
JOIN dbo.items_in_orders_1 as IO1 on S.stockid = IO1.stockid  
JOIN dbo.items_in_orders_2 as IO2 on IO1.itemid = IO2.itemid  
--select ratings = 5 to subsequently apply count(*) and check if there are at least 10 ratin  
WHERE rating=5  
--add filter for ratings in August  
AND IO2.feedback_date_time >= '2022-08-01'  
AND IO2.feedback_date_time <= '2022-08-31'  
--include group by for the aggregation statement  
GROUP BY P.pubid  
--filter publications with at least 10 reviews of 5  
HAVING COUNT(*) >= 10
```

pubid	avg_rating
PUB1	5
PUB2	5

The query joins the publication, stocks_in_bookstores, items_in_orders_1, and items_in_orders_2 tables to retrieve the relevant rating information. The AVG function is used to calculate the average rating of each publication based on the ratings in the items_in_orders_2 table that fall within the specified date range and have a value of 5. The results are grouped by the pubid column from the publication table and filtered using a HAVING clause to include only publications with at least 10 ratings. The final result is sorted in descending order by the average rating.

3. Query 3

```
SELECT P.pubid, AVG(DATEDIFF(day, O.date_time, IO.delivery_date)) AS
avg_days_to_delivery
FROM dbo.publication AS P
JOIN dbo.stocks_in_bookstores AS S ON P.pubid = S.pubid
JOIN dbo.items_in_orders_1 AS IO ON S.stockid = IO.stockid
JOIN dbo.orders AS O ON IO.orderid = O.orderid
WHERE IO.delivery_date IS NOT NULL AND
      MONTH(O.date_time) = 6 AND
      YEAR(O.date_time) = 2022
GROUP BY P.pubid;
```



The screenshot shows a SQL Server query window with the following SQL code:

```
--select pubids to return
--Use Microsoft SQL DATEDIFF method to find the difference in days between order date_time a
SELECT P.pubid, AVG(DATEDIFF(day, O.date_time, IO.delivery_date)) AS avg_days_to_delivery
--join publications with stocks_in_bookstores to find the stockid used in orders
FROM dbo.publication AS P
JOIN dbo.stocks_in_bookstores AS S ON P.pubid = S.pubid
--join with items_in_orders_1 to find the orderid used in orders
JOIN dbo.items_in_orders_1 AS IO ON S.stockid = IO.stockid
--join with orders to find date of order placing
JOIN dbo.orders AS O ON IO.orderid = O.orderid
--add constraints, where item is delivered and the date is June 2022
WHERE IO.delivery_date IS NOT NULL AND
      MONTH(O.date_time) = 6 AND
      YEAR(O.date_time) = 2022
--group by statement for the aggregation
GROUP BY P.pubid;
```

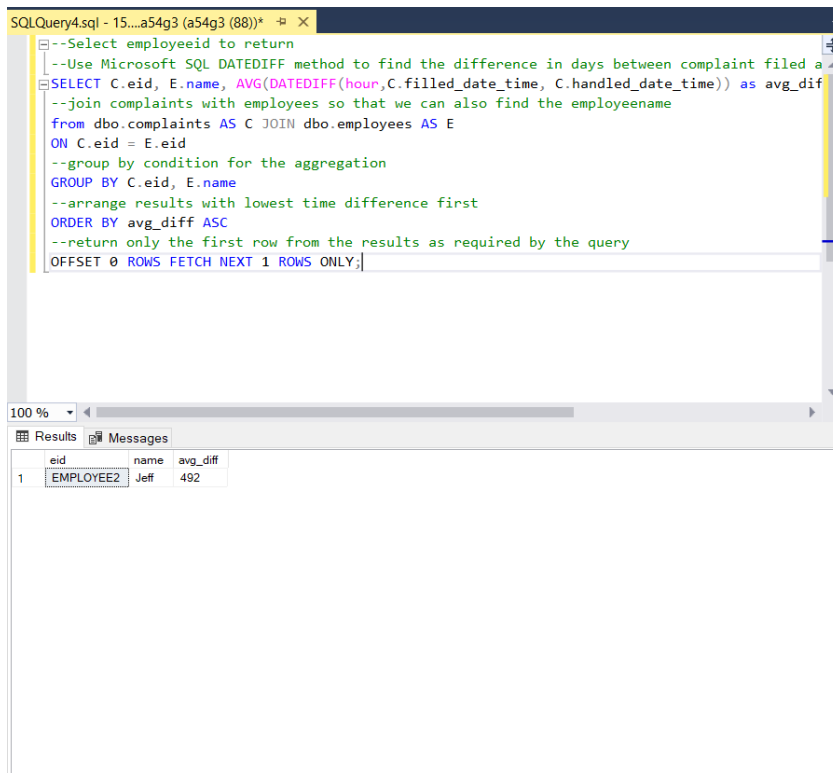
Below the query window, the 'Results' tab is active, displaying the following data:

	pubid	avg_days_to_delivery
1	PUB1	14
2	PUB2	14
3	PUB2001	5
4	PUB2006	6
5	PUB3	14

The query joins the publication, stocks_in_bookstores, items_in_orders_1, and orders tables to retrieve the relevant date information using WHERE to filter for orders in June 2022. The DATEDIFF function is used to calculate the difference in days between the date_time column from the orders table and the delivery_date column from the items_in_orders_1 table for each row. The AVG function is then used to calculate the average of these differences for each publication. The results are grouped by the pubid column from the publication table.

4. Query 4

```
SELECT C.eid, E.name, AVG(DATEDIFF(hour,C.filled_date_time,
C.handled_date_time)) as avg_diff
from dbo.complaints AS C JOIN dbo.employees AS E
ON C.eid = E.eid
GROUP BY C.eid, E.name
ORDER BY avg_diff ASC
OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY;
```



The screenshot shows a SQL Server Enterprise Manager window with a query editor and a results pane. The query editor contains the following SQL code:

