



**NANYANG  
TECHNOLOGICAL  
UNIVERSITY**  
**SINGAPORE**

**Lab 5 - Final Demonstration (SQL Query Solutions)**

Group Members (Z49 - Group 1)	
Yang Yichen	U2121540A
Oza Tirth Tusharbhai	U2123973F
Tan Jui Kit Justin	U2120886D
Weng Pei He	U2120920A
Lim Zheng Yong Shawn	U2121495D
Yap Shen Hwei	U2140630E

## SQL DDL Commands for Table Creation

Relation	Creation Query	Insertion Query Example
BOOKSTORE	<pre>CREATE TABLE BOOKSTORE (   ID CHAR(3)   PRIMARY KEY (ID) );</pre>	<pre>INSERT INTO BOOKSTORE VALUES ('B01');  INSERT INTO BOOKSTORE VALUES ('B02');</pre>
CUSTOMERS	<pre>CREATE TABLE CUSTOMERS (   CID CHAR(3),   Name VARCHAR(50),   PRIMARY KEY (CID) );</pre>	<pre>INSERT INTO CUSTOMERS VALUES('C01', 'Sarah Lim Xi Liu');  INSERT INTO CUSTOMERS VALUES('C01', 'Mohammad Ali');</pre>
PUBLICATION	<pre>CREATE TABLE PUBLICATION (   PubID CHAR(3),   Publisher VARCHAR(50),   Year INT,   PRIMARY KEY (PubID) );</pre>	<pre>INSERT INTO PUBLICATION VALUES('P01', 'Penguin', 2010);  INSERT INTO PUBLICATION VALUES('P02', 'Oxford', 2000);</pre>
BOOKS	<pre>CREATE TABLE BOOKS (   PubID CHAR(3),   Title VARCHAR(100),   PRIMARY KEY (PubID),   FOREIGN KEY(PubID) REFERENCES PUBLICATION(PubID) );</pre>	<pre>INSERT INTO BOOKS VALUES('P01', 'Frankenstein');  INSERT INTO BOOKS VALUES('P02', 'The Selfish Gene');</pre>
MAGAZINES	<pre>CREATE TABLE MAGAZINES (   PubID CHAR(3),   Issue INT,   Title VARCHAR(100),   PRIMARY KEY (PubID),   FOREIGN KEY(PubID) REFERENCES PUBLICATION(PubID) );</pre>	<pre>INSERT INTO MAGAZINES VALUES('P11', 1, 'Readers' Digest');  INSERT INTO MAGAZINES VALUES('P12', 2, 'Getting into Oxford');</pre>

	);	
ORDERS2	<pre>CREATE TABLE ORDERS2 (     OrderID CHAR(3),     Date_time DATE NOT NULL,     Shipping_address     VARCHAR(100) NOT NULL,     CID CHAR(3) NOT NULL,     PRIMARY KEY (OrderID),     UNIQUE (CID, Date_time),     FOREIGN KEY(CID) REFERENCES     CUSTOMERS(CID) );</pre>	<pre>INSERT INTO ORDERS2 VALUES('001', '2022-08-05', 'Jurong west st 52, 640536', 'C05');  INSERT INTO ORDERS2 VALUES('002', '2022-08-07', '50 Nanyang Ave, 639798', 'C03');</pre>
EMPLOYEES	<pre>CREATE TABLE EMPLOYEES (     EID CHAR(3),     Name VARCHAR(50),     Salary DECIMAL(10,2),     PRIMARY KEY (EID) );</pre>	<pre>INSERT INTO EMPLOYEES VALUES('E01', 'Tom Dickinson', 42000.0);  INSERT INTO EMPLOYEES VALUES('E02', 'Dick Harry', 69000.0);</pre>
COMPLAINTS	<pre>CREATE TABLE COMPLAINTS (     ID CHAR(3),     Text TEXT,     Filed_date_time DATE NOT     NULL,     EID CHAR(3) NULL,     Handled_date_time DATE NULL,     CID CHAR(3),     PRIMARY KEY (ID),     UNIQUE (CID,     Filed_date_time),     UNIQUE (EID,     Handled_date_time),     FOREIGN KEY(CID) REFERENCES     CUSTOMERS(CID),     FOREIGN KEY(EID) REFERENCES     EMPLOYEES(EID) );</pre>	<pre>INSERT INTO COMPLAINTS VALUES ('Z01', 'Most unaesthetic bookstore out there!', '2022-09-10', 'E03', '2022-09-11', 'C01');  INSERT INTO COMPLAINTS VALUES ('Z02', 'Book delivery took too long', '2022-09-15', 'E04', '2022-09-16', 'C02');</pre>

COMPLAINTS_ON_BOOKSTORE	<pre>CREATE TABLE COMPLAINTS_ON_BOOKSTORE (     ComplaintID CHAR(3),     BookstoreID CHAR(3) NOT NULL,     PRIMARY KEY (ComplaintID),     FOREIGN KEY(ComplaintID) REFERENCES COMPLAINTS(ID),     FOREIGN KEY(BookstoreID) REFERENCES BOOKSTORE(ID) );</pre>	<pre>INSERT INTO COMPLAINTS_ON_BOOKSTORE VALUES ('Z01', 'B01');  INSERT INTO COMPLAINTS_ON_BOOKSTORE VALUES ('Z03', 'B02');</pre>
COMPLAINTS_ON_ORDERS	<pre>CREATE TABLE COMPLAINTS_ON_ORDERS (     ComplaintID CHAR(3),     OrderID CHAR(3) NOT NULL,     PRIMARY KEY (ComplaintID),     FOREIGN KEY(ComplaintID) REFERENCES COMPLAINTS(ID),     FOREIGN KEY(OrderID) REFERENCES ORDERS2(OrderID) );</pre>	<pre>INSERT INTO COMPLAINTS_ON_ORDERS VALUES ('Z01', 'O01');  INSERT INTO COMPLAINTS_ON_ORDERS VALUES ('Z02', 'O02');</pre>
STOCKS_IN_BOOKSTORE	<pre>CREATE TABLE STOCKS_IN_BOOKSTORE (     Stock_ID CHAR(3),     BookstoreID CHAR(3) NOT NULL,     PubID CHAR(3) NOT NULL,     Stock_price DECIMAL(10,2),     Stock_qty INT NOT NULL,     PRIMARY KEY (Stock_ID),     UNIQUE (BookstoreID, PubID),     FOREIGN KEY (BookstoreID) REFERENCES BOOKSTORE(ID),     FOREIGN KEY (PubID) REFERENCES PUBLICATION(PubID) );</pre>	<pre>INSERT INTO STOCKS_IN_BOOKSTORE VALUES('S01', 'B01', 'P07', 69.99, 42);  INSERT INTO STOCKS_IN_BOOKSTORE VALUES('S03', 'B02', 'P03', 42.99, 30);</pre>
PRICE_HISTORY	<pre>CREATE TABLE PRICE_HISTORY (     Stock_ID CHAR(3),     Price DECIMAL(10,2) NOT NULL,</pre>	<pre>INSERT INTO PRICE_HISTORY VALUES('S01', 80.99, '2021-01-01', '2022-08-01');</pre>

