NANYANG TECHNOLOGICAL UNIVERSITY
SINGAPORE

Lab 5: Final demonstration

SC2207 Introduction to Databases
Assoc. Prof. Wee Keong Ng, Cong Gao
April 4, 2023

Team 1

Full Name	Individual Contribution to Lab 5 Submission	Percentage of Contribution	Signature
Anusree Sreekumar	Analyzed viability of DDL commands using potential queries + propose relevant edits	14.3%	
Aruna D/O Vadivelu	Construction of multiple components	14.3%	
Ho Wei Hao	Group Leader Directed the logic of the construction of database and solutions to query to allow for smooth flow	14.3%	
Kevin Yok	Helped to prepare part of the query solutions	14.3%	
Liv Tan Ker Jin	Participated in the development of query solutions and proposed relevant edits required for the database	14.3%	
Ng Si En	Helped with development of different aspects of database and answered queries raised by members	14.3%	
Shi Xinyu	Construction of database	14.3%	

Contents

1 Introduction

Relational databases have become the standard for data storage and management in today's technological landscape. One of the key features of a relational database is the use of a schema, which defines the structure of the data and how it is related to other data within the database. The implementation of a relational schema is a crucial step in creating a functional and efficient database that can handle large amounts of data while maintaining data integrity.

In this report, we transform the relational schema defined in lab report 3 into tables in the database using Data Definition Language (DDL) commands and proceed to populate the tables with a range of possible records using Data Manipulation Language (DML) commands. Subsequently, we demonstrate the use of Data Query Language (DQL) commands to solve the queries as specified in the lab report.

2 Implementation of Database Schema

In this section, we demonstrate the SQL DDL commands used to create the databases as specified in lab report 3.

```
CREATE DATABASE IF NOT EXISTS ECDS_Group1;
USE ECDS_Group1;

CREATE TABLE Publication(
      PubID int NOT NULL,
      Publisher VarChar(30) NOT NULL,
      Year int NOT NULL,
      PRIMARY KEY (PubID)
);

CREATE TABLE Books(
      PubID int
      Title VarChar(30) NOT NULL,
      PRIMARY KEY (PubID)
      FOREIGN KEY (PubID) REFERENCES Publication (PubID) ON DELETE CASCADE
      ON UPDATE CASCADE
);

CREATE TABLE Magazines (
      PubID int,
      Title VarChar(30) NOT NULL,
      Issue int NOT NULL,
      PRIMARY KEY(PubID)
      FOREIGN KEY (PubID) REFERENCES Publication (PubID) ON DELETE CASCADE
      ON UPDATE CASCADE
);


CREATE TABLE Bookstore(
      BookstoreID int NOT NULL,
      PRIMARY KEY (BookstoreID)
);
```

```
CREATE TABLE StocksInBookstores(
      PubID int NOT NULL,
      BookstoreID int NOT NULL ,
      StockID int NOT NULL ,
      Stock_Qty int NOT NULL,
      Stock_Price int NOT NULL,
      PRIMARY KEY (BookstoreID, StockID)
      FOREIGN KEY (BookstoreID) REFERENCES Bookstore(BookstoreID) ON
      DELETE RESTRICT ON UPDATE CASCADE,
      FOREIGN KEY (PubID) REFERENCES Publication(PubID) ON DELETE RESTRICT
      ON UPDATE CASCADE
)

CREATE TABLE PriceHistory(
      Startdate date NOT NULL,
      Enddate date,
      StockID int NOT NULL,
      BookstoreID int NOT NULL,
      Price int NOT NULL,
      PRIMARY KEY (Startdate, Enddate, StockID, BookstoreID),
      FOREIGN KEY(BookstoreID, StockID) REFERENCES
      StocksInBookstores(BookstoreID, StockID) ON DELETE RESTRICT ON
      UPDATE CASCADE,
);

CREATE TABLE Customers(
      CustomerID int NOT NULL,
      Name varchar(50) NOT NULL,
      PRIMARY KEY (CustomerID)
);
```

```
CREATE TABLE ItemsInOrders(
      OrderID int NOT NULL,
      StockID int NOT NULL,
      ItemID int not null,
      ItemPrice int NOT NULL,
      ItemQty int NOT NULL,
      Deliverydate date,
      Comment varchar(200),
      Rating int,
      RatingDateTime datetime NOT NULL,
      BookstoreID int NOT NULL,
      PRIMARY KEY (OrderID, ItemID),
      FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE RESTRICT
      ON UPDATE CASCADE,
      FOREIGN KEY (BookstoreID, StockID) REFERENCES
      StocksInBookstores(BookstoreID, StockID) ON DELETE RESTRICT ON
      UPDATE CASCADE,
);

CREATE TABLE OrderStatus (
      Status_Date date NOT NULL,
      itemID int NOT NULL,
      currentstatus varchar(30)  DEFAULT “Being Processed” NOT NULL,
      orderID int NOT NULL,
      PRIMARY KEY (OrderDate, itemID, orderID),
      FOREIGN KEY (orderID) REFERENCES Orders (OrderID) ON DELETE RESTRICT
      ON UPDATE CASCADE
);
```

```
CREATE TABLE Complaints (
      ComplaintID int NOT NULL,
      textcontent varchar(500) NOT NULL,
      fileddatetime datetime NOT NULL,
      customerID int NOT NULL,
      employeeID int ,
      Handleddatetime datetime,
      PRIMARY KEY(ComplaintID),
      FOREIGN KEY(customerID) REFERENCES customers(customerID) ON DELETE
      RESRICT ON UPDATE CASCADE,
      FOREIGN KEY(employeeID) REFERENCES employees(employeeID) ON DELETE
      RESRICT ON UPDATE CASCADE
);

CREATE TABLE ComplaintsOnBookstore (
      ComplaintID int NOT NULL,
      BookstoreID int NOT NULL,
      PRIMARY KEY (ComplaintID),
      FOREIGN KEY (ComplaintID) REFERENCES Complaints(ComplaintID) ON
      DELETE RESRICT ON UPDATE CASCADE,
      FOREIGN KEY (BookstoreID) REFERENCES Bookstore(BookstoreID) ON
      DELETE RESRICT ON UPDATE CASCADE,
);

CREATE TABLE ComplaintsOnOrders (
      ComplaintID int NOT NULL,
      OrderID int NOT NULL,
      PRIMARY KEY (ComplaintID),
      FOREIGN KEY (ComplaintID) REFERENCES Complaints(ComplaintID) ON
      DELETE RESRICT ON UPDATE CASCADE,
      FOREIGN KEY (OrderID) REFERENCES Orders(OrderID) ON DELETE RESRICT
      ON UPDATE CASCADE
);

CREATE TABLE Employees(
      EmployeeID int NOT NULL,
      Name varchar(50) NOT NULL,
      Salary int NOT NULL,
      PRIMARY KEY (EmployeeID)
);
```

```
CREATE TABLE ComplaintStatus(
      ComDate date NOT NULL,
      Complaintstatus varchar(50) NOT NULL DEFAULT ‘Pending’,
      ComplaintID int NOT NULL,
      PRIMARY KEY (ComplaintID, ComDate),
      FOREIGN KEY (ComplaintID) REFERENCES Complaints(ComplaintID) ON
      DELETE RESRICT ON UPDATE CASCADE
);
```

```
CREATE TABLE Orders (
      OrderID int NOT NULL,
      Date_time datetime NOT NULL,
      Shipping_address varchar(80) NOT NULL,
      CustomerID int NOT NULL,
      PRIMARY KEY (OrderID),
      FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID) ON UPDATE
      CASCADE
);
```