```
--Select employeeid to return
--Use Microsoft SQL DATEDIFF method to find the difference in days between complaint filed a
SELECT C.eid, E.name, AVG(DATEDIFF(hour,C.filled_date_time, C.handled_date_time)) as avg_diff
--join complaints with employees so that we can also find the employee name
from dbo.complaints AS C JOIN dbo.employees AS E
ON C.eid = E.eid
--group by condition for the aggregation
GROUP BY C.eid, E.name
--arrange results with lowest time difference first
ORDER BY avg_diff ASC
--return only the first row from the results as required by the query
OFFSET 0 ROWS FETCH NEXT 1 ROWS ONLY;
```

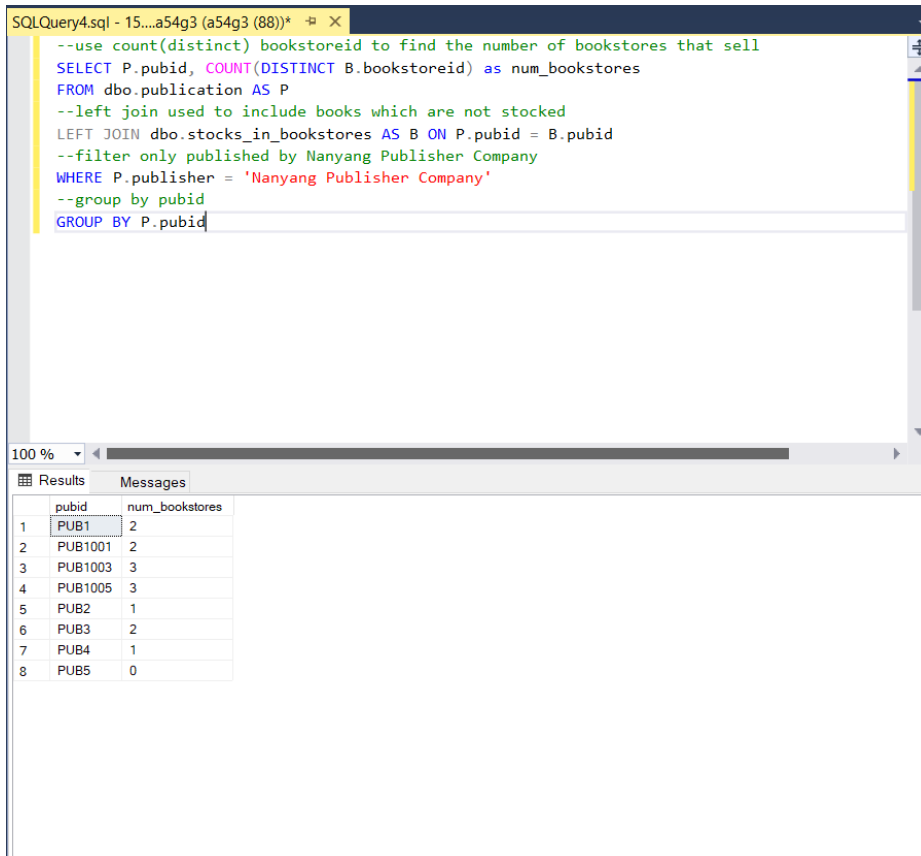
The results pane shows a single row of data:

	eid	name	avg_diff
1	EMPLOYEE2	Jeff	492

The query joins the complaints and employees tables to retrieve the relevant information. The DATEDIFF function is used to calculate the difference in hours between the two date columns for each row. The AVG function is then used to calculate the average of these differences for each employee. The results are grouped by the eid and name columns from the employees table and sorted in ascending order by the average time. The OFFSET and FETCH NEXT clauses are used to return only the first row of the result set, which corresponds to the employee with the lowest average time.

5. Query 5

```
SELECT P.pubid, COUNT(DISTINCT B.bookstoreid) as num_bookstores
FROM dbo.publication AS P
LEFT JOIN dbo.stocks_in_bookstores AS B ON P.pubid = B.pubid
WHERE P.publisher = 'Nanyang Publisher Company'
GROUP BY P.pubid
```



The screenshot shows a SQL Server query window with the following SQL code:

```
--use count(distinct) bookstoreid to find the number of bookstores that sell
SELECT P.pubid, COUNT(DISTINCT B.bookstoreid) as num_bookstores
FROM dbo.publication AS P
--left join used to include books which are not stocked
LEFT JOIN dbo.stocks_in_bookstores AS B ON P.pubid = B.pubid
--filter only published by Nanyang Publisher Company
WHERE P.publisher = 'Nanyang Publisher Company'
--group by pubid
GROUP BY P.pubid
```

The Results pane displays the following data:

	pubid	num_bookstores
1	PUB1	2
2	PUB1001	2
3	PUB1003	3
4	PUB1005	3
5	PUB2	1
6	PUB3	2
7	PUB4	1
8	PUB5	0

The query joins the publication and stocks_in_bookstores tables to retrieve the relevant information. The COUNT and DISTINCT functions are used to count the number of distinct bookstoreid values for each publication. The results are grouped by the pubid column from the publication table and filtered using a WHERE clause to include only publications from the specified publisher. LEFT JOIN is used to ensure that those publications who are published by Nanyang Publisher Company but not in stock are also showing in the results.

6. Query 6

```
WITH revenue_by_bookstore AS (
```



```

SELECT
sib.bookstoreid,
SUM(io.item_price * io.item_qty) AS total_revenue
FROM dbo.items_in_orders_1 AS io
JOIN dbo.stocks_in_bookstores AS sib ON io.stockid = sib.stockid
WHERE MONTH(io.delivery_date) = 8 AND YEAR(io.delivery_date) = 2022
GROUP BY sib.bookstoreid
)
SELECT
rb.bookstoreid,
rb.total_revenue
FROM revenue_by_bookstore AS rb
WHERE rb.total_revenue = (
SELECT MAX(inner_rb.total_revenue)
FROM revenue_by_bookstore AS inner_rb
)

```

SQLQuery4.sql - 15...a54g3 (a54g3 (88))* X

```

WITH revenue_by_bookstore AS (
    SELECT
    sib.bookstoreid,
    SUM(io.item_price * io.item_qty) AS total_revenue
    FROM dbo.items_in_orders_1 AS io
    JOIN dbo.stocks_in_bookstores AS sib ON io.stockid = sib.stockid
    WHERE MONTH(io.delivery_date) = 8 AND YEAR(io.delivery_date) = 2022
    GROUP BY sib.bookstoreid
)
SELECT
rb.bookstoreid,
rb.total_revenue
FROM revenue_by_bookstore AS rb
WHERE rb.total_revenue = (
    SELECT MAX(inner_rb.total_revenue)
    FROM revenue_by_bookstore AS inner_rb
)

```

100 %

Results Messages

	bookstoreid	total_revenue
1	BS1001	96
2	BS1002	96
3	BS1003	96

Query executed successfully. 155.69.100.36 (14.0 RTM) | a54g3 (88) | a54g3 | 00:00:00 | 3 rows

The query first calculates total revenue of each bookstore by joining the "items_in_orders_1" and "stocks_in_bookstores" tables. Then, it filters by the delivery date in August 2022, and grouping the results by bookstore ID.

The resulting table is then used in the outer query to retrieve the bookstore ID and total revenue. From there, we can find the bookstore(s) with the highest revenue in August 2022.

7. Query 7

