



INTRODUCTION TO DATABASE

CT042-3-1-IDB

Module Lecturer : Farhana Illiani Binti Hassan

Date Assigned : 18th September 2023

Due Date : 20th November 2023

Group Member:

Student Name	TP Number	Role
Shifra Shuaib Ali	TP073263	Group Leader
Leong Ciao Shi	TP072423	Group Member
Viishnu Sree Ganesh	TP067756	Group Member

Table of Contents:

1.0 Introduction	4
2.0 Database Schema.....	4
2.1 Entity Relationship Diagram (ERD)	4
2.2 Database Diagram.....	5
3.0 SQL-Data Definition Language (DDL)	6
3.1 Create and Insert Data for Manager Table	7
3.2 Create and Insert Data for Publisher Table	8
3.3 Create and Insert Data for Member Table	9
3.4 Create and Insert Data for Book Table	10
3.5 Create and Insert Data for Cart Table.....	11
3.6 Create and Insert Data for Cart Detail Table	12
3.7 Create and Insert Data for Member Receipt Table	13
3.8 Create and Insert Data for Receipt Detail Table	14
3.9 Create and Insert Data for Order List Table.....	15
3.10 Create and Insert Data for BoolTable	16
3.11 Create and Insert Data for Rating Table	17
4.0 SQL-Data Manipulation Language (DML)	18
4.1 Finding the total number of books published by each publisher	19
4.2 Finding the books in the shopping cart which has not been checked out	20
4.3 Finding the books which has the highest average ratings.....	20
4.4 Finding the total number of feedback from each member.....	21
4.5 Finding the publisher who published the most books	22
4.6 Finding the total number of books ordered by the store manager from every publisher	23
4.7 Finding the genre of book which is most in stock	24
4.8 Finding the bestselling book	24
4.9 Finding the members who have spent the most on buying books	25
4.11 List of purchased books that have not been delivered to the members	26
4.12 Members who have more than two orders	26
Conclusion.....	27
Workload Matrix.....	28
References:	29

1.0 Introduction

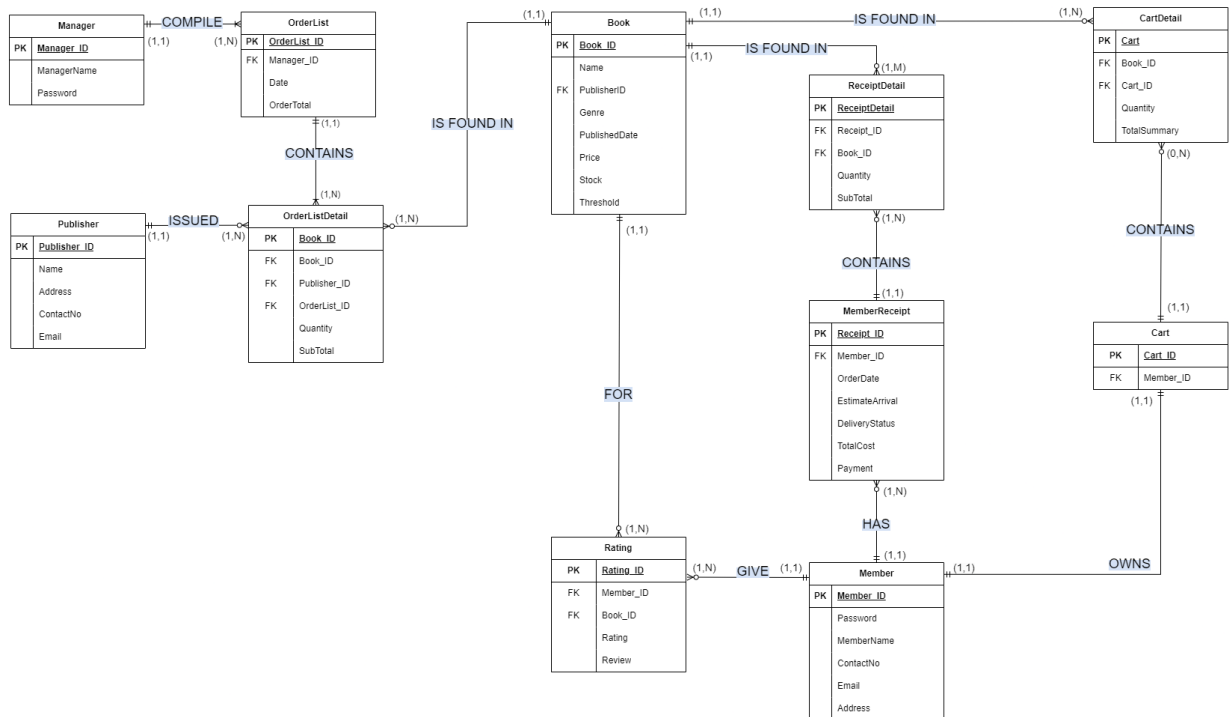
This documentation provides step-by-step guidance on the implementing and maximizing the potential of Database Management System (DBMS) procedures to in-depth feature explanations, for the E-Bookstore.