3 Population of Database

In this section we demonstrate the commands used in the population of the database with a wide variety of records. We inserted a broad variety of data to make our datasets and analysis robust.

```
INSERT INTO Publication(PubID, Publisher, PubYear)
VALUES
(23154879, 'SPH', 2022)
(24587692, ‘Nanyang Publisher Company’, 2020)
(34567890, ‘Ethos Books’,2020)
(35468921, 'SPH', 2021)
(36485721, ‘Nanyang Publisher Company', 2018)
(46548732, 'SPH', 2021)
(54876953, ‘Nanyang Publisher Company’, 2019)
(64587236, ‘Penguin', 2018),
(76854921, 'SPH', 2020),
(97772093, ‘Penguin’, 2015),
(98541276, ‘Penguin’, 2021);
```

```
INSERT INTO Books(PubID, Title)
VALUES
(34567890, ‘Harry Porter Finale’),
(54876953, ‘Harry Porter Book 1’),
(97772093, ‘Pachinko’),
(23154879, ‘Harry Porter Book 2’),
(35468921, ‘One Boy And The World’),
(24587692, ‘Burning Poem’);
```

```
INSERT INTO Magazines(PubID, Title, Issue)
VALUES
(76854921,'Sport Monthly Magazine', 87),
(36485721, 'Jazz Daily', 56),
(64587236, 'Animal Review', 41),
(98541276, 'Comic Digest', 23),
(46548732, 'Fitness Guide', 12);
```

```
INSERT INTO bookstore(BookstoreID)
VALUES
(353629),
(589384),
(457705),
(223296),
(513051),
(949693),
(197442),
(290976),
(834131),
(594173);


INSERT INTO StocksInBookstores(BookstoreID, StockID, PubID, Stock_Qty,
Stock_Price)
VALUES
(197442, 354681, 36485721, 10, 1),
(197442, 653241, 76854921, 23, 2),
(223296, 767281, 76854921, 22, 3),
(223296, 767282, 97772093, 15, 2),
(290976, 345276, 64587236, 56, 6),
(290976, 345277, 23154879, 50, 5),
(290976, 546387, 35468921, 78, 9),
(290976, 546388, 98541276, 80, 9),
(353629, 657482, 98541276, 72, 6),
(353629, 657483, 54876953, 23, 2),
(353629, 837261, 23154879, 67, 3),
(353629, 837262, 64587236, 50, 4),
(457705, 123456, 36485721, 22, 3),
(457705, 123459, 54876953, 16, 2),
(457705, 456789, 46548732, 9, 1),
(457705, 456790, 24587692, 12, 3),
(457705, 456791, 76854921, 20, 2),
(197442, 215468, 34567890, 10, 4),
(949693, 563468, 34567890, 6, 4);
```

```
INSERT INTO PriceHistory(Startdate, Enddate, StockID, BookstoreID,
Price)
VALUES
('2022-05-01', '2022-05-31', 653241, 197442, 31),
('2022-06-01', '2022-06-30', 653241, 197442, 30),
('2022-07-01', '2022-07-31', 653241, 197442, 32),
('2022-08-01', '2022-08-28', 653241, 197442, 34),

('2022-05-01', '2022-05-31', 354681, 197442, 19),
('2022-06-01', '2022-06-30', 354681, 197442, 20),
('2022-07-01', '2022-07-31', 354681, 197442, 24),
('2022-08-01', '2022-08-28', 354681, 197442, 23),

('2022-05-01', '2022-05-31', 767281, 223296, 8),
('2022-06-01', '2022-06-30', 767281, 223296, 10),
('2022-07-01', '2022-07-31', 767281, 223296, 9),
('2022-08-01', '2022-08-28', 767281, 223296, 10),

('2022-07-01', '2022-07-31', 123456, 457705, 24),
('2022-08-01', '2022-08-28', 123456, 457705, 22),

('2022-08-01', '2022-08-28', 345276, 290976, 56),
('2022-08-01', '2022-08-28', 546387, 290976, 78),
('2022-08-01', '2022-08-28', 546388, 290976, 80),
('2022-08-01', '2022-08-28', 837261, 353629, 67),
('2022-08-01', '2022-08-28', 837262, 353629, 50),
('2022-08-01', '2022-08-28', 657482, 353629, 23),
('2022-08-01', '2022-08-28', 456789, 457705, 9),
('2022-08-01', '2022-08-28', 123459, 457705, 16),

('2022-07-01', '2022-07-31', 456791, 457705, 22),
('2022-08-01', '2022-08-28', 456791, 457705, 20),
('2022-08-01', '2022-08-31', 215468, 197442, 54),
('2022-08-01', '2022-08-31', 563468, 949693, 56);
```

```
INSERT INTO Customers(CustomerID, Name)
VALUES
(54876213,'Levi Raymond'),
(12349045, ‘Jesse Lee’),
(65412879,'Gabriela Gomez'),
(32154769,'Jaycee Alvarez'),
(25146358,'Jaylyn Bernard'),
(13254769,'Myles Henson'),
(98541275,'Leroy Edwards'),
(41754398,'Natasha Freeman'),
(75632148,'Derek Wang'),
(86214357,'Jane Tate'),
(34126587,'Paula Cox');


INSERT INTO Employees(employeeID, name, salary)
VALUES
(40986608, ‘Talaat Lilibeth’, 1800),
(97729245, 'Per Nina', 2400),
(48381199, 'Jonathan Artūrs', 1000),
(89480874, 'Jayne Peter', 1500),
(72724744 , 'Leontina Asami', 4000),
(45232808, 'Beau Missie', 3000),
(80954251, 'Maggie Danai', 2000),
(40986608, 'Talaat Lilibeth', 1800),
(76666878, 'Kendall Laxmi', 2000),
(68752993, 'Charmion Naenia', 2500);


INSERT INTO Orders (OrderID, Date_time, Shipping_address, CustomerID)
VALUES
(96525998, '2022-06-23 10:22:03 AM', '5 Tyersall Park', 13254769),
(34569002,  '2022-06-25 03:22:03 PM', '29 Bukit Panjang Ring Road',
25146358),
(20494850, '2022-06-25 01:21:03 PM', '31 Jurong East Road', 32154769),
(20220020, '2022-07-21 09:21:03 AM', '3 Pasir Panjang Lane', 34126587),
(17872689, '2022-08-28 10:21:03 PM', '550 Jurong West' , 41754398),
(24681011, '2022-08-15 01:34:03 PM', '33 Dempsey Hill', 54876213),
(36912158, '2022-07-31 06:57:03 PM', '10 Marina Boulevard', 65412879),
```

```
(13579113, '2022-08-15 03:21:03 AM', '33 Dempsey Hill', 75632148),
(87829188, '2022-08-15 05:21:03 AM', '33 Woodlands Grove', 86214357),
(28494905, '2022-08-15 11:21:03 PM', '20 Admiralty Road', 98541275);


INSERT INTO OrderStatus (StatusDate, itemID, currentstatus, orderID)
VALUES

('2022-08-15', 3099121, 'Delivered', 13579113),
('2022-08-15', 63729102, 'Delivered', 17872689),
('2022-07-24', 3938434, 'Being Processed', 20220020),
('2022-07-24', 99999999, 'Being Processed', 20220020),
('2022-07-25', 11111111, 'Being Processed', 20220020),
('2022-07-25', 29393939, 'Being Processed', 20220020),
('2022-07-25', 55555555, 'Being Processed', 20220020),
('2022-07-27', 44444444, 'Being Processed', 20220020),
('2022-07-28', 22222222, 'Being Processed', 20220020),
('2022-07-28', 66666666, 'Being Processed', 20220020),
('2022-07-29', 23445556, 'Being Processed', 20220020),
('2022-07-29', 33333333, 'Being Processed', 20220020),
('2022-07-29', 77777777, 'Being Processed', 20220020),
('2022-07-30', 88888888, 'Being Processed', 20220020),
('2022-08-01', 3938434, 'Delivered', 20220020),
('2022-08-01', 99999999, 'Delivered', 20220020),
('2022-08-01', 11111111, 'Delivered', 20220020),
('2022-08-01', 29393939, 'Delivered', 20220020),
('2022-08-01', 55555555, 'Delivered', 20220020),
('2022-08-01', 44444444, 'Delivered', 20220020),
('2022-08-01', 22222222, 'Delivered', 20220020),
('2022-08-01', 66666666, 'Delivered', 20220020),
('2022-08-01', 23445556, 'Delivered', 20220020),
('2022-08-01', 33333333, 'Delivered', 20220020),
('2022-08-01', 77777777, 'Delivered', 20220020),
('2022-08-01', 88888888, 'Delivered', 20220020)
('2022-06-26', 39494505, 'Being Processed', 20494850),
('2022-06-27', 36289172, 'Being Processed', 20494850),
('2022-07-01', 39494505, 'Delivered', 20494850),
('2022-08-15', 36289172, 'Delivered', 20494850),
```

```
('2022-08-15', 32018301, 'Delivered', 24681011),
('2022-08-20', 72883791, 'Delivered', 28494905),
('2022-06-27', 49030321, 'Being Processed', 34569002),
('2022-06-30', 49030321, 'Delivered', 34569002),
('2022-07-31', 4049495, 'Being Processed', 36912158),
('2022-08-02', 33334949, 'Being Processed', 36912158),
('2022-08-02', 48333949, 'Being Processed', 36912158),
('2022-08-02', 48334949, 'Being Processed', 36912158),
('2022-08-02', 48422249, 'Being Processed', 36912158),
('2022-08-02', 48433949, 'Being Processed', 36912158),
('2022-08-02', 48493339, 'Being Processed', 36912158),
('2022-08-02', 48493349, 'Being Processed', 36912158),
('2022-08-02', 48494455, 'Being Processed', 36912158),
('2022-08-02', 48494944, 'Being Processed', 36912158),
('2022-08-02', 48494949, 'Being Processed', 36912158),
('2022-08-02', 55494949, 'Being Processed', 36912158),
('2022-08-02', 99303494, 'Being Processed', 36912158),
('2022-08-05', 46694949, 'Delivered', 36912158),
('2022-08-05', 4049495, 'Delivered', 36912158),
('2022-08-05', 33334949, 'Delivered', 36912158),
('2022-08-05', 48333949, 'Delivered', 36912158),
('2022-08-05', 48334949, 'Delivered', 36912158),
('2022-08-05', 48422249, 'Delivered', 36912158),
('2022-08-05', 48433949, 'Delivered', 36912158),
('2022-08-05', 48493339, 'Delivered', 36912158),
('2022-08-05', 48493349, 'Delivered', 36912158),
('2022-08-05', 48494455, 'Delivered', 36912158),
('2022-08-05', 48494944, 'Delivered', 36912158),
('2022-08-05', 48494949, 'Delivered', 36912158),
('2022-08-05', 55494949, 'Delivered', 36912158),
('2022-08-05', 99303494, 'Delivered', 36912158),
('2022-08-15', 23230101, 'Delivered', 87829188),
('2022-06-28', 4564332, 'Delivered', 96525998),
('2022-06-28', 12434322, 'Delivered', 96525998),
('2022-06-28', 12434342, 'Delivered', 96525998),
('2022-06-27', 38449920, 'Being Processed', 96525998),
('2022-06-23', 1234522, 'Being Processed', 96525998),
('2022-06-23', 1322422, 'Being Processed', 96525998),
('2022-06-27', 1323432, 'Being Processed', 96525998),
```

```
('2022-06-27', 2111222, 'Being Processed', 96525998),
('2022-06-27', 2123322, 'Being Processed', 96525998),
('2022-06-27', 2334222, 'Being Processed', 96525998),
('2022-06-27', 2344022, 'Being Processed', 96525998),