	<pre> Start_date DATE NOT NULL, End_date DATE, PRIMARY KEY (Stock_ID, Start_date), UNIQUE (Stock_ID, End_date), FOREIGN KEY (Stock_ID) REFERENCES STOCKS_IN_BOOKSTORE(Stock_ID), CONSTRAINT chk_date CHECK (End_date &gt;= Start_date OR End_date IS NULL) ); </pre>	<pre> INSERT INTO PRICE_HISTORY VALUES('S01', 74.99, '2022-08-01', '2022-08-03'); </pre>
COMPLAINT_STATUS	<pre> CREATE TABLE COMPLAINT_STATUS ( Complaint_ID CHAR(3), Date DATETIME NOT NULL, State VARCHAR(50) NOT NULL, PRIMARY KEY (Complaint_ID, Date), FOREIGN KEY (Complaint_ID) REFERENCES COMPLAINTS(ID) ); </pre>	<pre> INSERT INTO COMPLAINT_STATUS VALUES ('Z01', '2022-09-10 10:50:09', 'Being processed');  INSERT INTO COMPLAINT_STATUS VALUES ('Z01', '2022-09-11 00:00:00', 'Addressed'); </pre>
ITEMS_IN_ORDERS_1	<pre> CREATE TABLE ITEMS_IN_ORDERS_1 ( ItemID CHAR(3), StockID CHAR(3) NOT NULL, ItemPrice DECIMAL(10,2) NOT NULL, ItemQty INT NOT NULL, DeliveryDate DATE NOT NULL, OrderID CHAR(3) NOT NULL, PRIMARY KEY (ItemID), FOREIGN KEY (StockID) REFERENCES STOCKS_IN_BOOKSTORE(Stock_ID), FOREIGN KEY (OrderID) REFERENCES ORDERS2(OrderID) ); </pre>	<pre> INSERT INTO ITEMS_IN_ORDERS_1 VALUES ('001', '2022-06-01', 'Jurong west st 52, 640536', 'C05');  INSERT INTO ITEMS_IN_ORDERS_1 VALUES ('002', '2022-06-01', '50 Nanyang Ave, 639798', 'C03'); </pre>
ORDER_ITEM_STATUS	<pre> CREATE TABLE ORDER_ITEM_STATUS ( Item_ID CHAR(3), </pre>	<pre> INSERT INTO ORDER_ITEM_STATUS </pre>

	<pre> Date DATETIME NOT NULL, State VARCHAR(50) NOT NULL, PRIMARY KEY (Item_ID, Date), FOREIGN KEY (Item_ID) REFERENCES ITEMS_IN_ORDERS_1(ItemID) ); </pre>	<pre> VALUES ('I01', '2022-06-03 00:00:00.000', Delivered');  INSERT INTO ORDER_ITEM_STATUS VALUES('I02', '2022-06-03 00:00:00.000', Delivered'); </pre>
CUSTOMER_FEED BACK	<pre> CREATE TABLE CUSTOMER_FEEDBACK( ItemID CHAR(3), CID CHAR(3) NOT NULL, Date DATETIME NOT NULL, Comment TEXT, Rating INT, PRIMARY KEY (ItemID, CID, Date), FOREIGN KEY (ItemID) REFERENCES ITEMS_IN_ORDERS_1(ItemID), FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID) ); </pre>	<pre> INSERT INTO CUSTOMER_FEEDBACK VALUES('I01', 'C01', '2022-06-30 00:00:00.000', NULL, NULL);  INSERT INTO CUSTOMER_FEEDBACK VALUES('I03', 'C03', '2022-06-30 00:00:00.000', 'Good!', 4); </pre>
CUSTOMER_ORDE RS	<pre> CREATE TABLE CUSTOMER_ORDERS( OrderID CHAR(3), CID CHAR(3), PRIMARY KEY (OrderID), FOREIGN KEY (CID) REFERENCES CUSTOMERS(CID) ); </pre>	<pre> INSERT INTO CUSTOMER_ORDERS VALUES('O01', 'C01');  INSERT INTO CUSTOMER_ORDERS VALUES('O02', 'C02'); </pre>

## Appendix B - SQL Queries

1. Find the average price of "Harry Potter Finale" on Amazon from 1 August 2022 to 31 August 2022.

-- This code finds the average price of "Harry Potter Finale" on Amazon from 1 August 2022 to 31 August 2022.

### Code:

```
-- Obtain the average price
SELECT AVG(A.Price) AS avgPrice
FROM
-- A relation containing all the Prices of the book from 1 August 2022
-- to 31 August 2022
(SELECT PH.Price
   FROM PRICE_HISTORY AS PH JOIN STOCKS_IN_BOOKSTORE AS SIB
   ON PH.Stock_ID = SIB.Stock_ID
   WHERE PH.Start_date >= '2022-08-01 00:00:00' AND (PH.End_date <=
'2022-08-31 23:59:59' OR PH.End_date is NULL) AND SIB.PubID = (SELECT
PubID
   FROM BOOKS WHERE Title = 'Harry Potter Finale')) AS A;
```

### Final Result:

	avgPrice
1	69.240000

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

-- This SQL code finds the PUBLICATIONS that received at least 10 ratings of “5” August 2022, and rank them by their average ratings.  
-- The relations used here are: ITEMS\_IN\_ORDERS\_1, CUSTOMER\_FEEDBACK, STOCKS\_IN\_BOOKSTORE  
-- The WHERE clause joins these tables on their corresponding IDs, and filters only the books rated 5 and above in August 2022.  
-- Finally you get the answer as seen in the table.

**Code:**

```
-- Create view for publications that have at least 10 ratings of 5 in
august 2022
CREATE VIEW pub_10_ratings_5 AS
SELECT T.PubID
FROM (
SELECT SIB.PubID, CF.ItemID
FROM
ITEMS_IN_ORDERS_1 AS II01 JOIN CUSTOMER_FEEDBACK AS CF ON II01.ItemID =
CF.ItemID
JOIN STOCKS_IN_BOOKSTORE AS SIB ON II01.StockID = SIB.Stock_ID
WHERE CF.Rating = 5 AND CF.DateTime > '2022.08.01 00:00:00' AND
CF.DateTime <= '2022.08.31 23:59:59'
) AS T
GROUP BY T.PubID
HAVING COUNT(*) >= 10
```

pub\_10\_ratings\_5

	PubID
1	P05
2	P07

```
-- Obtain the average ratings by taking all the PubIDs satisfying the
conditions, in STOCKS_IN_BOOKSTORE
SELECT SIB.PubID, AVG(CF.Rating) AS AvgRating
FROM STOCKS_IN_BOOKSTORE AS SIB
JOIN ITEMS_IN_ORDERS_1 AS II01 ON II01.StockID = SIB.Stock_ID
```



```
JOIN CUSTOMER_FEEDBACK AS CF ON CF.ItemID = II01.ItemID
WHERE SIB.PubID IN (SELECT PubID FROM pub_10_ratings_5)
GROUP BY SIB.PubID
-- Rank them in Descending order
ORDER BY AvgRating DESC;
```

**Final Result:**

	PubID	AvgRating
1	P05	5
2	P07	4

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

-- This SQL code finds the average time from ordering date to the delivery date.  
-- The relations used here are: ORDERS2, ITEMS\_IN\_ORDERS\_1, STOCKS\_IN\_BOOKSTORE  
-- We initially join ORDERS2 and ITEMS\_IN\_ORDERS\_1 to get when a particular StockID was ordered and when it was delivered together with STOCKS\_IN\_BOOKSTORE to get the PubID. The WHERE clause helps us filter the order dates within the june period. Next, we calculated the difference in order date and the delivery date to get delivery\_hours and divided it by the number of deliveries for each publication.  
-- Finally you get the answer as seen in the table.

**Code:**

```
SELECT PubID, COUNT(PubID) deliveries, SUM(delivery_hours) totalhours,
SUM(delivery_hours)/COUNT(PubID) average_time_in_hours
FROM(
SELECT PubID, DATEDIFF(hour, Date_time, DeliveryDate) delivery_hours
FROM (select i.StockID, Date_time, DeliveryDate
FROM ORDERS2 as o JOIN ITEMS_IN_ORDERS_1 as i
ON o.OrderID = i.OrderID) AS a JOIN STOCKS_IN_BOOKSTORE AS b
ON a.StockID = b.Stock_ID
WHERE Date_time >= '2022-06-01' AND Date_time <= '2022-06-30'
) AS A
GROUP BY PubID
```

**Final Result:**

	PubID	deliveries	totalhours	average_time_in_hours
1	P03	1	37	37
2	P05	1	60	60
3	P06	10	695	69
4	P07	2	90	45

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.*

-- This SQL code finds the employee with the smallest latency.  
-- First (Common Table Expression) CTE OpenedComplaints creates a table with Complaint\_ID and the Smaller date among the filed date and the Addressed date, to select the filed date. This hence results in a table of complaint\_ID and their filed date.  
-- The second CTE ProcessedComplaints calculates the time taken to process each complaint, by subtracting the date the complaint was opened from the date it was addressed, Grouped by Employee ID and Complaint\_ID.  
-- The third CTE Latency will then take the values from ProcessedComplaints, and find the average number of days taken by each EID, from opening of complaint to addressing of complaint, and group the result by EID.  
-- Finally, the query finds the Employer name by joining EMPLOYEES, from the corresponding EID, where the Latency is the smallest, and lists the EID, Name and Latency of the employer.