2.0 Database Schema

A database schema is a blueprint that defines the structure of a database, specifying tables, attributes, data types, and relationships. It serves as a roadmap for organizing and linking data, ensuring data integrity and efficient retrieval. Constraints and keys enforce rules, contributing to a systematic and organized representation of information within a relational database.

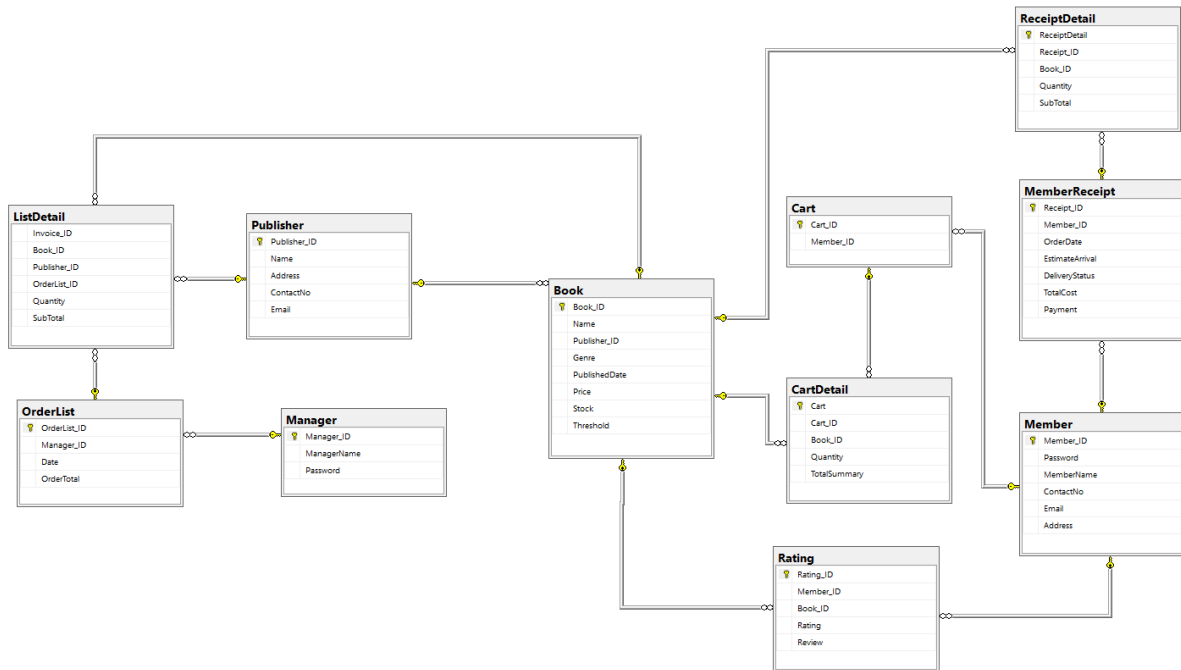
2.1 Entity Relationship Diagram (ERD)

The ERD below shows the updated version for the diagram which has been previously submitted in part one.



2.2 Database Diagram

The diagram below shows the database diagram generated from the DBMS.