```
SELECT T.cid AS Customer, title AS Title, Q.HighestPub AS Price
FROM items_in_orders_4 AS T
JOIN
    (SELECT DISTINCT Z.cid AS SELECTedCus
     FROM complaints AS Z
     GROUP BY Z.cid
     HAVING Count(Z.cid) =
      (SELECT Max(NumofCom) AS MaxComplaints
       FROM (SELECT Y.cid, Count(Y.complaintsid) AS NumOfCom
        FROM complaints AS Y
        GROUP BY Y.cid) AS X)) AS S
ON T.cid = S.SELECTedCus
JOIN
    (SELECT T.cid, max(item_price) AS HighestPub
     FROM items_in_orders_1 AS W JOIN items_in_orders_4 AS T
     ON W.orderid = T.orderid
     GROUP BY T.cid) AS Q
ON Q.cid = T.cid
JOIN items_in_orders_1 AS W
ON T.orderid = W.orderid
JOIN stocks_in_bookstores AS V
ON W.stockid = V.stockid
JOIN magazines AS U
ON V.pubid = U.pubid
WHERE Q.HighestPub = W.item_price
UNION ALL
SELECT T.cid AS Customer, title AS Title, Q.HighestPub AS Price
FROM items_in_orders_4 AS T
JOIN
    (SELECT DISTINCT Z.cid AS SelectedCus
     FROM complaints AS Z
     GROUP BY Z.cid
     HAVING Count(Z.cid) =
      (SELECT Max(NumofCom) AS MaxComplaints
       FROM (SELECT Y.cid, Count(Y.complaintsid) AS NumOfCom
        FROM complaints AS Y
        GROUP BY Y.cid) AS X)) AS S
```

```

ON T.cid = S.SELECTedCus
JOIN
    (SELECT T.cid, max(item_price) AS HighestPub
     FROM items_in_orders_1 AS W JOIN items_in_orders_4 AS T
     ON W.orderid = T.orderid
     GROUP BY T.cid) AS Q
ON Q.cid = T.cid
JOIN items_in_orders_1 AS W
ON T.orderid = W.orderid
JOIN stocks_in_bookstores AS V
ON W.stockid = V.stockid
JOIN books AS U
ON V.pubid = U.pubid
WHERE Q.HighestPub = W.item_price

```

The screenshot shows a SQL query editor window with the following query:

```

FROM (SELECT T.cid, COUNT(Y.complaintsid) AS NumOfCom
FROM complaints AS Y
GROUP BY Y.cid) AS S
ON T.cid = S.SELECTedCus
JOIN
    (SELECT T.cid, max(item_price) AS HighestPub
     FROM items_in_orders_1 AS W JOIN items_in_orders_4 AS T
     ON W.orderid = T.orderid
     GROUP BY T.cid) AS Q
ON Q.cid = T.cid
JOIN items_in_orders_1 AS W
ON T.orderid = W.orderid
JOIN stocks_in_bookstores AS V
ON W.stockid = V.stockid
JOIN books AS U
ON V.pubid = U.pubid
WHERE Q.HighestPub = W.item_price

```

The results pane shows the following data:

	Customer	Title	Price
1	CUSTOMER102	EatHub	93
2	CUSTOMER103	EatHub	93
3	CUSTOMER100	Book B	65
4	CUSTOMER101	Book D	69

Query executed successfully. 155.69.100.36 (14.0 RTM) | a54g3 (88) | a54g3 | 00:00:00 | 4 rows

The query primarily utilizes subquery functions to firstly find the customers with the most number of complaints and secondly, identify the most expensive items they purchase.

The subqueries happen in the FROM statement and are then joined with 4 other tables. The tables being items_in_orders_4 to link the cid, items_in_orders_1 to link the orderid, stocks_in_bookstores to link the stock-id and finally magazines to link the pubid.

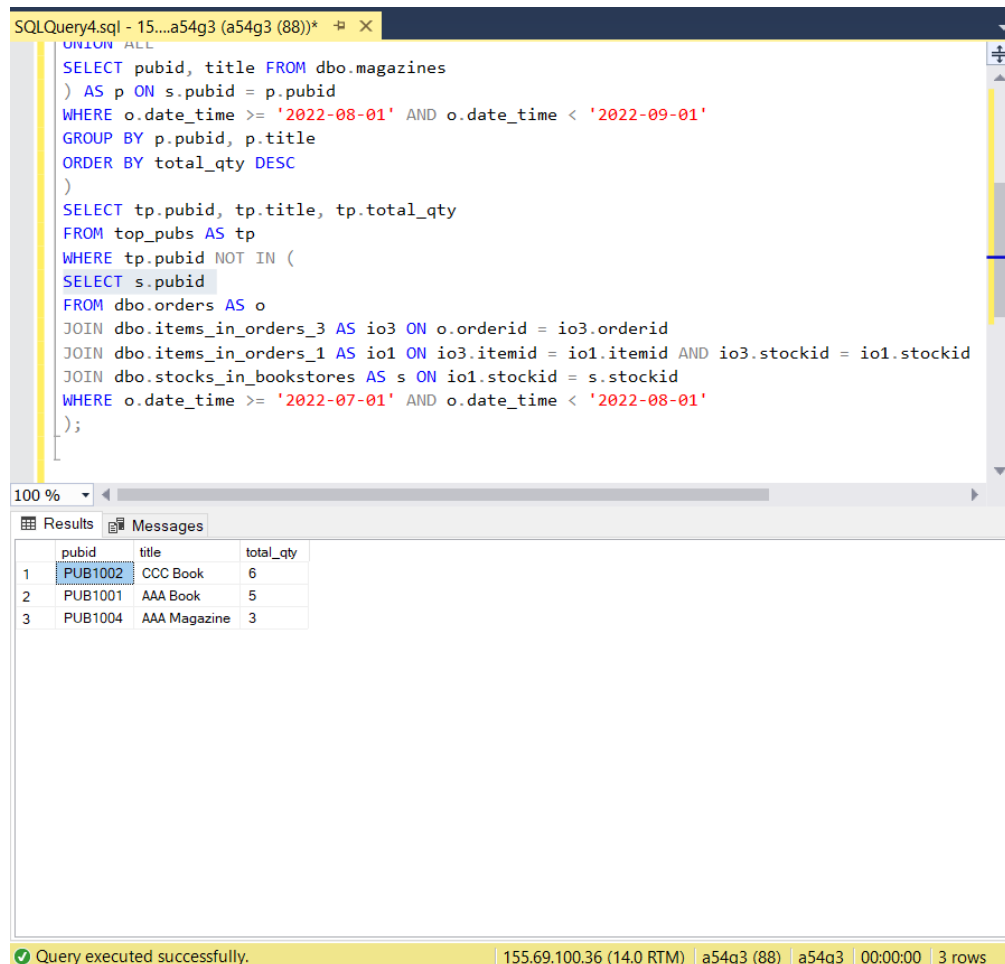
The books table on the other hand would be impractical to join as it would introduce another title attribute as a separate column. To place the magazine and book title under one column, UNION ALL was applied.

Under the merged table, we then displayed the customer id, publication title and the price that the customer bought it.

8. Query 8