('2022-06-27', 3034022, 'Being Processed', 96525998),
('2022-06-27', 4132432, 'Being Processed', 96525998),
('2022-06-27', 4564522, 'Being Processed', 96525998),
('2022-06-27', 12432122, 'Being Processed', 96525998),
('2022-06-27', 12443422, 'Being Processed', 96525998),
('2022-06-28', 38449920, 'Delivered', 96525998),
('2022-06-28', 1234522, 'Delivered', 96525998),
('2022-06-28', 1322422, 'Delivered', 96525998),
('2022-06-28', 1323432, 'Delivered', 96525998),
('2022-06-28', 2111222, 'Delivered', 96525998),
('2022-06-28', 2123322, 'Delivered', 96525998),
('2022-06-28', 2334222, 'Delivered', 96525998),
('2022-06-28', 2344022, 'Delivered', 96525998),
('2022-06-28', 3034022, 'Delivered', 96525998),
('2022-06-28', 4132432, 'Delivered', 96525998),
('2022-06-28', 4564522, 'Delivered', 96525998),
('2022-06-28', 12432122, 'Delivered', 96525998),
('2022-06-28', 12443422, 'Delivered', 96525998);
```

```
INSERT INTO Complaints(ComplaintID, textcontent, fileddatetime,
customerID, employeeID, Handleddatetime)
VALUES
(65487213, 'Wrong books delivered.','2022-06-14', 54876213, 40986608,
'2022-06-17'),
(98746521, 'Books not packed properly', '2022-08-23', 65412879, 97729245,
'2022-08-28'),
(32641578, 'Dull, predictable, repetitive stories written in a confusing
style.', '2022-07-14', 32154769, 48381199, '2022-07-18'),
```

```
(32654128, 'Keeping readers waiting for a climax that never comes.',
'2022-08-06', 25146358, 89480874, '2022-08-07'),
(23654198, 'Lazy writing, flimsy and unbelievable storyline.',
'2022-06-04', 75632148, 72724744, '2022-06-09'),
(12365487, 'Writers with no clue of human nature.', '2022-07-22',
34126587, 45232808, '2022-07-26'),
(42541876, 'Unrelatable characters who are difficult to care about.',
'2022-06-27', 41754398, 89480874, '2022-06-30'),
(76521438, 'Extra copies of the books received.', '2022-08-25', 65412879,
40986608,'2022-08-29'),
(85463214, ‘Received the wrong books, '2022-07-18', 65412879, 89480874,
'2022-07-24'),
(53214569, ‘Book delivered to wrong address', '2022-06-17', 34126587,
45232808, '2022-06-25');


INSERT INTO ComplaintsOnOrders(ComplaintID, OrderID)
VALUES
(12365487, 20220020),
(23654198, 13579113),
(32641578, 20494850),
(32654128, 34569002),
(42541876, 17872689);


INSERT INTO ComplaintsOnBookstore(ComplaintID, BookstoreID)
VALUES
(53214569, 513051),
(65487213, 589384),
(76521438, 594173),
(85463214, 834131),
(98746521, 949693);


INSERT INTO ComplaintStatus(ComDate, Complaintstatus, ComplaintID)
VALUES
('2022-07-22', Pending, 12365487),
('2022-07-23', Pending, 12365487),
('2022-07-24', ‘Pending', 12365487),
('2022-07-25', 'Pending', 12365487),
('2022-07-26', 'Complete', 12365487),
```

```
('2022-06-04', 'Pending', 23654198),
('2022-06-05', 'Pending', 23654198),
('2022-06-06', 'Pending', 23654198),
('2022-06-07', 'Pending', 23654198),
('2022-06-08', 'Pending', 23654198),
('2022-06-09', 'Complete', 23654198),
('2022-07-14', Pending, 32641578),
('2022-07-15', Pending, 32641578),
('2022-07-16', Pending, 32641578),
('2022-07-17', Pending, 32641578),
('2022-07-18', 'Complete', 32641578),
('2022-08-06', 'Pending', 32654128),
('2022-08-07', 'Complete', 32654128),
('2022-06-27', 'Pending', 42541876),
('2022-06-28', 'Pending', 42541876),
('2022-06-29', 'Pending', 42541876),
('2022-06-30', 'Complete', 42541876),
('2022-06-17', ‘Pending’, 53214569),
('2022-06-18', ‘Pending’, 53214569),
('2022-06-19', ‘Pending’, 53214569),
('2022-06-20', ‘Pending’, 53214569),
('2022-06-21', ‘Pending’, 53214569),
('2022-06-22', ‘Pending’, 53214569),
('2022-06-23', ‘Pending’, 53214569),
('2022-06-24', ‘Pending’, 53214569),
('2022-06-25', 'Complete', 53214569),
('2022-06-14', ‘Pending’, 65487213),
('2022-06-15', ‘Pending’, 65487213),
('2022-06-16', ‘Pending’, 65487213),
('2022-06-17', 'Complete', 65487213),
('2022-08-25', 'Pending', 76521438),
('2022-08-26', 'Pending', 76521438),
('2022-08-27', 'Pending', 76521438),
('2022-08-28', 'Pending', 76521438),
('2022-08-29', 'Complete', 76521438),
('2022-07-18', 'Pending', 85463214),
('2022-07-19', 'Pending', 85463214),
('2022-07-20', 'Pending', 85463214),
('2022-07-21', 'Pending', 85463214),
('2022-07-22', 'Pending', 85463214),
('2022-07-23', 'Pending', 85463214),
```

```
('2022-07-24', 'Complete', 85463214),
('2022-08-23', 'Pending', 98746521),
('2022-08-24', 'Pending', 98746521),
('2022-08-25', 'Pending', 98746521),
('2022-08-26', 'Pending', 98746521),
('2022-08-27', 'Pending', 98746521),
('2022-08-28', 'Complete', 98746521);
```

```
INSERT INTO ItemsInOrders(OrderID, StockID, ItemID, ItemPrice, ItemQty,
Deliverydate, comment, rating, RatingDateTime, BookstoreID)
VALUES
(13579113, 123456, 3099121, 23, 1, '2022-08-15', 'Very nice', 5,
'2022-08-15 10:37:03.000', 457705),
(17872689, 653241, 63729102, 22, 1, '2022-08-15', 'Very bad', 2,
'2022-08-15 10:37:03.000', 197442),
(20220020, 657482, 3938434, 22, 1, '2022-08-01', 'Soso',5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657483, 11111111,  22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 22222222, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 23445556, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 767281, 29393939,  28, 1, '2022-08-01', 'Great book!', 5,
'2022-08-01 10:38:03.000', 223296),
(20220020, 657482, 33333333, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 44444444, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 55555555, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 66666666, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 77777777, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000',   353629),
(20220020, 657482, 88888888, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
(20220020, 657482, 99999999, 22, 1, '2022-08-01', 'Soso', 5, '2022-08-01
10:35:03.000', 353629),
```

```
(20494850, 653241, 36289172, 12, 1, '2022-08-15', 'Very bad', 1,
'2022-08-15 10:37:03.000', 197442),
(20494850, 345276, 39494505, 9, 1, '2022-07-01', 'Nice!', 5, '2022-07-01
10:38:03.000', 290976),
(24681011, 653241, 32018301, 22, 1, '2022-08-15', 'So so', 3, '2022-08-15
10:37:03.000', 197442),
(28494905, 653241, 72883791, 15, 1, '2022-08-20', 'not nice', 2,
'2022-08-20 10:37:03.000', 197442),
(34569002, 354681, 49030321, 18, 1, '2022-06-30', 'Perfect as per
usual.Thanks!', 5, '2022-06-30 10:27:03.000', 197442),
(36912158, 653241, 4049495, 25, 1, '2022-08-05', 'Ok', 5, '2022-08-06
10:27:03.000', 197442),
(36912158, 767281, 33334949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 46694949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000',  223296),
(36912158, 767281, 48333949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48334949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48422249, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48433949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48493339, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48493349, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48494455, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48494944, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 48494949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 767281, 55494949, 13, 1, '2022-08-05', 'Can be better', 5,
'2022-08-06 10:35:03.000', 223296),
(36912158, 837261, 99303494, 15, 2, '2022-08-05', 'Not Bad', 5,
'2022-08-06 10:33:03.000', 353629),
(87829188, 123456, 23230101, 22, 1, '2022-08-15', 'Very nice', 5,
'2022-08-15 10:37:03.000', 457705),
```

```
(96525998, 653241, 1234522, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 1322422, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 1323432, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2111222, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2123322, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2334222, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2344022, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 3034022, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 4132432, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 4564332, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2334223, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 2344023, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 3034023, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 4132433, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 4564523, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 12432123, 25, 1, '2022-06-28', 'Amazing Book!', 4,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 12434323, 25, 1, '2022-06-28', 'Amazing Book!', 5,
'2022-08-01 10:22:03.000', 197442),
(96525998, 653241, 12434343, 25, 1, '2022-06-28', 'Amazing Book!', 3,
'2022-08-01 10:22:03.000', 197442),
(96525998, 456791, 12443423, 25, 1, '2022-06-28', 'Amazing Book!', 4,
'2022-08-01 10:22:03.000', 457705),
(96525998, 123456, 38449920, 20, 1, '2022-06-29', 'Slightly Damaged :(',
3, '2022-08-01 10:22:03.000', 457705);
```