3.0 SQL-Data Definition Language (DDL)

Function	Description	Code
CREATE DATABASE	Creates a new database.	CREATE DATABASE IDB_Assignment;
USE	Switches to a specific database.	USE IDB_Assignment;
CREATE TABLE	Defines the structure of a new table.	CREATE TABLE Manager (Manager_ID nvarchar(10) PRIMARY KEY NOT NULL, ManagerName char(30), Password nvarchar(20));
INSERT INTO	Adds new records into a table.	INSERT INTO Manager (Manager_ID, ManagerName, Password) VALUES ('M01', 'Sam Smith', 'sam1234');

3.1 Create and Insert Data for Manager Table

ManagerID (PK)	ManagerName	Password
M01	Sam Smith	sam1234
M02	John Legend	john456
M03	Taylor Swift	taylor789
M04	Avril Lagine	avril123
M05	Akon	akon123

```
4  --Manager Table
5  CREATE TABLE Manager (
6      Manager_ID nvarchar(10) PRIMARY KEY NOT NULL,
7      ManagerName char (30),
8      Password nvarchar(20)
9  );
10 INSERT INTO Manager (Manager_ID,ManagerName,Password)
11 VALUES
12 ('M01','Sam Smith','sam1234'),
13 ('M02','John Legend','john456'),
14 ('M03','Taylor Swift','taylor789'),
15 ('M04','Avril Lagine','avril123'),
16 ('M05','Akon','akon123')
17
```

Results Messages

	Manager_ID	ManagerName	Password
1	M01	Sam Smith	sam1234
2	M02	John Legend	john456
3	M03	Taylor Swift	taylor789
4	M04	Avril Lagine	avril123
5	M05	Akon	akon123

3.2 Create and Insert Data for Publisher Table

Publisher_ID (PK)	Name	Address	ContactNo	Email
P201	ABJ Company	Bukit Jalil	0146589745	abjco@work.my
P202	JZ Company	Penang	0128546589	jzco@work.my
P203	JANT Company	Ipoh	0103257465	jantco@work.my
P204	QRET Company	Kuching	0154589631	qretco@work.my
P205	SAKU Company	Sabah	0114563258	sakuco@work.my
P206	APD Company	Melaka	0175864528	apdco@work.my

```

20  --Publisher
21  CREATE TABLE Publisher (
22      Publisher_ID nvarchar(10) PRIMARY KEY NOT NULL,
23      Name char(30),
24      Address char (15),
25      ContactNo varchar(20),
26      Email nvarchar(20)
27  );
28  INSERT INTO Publisher (Publisher_ID,Name,Address,ContactNo,Email)
29  VALUES
30  ('P201','ABJ Company','Bukit Jalil','0146589745','abjco@work.my'),
31  ('P202','JZ Company','Penang','0128546589','jzco@work.my'),
32  ('P203','JANT Company','Ipoh','0103257465','jantco@work.my'),
33  ('P204','QRET Company','Kuching','0154589631','qretco@work.my'),
34  ('P205','SAKU Company','Sabah','0114563258','sakuco@work.my'),
35  ('P206','APD Company','Melaka','0175864528','apdco@work.my')
36

```

Results Messages

	Publisher_ID	Name	Address	ContactNo	Email
1	P201	ABJ Company	Bukit Jalil	0146589745	abjco@work.my
2	P202	JZ Company	Penang	0128546589	jzco@work.my
3	P203	JANT Company	Ipoh	0103257465	jantco@work.my
4	P204	QRET Company	Kuching	0154589631	qretco@work.my
5	P205	SAKU Company	Sabah	0114563258	sakuco@work.my
6	P206	APD Company	Melaka	0175864528	apdco@work.my