```
WITH top_pubs AS (  
  SELECT TOP 3  
    p.pubid,  
    p.title,  
    SUM(io1.item_qty) AS total_qty  
  FROM dbo.orders AS o  
  JOIN dbo.items_in_orders_3 AS io3 ON o.orderid = io3.orderid  
  JOIN dbo.items_in_orders_1 AS io1 ON io3.itemid = io1.itemid AND  
    io3.stockid = io1.stockid  
  JOIN dbo.stocks_in_bookstores AS s ON io1.stockid = s.stockid  
  JOIN (  
    SELECT pubid, title FROM dbo.books  
    UNION ALL  
    SELECT pubid, title FROM dbo.magazines  
  ) AS p ON s.pubid = p.pubid  
  WHERE o.date_time >= '2022-08-01' AND o.date_time < '2022-09-01'  
  GROUP BY p.pubid, p.title  
  ORDER BY total_qty DESC  
)  
SELECT tp.pubid, tp.title, tp.total_qty  
FROM top_pubs AS tp  
WHERE tp.pubid NOT IN (  
  SELECT s.pubid  
  FROM dbo.orders AS o  
  JOIN dbo.items_in_orders_3 AS io3 ON o.orderid = io3.orderid  
  JOIN dbo.items_in_orders_1 AS io1 ON io3.itemid = io1.itemid AND  
    io3.stockid = io1.stockid  
  JOIN dbo.stocks_in_bookstores AS s ON io1.stockid = s.stockid  
  WHERE o.date_time >= '2022-07-01' AND o.date_time < '2022-08-01'
```

);



The screenshot shows a SQL Server Enterprise Manager window with a query titled "SQLQuery4.sql - 15....a54g3 (a54g3 (88))*". The query is a complex SQL statement using a Common Table Expression (CTE) named "top_pubs". The CTE selects the top three publications based on total quantity sold in August 2022, joining tables "magazines", "orders", "items_in_orders_3", "items_in_orders_1", and "stocks_in_bookstores". The main query then selects the "pubid", "title", and "total_qty" columns from the "top_pubs" CTE, but only for publications that were never purchased by any customers in July 2022, using a "NOT IN" clause to filter out publications that were purchased in July 2022.

```
WITH top_pubs AS (
    SELECT pubid, title FROM dbo.magazines
) AS p ON s.pubid = p.pubid
WHERE o.date_time >= '2022-08-01' AND o.date_time < '2022-09-01'
GROUP BY p.pubid, p.title
ORDER BY total_qty DESC
)
SELECT tp.pubid, tp.title, tp.total_qty
FROM top_pubs AS tp
WHERE tp.pubid NOT IN (
    SELECT s.pubid
    FROM dbo.orders AS o
    JOIN dbo.items_in_orders_3 AS io3 ON o.orderid = io3.orderid
    JOIN dbo.items_in_orders_1 AS io1 ON io3.itemid = io1.itemid AND io3.stockid = io1.stockid
    JOIN dbo.stocks_in_bookstores AS s ON io1.stockid = s.stockid
    WHERE o.date_time >= '2022-07-01' AND o.date_time < '2022-08-01'
);
```

	pubid	title	total_qty
1	PUB1002	CCC Book	6
2	PUB1001	AAA Book	5
3	PUB1004	AAA Magazine	3

Query executed successfully. | 155.69.100.36 (14.0 RTM) | a54g3 (88) | a54g3 | 00:00:00 | 3 rows

The query starts by defining a common table expression (CTE) called "top_pubs". The CTE selects the top three publications based on total quantity sold in August 2022. It does this by joining several tables: "orders", "items_in_orders_3", "items_in_orders_1", "stocks_in_bookstores", "books", and "magazines". The CTE groups the results by publication and orders them by total quantity sold in descending order.

The main query then selects the pubid, title, and total_qty columns from the "top_pubs" CTE, but only for publications that were never purchased by any customers in July 2022. It does this by using a subquery that selects the pubid of all publications that were purchased in July 2022. The main query then filters out these publications using a "NOT IN" clause.

9. Query 9

WITH monthly_sales AS (

```

SELECT p.pubid, MONTH(o.date_time) AS month, SUM(io1.item_qty) AS sales
FROM dbo.publication p
JOIN dbo.stocks_in_bookstores s ON p.pubid = s.pubid
JOIN dbo.items_in_orders_3 io3 ON s.stockid = io3.stockid
JOIN dbo.items_in_orders_1 io1 ON io3.itemid = io1.itemid
JOIN dbo.orders o ON io1.orderid = o.orderid
GROUP BY p.pubid, MONTH(o.date_time)
),
ranked_sales AS (
SELECT pubid, month, sales,
RANK() OVER (PARTITION BY pubid ORDER BY sales) AS sales_rank
FROM monthly_sales
)
SELECT pubid
FROM ranked_sales
WHERE sales_rank <= 3
GROUP BY pubid
HAVING COUNT(*) = 3;

```

The screenshot shows a SQL Server Enterprise Manager window with a query titled 'SQLQuery4.sql - 15...a54g3 (a54g3 (88))'. The query text is identical to the one in the first block. Below the query window, the 'Results' tab is active, displaying a table with 7 rows and 1 column named 'pubid'. The values in the 'pubid' column are PUB1, PUB2001, PUB2004, PUB2005, PUB2006, PUB2007, and PUB2008. At the bottom of the window, a status bar indicates 'Query executed successfully.', '155.69.100.36 (14.0 RTM)', 'a54g3 (88)', 'a54g3', '00:00:00', and '7 rows'.

pubid
1
PUB1
2
PUB2001
3
PUB2004
4
PUB2005
5
PUB2006
6
PUB2007
7
PUB2008

The query selects the publications that have been ranked in the top three by sales for every month by calculating the monthly sales for each publication and then ranking those sales by month and publication. The query then selects the pubid values that have been ranked in the top three for exactly 3 months. It accomplishes this using two common table expressions.

The first common table expression (CTE) is called "monthly_sales". This CTE calculates the total sales for each publication in each month. It does this by joining several tables: "publication", "stocks_in_bookstores", "items_in_orders_3", "items_in_orders_1", and "orders". The CTE groups the results by publication and month and calculates the sum of item_qty for each group.

The second CTE is called "ranked_sales". This CTE ranks the sales for each publication in each month using the RANK() function. It does this by partitioning the results by pubid and ordering them by sales.

The main query then selects the pubid column from the "ranked_sales" CTE for publications that have been ranked in the top three for every month. It does this by filtering the results using a "WHERE" clause that selects the rows where the sales_rank is less than or equal to 3. The query groups the results by pubid and selects only the pubid values that have been ranked in the top three for exactly 3 months using a "HAVING" clause.

Printout of Table Records

Please refer to the attached CSV files for further details of our table contents.

1. Books