4 Showcase of Database

Books

	PubID	Title
1	34567890	Harry Porter Finale
2	54876953	Harry Porter Book 1
3	97772093	Pachinko
4	23154879	Harry Porter Book 2
5	35468921	One Boy And The World
6	24587692	Burning Poem

Bookstore

	BookstoreID
1	197442
2	223296
3	290976
4	353629
5	457705
6	513051
7	589384
8	594173
9	834131
10	949693

Complaints

	ComplaintID	textcontent	fileddatetime	customerID	employeeID	Handleddatetime
1	12365487	Writers with no clue of human nature.	2022-07-22 00:00:00.000	34126587	45232808	2022-07-26 00:00:00.000
2	23654198	Lazy writing, flimsy and unbelievable storyline.	2022-06-04 00:00:00.000	75632148	72724744	2022-06-09 00:00:00.000
3	32641578	Dull, predictable, repetitive stories written in a co...	2022-07-14 00:00:00.000	32154769	48381199	2022-07-18 00:00:00.000
4	32654128	Keeping readers waiting for a climax that never ...	2022-08-06 00:00:00.000	25146358	89480874	2022-08-07 00:00:00.000
5	42541876	Unrelatable characters who are difficult to care ...	2022-06-27 00:00:00.000	41754398	89480874	2022-06-30 00:00:00.000
6	53214569	Book delievered to wrong address	2022-06-17 00:00:00.000	34126587	45232808	2022-06-25 00:00:00.000
7	65487213	Wrong books delivered	2022-06-14 00:00:00.000	54876213	40986608	2022-06-17 00:00:00.000
8	76521438	Extra copies of the books received	2022-08-25 00:00:00.000	65412879	40986608	2022-08-29 00:00:00.000
9	85463214	Received the wrong books	2022-07-18 00:00:00.000	65412879	89480874	2022-07-24 00:00:00.000
10	98746521	Books not packed properly	2022-08-23 00:00:00.000	65412879	97729245	2022-08-28 00:00:00.000

ComplaintsOnBookstores

	ComplaintID	BookstoreID
1	53214569	197442
2	65487213	949693
3	76521438	457705
4	85463214	197442
5	98746521	197442

ComplaintsOnOrders

	ComplaintID	OrderID
1	12365487	20220020
2	23654198	13579113
3	32641578	20494850
4	32654128	34569002
5	42541876	17872689

ComplaintStatus

	ComDate	Complaintstatus	ComplaintID
1	2022-07-22	Pending	12365487
2	2022-07-23	Pending	12365487
3	2022-07-24	Pending	12365487
4	2022-07-25	Pending	12365487
5	2022-07-26	Complete	12365487
6	2022-06-04	Pending	23654198
7	2022-06-05	Pending	23654198
8	2022-06-06	Pending	23654198
9	2022-06-07	Pending	23654198
10	2022-06-08	Pending	23654198
11	2022-06-09	Complete	23654198
12	2022-07-14	Pending	32641578
13	2022-07-15	Pending	32641578
14	2022-07-16	Pending	32641578
15	2022-07-17	Pending	32641578
16	2022-07-18	Complete	32641578
17	2022-08-06	Pending	32654128
18	2022-08-07	Complete	32654128
19	2022-06-27	Pending	42541876
20	2022-06-28	Pending	42541876
21	2022-06-29	Pending	42541876
22	2022-06-30	Complete	42541876
23	2022-06-17	Pending	53214569
24	2022-06-18	Pending	53214569
25	2022-06-19	Pending	53214569
26	2022-06-20	Pending	53214569
27	2022-06-21	Pending	53214569
28	2022-06-22	Pending	53214569
29	2022-06-23	Pending	53214569
30	2022-06-24	Pending	53214569