3.3 Create and Insert Data for Member Table

Member_ID (PK)	Password	MemberName	ContactNo	Email	Address
I001	lim1234	Audrey Lim	0127851258	audrey@gmail.com	Kuching
I002	kho1234	Mary Kho	0153268575	mary@gmail.com	Puchong
I003	crocker1234	Issabelle Crocker	0149854568	issabelle@gmail.com	Kajang
I004	yang1234	Sia Jin Yang	0168961854	sia@gmail.com	Semenyih
I005	ali1234	Mishya Ali	0176589346	mishya@gmail.com	Bukit Jalil
I006	izzat1234	Mohd Izzat	0180159634	mohd@gmail.com	Kuala Lumpur
I007	bong1234	Rose Bong	0198541286	rose@gmail.com	Ampang

```

38  --Member
39  CREATE TABLE Member(
40      Member_ID nvarchar(10) PRIMARY KEY NOT NULL,
41      Password nvarchar(20),
42      MemberName char(30),
43      ContactNo varchar(20),
44      Email nvarchar(30),
45      Address char(20)
46  );
47  INSERT INTO Member (Member_ID,Password,MemberName,ContactNo,Email,Address)
48  VALUES
49  ('I001','lim1234','Audrey Lim','0127851258','audrey@gmail.com','Kuching'),
50  ('I002','kho1234','Mary Kho','0153268575','mary@gmail.com','Puchong'),
51  ('I003','crocker1234','Issabelle Crocker','0149854568','issabelle@gmail.com','Kajang'),
52  ('I004','yang1234','Sia Jin Yang','0168961854','sia@gmail.com','Semenyih'),
53  ('I005','ali1234','Mishya Ali','0176589346','mishya@gmail.com','Bukit Jalil'),
54  ('I006','izzat1234','Mohd Izzat','0180159634','mohd@gmail.com','Kuala Lumpur'),
55  ('I007','bong1234','Rose Bong','0198541286','rose@gmail.com','Ampang')

```

Results Messages

	Member_ID	Password	MemberName	ContactNo	Email	Address
1	I001	lim1234	Audrey Lim	0127851258	audrey@gmail.com	Kuching
2	I002	kho1234	Mary Kho	0153268575	mary@gmail.com	Puchong
3	I003	crocker1234	Issabelle Crocker	0149854568	issabelle@gmail.com	Kajang
4	I004	yang1234	Sia Jin Yang	0168961854	sia@gmail.com	Semenyih
5	I005	ali1234	Mishya Ali	0176589346	mishya@gmail.com	Bukit Jalil
6	I006	izzat1234	Mohd Izzat	0180159634	mohd@gmail.com	Kuala Lumpur
7	I007	bong1234	Rose Bong	0198541286	rose@gmail.com	Ampang

3.4 Create and Insert Data for Book Table

Book_ID (PK)	Name	Publisher_ID (FK)	Genre	PublishedDate	Price	Stock	Threshold
B1	The Girl With No Name	P201	Mystery	1-Jan-16	54	40	10
B2	In The Name of Love	P202	Romance	9-Jan-15	60	15	10
B3	Call Me By Your Name	P202	Romance	23-Jan-07	45	20	15
B4	The Goldfinch	P203	Mystery	22-Oct-13	65	30	10
B5	A Little Life	P203	Fiction	10-Mar-15	76	12	10
B6	Life of Pi	P204	Adventure	11-Sep-01	70	25	15
B7	The Vanishing Half	P205	Finction	2-Jun-20	65	15	10
B8	The Hating Game	P206	Romance	9-Aug-16	90	13	10
B9	Diary of Wimpy Kid	NULL	Fantasy	25-Dec-99	100	15	10

```

59  --Book
60  CREATE TABLE Book (
61      Book_ID nvarchar(10) PRIMARY KEY NOT NULL,
62      Name nvarchar(50),
63      Publisher_ID nvarchar(10) FOREIGN KEY REFERENCES Publisher(Publisher_ID),
64      Genre char(15),
65      PublishedDate date,
66      Price int,
67      Stock int,
68      Threshold int
69  );
70
71  INSERT INTO Book (Book_ID,Name,Publisher_ID,Genre,PublishedDate,Price,Stock,Threshold)
72  VALUES
73  ('B1','The Girl With No Name','P201','Mystery','1 Jan 2016',54,40,10),
74  ('B2','In The Name of Love','P202','Romance','9 Jan 2015',60,15,10),
75  ('B3','Call Me By Your Name','P202','Romance','23 Jan 2007',45,20,15),
76  ('B4','The Goldfinch','P203','Mystery','22 Oct 2013',65,30,10),
77  ('B5','A Little Life','P203','Fiction','10 Mar 2015',76,12,10),
78  ('B6','Life of Pi','P204','Adventure','11 Sep 2001',70,25,15),
79  ('B7','The Vanishing Half','P205','Finction','2 Jun 2020',65,15,10),
80  ('B8','The Hating Game','P206','Romance','9 Aug 2016',90,13,10),
81  ('B9','Diary of Wimpy Kid',NULL,'Fantasy','25 Dec 1999',100,15,10)