**Code:**

```
WITH OpenedComplaints AS (  
    SELECT  
        Complaint_ID,  
        MIN(Date) AS Date  
    FROM  
        COMPLAINT_STATUS  
    WHERE  
        State = 'Addressed'  
    GROUP BY  
        Complaint_ID  
),  
ProcessedComplaints AS (  
    SELECT  
        c.EID,  
        DAY((SELECT Date FROM COMPLAINT_STATUS WHERE State='Addressed' AND  
Complaint_ID=cs.Complaint_ID)) - DAY((SELECT Date FROM COMPLAINT_STATUS  
WHERE State='Being processed' AND Complaint_ID=cs.Complaint_ID)) AS  
DaysTaken  
    FROM  
        COMPLAINT_STATUS cs
```

```

        INNER JOIN OpenedComplaints opened ON cs.Complaint_ID =
opened.Complaint_ID
        INNER JOIN COMPLAINTS c ON cs.Complaint_ID = c.ID
    WHERE
        cs.State = 'Being processed'
    GROUP BY
        c.EID,
        cs.Complaint_ID
),
Latency AS (
    SELECT
        EID,
        AVG(DaysTaken) AS AvgLatency
    FROM ProcessedComplaints
    GROUP BY EID
)
SELECT
    L.EID,
    E.Name,
    L.AvgLatency
FROM
    Latency L
    INNER JOIN EMPLOYEES E ON L.EID = E.EID
WHERE
    AvgLatency = (SELECT MIN(AvgLatency) FROM Latency);

```

### Final Result:

Results		Messages	
	EID	Name	AvgLatency
1	E04	Mark Wahlberg	1

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 Amazon that sell them.

(i)

We obtain the publication IDs from PUBLICATION, and their titles from BOOK and MAGAZINE. Then we can simply select the publications published by Nanyang Publisher Company.

**Code:**

```
SELECT P.PubID, T.Title, Publisher
FROM PUBLICATION P, (SELECT * FROM BOOKS UNION SELECT PubID, Title FROM
MAGAZINES) AS T
WHERE Publisher = 'Nanyang Publisher Company'
AND P.PubID = T.PubID
```

**Final Result:**

	PubID	Title	Publisher
1	P03	Organic Chemistry	Nanyang Publisher Company
2	P08	Getting into Oxford	Nanyang Publisher Company
3	P09	U Magazine	Nanyang Publisher Company
4	P10	Student Wellbeing	Nanyang Publisher Company
5	P11	Paris Fashion Week 2023	Nanyang Publisher Company

(ii)

We first select the PubID and title from the previous query as a temporary view NanyangPubs. We can then use NanyangPubs to find the number of distinct bookstores that sell the publication by grouping and aggregation.

**Code:**

```
WITH NanyangPubs AS(
SELECT P.PubID, T.Title
FROM PUBLICATION P, (SELECT * FROM BOOKS UNION SELECT PubID, Title FROM
MAGAZINES) AS T
WHERE Publisher = 'Nanyang Publisher Company'
AND P.PubID = T.PubID)

SELECT N.PubID, N.Title, COUNT(DISTINCT S.BookstoreID) AS NumBookstores
FROM STOCKS_IN_BOOKSTORE S, NanyangPubs N
WHERE S.PubID = N.PubID
```

**Final Result:**

	PubID	Title	NumBookstores
1	P03	Organic Chemistry	5
2	P08	Getting into Oxford	1
3	P09	U Magazine	2
4	P10	Student Wellbeing	1
5	P11	Paris Fashion Week 2023	1

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

Create View for Orders in August 2022

```
-- Orders in Aug 2022
CREATE VIEW orders_in_aug AS
SELECT OrderID
FROM ORDERS2
WHERE Date_time >= '2022.08.01 00:00:00' AND Date_time <= '2022.08.31
23:59:59';
```

orders\_in\_aug (showing only a few):

	OrderID
1	O45
2	O48
3	O46
4	O47
5	O51
6	O50

```
-- Table containing the Bookstores that sold in August 2022, and their
total revenues in Aug 2022
CREATE VIEW total_rev_1 AS(
SELECT SIB.BookstoreID, SUM(II01.ItemPrice * II01.ItemQty) AS
totalRevenue
FROM STOCKS_IN_BOOKSTORE AS SIB JOIN ITEMS_IN_ORDERS_1 AS II01 ON
SIB.Stock_ID = II01.StockID
WHERE II01.OrderID IN (SELECT OrderID FROM orders_in_aug)
GROUP BY SIB.BookstoreID)
```

total\_rev\_1:

	BookstoreID	totalRevenue
1	B01	649.50
2	B02	949.05
3	B03	1708.10
4	B04	2362.03

Then, we execute the query to find the bookstore with the most revenue:

```
SELECT T.BookstoreID, T.totalRevenue
FROM total_rev_1 AS T
```

```
WHERE T.totalRevenue = (SELECT MAX(totalRevenue) FROM total_rev_1);
```

**Final Result:**

	BookstoreID	totalRevenue
1	B04	2362.03



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

```
-- This SQL code finds the most expensive publication he/she has
purchased, for the customers with the most number of complaints.
-- The relations used here are COMPLAINTS, CUSTOMER_ORDERS,
ITEMS_IN_ORDERS_1 and STOCKS_IN_BOOKSTORE.
-- First, we create view noisy_customer to get the customer with the
most complaints, then we create another view stocks_bought_by_noisy
to get the StockID of the stuff he bought. Next, we join
stocks_bought_by_noisy with STOCKS_IN_BOOKSTORE to get the PubID of
the most expensive publication he purchased.
-- Finally you get the answer as seen in the table.
```

#### **Code:**

##### **NOISIEST CUSTOMER**

```
CREATE VIEW noisy_customer as
SELECT CID
FROM COMPLAINTS
GROUP BY CID
HAVING COUNT(*) =(
SELECT MAX(Number_of_complaints)
FROM (
SELECT CID,COUNT(*) AS Number_of_complaints
FROM COMPLAINTS
GROUP BY CID)
COMPLAINTS)
```

##### **STUFF HE BOUGHT**

```
CREATE VIEW stocks_bought_by_noisy AS
SELECT StockID, ItemPrice
FROM ((SELECT o.CID, o.OrderID
FROM CUSTOMER_ORDERS AS o JOIN noisy_customer AS c ON o.CID = c.CID) AS A
JOIN ITEMS_IN_ORDERS_1 AS B
ON A.OrderID = B.OrderID)
```

##### **MOST EXPENSIVE PUBLICATION**

```
SELECT PubID
FROM(
```

```
(SELECT DISTINCT StockID
FROM stocks_bought_by_noisy
WHERE ItemPrice = (SELECT MAX(ItemPrice) FROM stocks_bought_by_noisy)) AS A
JOIN
STOCKS_IN_BOOKSTORE AS B
ON A.StockID = B.Stock_ID)
```

**Final Result:**

Results		Messages	
	PubID		
1	P07		

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.*

Firstly, we can find the publications in August. This is the temporary view AugPurchases, which uses the item Stock ID and Order date-time to find the number of purchases of each publication in August, and PubID and ItemQty to track their purchases. The output of this view is as follows:

	PubID	Purchases
1	P03	95
2	P05	197
3	P07	190
4	P11	50

With this view, we can find the top 3 publications in August.