31	2022-06-25	Complete	53214569
32	2022-06-14	Pending	65487213
33	2022-06-15	Pending	65487213
34	2022-06-16	Pending	65487213
35	2022-06-17	Complete	65487213
36	2022-08-25	Pending	76521438
37	2022-08-26	Pending	76521438
38	2022-08-27	Pending	76521438
39	2022-08-28	Pending	76521438
40	2022-08-29	Complete	76521438
41	2022-07-18	Pending	85463214
42	2022-07-19	Pending	85463214
43	2022-07-20	Pending	85463214
44	2022-07-21	Pending	85463214
45	2022-07-22	Pending	85463214
46	2022-07-23	Pending	85463214
47	2022-07-24	Complete	85463214
48	2022-08-23	Pending	98746521
49	2022-08-24	Pending	98746521
50	2022-08-25	Pending	98746521
51	2022-08-26	Pending	98746521
52	2022-08-27	Pending	98746521
53	2022-08-28	Complete	98746521

Customers

	CustomerID	Name
1	12349045	Jesse Lee
2	13254769	Myles Henson
3	25146358	Jaylyn Bernard
4	32154769	Jaycee Alvarez
5	34126587	Paula Cox
6	41754398	Natasha Freeman
7	54876213	Levi Raymond
8	65412879	Gabriela Gomez
9	75632148	Derek Wang
10	86214357	Jane Tate
11	98541275	Leroy Edwards

Employees

	EmployeeID	Name	Salary
1	40986608	Talaat Lilibeth	1800
2	45232808	Beau Missie	3000
3	48381199	Jonathan Arturs	1000
4	68752993	Charmion Naenia	2500
5	72724744	Leontina Asami	4000
6	74161460	Zora Aeliana	2500
7	76666878	Kendall Laxmi	2000
8	80954251	Maggie Danai	2000
9	89480874	Jayne Peter	1500
10	97729245	Per Nina	2400

Items in orders

	OrderID	StockID	ItemID	ItemPrice	ItemQty	Deliverydate	Comment	Rating	RatingDateTime	BookstoreID
1	13579113	123456	3099121	23	1	2022-08-15	Very nice	5	2022-08-15 10:37:03.000	457705
2	13579113	215468	11023949	13	1	2022-08-06	not too good!	1	2022-08-07 10:57:12.000	197442
3	17872689	215468	45458458	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
4	17872689	215468	55960004	13	1	2022-08-06	Nice!	5	2022-08-07 10:57:12.000	197442
5	17872689	653241	63729102	22	1	2022-08-15	Very bad	2	2022-08-15 10:37:03.000	197442
6	20220020	215468	3494949	13	1	2022-08-06	Satisfied!	5	2022-08-07 10:57:12.000	197442
7	20220020	657482	3938434	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
8	20220020	657483	11111111	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
9	20220020	657482	22222222	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
10	20220020	657482	23445556	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
11	20220020	767281	29393939	28	1	2022-08-01	Great book!	5	2022-08-01 10:38:03.000	223296
12	20220020	657482	33333333	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
13	20220020	215468	39948020	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
14	20220020	657482	44444444	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
15	20220020	657482	55555555	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
16	20220020	657482	66666666	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
17	20220020	657482	77777777	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
18	20220020	657482	88888888	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
19	20220020	657482	99999999	22	1	2022-08-01	Soso	5	2022-08-01 10:35:03.000	353629
20	20494850	653241	36289172	12	1	2022-08-15	Very bad	1	2022-08-15 10:37:03.000	197442
21	20494850	345276	39494505	9	1	2022-07-01	Nice!	5	2022-07-01 10:38:03.000	290976
22	24681011	653241	32018301	22	1	2022-08-15	So so	3	2022-08-15 10:37:03.000	197442
23	28494905	653241	72883791	15	1	2022-08-20	not nice	2	2022-08-20 10:37:03.000	197442
24	34569002	215468	23040595	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
25	34569002	215468	33005968	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
26	34569002	215468	48893002	13	1	2022-08-06	Nubbad!	5	2022-08-07 10:57:12.000	197442
27	34569002	354681	49030321	35	1	2022-06-30	Perfect as per usual.Thanks!	5	2022-06-30 10:27:03.000	197442
28	36912158	215468	294948	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
29	36912158	653241	4049495	25	1	2022-08-05	Ok	5	2022-08-06 10:27:03.000	197442
30	36912158	767281	33334949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296

31	36912158	767281	46694949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
32	36912158	767281	48333949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
33	36912158	767281	48334949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
34	36912158	767281	48422249	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
35	36912158	767281	48433949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
36	36912158	767281	48493339	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
37	36912158	767281	48493349	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
38	36912158	767281	48494455	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
39	36912158	767281	48494944	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
40	36912158	767281	48494949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
41	36912158	215468	50099694	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442
42	36912158	767281	55494949	13	1	2022-08-05	Can be better	5	2022-08-06 10:35:03.000	223296
43	36912158	837261	99303494	15	2	2022-08-05	Not Bad	5	2022-08-06 10:33:03.000	353629
44	87829188	123456	23230101	22	1	2022-08-15	Very nice	5	2022-08-15 10:37:03.000	457705
45	96525998	653241	1234522	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
46	96525998	653241	1322422	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
47	96525998	653241	1323432	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
48	96525998	653241	2111222	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
49	96525998	653241	2123322	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
50	96525998	653241	2334222	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
51	96525998	653241	2334223	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
52	96525998	653241	2344022	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
53	96525998	653241	2344023	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
54	96525998	653241	3034022	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
55	96525998	653241	3034023	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
56	96525998	653241	4132432	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
57	96525998	653241	4132433	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
58	96525998	653241	4564332	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
59	96525998	653241	4564523	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
60	96525998	653241	12432123	25	1	2022-06-28	Amazing Book!	4	2022-08-01 10:22:03.000	197442
61	96525998	653241	12434323	25	1	2022-06-28	Amazing Book!	5	2022-08-01 10:22:03.000	197442
62	96525998	653241	12434343	25	1	2022-06-28	Amazing Book!	3	2022-08-01 10:22:03.000	197442
63	96525998	456791	12443423	25	1	2022-06-28	Amazing Book!	4	2022-08-01 10:22:03.000	457705
64	96525998	215468	20049495	13	1	2022-08-06	okok	1	2022-08-07 10:57:12.000	197442
65	96525998	123456	38449920	20	1	2022-06-29	Slightly Damaged :(	3	2022-08-01 10:22:03.000	457705
66	96525998	215468	55969042	13	1	2022-08-06	Great!	5	2022-08-07 10:57:12.000	197442

Magazines

	PubID	Title	Issue
1	76854921	Sport Monthly Magazine	87
2	36485721	Jazz Daily	56
3	64587236	Animal Review	41
4	98541276	Comic Digest	23
5	46548732	Fitness Guide	12

Orders

	OrderID	Date_time	Shipping_address	CustomerID
1	13579113	2022-08-15 03:21:03.000	33 Dempsey Hill	75632148
2	17872689	2022-08-14 22:21:03.000	550 Jurong West	41754398
3	20220020	2022-07-21 09:21:03.000	3 Pasir Panjang Lane	34126587
4	20494850	2022-06-25 13:21:03.000	31 Jurong East Road	32154769
5	24681011	2022-08-15 13:34:03.000	33 Dempsey Hill	54876213
6	28494905	2022-08-15 23:21:03.000	20 Admiralty Road	98541275
7	34569002	2022-06-25 15:22:03.000	29 Bukit Panjang Ring Road	25146358
8	36912158	2022-07-31 18:57:03.000	10 Marina Boulevard	65412879
9	87829188	2022-08-15 05:21:03.000	33 Woodlands Grove	86214357
10	96525998	2022-06-23 10:22:03.000	5 Tyersall Park	13254769