```

Results Messages

	Book_ID	Name	Publisher_ID	Genre	PublishedDate	Price	Stock	Threshold
1	B1	The Girl With No Name	P201	Mystery	2016-01-01	54	40	10
2	B2	In The Name of Love	P202	Romance	2015-01-09	60	15	10
3	B3	Call Me By Your Name	P202	Romance	2007-01-23	45	20	15
4	B4	The Goldfinch	P203	Mystery	2013-10-22	65	30	10
5	B5	A Little Life	P203	Fiction	2015-03-10	76	12	10
6	B6	Life of Pi	P204	Adventure	2001-09-11	70	25	15
7	B7	The Vanishing Half	P205	Finction	2020-06-02	65	15	10
8	B8	The Hating Game	P206	Romance	2016-08-09	90	13	10
9	B9	Diary of Wimpy Kid	NULL	Fantasy	1999-12-25	100	15	10

3.5 Create and Insert Data for Cart Table

Cart_ID (PK)	Member_ID (FK)
1	I001
2	I002
3	I003
4	I004
5	I005
6	I006
7	I007

```
184 --Cart
185 CREATE TABLE Cart(
186     Cart_ID varchar(10) PRIMARY KEY NOT NULL,
187     Member_ID nvarchar (10) FOREIGN KEY REFERENCES Member(Member_ID)
188 );
189 INSERT INTO Cart (Cart_ID,Member_ID)
190 VALUES
191 ('1','I001'),
192 ('2','I002'),
193 ('3','I003'),
194 ('4','I004'),
195 ('5','I005'),
196 ('6','I006'),
197 ('7','I007');
```

Results Messages

	Cart_ID	Member_ID
1	1	I001
2	2	I002
3	3	I003
4	4	I004
5	5	I005
6	6	I006
7	7	I007

3.6 Create and Insert Data for Cart Detail Table

Cart (PK)	Cart_ID (FK)	Book_ID (FK)	Quantity	TotalSummary
1	1	B8	1	225
2	1	B6	1	225
3	1	B4	1	225
4	2	B5	1	76
5	3	B6	1	70

```
201  --CartDetail
202  CREATE TABLE CartDetail(
203      Cart varchar(5) PRIMARY KEY NOT NULL,
204      Cart_ID varchar(10) FOREIGN KEY REFERENCES Cart(Cart_ID),
205      Book_ID nvarchar(10) FOREIGN KEY REFERENCES Book(Book_ID),
206      Quantity int,
207      TotalSummary int
208  );
209  INSERT INTO CartDetail(Cart, Cart_ID, Book_ID, Quantity, TotalSummary)
210  VALUES
211  ('1', '1', 'B8', 1, 225),
212  ('2', '1', 'B6', 1, 225),
213  ('3', '1', 'B4', 1, 225),
214  ('4', '2', 'B5', 1, 76),
215  ('5', '3', 'B6', 1, 70)
```

Results Messages

	Cart	Cart_ID	Book_ID	Quantity	TotalSummary
1	1	1	B8	1	225
2	2	1	B6	1	225
3	3	1	B4	1	225
4	4	2	B5	1	76
5	5	3	B6	1	70