	PubID
1	P03
2	P05
3	P07

Next, we can find publications that are not purchased in July using NOT EXISTS.

	PubID
1	P01
2	P02
3	P03
4	P04
5	P07
6	P08
7	P09
8	P10
9	P11
10	P12

Finally, we INTERSECT the previous 2 relations to give the final output.

**Code:**

```
WITH AugPurchases AS
```

```

(SELECT S.PubID AS PubID, SUM(I.ItemQty) AS Purchases
FROM ITEMS_IN_ORDERS_1 I, ORDERS2 O, STOCKS_IN_BOOKSTORE S
WHERE S.Stock_ID = I.StockID
AND I.OrderID = O.OrderID
AND O.Date_time >= '2022-08-01' AND O.Date_time <= '2022-08-31'
GROUP BY S.PubID)

SELECT A.PubID
FROM AugPurchases A
WHERE
    (SELECT COUNT(*)
     FROM AugPurchases B
     WHERE A.Purchases < B.Purchases) < 3 /*Assuming that top 3 is by
position*/

INTERSECT

SELECT P.PubID
FROM PUBLICATION P
WHERE NOT EXISTS
(SELECT S7.PubID
FROM ITEMS_IN_ORDERS_1 I7, ORDERS2 O7, STOCKS_IN_BOOKSTORE S7
WHERE S7.Stock_ID = I7.StockID
AND I7.OrderID = O7.OrderID
AND O7.Date_time >= '2022-07-01' AND O7.Date_time <= '2022-07-31'
AND S7.PubID = P.PubID)

```

**Final Result:**

	PubID
1	P03
2	P07

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

-- This SQL code finds the publications that are increasingly being purchased over at least 3 months.  
-- In the first Common Table Expression MonthlyPurchases, ORDERS2, ITEM\_IN\_ORDERS\_1 and STOCKS\_IN\_BOOKSTORE are joined, to get the pubID from STOCKS\_IN\_BOOKSTORE, month of purchase from ORDERS2, and quantity sold from ITEM\_IN\_ORDERS\_1.  
-- The second subquery Assigns a row number 'MonthNo' that will be used to check the criteria.  
-- The query at last uses the Occurrence CTE to find the publications that satisfies the criteria. OC3 is used to represent the current month, OC2 representing last month, and OC1 representing the month before the previous.  
-- OC2 ON OC1.PubID = OC2.PubID AND OC1.MonthNo = OC2.MonthNo - 1, OC1.MonthNo = OC3.MonthNo - 2 Ensures the previous condition is met.  
-- OC2.TotalSale > OC1.TotalSale and OC3.TotalSales > OC2.TotalSales ensure the sales of publication are increasing over three months.  
-- At the end, the Distinct publications that are increasingly being purchased over at least 3 months will be displayed.

**Code:**

```
WITH MonthlyPurchases AS (  
    SELECT s.PubID, CONVERT(VARCHAR(7), O.Date_time, 120) AS Date,  
    SUM(io.ItemQty) AS Purchases  
    FROM ORDERS2 O  
    JOIN ITEMS_IN_ORDERS_1 io ON O.OrderID = io.OrderID  
    JOIN STOCKS_IN_BOOKSTORE s ON io.StockID = s.Stock_ID  
    GROUP BY s.PubID, CONVERT(VARCHAR(7), O.Date_time, 120)  
),  
occurrence AS (  
    SELECT mp.PubID,  
    mp.Date,  
    SUM(mp.Purchases) AS TotalSales,  
    ROW_NUMBER() OVER (  
        PARTITION BY mp.PubID  
        ORDER BY mp.Date  
    ) AS MonthNo  
    FROM MonthlyPurchases mp  
    GROUP BY mp.PubID, mp.Date  
)  
SELECT DISTINCT OC3.PubID, COALESCE( b.Title, m.Title) AS Title
```

```

FROM occurrence OC1
  JOIN occurrence OC2 ON OC1.PubID = OC2.PubID AND OC1.MonthNo =
OC2.MonthNo - 1
  JOIN occurrence OC3 ON OC1.PubID = OC3.PubID AND OC1.MonthNo =
OC3.MonthNo - 2
  LEFT JOIN Books b ON OC1.PubID = b.PubID
  LEFT JOIN Magazines m ON OC1.PubID = m.PubID
WHERE OC3.TotalSales > OC2.TotalSales
      AND OC2.TotalSales > OC1.TotalSales;

```

### **Final Result:**

Results		Messages
	PubID	Title
1	P05	How to Survive SCSE

## Print-out of Table Records:

### 1. BOOKSTORE

Results		Messages
	ID	
1	B01	
2	B02	
3	B03	
4	B04	
5	B05	

### 2. CUSTOMERS

Results			Messages
	CID	Name	
1	C01	Sarah Lim Xi Liu	
2	C02	Mohammad Ali	
3	C03	John Cena	
4	C04	Hannah Montana	
5	C05	Anonymous Tiger	

### 3. PUBLICATION

<div><div><div>Results</div><div>Messages</div></div></div>			
	PubID	Publisher	Year
1	P01	Oxford	2000
2	P02	Vogue	2010
3	P03	Nanyang Publisher Company	2020
4	P04	Dolphin	2022
5	P05	Pearson	2023
6	P06	Penguin	2010
7	P07	Lanny	2015
8	P08	Nanyang Publisher Company	2021
9	P09	Nanyang Publisher Company	2022
10	P10	Nanyang Publisher Company	2022
11	P11	Nanyang Publisher Company	2023
12	P12	Trusted Media Brands	2023

### 4. BOOKS

<div><div><div>Results</div><div>Messages</div></div></div>		
	PubID	Title
1	P01	Frankenstein
2	P02	The Selfish Gene
3	P03	Organic Chemistry
4	P04	Secrets of Singapore
5	P05	How to Survive SCSE
6	P06	Harry Porter and the Bellhop
7	P07	Harry Porter Finale



## 5. MAGAZINES

Results Messages			
	PubID	Issue	Title
1	P08	2	Getting into Oxford
2	P09	3	U Magazine
3	P10	4	Student Wellbeing
4	P11	5	Paris Fashion Week 2023
5	P12	1	Readers Digest

## 6. EMPLOYEES

Results Messages			
	EID	Name	Salary
1	E01	Tom Dickinson	42000.00
2	E02	Dick Harry	69000.00
3	E03	Coco Chanel	42069.00
4	E04	Mark Walhberg	123456.00
5	E05	Thomas Lemar	69696.00

## 7. COMPLAINTS

Results Messages						
	ID	Text	Filed_date_time	EID	Handled_date_time	CID
1	Z01	Most aesthetic bookstore out there!	2022-09-10 10:34:09.000	E03	2022-09-12 00:00:00.000	C01
2	Z02	Book delivery took too long	2022-09-15 10:34:09.000	E04	2022-09-16 10:34:09.000	C02
3	Z03	Bookstore was very disorganised	2022-09-10 10:34:09.000	E01	2022-09-17 11:20:00.000	C04
4	Z04	Order did not arrive on time	2022-08-25 10:34:09.000	E02	2022-08-27 17:24:09.000	C03
5	Z05	Bookstore too big	2022-08-20 11:30:10.000	E05	2022-08-25 22:34:09.000	C01
6	Z06	Salesperson was rude	2022-08-21 10:34:09.000	NULL	NULL	C01