```
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB1', N'Harry Porter
Finale');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB1001', N'AAA Book');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB1002', N'CCC Book');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB1003', N'BBB Book');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB2', N'Book B');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB2001', N'Very Popular
Book');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB3', N'Book C');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB4', N'Book D');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB5', N'Book E');
INSERT INTO dbo.books (pubid, title) VALUES (N'PUB6', N'Book F');
```

2. Bookstore

```
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS1');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS1001');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS1002');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS1003');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS1004');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS2');
INSERT INTO dbo.bookstore (bookstoreid) VALUES (N'BS3');
```

3. Complaints on Bookstore

```
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT1', N'BS1');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT109', N'BS1');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT110', N'BS1');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT111', N'BS1');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT112', N'BS2');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT113', N'BS2');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
```



```

(N'COMPLAINT114', N'BS2');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT115', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT116', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT117', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT121', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT122', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT123', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT2', N'BS1');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT3', N'BS2');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT4', N'BS2');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT5', N'BS3');
INSERT INTO dbo.complaints_on_bookstore (complaintsid, bookstoreid) VALUES
(N'COMPLAINT6', N'BS3');

```

4. Complaints on Orders

```

INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT100', N'ORDER100');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT101', N'ORDER101');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT102', N'ORDER102');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT103', N'ORDER103');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT104', N'ORDER104');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT105', N'ORDER105');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT106', N'ORDER106');
INSERT INTO dbo.complaints_on_orders (complaintsid, orderid) VALUES
(N'COMPLAINT107', N'ORDER107');

```

```

INSERT INTO dbo.complaints_on_orders (complaintsid,orderid) VALUES
(N'COMPLAINT108', N'ORDER108');
INSERT INTO dbo.complaints_on_orders (complaintsid,orderid) VALUES
(N'COMPLAINT118', N'ORDER109');
INSERT INTO dbo.complaints_on_orders (complaintsid,orderid) VALUES
(N'COMPLAINT119', N'ORDER110');
INSERT INTO dbo.complaints_on_orders (complaintsid,orderid) VALUES
(N'COMPLAINT120', N'ORDER111');

```

5. Complaints

```

INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT1', N'CUSTOMER1',
N'EMPLOYEE1', N'2022-06-15 00:52:50.000', N'Bad bookstore', N'2022-06-01
00:54:23.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT100', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-05-11 10:12:00.000', N'Why the book not in english',
N'2022-04-20 01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT101', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-06-15 07:30:27.000', N'The book is torn, wth',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT102', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-06-20 13:50:43.000', N'The book words so small',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT103', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-05-12 10:12:00.000', N'Cliff hanger bad', N'2022-04-20
01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT104', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-06-16 07:30:27.000', N'wrong book version',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT105', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-06-21 13:50:43.000', N'book came damaged',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT106', N'CUSTOMER102',
N'EMPLOYEE2', N'2022-05-13 10:12:00.000', N'magazine is old', N'2022-04-20

```

```

01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT107', N'CUSTOMER102',
N'EMPLOYEE1', N'2022-06-17 07:30:27.000', N'wrong magazine issue',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT108', N'CUSTOMER102',
N'EMPLOYEE2', N'2022-06-22 13:50:43.000', N'magazine was torn',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT109', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-05-11 10:12:00.000', N'Why the bookstore not in
english', N'2022-04-20 01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT110', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-06-15 07:30:27.000', N'The booksore in disrepair,
wth', N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT111', N'CUSTOMER100',
N'EMPLOYEE2', N'2022-06-20 13:50:43.000', N'The bookstore so small',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT112', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-05-12 10:12:00.000', N'Bookstore ugly', N'2022-04-20
01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT113', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-06-16 07:30:27.000', N'wrong bookstore', N'2022-05-25
00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT114', N'CUSTOMER101',
N'EMPLOYEE1', N'2022-06-21 13:50:43.000', N'bookstore not fun',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT115', N'CUSTOMER102',
N'EMPLOYEE2', N'2022-05-13 10:12:00.000', N'bookstore is old', N'2022-04-20
01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT116', N'CUSTOMER102',
N'EMPLOYEE1', N'2022-06-17 07:30:27.000', N'bookstore is confusing',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT117', N'CUSTOMER102',

```

```
N'EMPLOYEE2', N'2022-06-22 13:50:43.000', N'bad customer service',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT118', N'CUSTOMER103',
N'EMPLOYEE2', N'2022-05-13 10:12:00.000', N'magazine is old', N'2022-04-20
01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT119', N'CUSTOMER103',
N'EMPLOYEE1', N'2022-06-17 07:30:27.000', N'wrong magazine issue',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT120', N'CUSTOMER103',
N'EMPLOYEE2', N'2022-06-22 13:50:43.000', N'magazine was torn',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT121', N'CUSTOMER103',
N'EMPLOYEE2', N'2022-05-13 10:12:00.000', N'bookstore is old', N'2022-04-20
01:50:40.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT122', N'CUSTOMER103',
N'EMPLOYEE1', N'2022-06-17 07:30:27.000', N'bookstore is confusing',
N'2022-05-25 00:30:15.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT123', N'CUSTOMER103',
N'EMPLOYEE2', N'2022-06-22 13:50:43.000', N'bad customer service',
N'2022-05-29 12:35:00.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT2', N'CUSTOMER2',
N'EMPLOYEE1', N'2022-06-16 00:52:50.000', N'Very bad bookstore',
N'2022-06-02 00:54:23.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT3', N'CUSTOMER3',
N'EMPLOYEE1', N'2022-06-17 00:52:50.000', N'Extremely dirty bookstore',
N'2022-06-03 00:54:23.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT4', N'CUSTOMER1',
N'EMPLOYEE2', N'2022-06-14 00:52:50.000', N'Bad customer service',
N'2022-06-04 00:54:23.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
complain_text, filled_date_time) VALUES (N'COMPLAINT5', N'CUSTOMER2',
N'EMPLOYEE2', N'2022-06-15 00:53:50.000', N'Bad books', N'2022-06-05
00:54:23.000');
INSERT INTO dbo.complaints (complaintsid, cid, eid, handled_date_time,
```

```
complain_text, filled_date_time) VALUES (N'COMPLAINT6', N'CUSTOMER3',  
N'EMPLOYEE2', N'2022-06-16 00:54:50.000', N'No books in store',  
N'2022-06-06 00:54:23.000');
```

6. Customers