OrderStatus

	StatusDate	itemID	currentstatus	orderID
1	2022-06-23	1234522	Being Processed	96525998
2	2022-06-23	1322422	Being Processed	96525998
3	2022-06-26	39494505	Being Processed	20494850
4	2022-06-27	1323432	Being Processed	96525998
5	2022-06-27	2111222	Being Processed	96525998
6	2022-06-27	2123322	Being Processed	96525998
7	2022-06-27	2334222	Being Processed	96525998
8	2022-06-27	2344022	Being Processed	96525998
9	2022-06-27	3034022	Being Processed	96525998
10	2022-06-27	4132432	Being Processed	96525998
11	2022-06-27	4564522	Being Processed	96525998
12	2022-06-27	12432122	Being Processed	96525998
13	2022-06-27	12443422	Being Processed	96525998
14	2022-06-27	36289172	Being Processed	20494850
15	2022-06-27	38449920	Being Processed	96525998
16	2022-06-27	49030321	Being Processed	34569002
17	2022-06-28	1234522	Delivered	96525998
18	2022-06-28	1322422	Delivered	96525998
19	2022-06-28	1323432	Delivered	96525998
20	2022-06-28	2111222	Delivered	96525998
21	2022-06-28	2123322	Delivered	96525998
22	2022-06-28	2334222	Delivered	96525998
23	2022-06-28	2344022	Delivered	96525998
24	2022-06-28	3034022	Delivered	96525998
25	2022-06-28	4132432	Delivered	96525998
26	2022-06-28	4564332	Delivered	96525998
27	2022-06-28	4564522	Delivered	96525998
28	2022-06-28	12432122	Delivered	96525998
29	2022-06-28	12434322	Delivered	96525998
30	2022-06-28	12434342	Delivered	96525998
31	2022-06-28	12443422	Delivered	96525998
32	2022-06-28	38449920	Delivered	96525998
33	2022-06-30	49030321	Delivered	34569002
34	2022-07-01	39494505	Delivered	20494850
35	2022-07-24	3938434	Being Processed	20220020
36	2022-07-24	99999999	Being Processed	20220020
37	2022-07-25	11111111	Being Processed	20220020
38	2022-07-25	29393939	Being Processed	20220020
39	2022-07-25	55555555	Being Processed	20220020
40	2022-07-27	44444444	Being Processed	20220020
41	2022-07-28	22222222	Being Processed	20220020
42	2022-07-28	66666666	Being Processed	20220020
43	2022-07-29	23445556	Being Processed	20220020
44	2022-07-29	33333333	Being Processed	20220020
45	2022-07-29	77777777	Being Processed	20220020
46	2022-07-30	88888888	Being Processed	20220020
47	2022-07-31	4049495	Being Processed	36912158
48	2022-08-01	3938434	Delivered	20220020
49	2022-08-01	11111111	Delivered	20220020
50	2022-08-01	22222222	Delivered	20220020
51	2022-08-01	23445556	Delivered	20220020
52	2022-08-01	29393939	Delivered	20220020
53	2022-08-01	33333333	Delivered	20220020
54	2022-08-01	44444444	Delivered	20220020

	StatusDate	itemID	currentstatus	orderID
55	2022-08-01	55555555	Delivered	20220020
56	2022-08-01	66666666	Delivered	20220020
57	2022-08-01	77777777	Delivered	20220020
58	2022-08-01	88888888	Delivered	20220020
59	2022-08-01	99999999	Delivered	20220020
60	2022-08-02	33334949	Being Processed	36912158
61	2022-08-02	48333949	Being Processed	36912158
62	2022-08-02	48334949	Being Processed	36912158
63	2022-08-02	48422249	Being Processed	36912158
64	2022-08-02	48433949	Being Processed	36912158
65	2022-08-02	48493339	Being Processed	36912158
66	2022-08-02	48493349	Being Processed	36912158
67	2022-08-02	48494455	Being Processed	36912158
68	2022-08-02	48494944	Being Processed	36912158
69	2022-08-02	48494949	Being Processed	36912158
70	2022-08-02	55494949	Being Processed	36912158
71	2022-08-02	99303494	Being Processed	36912158
72	2022-08-05	4049495	Delivered	36912158
73	2022-08-05	33334949	Delivered	36912158
74	2022-08-05	46694949	Delivered	36912158
75	2022-08-05	48333949	Delivered	36912158
76	2022-08-05	48334949	Delivered	36912158
77	2022-08-05	48422249	Delivered	36912158
78	2022-08-05	48433949	Delivered	36912158
79	2022-08-05	48493339	Delivered	36912158
80	2022-08-05	48493349	Delivered	36912158
81	2022-08-05	48494455	Delivered	36912158
82	2022-08-05	48494944	Delivered	36912158
83	2022-08-05	48494949	Delivered	36912158
84	2022-08-05	55494949	Delivered	36912158
85	2022-08-05	99303494	Delivered	36912158
86	2022-08-15	3099121	Delivered	13579113
87	2022-08-15	23230101	Delivered	87829188
88	2022-08-15	32018301	Delivered	24681011
89	2022-08-15	36289172	Delivered	20494850
90	2022-08-15	63729102	Delivered	17872689
91	2022-08-20	72883791	Delivered	28494905

PriceHistory

	Startdate	Enddate	StockID	BookstoreID	Price
1	2022-05-01	2022-05-31	354681	197442	19
2	2022-05-01	2022-05-31	653241	197442	31
3	2022-05-01	2022-05-31	767281	223296	8
4	2022-06-01	2022-06-30	354681	197442	20
5	2022-06-01	2022-06-30	653241	197442	30
6	2022-06-01	2022-06-30	767281	223296	10
7	2022-07-01	2022-07-31	123456	457705	24
8	2022-07-01	2022-07-31	354681	197442	24
9	2022-07-01	2022-07-31	456791	457705	22
10	2022-07-01	2022-07-31	653241	197442	32
11	2022-07-01	2022-07-31	767281	223296	9
12	2022-08-01	2022-08-28	123456	457705	22
13	2022-08-01	2022-08-28	123459	457705	16
14	2022-08-01	2022-08-28	345276	290976	56
15	2022-08-01	2022-08-28	354681	197442	23
16	2022-08-01	2022-08-28	456789	457705	9
17	2022-08-01	2022-08-28	456791	457705	20
18	2022-08-01	2022-08-28	546387	290976	78
19	2022-08-01	2022-08-28	546388	290976	80
20	2022-08-01	2022-08-28	653241	197442	34
21	2022-08-01	2022-08-28	657482	353629	23
22	2022-08-01	2022-08-28	767281	223296	10
23	2022-08-01	2022-08-28	837261	353629	67
24	2022-08-01	2022-08-28	837262	353629	50
25	2022-08-01	2022-08-31	215468	197442	54
26	2022-08-01	2022-08-31	563468	949693	56

Publication

	PubID	Publisher	PubYear
1	23154879	SPH	2022
2	24587692	Nanyang Publisher Company	2020
3	34567890	Ethos Books	2018
4	35468921	SPH	2021
5	36485721	Nanyang Publisher Company	2018
6	46548732	SPH	2021
7	54876953	Nanyang Publisher Company	2019
8	64587236	Penguin	2018
9	76854921	SPH	2020
10	97772093	Penguin	2015
11	98541276	Penguin	2021

StocksinBookStores

	PubID	BookstoreID	StockID	Stock_Qty	Stock_Price
1	34567890	197442	215468	10	4
2	36485721	197442	354681	1	10
3	76854921	197442	653241	2	23
4	76854921	223296	767281	3	22
5	97772093	223296	767282	2	15
6	64587236	290976	345276	6	56
7	23154879	290976	345277	5	50
8	35468921	290976	546387	9	78
9	98541276	290976	546388	9	80
10	98541276	353629	657482	6	72
11	54876953	353629	657483	2	23
12	23154879	353629	837261	3	67
13	64587236	353629	837262	4	50
14	36485721	457705	123456	3	22
15	54876953	457705	123459	2	16
16	46548732	457705	456789	1	9
17	24587692	457705	456790	3	12
18	76854921	457705	456791	2	20
19	34567890	949693	563468	6	4

5 SQL Queries

In this section, we demonstrate the SQL statements, accompanied by the outputs and explanation to solve the queries specified in Appendix B of the Lab Manual.