## 8. ORDERS2

	OrderID	Date_time	Shipping_address	CID
1	O01	2022-06-01 10:34:09.000	Jurong west st 52, 640536	C05
2	O02	2022-06-01 11:34:09.000	50 Nanyang Ave, 639798	C03
3	O03	2022-06-01 20:34:09.000	Pasir Laba Road, 637901	C04
4	O04	2022-06-01 12:34:09.000	SAFTI Military Institute, 634364	C01
5	O05	2022-06-02 13:34:09.000	97 Cashew Rd, 679668	C02
6	O06	2022-06-02 00:00:00.000	SAFTI Military Institute, 634364	C01
7	O07	2022-06-02 13:34:09.000	SAFTI Military Institute, 634364	C01
8	O08	2022-06-02 12:34:09.000	SAFTI Military Institute, 634364	C01
9	O09	2022-06-03 20:34:09.000	SAFTI Military Institute, 634364	C01
10	O10	2022-06-03 10:20:20.000	Jurong west st 52, 640536	C05
11	O11	2022-06-03 13:34:23.000	SAFTI Military Institute, 634364	C01
12	O12	2022-06-03 16:59:10.000	SAFTI Military Institute, 634364	C01
13	O13	2022-06-04 01:10:00.000	SAFTI Military Institute, 634364	C01
14	O14	2022-06-04 23:51:27.000	SAFTI Military Institute, 634364	C01
15	O15	2022-07-04 12:20:12.000	SAFTI Military Institute, 634364	C01
16	O16	2022-07-04 13:13:58.000	Jurong West st 52, 640536	C05
17	O17	2022-07-05 11:54:12.000	Jurong West st 52, 640536	C05
18	O18	2022-07-05 12:01:12.000	Jurong West st 52, 640536	C05
19	O19	2022-07-05 13:21:32.000	Jurong West st 52, 640536	C05
20	O20	2022-07-05 23:53:44.000	Jurong West st 52, 640536	C05
21	O21	2022-07-06 15:01:12.000	Jurong West st 52, 640536	C05
22	O22	2022-07-06 12:23:12.000	Jurong West st 52, 640536	C05
23	O23	2022-07-06 12:01:54.000	Jurong West st 52, 640536	C05
24	O24	2022-07-06 03:01:12.000	Jurong West st 52, 640536	C05
25	O25	2022-07-07 12:12:54.000	Jurong West st 52, 640536	C05
26	O26	2022-07-07 12:01:32.000	Jurong West st 52, 640536	C05
27	O27	2022-07-07 19:42:12.000	Jurong West st 52, 640536	C05
28	O28	2022-07-07 18:01:12.000	Jurong West st 52, 640536	C05
29	O29	2022-07-08 12:22:12.000	Jurong West st 52, 640536	C05
30	O30	2022-07-08 12:01:43.000	Jurong West st 52, 640536	C05
31	O31	2022-07-08 20:43:12.000	Jurong West st 52, 640536	C05
32	O32	2022-07-08 21:01:12.000	Jurong West st 52, 640536	C05
33	O33	2022-07-08 22:22:22.000	Jurong West st 52, 640536	C05
34	O34	2022-07-09 13:11:33.000	Jurong West st 52, 640536	C05
35	O35	2022-07-09 14:01:12.000	Jurong West st 52, 640536	C05

36	O36	2022-07-09 01:01:12.000	Jurong West st 52, 640536	C05
37	O37	2022-07-10 02:17:12.000	Jurong West st 52, 640536	C05
38	O38	2022-07-10 03:55:12.000	Jurong West st 52, 640536	C05
39	O39	2022-07-10 04:25:12.000	Jurong West st 52, 640536	C05
40	O40	2022-07-11 05:22:12.000	Jurong West st 52, 640536	C05
41	O41	2022-07-11 07:01:12.000	Jurong West st 52, 640536	C05
42	O42	2022-07-11 08:01:12.000	Jurong West st 52, 640536	C05
43	O43	2022-07-12 14:01:12.000	Jurong West st 52, 640536	C05
44	O44	2022-07-12 15:10:54.000	Jurong West st 52, 640536	C05
45	O45	2022-08-12 12:50:06.000	Jurong West st 52, 640536	C05
46	O46	2022-08-13 12:49:17.000	Jurong West st 52, 640536	C05
47	O47	2022-08-13 13:48:01.000	Jurong West st 52, 640536	C05
48	O48	2022-08-13 12:47:02.000	Jurong West st 52, 640536	C05
49	O49	2022-08-14 16:46:03.000	Jurong West st 52, 640536	C05
50	O50	2022-08-14 12:45:04.000	Jurong West st 52, 640536	C05
51	O51	2022-08-14 12:44:05.000	Jurong West st 52, 640536	C05
52	O52	2022-08-15 17:43:06.000	Jurong West st 52, 640536	C05
53	O53	2022-08-15 18:42:07.000	Jurong West st 52, 640536	C05
54	O54	2022-08-15 19:41:08.000	Jurong West st 52, 640536	C05
55	O55	2022-08-16 20:40:09.000	Jurong West st 52, 640536	C05
56	O56	2022-08-16 21:39:10.000	Jurong West st 52, 640536	C05
57	O57	2022-08-16 22:38:11.000	Jurong West st 52, 640536	C05
58	O58	2022-08-17 23:37:12.000	Jurong West st 52, 640536	C05
59	O59	2022-08-17 00:36:13.000	Jurong West st 52, 640536	C05
60	O60	2022-08-17 04:35:14.000	Jurong West st 52, 640536	C05
61	O61	2022-08-18 05:34:15.000	Jurong West st 52, 640536	C05
62	O62	2022-08-18 06:33:16.000	Jurong West st 52, 640536	C05
63	O63	2022-08-18 07:32:17.000	Jurong West st 52, 640536	C05
64	O64	2022-08-19 08:31:18.000	Jurong West st 52, 640536	C05
65	O65	2022-08-19 09:30:19.000	Jurong West st 52, 640536	C05
66	O66	2022-08-19 10:29:20.000	Jurong West st 52, 640536	C05
67	O67	2022-08-20 11:28:21.000	Jurong West st 52, 640536	C05
68	O68	2022-08-20 12:27:22.000	Jurong West st 52, 640536	C05
69	O69	2022-08-20 13:26:23.000	Jurong West st 52, 640536	C05
70	O70	2022-08-21 14:25:24.000	Jurong West st 52, 640536	C05
71	O71	2022-08-21 15:24:25.000	Jurong West st 52, 640536	C05

72	O72	2022-08-22 16:23:26.000	Jurong West st 52, 640536	C05
73	O73	2022-08-22 17:22:27.000	Jurong West st 52, 640536	C05
74	O74	2022-08-23 18:21:28.000	Jurong West st 52, 640536	C05
75	O75	2022-08-23 19:19:29.000	Jurong West st 52, 640536	C05
76	O76	2022-08-24 20:18:30.000	Jurong West st 52, 640536	C05
77	O77	2022-08-24 21:17:31.000	Jurong West st 52, 640536	C05
78	O78	2022-08-25 22:16:32.000	Jurong West st 52, 640536	C05
79	O79	2022-08-25 23:15:33.000	Jurong West st 52, 640536	C05
80	O80	2022-08-26 00:14:34.000	Jurong West st 52, 640536	C05
81	O81	2022-08-26 01:13:35.000	Jurong West st 52, 640536	C05
82	O82	2022-08-27 02:12:36.000	Jurong West st 52, 640536	C05
83	O83	2022-08-27 03:23:37.000	Jurong West st 52, 640536	C05
84	O84	2022-08-28 04:54:38.000	Jurong West st 52, 640536	C05
85	O85	2022-08-28 05:22:39.000	Jurong West st 52, 640536	C05
86	O86	2022-08-29 06:21:40.000	Jurong West st 52, 640536	C05
87	O87	2022-08-29 07:51:41.000	Jurong West st 52, 640536	C05

## 9. COMPLAINTS\_ON\_BOOKSTORE

Results Messages		
	ComplaintID	BookstoreID
1	Z01	B01
2	Z03	B02
3	Z05	B03
4	Z06	B03