```
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1', N'Chaitanya');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER100', N'Irfan');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1001', N'Andy');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1002', N'Bob');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1003', N'Clara');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1004', N'Duncan');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER1005', N'Elly');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER101', N'Bobby');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER102', N'Sam');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER103', N'Jolly');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER2', N'XYZ');  
INSERT INTO dbo.customers (cid, name) VALUES (N'CUSTOMER3', N'John');
```

7. Employees

```
INSERT INTO dbo.employees (eid, name, salary) VALUES (N'EMPLOYEE1',  
N'Marc', 1050);  
INSERT INTO dbo.employees (eid, name, salary) VALUES (N'EMPLOYEE2',  
N'Jeff', 2210);
```

8. Items in Orders 1

```
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,  
delivery_date, stockid) VALUES (N'ITEM1', 10, 1, N'ORDER1', N'2022-06-15',  
N'STOCK2');  
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,  
delivery_date, stockid) VALUES (N'ITEM100', 25, 1, N'ORDER100',  
N'2022-04-19', N'STOCK100');  
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,  
delivery_date, stockid) VALUES (N'ITEM1001', 20, 3, N'ORDER1001',  
N'2022-08-02', N'STOCK1001');  
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,  
delivery_date, stockid) VALUES (N'ITEM1002', 18, 2, N'ORDER1001',  
N'2022-08-05', N'STOCK1002');  
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,  
delivery_date, stockid) VALUES (N'ITEM1003', 20, 3, N'ORDER1002',
```

```
N'2022-08-15', N'STOCK1004');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1004', 18, 2, N'ORDER1002',
N'2022-08-17', N'STOCK1005');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1005', 20, 3, N'ORDER1003',
N'2022-08-18', N'STOCK1007');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1006', 18, 2, N'ORDER1003',
N'2022-08-19', N'STOCK1008');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1007', 19, 1, N'ORDER1004',
N'2022-08-20', N'STOCK10010');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1008', 18, 1, N'ORDER1005',
N'2022-07-10', N'STOCK1005');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM1009', 19, 1, N'ORDER1005',
N'2022-07-10', N'STOCK1006');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM101', 15, 1, N'ORDER101',
N'2022-05-24', N'STOCK101');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM102', 65, 1, N'ORDER102',
N'2022-05-28', N'STOCK102');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM103', 60, 1, N'ORDER103',
N'2022-04-19', N'STOCK103');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM104', 27, 2, N'ORDER104',
N'2022-05-24', N'STOCK104');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM105', 69, 1, N'ORDER105',
N'2022-05-28', N'STOCK105');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM106', 88, 1, N'ORDER106',
N'2022-04-19', N'STOCK106');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM107', 71, 1, N'ORDER107',
N'2022-05-24', N'STOCK107');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM108', 93, 2, N'ORDER108',
N'2022-05-28', N'STOCK108');
```



```
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM109', 88, 1, N'ORDER109',
N'2022-04-19', N'STOCK109');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM110', 71, 1, N'ORDER110',
N'2022-05-24', N'STOCK110');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM111', 93, 2, N'ORDER111',
N'2022-05-28', N'STOCK111');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2', 15, 2, N'ORDER2', N'2022-06-16',
N'STOCK3');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2001', 10, 1, N'ORDER21',
N'2022-06-15', N'STOCK21');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2002', 10, 2, N'ORDER22',
N'2022-07-15', N'STOCK22');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2003', 10, 3, N'ORDER23',
N'2022-08-15', N'STOCK23');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2004', 10, 5, N'ORDER24',
N'2022-09-11', N'STOCK24');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2005', 10, 7, N'ORDER25',
N'2022-10-11', N'STOCK25');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2006', 10, 11, N'ORDER26',
N'2022-11-11', N'STOCK26');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2007', 10, 3, N'ORDER27',
N'2022-03-30', N'STOCK27');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2008', 10, 4, N'ORDER28',
N'2022-04-30', N'STOCK28');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM2009', 10, 5, N'ORDER29',
N'2022-05-30', N'STOCK29');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3', 20, 3, N'ORDER3', N'2022-06-17',
N'STOCK4');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
```

```

delivery_date, stockid) VALUES (N'ITEM3001', 10, 4, N'ORDER31',
N'2022-04-26', N'STOCK31');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3002', 10, 5, N'ORDER32',
N'2022-05-26', N'STOCK32');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3003', 10, 8, N'ORDER33',
N'2022-06-26', N'STOCK33');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3004', 10, 2, N'ORDER34',
N'2022-07-18', N'STOCK34');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3005', 10, 4, N'ORDER35',
N'2022-08-18', N'STOCK35');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3006', 10, 6, N'ORDER36',
N'2022-09-18', N'STOCK36');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3007', 10, 1, N'ORDER37',
N'2022-10-05', N'STOCK37');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3008', 10, 3, N'ORDER38',
N'2022-11-04', N'STOCK38');
INSERT INTO items_in_orders_1 (itemid, item_price, item_qty, orderid,
delivery_date, stockid) VALUES (N'ITEM3009', 10, 4, N'ORDER39',
N'2022-12-03', N'STOCK39');

```

9. Items in Orders 2

```

INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment1', 5, N'2022-08-01
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment2', 5, N'2022-08-02
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment3', 5, N'2022-08-03
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment4', 5, N'2022-08-04
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,

```



```
feedback_date_time) VALUES (N'ITEM1', N'Comment5', 5, N'2022-08-05
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment6', 5, N'2022-08-06
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment7', 5, N'2022-08-07
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment8', 5, N'2022-08-08
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment9', 5, N'2022-08-09
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1', N'Comment10', 5, N'2022-08-10
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment11', 5, N'2022-08-11
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment12', 5, N'2022-08-12
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment13', 5, N'2022-08-13
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment14', 5, N'2022-08-14
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment15', 5, N'2022-08-15
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment16', 5, N'2022-08-16
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment17', 5, N'2022-08-17
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment18', 5, N'2022-08-18
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment19', 5, N'2022-08-19
```

```
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment20', 5, N'2022-08-20
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2', N'Comment21', 4, N'2022-08-21
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM3', N'Comment22', 1, N'2022-09-29
00:41:27.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM3', N'Comment23', 5, N'2022-08-22
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM3', N'Comment24', 5, N'2022-08-23
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM3', N'Comment25', 5, N'2022-08-24
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM3', N'Comment26', 5, N'2022-08-25
00:38:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1001', null, 3, N'2022-08-02
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1002', null, 2, N'2022-08-05
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1003', null, 3, N'2022-08-15
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1004', null, 2, N'2022-08-17
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1005', null, 3, N'2022-08-18
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1006', null, 2, N'2022-08-19
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1007', null, 5, N'2022-08-20
00:48:42.000');
```