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

SQL Code:

```
SELECT AVG(t1.price) AS ave_price
FROM(
      SELECT s.PubID,
             pr.Startdate,
             pr.Enddate,
             pr.price,
             COALESCE(b.Title, m.Title) AS title
      FROM StocksInBookstores s
           JOIN PriceHistory pr
                ON (s.StockID = pr.StockID AND s.BookstoreID =
                pr.BookstoreID)
           LEFT OUTER JOIN Books b
                ON s.PubID = b.PubID
           LEFT OUTER JOIN Magazines m
                ON s.PubID = m.PubID
) AS t1
WHERE Title = 'Harry Porter Finale'
      AND Startdate ⩾ '2022-08-01'
      AND Enddate ⩽ '2022-08-31'
```

Output:

	ave_price
1	55

Explanation and Insights:

This SQL query selects the average price of the book "Harry Porter Finale" from Ahamazon from 1 August 2022 to 31 August 2022. Here is s a breakdown of the query:

1. The inner query joins the StocksInBookstores table with the PriceHistory table to get the price history of all publications that were stocked in all bookstores.
2. The inner query also uses the LEFT OUTER JOIN operator to join the Books and Magazines tables with the StocksInBookstores table, using the PubID field, to get the title of each book or magazine.
3. The result of the inner query is used as a subquery (i.e., a derived table) and aliased as "t1".
4. The outer query selects the average price of the book "Harry Porter Finale" by filtering the results of the subquery based on the book title and the start and end dates of August 2022.

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

SQL Code:

```
SELECT S.PubID,
       CAST(AVG(I.Rating) AS DECIMAL (12,1)) AS AvgRating
FROM ItemsinOrders I
     JOIN StocksInBookstores S
          ON S.BookstoreID = I.BookstoreID
          AND S.StockID = I.StockID
WHERE I.RatingDatetime ⩾ '2022-08-01'
      AND I.RatingDatetime < '2022-09-01'
GROUP BY S.PubID
HAVING COUNT(CASE WHEN I.Rating = 5 THEN 1 END) ⩾ 10
ORDER BY AVG(I.Rating) DESC;
```

Output:

	PubID	AvgRating
1	98541276	5.0
2	34567890	4.0
3	76854921	4.0

Explanation and Insights:

The query joins the ItemsinOrders table with the StocksInBookstores table on matching BookstoreID and StockID columns, and filters the result to only include ratings submitted during August 2022. The query then groups the results by PubID and calculates the average rating for each publisher. Finally, the query filters the results to only include publishers whose items received at least 10 ratings of 5 stars and sorts the results by the calculated average rating in descending order.

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

Assumption: Average time is measured in terms of days

SQL Code:

```
SELECT PubID,
    AVG(DATEDIFF(DAY, Orders.Date_Time, iio.Deliverydate)) AS
    avg_delivery_time
FROM ItemsInOrders iio
    JOIN Orders
        ON iio.OrderID = Orders.OrderID
    JOIN StocksInBookstores sib
        ON iio.BookStoreID = sib.BookstoreID
WHERE MONTH(orders.Date_time) = 6
AND YEAR(Orders.Date_time) = 2022
GROUP BY PubID
```

Output:

	PubID	avg_delivery_time
1	23154879	6
2	24587692	5
3	34567890	14
4	35468921	6
5	36485721	13
6	46548732	5
7	54876953	5
8	64587236	6
9	76854921	13
10	98541276	6

Explanation and Insights:

It accomplishes this by joining the ItemsInOrders and Orders tables on OrderID, and then joining the resulting table with the StocksInBookstores table on BookstoreID. It calculates the average delivery time for each publisher by using the AVG() function and the DATEDIFF() function to calculate the number of days between the order date and delivery date. Finally, it groups the results by PubID using the GROUP BY clause.

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.

Assumption: Latency is measured in terms of days

SQL Code:

```
SELECT TOP 1 Name,
             t.avg_days
FROM (
      SELECT e2.employeeID,
             AVG(DATEDIFF(DAY,fileddatetime, handleddatetime))
             AS avg_days
      FROM employees e2
      JOIN complaints c
            ON c.employeeID = e2.employeeID
      GROUP BY e2.employeeID
      ) AS t
JOIN employees e1
      ON e1.employeeID = t.employeeID
ORDER BY t.avg_days;
```

Output:

	Name	avg_days
1	Talaat Lilibeth	3

Explanation and Insights:

The query first calculates the average time it took for each employee to handle a complaint by joining the employees table with the complaints table on matching employeeID columns and using the DATEDIFF function to calculate the difference in days between the fileddatetime and handleddatetime columns. The result is then grouped by employeeID and the average number of days is calculated and aliased as avg_days.

The result of the subquery is then joined with the employees table on matching employeeID columns, allowing the query to retrieve the name of the employee with the lowest average time to handle a complaint. Finally, the query orders the results by the avg_days in ascending order and selects the top 1 record, which corresponds to the employee with the lowest average time to handle a complaint.

5. (i) Produce a list that contains all publications published by Nanyang Publisher Company.

SQL Code:

```
SELECT p.PubID,
       COALESCE(b.title, m.title) AS Title
FROM Publication p
     LEFT JOIN Books b
           ON p.PubID = b.PubID
     LEFT JOIN Magazines m
           ON p.PubID = m.PubID
WHERE p.Publisher = 'Nanyang Publisher Company';
```

Output:

	PubID	Title
1	24587692	Burning Poem
2	36485721	Jazz Daily
3	54876953	Harry Porter Book 1

Explanation and Insights:

The query first selects from the Publication table and Leftjoins the Books table on matching PubID columns, and then Leftjoins the Magazines table on matching PubID columns. This allows the query to retrieve both books and magazines for each publication, while still including publications that may have only books or only magazines (using LEFT JOIN instead of INNER JOIN).

The COALESCE function is used to retrieve the title of the publication, either from the Books table or the Magazines table. If a title exists in the Books table, it will be selected. Otherwise, if a title exists in the Magazines table, it will be selected. If neither table has a title, the result will be NULL.

The query then filters the result to only include publications from the 'Nanyang Publisher Company', based on the Publisher column in the Publication table.

(ii) For each of them, the number of bookstores on Ahamazon that sell them.

SQL Code:

```
SELECT p.PubID,
       COUNT(s.BookstoreID) AS Number_of_Bookstores
FROM Publication p
     JOIN StocksInBookStores s
          ON p.PubID = s.PubID
          AND p.Publisher = 'Nanyang Publisher Company'
GROUP BY p.PubID;
```

Output:

	PubID	Number_of_Bookstores
1	24587692	1
2	36485721	2
3	54876953	2

Explanation and Insights:

This SQL query retrieves the bookstore with the highest revenue for the month of August 2022. Here's a breakdown of how it works:

1. The inner query calculates the revenue for each bookstore by joining four tables: bookstore, orders, ItemsInOrders, and Stocksinbookstores. It filters the results to only include orders delivered in August 2022 (using the MONTH() and YEAR() functions), and groups the results by bookstore.BookstoreID. The result of this query is a table aliased as "alias" with two columns: BookstoreID and Revenue.
2. The outer query selects the top row from the "alias" table with the highest Revenue value and renames the BookstoreID column as alias.BookstoreID. The MAX() function is not used here because the ORDER BY clause sorts the results in descending order of Revenue, so the first row returned will have the highest Revenue value.

The result of this query will be a table with a single row and two columns: alias.BookstoreID and MaxRevenue. The value in the alias.BookstoreID column will be the ID of the bookstore with the highest revenue, and the value in the MaxRevenue column will be the actual revenue amount.

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

SQL Code:

```
SELECT TOP 1 alias.BookstoreID,
      Revenue AS MaxRevenue
FROM (
      SELECT bookstore.BookstoreID,
            SUM(ItemPrice * ItemQty) AS Revenue
      FROM bookstore,
            Orders,
            ItemsInOrders,
            Stocksinbookstores AS s
      WHERE month(Deliverydate) = 8
            AND year(Deliverydate) = 2022
            AND bookstore.BookstoreID = s.BookstoreID
            AND ItemsInOrders.OrderID = Orders.OrderID
            AND s.StockID = ItemsInOrders.StockID
GROUP BY bookstore.BookstoreID
) AS alias
ORDER BY revenue DESC;
```

Output:

	BookstoreID	MaxRevenue
1	197442	1006

Explanation and Insights:

The query first selects from a subquery that calculates the revenue for each bookstore based on orders delivered in August 2022. The subquery joins the bookstore table with the Orders table and the ItemsInOrders table on matching BookstoreID and OrderID columns, and then joins the Stocksinbookstores table on matching BookstoreID and StockID columns. The subquery

calculates the total revenue for each bookstore by multiplying the ItemPrice and ItemQty columns and then sums the result for each bookstore, grouping the results by BookstoreID. The result of the subquery is aliased as alias.