## 10. COMPLAINTS\_ON\_ORDERS

Results Messages		
	ComplaintID	OrderID
1	Z01	O01
2	Z02	O02

## 11. STOCKS\_IN\_BOOKSTORE

	Stock_ID	BookstoreID	PubID	Stock_price	Stock_qty
1	S01	B01	P07	60.99	42
2	S02	B02	P03	9.99	30
3	S03	B03	P07	8.99	10
4	S04	B04	P05	20.99	20
5	S05	B05	P06	42.00	69
6	S06	B01	P08	9.99	50
7	S07	B05	P09	10.99	14
8	S08	B01	P09	9.99	10
9	S09	B01	P11	12.99	12
10	S10	B02	P10	10.99	22
11	S11	B01	P03	40.99	6
12	S12	B03	P03	41.99	19
13	S13	B04	P03	40.99	7
14	S14	B05	P03	39.99	1

## 12. PRICE\_HISTORY

Results		Messages		
	Stock_ID	Price	Start_date	End_date
1	S01	80.99	2021-01-01	2022-08-01
2	S01	74.99	2022-08-01	2022-08-03
3	S01	69.99	2022-08-15	NULL
4	S03	70.99	2022-08-01	2022-08-18
5	S03	60.99	2022-08-18	NULL

### 13.COMPLAINT\_STATUS

<div> <div>Results</div> <div>Messages</div> </div>			
	Complaint_ID	Date	State
1	Z01	2022-09-10 10:50:09.000	Being processed
2	Z01	2022-09-12 00:00:00.000	Addressed
3	Z02	2022-09-15 10:40:09.000	Being processed
4	Z02	2022-09-16 10:34:09.000	Addressed
5	Z03	2022-09-10 10:38:09.000	Being processed
6	Z03	2022-09-17 11:20:00.000	Addressed
7	Z04	2022-08-25 10:35:09.000	Being processed
8	Z04	2022-08-27 17:24:09.000	Addressed
9	Z05	2022-08-20 11:35:10.000	Being processed
10	Z05	2022-08-25 22:34:09.000	Addressed
11	Z06	2022-08-25 10:34:09.000	Pending

# 14.ITEMS\_IN\_ORDERS\_1

	ItemID	StockID	ItemPrice	ItemQty	DeliveryDate	OrderID
1	I01	S01	60.99	10	2022-06-03 00:00:00.0...	O01
2	I02	S02	9.99	15	2022-06-03 00:00:00.0...	O02
3	I03	S03	8.99	20	2022-06-04 00:00:00.0...	O03
4	I04	S04	12.99	25	2022-06-04 00:00:00.0...	O04
5	I05	S05	11.99	30	2022-06-04 00:00:00.0...	O05
6	I06	S05	11.99	29	2022-06-04 00:00:00.0...	O06
7	I07	S05	11.99	28	2022-06-04 00:00:00.0...	O07
8	I08	S05	11.99	27	2022-06-07 00:00:00.0...	O08
9	I09	S05	11.99	26	2022-06-05 00:00:00.0...	O09
10	I10	S05	11.99	25	2022-06-05 00:00:00.0...	O10
11	I11	S05	11.99	24	2022-06-08 00:00:00.0...	O11
12	I12	S05	11.99	23	2022-06-08 00:00:00.0...	O12
13	I13	S05	11.99	22	2022-06-06 00:00:00.0...	O13
14	I14	S05	11.99	21	2022-06-11 00:00:00.0...	O14
15	I15	S05	11.99	20	2022-07-06 00:00:00.0...	O15
16	I16	S05	11.99	19	2022-07-06 00:00:00.0...	O16
17	I17	S05	11.99	18	2022-07-07 00:00:00.0...	O17
18	I18	S05	11.99	17	2022-07-07 00:00:00.0...	O18
19	I19	S05	11.99	16	2022-07-07 00:00:00.0...	O19
20	I20	S05	11.99	15	2022-07-11 00:00:00.0...	O20
21	I21	S05	11.99	14	2022-07-08 00:00:00.0...	O21
22	I22	S05	11.99	13	2022-07-08 00:00:00.0...	O22
23	I23	S05	11.99	12	2022-07-08 00:00:00.0...	O23
24	I24	S05	11.99	11	2022-07-11 00:00:00.0...	O24
25	I25	S05	11.99	10	2022-07-09 00:00:00.0...	O25
26	I26	S05	11.99	9	2022-07-09 00:00:00.0...	O26
27	I27	S05	11.99	8	2022-07-09 00:00:00.0...	O27
28	I28	S05	11.99	7	2022-07-10 00:00:00.0...	O28
29	I29	S05	11.99	6	2022-07-10 00:00:00.0...	O29
30	I30	S05	11.99	5	2022-07-10 00:00:00.0...	O30
31	I31	S05	11.99	4	2022-07-10 00:00:00.0...	O31
32	I32	S05	11.99	3	2022-07-10 00:00:00.0...	O32
33	I33	S05	11.99	2	2022-07-13 00:00:00.0...	O33
34	I34	S05	11.99	1	2022-07-11 00:00:00.0...	O34
35	I35	S04	12.99	24	2022-07-11 00:00:00.0...	O35
36	I36	S04	11.99	23	2022-07-13 00:00:00.0...	O36
37	I37	S04	11.99	22	2022-07-12 00:00:00.0...	O37
38	I38	S04	11.99	21	2022-07-12 00:00:00.0...	O38
39	I39	S04	11.99	20	2022-07-13 00:00:00.0...	O39
40	I40	S04	11.99	19	2022-07-13 00:00:00.0...	O40
41	I41	S04	11.99	18	2022-07-13 00:00:00.0...	O41
42	I42	S04	11.99	17	2022-07-15 00:00:00.0...	O42
43	I43	S04	11.99	16	2022-07-14 00:00:00.0...	O43
44	I44	S04	11.99	15	2022-07-14 00:00:00.0...	O44
45	I45	S04	11.99	14	2022-08-14 00:00:00.0...	O45
46	I46	S04	11.99	13	2022-08-15 00:00:00.0...	O46
47	I47	S04	11.99	12	2022-08-15 00:00:00.0...	O47
48	I48	S04	11.99	11	2022-08-18 00:00:00.0...	O48

49	I49	S04	11.99	10	2022-08-16 00:00:00.0...	O49
50	I50	S04	11.99	9	2022-08-16 00:00:00.0...	O50
51	I51	S04	11.99	100	2022-08-18 00:00:00.0...	O51
52	I52	S04	11.99	7	2022-08-17 00:00:00.0...	O52
53	I53	S04	11.99	6	2022-08-17 00:00:00.0...	O53
54	I54	S04	11.99	5	2022-08-18 00:00:00.0...	O54
55	I55	S04	11.99	4	2022-08-18 00:00:00.0...	O55
56	I56	S04	11.99	3	2022-08-18 00:00:00.0...	O56
57	I57	S04	11.99	2	2022-08-18 00:00:00.0...	O57
58	I58	S04	11.99	1	2022-08-19 00:00:00.0...	O58
59	I59	S03	8.99	19	2022-08-19 00:00:00.0...	O59
60	I60	S03	8.99	18	2022-08-25 00:00:00.0...	O60
61	I61	S03	8.99	17	2022-08-20 00:00:00.0...	O61
62	I62	S03	8.99	16	2022-08-20 00:00:00.0...	O62
63	I63	S03	8.99	15	2022-08-25 00:00:00.0...	O63
64	I64	S03	8.99	14	2022-08-21 00:00:00.0...	O64
65	I65	S03	8.99	13	2022-08-21 00:00:00.0...	O65
66	I66	S03	8.99	12	2022-08-25 00:00:00.0...	O66
67	I67	S03	8.99	11	2022-08-22 00:00:00.0...	O67
68	I68	S03	8.99	10	2022-08-22 00:00:00.0...	O68
69	I69	S03	8.99	9	2022-08-25 00:00:00.0...	O69
70	I70	S03	8.99	8	2022-08-23 00:00:00.0...	O70
71	I71	S03	8.99	7	2022-08-23 00:00:00.0...	O71
72	I72	S03	8.99	6	2022-08-24 00:00:00.0...	O72
73	I73	S03	8.99	5	2022-08-24 00:00:00.0...	O73
74	I74	S03	8.99	4	2022-08-25 00:00:00.0...	O74
75	I75	S03	8.99	3	2022-08-28 00:00:00.0...	O75
76	I76	S03	8.99	2	2022-08-26 00:00:00.0...	O76
77	I77	S03	8.99	1	2022-08-26 00:00:00.0...	O77
78	I78	S02	9.99	14	2022-08-28 00:00:00.0...	O78
79	I79	S02	9.99	13	2022-08-28 00:00:00.0...	O79
80	I80	S02	9.99	12	2022-08-28 00:00:00.0...	O80
81	I81	S02	9.99	11	2022-08-28 00:00:00.0...	O81
82	I82	S02	9.99	10	2022-08-29 00:00:00.0...	O82
83	I83	S02	9.99	9	2022-08-29 00:00:00.0...	O83
84	I84	S02	9.99	8	2022-08-30 00:00:00.0...	O84
85	I85	S02	9.99	7	2022-08-30 00:00:00.0...	O85
86	I86	S02	9.99	6	2022-09-02 00:00:00.0...	O86
87	I87	S02	9.99	5	2022-09-02 00:00:00.0...	O87
88	I88	S09	12.99	50	NULL	O88