```

INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1008', null, 2, N'2022-07-10
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM1009', null, 3, N'2022-07-10
00:48:42.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2001', N'Brilliant', 4, N'2022-06-15
23:21:23.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2001', N'Brilliant', 4, N'2022-07-15
23:21:23.000');
INSERT INTO dbo.items_in_orders_2 (itemid, comment, rating,
feedback_date_time) VALUES (N'ITEM2001', N'Brilliant', 4, N'2022-07-15
23:21:23.000');

```

10. Items in Orders 3

```

INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES (N'ORDER1',
N'STOCK2', N'ITEM1');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER100', N'STOCK100', N'ITEM100');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1001', N'STOCK1001', N'ITEM1001');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1001', N'STOCK1002', N'ITEM1002');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1002', N'STOCK1004', N'ITEM1003');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1002', N'STOCK1005', N'ITEM1004');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1003', N'STOCK1007', N'ITEM1005');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1003', N'STOCK1008', N'ITEM1006');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1004', N'STOCK10010', N'ITEM1007');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1005', N'STOCK1005', N'ITEM1008');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER1005', N'STOCK1006', N'ITEM1009');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES

```

```

(N'ORDER101', N'STOCK101', N'ITEM101');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER102', N'STOCK102', N'ITEM102');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER103', N'STOCK103', N'ITEM103');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER104', N'STOCK104', N'ITEM104');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER105', N'STOCK105', N'ITEM105');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER106', N'STOCK106', N'ITEM106');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER107', N'STOCK107', N'ITEM107');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER108', N'STOCK108', N'ITEM108');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER109', N'STOCK109', N'ITEM109');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER110', N'STOCK110', N'ITEM110');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER111', N'STOCK111', N'ITEM111');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES (N'ORDER2',
N'STOCK3', N'ITEM2');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER21', N'STOCK21', N'ITEM2001');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER22', N'STOCK22', N'ITEM2002');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER23', N'STOCK23', N'ITEM2003');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER24', N'STOCK24', N'ITEM2004');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER25', N'STOCK25', N'ITEM2005');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER26', N'STOCK26', N'ITEM2006');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER27', N'STOCK27', N'ITEM2007');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER28', N'STOCK28', N'ITEM2008');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER29', N'STOCK29', N'ITEM2009');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES (N'ORDER3',
N'STOCK4', N'ITEM3');

```

```

INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER31', N'STOCK31', N'ITEM3001');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER32', N'STOCK32', N'ITEM3002');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER33', N'STOCK33', N'ITEM3003');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER34', N'STOCK34', N'ITEM3004');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER35', N'STOCK35', N'ITEM3005');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER36', N'STOCK36', N'ITEM3006');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER37', N'STOCK37', N'ITEM3007');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER38', N'STOCK38', N'ITEM3008');
INSERT INTO items_in_orders_3 (orderid, stockid, itemid) VALUES
(N'ORDER39', N'STOCK39', N'ITEM3009');

```

11. Items in Orders 4

```

INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1',
N'CUSTOMER1');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER100',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1001',
N'CUSTOMER1001');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1002',
N'CUSTOMER1002');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1003',
N'CUSTOMER1003');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1004',
N'CUSTOMER1004');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER1005',
N'CUSTOMER1005');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER101',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER102',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER103',
N'CUSTOMER101');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER104',

```

```
N'CUSTOMER101');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER105',
N'CUSTOMER101');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER106',
N'CUSTOMER102');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER107',
N'CUSTOMER102');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER108',
N'CUSTOMER102');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER109',
N'CUSTOMER103');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER110',
N'CUSTOMER103');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER111',
N'CUSTOMER103');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER2',
N'CUSTOMER2');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER21',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER22',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER24',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER25',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER26',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER27',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER28',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER29',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER3',
N'CUSTOMER3');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER31',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER32',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER33',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER34',
N'CUSTOMER100');
```

```

INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER35',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER36',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER37',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER38',
N'CUSTOMER100');
INSERT INTO items_in_orders_4 (orderid, cid) VALUES (N'ORDER39',
N'CUSTOMER100');

```

12. Magazines

```

INSERT INTO dbo.magazines (pubid, title, issue) VALUES (N'PUB10',
N'PopClub', 54);
INSERT INTO dbo.magazines (pubid, title, issue) VALUES (N'PUB1004', N'AAA
Magazine', 1);
INSERT INTO dbo.magazines (pubid, title, issue) VALUES (N'PUB1005', N'BBB
Magazine', 2);
INSERT INTO dbo.magazines (pubid, title, issue) VALUES (N'PUB11',
N'EatHub', 24);

```

13. Orders

```

INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1', N'2022-06-01 22:45:13.000', N'123 Oak St', N'CUSTOMER1');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER100', N'2022-04-19 00:30:12.000', N'420 Dover St', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1001', N'2022-08-02 22:45:13.000', N'Blk 111 ABC Road',
N'CUSTOMER1001');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1002', N'2022-08-03 22:45:13.000', N'Blk 222 ABC Road',
N'CUSTOMER1002');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1003', N'2022-08-05 22:45:13.000', N'Blk 333 ABC Road',
N'CUSTOMER1003');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1004', N'2022-08-06 22:45:13.000', N'Blk 444 ABC Road',
N'CUSTOMER1004');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER1005', N'2022-08-03 22:45:13.000', N'Blk 555 ABC Road',
N'CUSTOMER1005');

```



```
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER101', N'2022-05-24 20:05:13.000', N'420 Dover St', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER102', N'2022-05-28 23:50:20.000', N'420 Dover St', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER103', N'2022-04-19 00:30:12.000', N'422 Devar St', N'CUSTOMER101');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER104', N'2022-05-24 20:05:13.000', N'422 Devar St', N'CUSTOMER101');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER105', N'2022-05-28 23:50:20.000', N'422 Devar St', N'CUSTOMER101');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER106', N'2022-04-19 00:30:12.000', N'320 Cover St', N'CUSTOMER102');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER107', N'2022-05-24 20:05:13.000', N'320 Cover St', N'CUSTOMER102');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER108', N'2022-05-28 23:50:20.000', N'320 Cover St', N'CUSTOMER102');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER109', N'2022-04-19 00:30:12.000', N'320 Flover St',
N'CUSTOMER103');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER110', N'2022-05-24 20:05:13.000', N'320 Flover St',
N'CUSTOMER103');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER111', N'2022-05-28 23:50:20.000', N'320 Flover St',
N'CUSTOMER103');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER2', N'2022-06-02 22:45:13.000', N'456 Pine St', N'CUSTOMER2');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER21', N'2022-06-15 22:21:23.000', N'Boon Lay Street 12',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER22', N'2022-07-15 22:21:23.000', N'Boon Lay Street 12',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER23', N'2022-08-15 22:21:23.000', N'Boon Lay Street 12',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER24', N'2022-09-11 22:21:23.000', N'Nanyang Crescent',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER25', N'2022-10-11 22:21:23.000', N'Nanyang Crescent',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
```