The outer query then selects the top 1 record from the result of the subquery, which corresponds to the bookstore with the highest revenue in August 2022. The result includes the ID of the bookstore (BookstoreID) and the maximum revenue (MaxRevenue). The ORDER BY clause sorts the result in descending order by revenue, allowing the TOP 1 function to select the record with the highest revenue.

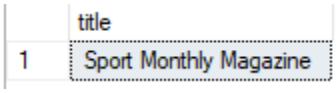
7. For customers that made the most number of complaints, find the most expensive publication he/she has ever purchased.

SQL Code:

```
SELECT TOP 1 COALESCE(b.Title, m.Title) AS title
FROM (
	SELECT *
	FROM ItemsInOrders iio
	WHERE OrderID = (
	SELECT OrderID
	FROM(
		SELECT TOP 1 customerID,
			COUNT(customerID) AS ComplaintCount
		FROM Complaints
		GROUP BY customerID
		ORDER BY ComplaintCount DESC
	) AS t1
	JOIN Orders
		ON o.CustomerID = t1.customerID
	)
) AS t2

JOIN StocksInBookstores s
	ON s.BookstoreID = t2.BookstoreID
	AND s.StockID = t2.StockID
LEFT OUTER JOIN Books b
	ON s.PubID = b.PubID
LEFT OUTER JOIN Magazines m
	ON s.PubID = m.PubID
ORDER BY ItemPrice DESC
```

Output:



	title
1	Sport Monthly Magazine

Explanation and Insights:

This SQL query retrieves the most expensive title of the publication made by the customer who have filed the most complaints. Here's a breakdown of the query:

1. The innermost subquery retrieves the customerID and the count of complaints for each customer, sorted in descending order of complaint count. The TOP 1 clause selects only the customer with the most complaints.
2. The middle subquery retrieves the OrderID of the customer selected in the previous subquery, by joining the Orders table with the result of the previous subquery.
3. The outermost subquery retrieves all items from the ItemsInOrders table that belong to the order selected in the previous subquery.
4. The result of the outermost subquery is aliased as "t2".
5. The main query joins the result of the outermost subquery with the StocksInBookstores table to retrieve the corresponding book or magazine for each item.
6. The LEFT OUTER JOIN operator is used to join the Books and Magazines tables with the StocksInBookstores table to retrieve the title of each book or magazine.
7. The result of the main query is sorted in descending order of item price, and the TOP 1 clause selects only the first row, which contains the item with the highest price.

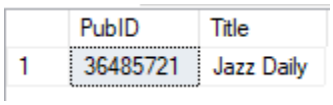
8. Find publications that have never been purchased by any customer in July 2022, but are the top 3 most purchased publications in August 2022.

SQL Code:

```
SELECT t1.PubID,
       COALESCE(b.title, m.title) AS Title
FROM (
      SELECT TOP 3 s2.PubID,
             COUNT(i2.StockID) AS number_of_sales
      FROM itemsinorders i2
      JOIN StocksInBookstores s2
            ON i2.StockID = s2.StockID
            AND i2.BookstoreID = s2.BookstoreID
      JOIN Orders o2
            ON o2.OrderID = i2.OrderID
      WHERE o2.Date_time ⩽ '2022-08-31 23:59:59'
            AND o2.Date_time ⩾ '2022-08-01 00:00:00'
      GROUP BY s2.PubID
      ORDER BY number_of_sales DESC
    ) AS t1
LEFT JOIN Books b
      ON t1.PubID = b.PubID
LEFT JOIN Magazines m
      ON t1.PubID = m.PubID
WHERE t1.PubID NOT IN (
      SELECT PubID
      FROM itemsinorders i1
      JOIN Orders o1
            ON o1.OrderID = i1.OrderID
      JOIN StocksInBookstores s1
            ON i1.StockID = s1.StockID
            AND i1.BookstoreID = s1.BOOKSTOREID
```

```
        WHERE o1.Date_time ⩽ '2022-07-31 23:59:59'
              AND o1.Date_time ⩾ '2022-07-01 00:00:00'
    )
```

Output:



	PubID	Title
1	36485721	Jazz Daily

Explanation and Insights:

1. The inner query retrieves the top 3 Publishers based on their number of book or magazine sales in August 2022. It joins the tables "itemsinorders", "StocksInBookstores", and "Orders" and filters the results to include only sales made in August 2022. It groups the results by the PubID column and orders the results by the number_of_sales column in descending order. The result of this query is a table aliased as "t1" with two columns: PubID and Title.
2. The outer query left joins the "t1" table with the "Books" and "Magazines" tables on the PubID column. It uses the COALESCE() function to return the title of the publisher's top-selling book or magazine, depending on which table has a match. The result of this query is a table with two columns: PubID and Title.
3. The WHERE clause excludes any publishers whose books or magazines were sold in July 2022. It achieves this by comparing the PubID column to the result of an inner query that retrieves PubIDs of sales made in July 2022.

In summary, this query retrieves the top-selling publishers in August 2022 and their top-selling book or magazine, but only if they did not sell any books or magazines in July 2022.

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

SQL Code:

```
CREATE VIEW v1 AS
SELECT sib.pubID,
      MONTH(o.date_time) AS month_of_purchase,
      SUM(itemqty) AS sales
FROM ItemsInOrders iio
JOIN Orders o
      ON iio.OrderID = o.OrderID
JOIN StocksInBookstores sib
      ON iio.BookstoreID = sib.BookstoreID
      AND iio.StockID = sib.StockID
GROUP BY sib.PubID, MONTH(o.date_time);
SELECT a1.pubID
FROM v1 AS a1
JOIN v1 AS a2
      ON a1.PubID = a2.PubID
      AND a1.month_of_purchase = a2.month_of_purchase+1
JOIN v1 AS a3
      ON a2.PubID =a3.PubID
      AND a2.month_of_purchase = a3.month_of_purchase + 1
WHERE  a3.sales < a2.sales
      AND a2.sales < a1.sales;
```

Outputs:

	pubID
1	34567890

Explanation and Insights:

1. The first part of the query creates a view named "v1" by joining the tables "ItemsInOrders", "Orders", and "StocksInBookstores". It retrieves the PubID, month_of_purchase, and total sales of each item in each order. The results are grouped by PubID and month_of_purchase. The resulting table is saved as the view "v1".
2. The second part of the query joins the "v1" view with itself three times (aliased as a1, a2, and a3) to compare sales data for three consecutive months for each publisher. It first joins a1 with a2 to compare sales for two consecutive months for each publisher. Then, it joins a2 with a3 to compare sales for three consecutive months for each publisher.
3. The WHERE clause filters the results to include only publishers whose sales increased for three consecutive months. Specifically, it checks that the sales for the third month (a3.sales) are less than the sales for the second month (a2.sales), and that the sales for the second month (a2.sales) are less than the sales for the first month (a1.sales).

In summary, this query creates a view of sales data for each publisher by month, and then finds the publishers whose sales increased for three consecutive months.

6 Conclusion

We would like to express our greatest gratitude to our lab professor, Assoc Prof Hui Siu Cheung, and our TA, Yuan Xi, for their utmost guidance and support for the entirety of the lab sessions of SC2207, as well as the course coordinators Prof Ng Wee Keong, Prof Quah Tong Seng and Prof Cong Gao for the knowledge and wisdom imparted to us. We have gained a broad base of knowledge on relational database design, implementation and analysis and a working knowledge of database querying using SQL, and have gained confidence to carry our knowledge gained from this course to real world implementations in the future.