## 15.ORDER\_ITEM\_STATUS

	Item_ID	Date	State
1	I01	2022-06-03 00:00:00.000	Delivered
2	I02	2022-06-03 00:00:00.000	Delivered
3	I03	2022-06-04 00:00:00.000	Delivered
4	I04	2022-06-04 00:00:00.000	Delivered
5	I05	2022-06-04 00:00:00.000	Delivered
6	I06	2022-06-04 00:00:00.000	Delivered
7	I07	2022-06-04 00:00:00.000	Delivered
8	I08	2022-06-07 00:00:00.000	Delivered
9	I09	2022-06-05 00:00:00.000	Delivered
10	I10	2022-06-05 00:00:00.000	Delivered
11	I11	2022-06-08 00:00:00.000	Delivered
12	I12	2022-06-08 00:00:00.000	Delivered
13	I13	2022-06-06 00:00:00.000	Delivered
14	I14	2022-06-11 00:00:00.000	Returned
15	I15	2022-07-06 00:00:00.000	Delivered
16	I16	2022-07-06 00:00:00.000	Delivered
17	I17	2022-07-07 00:00:00.000	Delivered
18	I18	2022-07-07 00:00:00.000	Delivered
19	I19	2022-07-07 00:00:00.000	Delivered
20	I20	2022-07-11 00:00:00.000	Delivered
21	I21	2022-07-08 00:00:00.000	Delivered
22	I22	2022-07-08 00:00:00.000	Delivered
23	I23	2022-07-08 00:00:00.000	Returned
24	I24	2022-07-11 00:00:00.000	Delivered
25	I25	2022-07-09 00:00:00.000	Delivered
26	I26	2022-07-09 00:00:00.000	Delivered
27	I27	2022-07-09 00:00:00.000	Delivered
28	I28	2022-07-10 00:00:00.000	Delivered
29	I29	2022-07-10 00:00:00.000	Delivered
30	I30	2022-07-10 00:00:00.000	Delivered
31	I31	2022-07-10 00:00:00.000	Delivered
32	I32	2022-07-10 00:00:00.000	Delivered
33	I33	2022-07-13 00:00:00.000	Delivered
34	I34	2022-07-11 00:00:00.000	Delivered
35	I35	2022-07-11 00:00:00.000	Delivered

36	I36	2022-07-13 00:00:00.000	Delivered
37	I37	2022-07-12 00:00:00.000	Delivered
38	I38	2022-07-12 00:00:00.000	Delivered
39	I39	2022-07-13 00:00:00.000	Delivered
40	I40	2022-07-13 00:00:00.000	Delivered
41	I41	2022-07-13 00:00:00.000	Delivered
42	I42	2022-07-15 00:00:00.000	Returned
43	I43	2022-07-14 00:00:00.000	Delivered
44	I44	2022-07-14 00:00:00.000	Delivered
45	I45	2022-08-14 00:00:00.000	Delivered
46	I46	2022-08-15 00:00:00.000	Delivered
47	I47	2022-08-15 00:00:00.000	Delivered
48	I48	2022-08-18 00:00:00.000	Delivered
49	I49	2022-08-16 00:00:00.000	Delivered
50	I50	2022-08-16 00:00:00.000	Delivered
51	I51	2022-08-18 00:00:00.000	Delivered
52	I52	2022-08-17 00:00:00.000	Delivered
53	I53	2022-08-17 00:00:00.000	Delivered
54	I54	2022-08-18 00:00:00.000	Delivered
55	I55	2022-08-18 00:00:00.000	Delivered
56	I56	2022-08-18 00:00:00.000	Delivered
57	I57	2022-08-18 00:00:00.000	Delivered
58	I58	2022-08-19 00:00:00.000	Delivered
59	I59	2022-08-19 00:00:00.000	Delivered
60	I60	2022-08-25 00:00:00.000	Delivered
61	I61	2022-08-20 00:00:00.000	Delivered
62	I62	2022-08-20 00:00:00.000	Delivered
63	I63	2022-08-25 00:00:00.000	Delivered
64	I64	2022-08-21 00:00:00.000	Delivered
65	I65	2022-08-21 00:00:00.000	Returned
66	I66	2022-08-25 00:00:00.000	Delivered
67	I67	2022-08-22 00:00:00.000	Delivered
68	I68	2022-08-22 00:00:00.000	Delivered
69	I69	2022-08-25 00:00:00.000	Delivered
70	I70	2022-08-23 00:00:00.000	Delivered
71	I71	2022-08-23 00:00:00.000	Delivered

72	172	2022-08-24 00:00:00.000	Delivered
73	173	2022-08-24 00:00:00.000	Delivered
74	174	2022-08-25 00:00:00.000	Delivered
75	175	2022-08-28 00:00:00.000	Delivered
76	176	2022-08-26 00:00:00.000	Delivered
77	177	2022-08-26 00:00:00.000	Delivered
78	178	2022-08-28 00:00:00.000	Delivered
79	179	2022-08-28 00:00:00.000	Delivered
80	180	2022-08-28 00:00:00.000	Returned
81	181	2022-08-28 00:00:00.000	Delivered
82	182	2022-08-29 00:00:00.000	Delivered
83	183	2022-08-29 00:00:00.000	Delivered
84	184	2022-08-30 00:00:00.000	Delivered
85	185	2022-08-30 00:00:00.000	Delivered
86	186	2022-09-02 00:00:00.000	Being Proce...
87	187	2022-09-02 00:00:00.000	Being Proce...
88	188	2022-09-03 00:00:00.000	Shipping

## 16.CUSTOMER\_FEEDBACK

	ItemID	CID	DateTime	Comment	Rating
1	I01	C01	2022-06-30 00:00:00.000	NULL	NULL
2	I02	C02	2022-06-30 00:00:00.000	NULL	NULL
3	I03	C03	2022-06-30 00:00:00.000	Good!	4
4	I04	C04	2022-06-30 00:00:00.000	Great!	5
5	I05	C05	2022-06-30 00:00:00.000	NULL	NULL
6	I06	C01	2022-06-30 00:00:00.000	Great!	5
7	I07	C01	2022-06-30 00:00:00.000	Great!	5
8	I08	C01	2022-06-30 00:00:00.000	Great!	5
9	I09	C01	2022-06-30 00:00:00.000	Great!	5
10	I10	C05	2022-06-30 00:00:00.000	Great!	5
11	I11	C01	2022-06-30 00:00:00.000	Great!	5
12	I12	C01	2022-06-30 00:00:00.000	Great!	5
13	I13	C01	2022-06-30 00:00:00.000	Great!	5
14	I14	C01	2022-06-30 00:00:00.000	Great!	5
15	I15	C01	2022-07-30 00:00:00.000	Great!	5
16	I16	C05	2022-07-30 00:00:00.000	Great!	5
17	I17	C05	2022-07-30 00:00:00.000	Great!	5
18	I18	C05	2022-07-30 00:00:00.000	Great!	5
19	I19	C05	2022-07-30 00:00:00.000	Great!	5
20	I20	C05	2022-07-30 00:00:00.000	Great!	5
21	I21	C05	2022-07-30 00:00:00.000	Great!	5
22	I22	C05	2022-07-30 00:00:00.000	Great!	5
23	I23	C05	2022-07-30 00:00:00.000	Great!	5
24	I24	C05	2022-07-30 00:00:00.000	Great!	5
25	I25	C05	2022-07-30 00:00:00.000	Great!	5
26	I26	C05	2022-07-30 00:00:00.000	Great!	5
27	I27	C05	2022-07-30 00:00:00.000	Great!	5
28	I28	C05	2022-07-30 00:00:00.000	Great!	5
29	I29	C05	2022-07-30 00:00:00.000	Great!	5
30	I30	C05	2022-07-30 00:00:00.000	Great!	5
31	I31	C05	2022-07-30 00:00:00.000	Great!	5
32	I32	C05	2022-07-30 00:00:00.000	Great!	5
33	I33	C05	2022-07-30 00:00:00.000	Great!	5
34	I34	C05	2022-07-30 00:00:00.000	Great!	5
35	I35	C05	2022-07-30 00:00:00.000	Great!	5