```

(N'ORDER26', N'2022-11-11 22:21:23.000', N'Nanyang Crescent',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER27', N'2022-03-30 15:28:05.000', N'Punggol St 21', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER28', N'2022-04-30 15:28:19.000', N'Punggol St 21', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER29', N'2022-05-30 15:28:27.000', N'Punggol St 21', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER3', N'2022-06-03 22:45:13.000', N'789 Tree St', N'CUSTOMER3');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER31', N'2022-04-26 09:36:03.000', N'Clementi Drive',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER32', N'2022-05-26 15:36:17.000', N'Clementi Drive',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER33', N'2022-06-26 15:36:27.000', N'Clementi Drive',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER34', N'2022-07-18 02:37:58.000', N'Serangoon Gardens',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER35', N'2022-08-18 15:41:20.000', N'Serangoon Gardens',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER36', N'2022-09-18 21:41:32.000', N'Serangoon Gardens',
N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER37', N'2022-10-05 20:51:22.000', N'Changi Green', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER38', N'2022-11-04 15:54:38.000', N'Changi Green', N'CUSTOMER100');
INSERT INTO orders (orderid, date_time, shipping_address, cid) VALUES
(N'ORDER39', N'2022-12-03 15:54:47.000', N'Changi Green', N'CUSTOMER100');

```

14. Order Status

```

INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES (N'ITEM1',
N'2022-06-15', N'delivered');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1001', N'2022-08-01', N'delivered');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES

```

```

(N'ITEM1002', N'2022-08-05', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1003', N'2022-08-15', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1004', N'2022-08-17', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1005', N'2022-08-18', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1006', N'2022-08-19', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1007', N'2022-08-20', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1008', N'2022-07-10', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES
(N'ITEM1009', N'2022-07-10', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES (N'ITEM2',
N'2022-06-16', N'delivered
');
INSERT INTO dbo.orderstatus (itemid, status_date, state) VALUES (N'ITEM3',
N'2022-06-17', N'delivered
');

```

15. Price History

```

INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1', 10, N'2022-08-01', N'2022-08-15');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1', 15, N'2022-08-16', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1001', 20, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK10010', 19, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK10011', 21, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK10012', 18, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES

```

```

(N'STOCK1002', 18, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1003', 17, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1004', 20, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1005', 18, N'2022-07-01', N'2022-07-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1005', 18, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1006', 19, N'2022-07-01', N'2022-07-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1006', 21, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1007', 20, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1008', 18, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK1009', 20, N'2022-08-01', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK2', 10, N'2022-06-01', N'2022-06-30');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK2', 12.5, N'2022-08-01', N'2022-08-15');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK2', 20, N'2022-08-16', N'2022-08-31');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK3', 15, N'2022-06-01', N'2022-06-30');
INSERT INTO dbo.price_history (stockid, price, start_date, end_date) VALUES
(N'STOCK4', 20, N'2022-06-01', N'2022-06-30');

```

16. Publication

```

INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1', N'Nanyang
Publisher Company', 2001);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB10', N'Galaxy
Mag Company', 2012);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1001',
N'Nanyang Publisher Company', 2021);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1002', N'BBB
Publisher Company', 2020);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1003',
N'Nanyang Publisher Company', 2019);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1004', N'CCC

```

```

Publisher Company', 2020);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB1005',
N'Nanyang Publisher Company', 2021);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB11', N'Happy
TreeF Company', 2013);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2', N'Nanyang
Publisher Company', 2002);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2001',
N'Popular Publisher Company', 2020);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2004',
N'Penguin Publishing Company', 2011);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2005',
N'Macmillan Company', 2004);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2006',
N'Reader's Digest', 2001);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2007', N'NBS
Publisher Company', 2001);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB2008',
N'Kinokuniya Publishing Company', 2003);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB3', N'Nanyang
Publisher Company', 2003);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB4', N'Nanyang
Publisher Company', 2004);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB5', N'Nanyang
Publisher Company', 2005);
INSERT INTO PUBLICATION (pubid, publisher, year) VALUES (N'PUB6', N'SCSE
Publisher Company', 2006);

```

17. Stocks in Bookstores

```





INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1', 15, 10, N'BS1', N'PUB1');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK100', 20, 10, N'BS2', N'PUB1');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1001', 10, 20, N'BS1001', N'PUB1001');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK10010', 10, 20, N'BS1004', N'PUB1002');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK10011', 10, 20, N'BS1004', N'PUB1003');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK10012', 10, 20, N'BS1004', N'PUB1005');

```

```
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1002', 10, 20, N'BS1001', N'PUB1002');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1003', 10, 20, N'BS1001', N'PUB1003');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1004', 10, 20, N'BS1002', N'PUB1002');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1005', 10, 20, N'BS1002', N'PUB1004');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1006', 10, 20, N'BS1002', N'PUB1005');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1007', 10, 20, N'BS1003', N'PUB1003');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1008', 10, 20, N'BS1003', N'PUB1001');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK1009', 10, 20, N'BS1003', N'PUB1005');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK101', 10, 8, N'BS1', N'PUB1');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK102', 60, 10, N'BS1', N'PUB2');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK103', 60, 10, N'BS1', N'PUB3');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK104', 20, 8, N'BS2', N'PUB6');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK105', 60, 10, N'BS1', N'PUB4');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK106', 80, 10, N'BS1', N'PUB10');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK107', 70, 8, N'BS2', N'PUB11');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK108', 90, 10, N'BS1', N'PUB11');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK109', 80, 10, N'BS1', N'PUB10');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK110', 70, 8, N'BS2', N'PUB11');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK111', 90, 10, N'BS1', N'PUB11');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK2', 20, 8, N'BS2', N'PUB1');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
bookstoreid, pubid) VALUES (N'STOCK21', 10, 21, N'BS1', N'PUB2001');
INSERT INTO stocks_in_bookstores (stockid, stock_price, stock_qty,
```

[illegible]

APPENDIX C: INDIVIDUAL CONTRIBUTION FORM

Full Name	Individual Contribution to Lab 5 Submission	Percentage of Contribution	Signature
Jadhav Chaitanya Dhananjay	Created tables and did queries	20%	
Palaniswamy Tarun Kumar	Created tables and did queries	20%	
Muhammad Irfan bin Ameer Hamzah	Created tables and did queries	20%	
Zhang Xiaoyang	Created tables and did queries	20%	
Rishabh Alexander John	Created tables and did queries	20%	