36	I36	C05	2022-07-30 00:00:00.000	Great!	5
37	I37	C05	2022-07-30 00:00:00.000	Great!	5
38	I38	C05	2022-07-30 00:00:00.000	Great!	5
39	I39	C05	2022-07-30 00:00:00.000	Great!	5
40	I40	C05	2022-07-30 00:00:00.000	Great!	5
41	I41	C05	2022-07-30 00:00:00.000	Great!	5
42	I42	C05	2022-07-30 00:00:00.000	Great!	5
43	I43	C05	2022-07-30 00:00:00.000	Great!	5
44	I44	C05	2022-07-30 00:00:00.000	Great!	5
45	I45	C05	2022-08-30 00:00:00.000	Great!	5
46	I46	C05	2022-08-30 00:00:00.000	Great!	5
47	I47	C05	2022-08-30 00:00:00.000	Great!	5
48	I48	C05	2022-08-30 00:00:00.000	Great!	5
49	I49	C05	2022-08-30 00:00:00.000	Great!	5
50	I50	C05	2022-08-30 00:00:00.000	Great!	5
51	I51	C05	2022-08-30 00:00:00.000	Great!	5
52	I52	C05	2022-08-30 00:00:00.000	Great!	5
53	I53	C05	2022-08-30 00:00:00.000	Great!	5
54	I54	C05	2022-08-30 00:00:00.000	Great!	5
55	I55	C05	2022-08-30 00:00:00.000	Great!	5
56	I56	C05	2022-08-30 00:00:00.000	Great!	5
57	I57	C05	2022-08-30 00:00:00.000	Great!	5
58	I58	C05	2022-08-30 00:00:00.000	Great!	5
59	I59	C05	2022-08-30 00:00:00.000	Great!	5
60	I60	C05	2022-08-30 00:00:00.000	Great!	5
61	I61	C05	2022-08-30 00:00:00.000	Great!	5
62	I62	C05	2022-08-30 00:00:00.000	Great!	5
63	I63	C05	2022-08-30 00:00:00.000	Great!	5
64	I64	C05	2022-08-30 00:00:00.000	Great!	5
65	I65	C05	2022-08-30 00:00:00.000	Great!	5
66	I66	C05	2022-08-30 00:00:00.000	Great!	5
67	I67	C05	2022-08-30 00:00:00.000	Great!	5
68	I68	C05	2022-08-30 00:00:00.000	Great!	5
69	I69	C05	2022-08-30 00:00:00.000	Great!	5
70	I70	C05	2022-08-30 00:00:00.000	Great!	5
71	I71	C05	2022-08-30 00:00:00.000	Great!	5

72	I72	C05	2022-08-30 00:00:00.000	Great!	5
73	I73	C05	2022-08-30 00:00:00.000	Great!	5
74	I74	C05	2022-08-30 00:00:00.000	Great!	5
75	I75	C05	2022-08-30 00:00:00.000	Great!	5
76	I76	C05	2022-08-30 00:00:00.000	Great!	5
77	I77	C05	2022-08-30 00:00:00.000	Great!	5
78	I78	C05	2022-08-30 00:00:00.000	Great!	5
79	I79	C05	2022-08-30 00:00:00.000	Great!	5
80	I80	C05	2022-08-30 00:00:00.000	Great!	5
81	I81	C05	2022-08-30 00:00:00.000	Great!	5
82	I82	C05	2022-08-30 00:00:00.000	Great!	5
83	I83	C05	2022-08-30 00:00:00.000	Great!	5
84	I84	C05	2022-08-30 00:00:00.000	Great!	5
85	I85	C05	2022-08-30 00:00:00.000	Great!	5
86	I86	C05	2022-08-30 00:00:00.000	Great!	5
87	I87	C05	2022-08-30 00:00:00.000	Great!	5

## 17.CUSTOMER\_ORDERS

	OrderID	CID
1	O01	C01
2	O02	C02
3	O03	C03
4	O04	C04
5	O05	C05
6	O06	C01
7	O07	C01
8	O08	C01
9	O09	C01
10	O10	C05
11	O11	C01
12	O12	C01
13	O13	C01
14	O14	C01
15	O15	C01
16	O16	C05
17	O17	C05
18	O18	C05
19	O19	C05
20	O20	C05
21	O21	C05
22	O22	C05
23	O23	C05
24	O24	C05
25	O25	C05
26	O26	C05
27	O27	C05
28	O28	C05
29	O29	C05
30	O30	C05
31	O31	C05
32	O32	C05
33	O33	C05
34	O34	C05
35	O35	C05

36	O36	C05
37	O37	C05
38	O38	C05
39	O39	C05
40	O40	C05
41	O41	C05
42	O42	C05
43	O43	C05
44	O44	C05
45	O45	C05
46	O46	C05
47	O47	C05
48	O48	C05
49	O49	C05
50	O50	C05
51	O51	C05
52	O52	C05
53	O53	C05
54	O54	C05
55	O55	C05
56	O56	C05
57	O57	C05
58	O58	C05
59	O59	C05
60	O60	C05
61	O61	C05
62	O62	C05
63	O63	C05
64	O64	C05
65	O65	C05
66	O66	C05
67	O67	C05
68	O68	C05
69	O69	C05
70	O70	C05
71	O71	C05



72	O72	C05
73	O73	C05
74	O74	C05
75	O75	C05
76	O76	C05
77	O77	C05
78	O78	C05
79	O79	C05
80	O80	C05
81	O81	C05
82	O82	C05
83	O83	C05
84	O84	C05
85	O85	C05
86	O86	C05
87	O87	C05

### **Any additional efforts:**

1. Since not every customer complained about every item they bought, and as ItemID is globally unique, we additionally established a second table named CUSTOMER FEEDBACKS by decomposition. Only one feedback will be connected with it on the feedback database,
2. Since certain queries may be performed again with different values for attribute, we generated views for some of the queries. As a result, employing a view enhances the database. For example, in query 7, we created 2 additional views noisy\_customer and stocks\_bought\_by\_noisy in order to visualize the publications bought by the customer with the most complaints before getting our results.
3. We populated the tables using large amounts of realistic data, also used multiple value insertion technique.
4. Used constraints to ensure the validity of our data. For example, for PRICE\_HISTORY, we made sure that Start\_date is not later than End\_date.

### Appendix C - Individual Contribution Form

Full Name	Individual Contributions to Lab 5 Submission	Percentage Of Contribution	Signature
Weng Pei He	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Pei He
Tan Jui Kit Justin	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Justin
Oza Tirth Tusharbai	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Tirth
Yang Yichen	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Yichen
Lim Zheng Yong Shawn	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Shawn
Yap Shen Hwei	Help make the group project report and formulate/brainstorm about the queries.	16.67%	Shen Hwei