3.7 Create and Insert Data for Member Receipt Table

Receipt_ID (PK)	Member_ID (FK)	OrderDate	EstimateArrival	DeliveryStatus	TotalCost	Payment
501	I001	1-Dec-22	7-Dec-22	Delivered	99	Online Banking
502	I001	20-Dec-22	26-Dec-22	Delivered	70	COD
503	I002	1-Jan-23	7-Jan-23	Delivered	60	Debit Card
504	I003	4-Jan-23	10-Jan-23	Not Delivered	130	Online Banking
505	I004	7-Jan-23	14-Jan-23	Not Delivered	90	COD
506	I001	20-Feb-23	26-Feb-23	Delivered	70	Online Banking

```

144 --MemberReceipt
145 CREATE TABLE MemberReceipt(
146     Receipt_ID varchar(10) PRIMARY KEY NOT NULL,
147     Member_ID nvarchar(10) FOREIGN KEY REFERENCES Member(Member_ID),
148     OrderDate date,
149     EstimateArrival date,
150     DeliveryStatus char(20),
151     TotalCost int,
152     Payment char(15)
153 );
154 INSERT INTO MemberReceipt(Receipt_ID,Member_ID,OrderDate,EstimateArrival,DeliveryStatus,TotalCost,Payment)
155 VALUES
156 ('501','I001','1 Dec 2022','7 Dec 2022','Delivered',99,'Online Banking'),
157 ('502','I001','20 Dec 2022','26 Dec 2022','Delivered',70,'COD'),
158 ('503','I002','1 Jan 2023','7 Jan 2023','Delivered',60,'Debit Card'),
159 ('504','I003','4 Jan 2023','10 Jan 2023','Not Delivered',130,'Online Banking'),
160 ('505','I004','7 Jan 2023','14 Jan 2023','Not Delivered',90,'COD'),
161 ('506','I001','20 Feb 2023','26 Feb 2023','Delivered',70,'Online Banking')

```

Results Messages

	Receipt_ID	Member_ID	OrderDate	EstimateArrival	DeliveryStatus	TotalCost	Payment
1	501	I001	2022-12-01	2022-12-07	Delivered	99	Online Banking
2	502	I001	2022-12-20	2022-12-26	Delivered	70	COD
3	503	I002	2023-01-01	2023-01-07	Delivered	60	Debit Card
4	504	I003	2023-01-04	2023-01-10	Not Delivered	130	Online Banking
5	505	I004	2023-01-07	2023-01-14	Not Delivered	90	COD
6	506	I001	2023-02-20	2023-02-26	Delivered	70	Online Banking

3.8 Create and Insert Data for Receipt Detail Table

ReceiptDetail (PK)	Receipt_ID (FK)	Book_ID (FK)	Quantity	SubTotal
5001	501	B1	1	54
5002	501	B3	1	45
5003	502	B6	1	70
5004	503	B2	1	60
5005	504	B4	1	65
5006	504	B7	1	65
5007	505	B8	1	90
5008	506	B6	1	70

```
165 --ReceiptDetail
166 CREATE TABLE ReceiptDetail(
167     ReceiptDetail varchar(10) PRIMARY KEY NOT NULL,
168     Receipt_ID varchar(10) FOREIGN KEY REFERENCES MemberReceipt(Receipt_ID),
169     Book_ID nvarchar(10) FOREIGN KEY REFERENCES Book(Book_ID),
170     Quantity int,
171     SubTotal int
172 );
173 INSERT INTO ReceiptDetail(ReceiptDetail,Receipt_ID,Book_ID,Quantity,SubTotal)
174 VALUES
175 ('5001','501','B1',1,54),
176 ('5002','501','B3',1,45),
177 ('5003','502','B6',1,70),
178 ('5004','503','B2',1,60),
179 ('5005','504','B4',1,65),
180 ('5006','504','B7',1,65),
181 ('5007','505','B8',1,90),
182 ('5008','506','B6',1,70)
183
```

Results Messages

	ReceiptDetail	Receipt_ID	Book_ID	Quantity	SubTotal
1	5001	501	B1	1	54
2	5002	501	B3	1	45
3	5003	502	B6	1	70
4	5004	503	B2	1	60
5	5005	504	B4	1	65
6	5006	504	B7	1	65
7	5007	505	B8	1	90
8	5008	506	B6	1	70

3.9 Create and Insert Data for Order List Table

OrderList_ID (PK)	Manager_ID (FK)	Date	OrderTotal
2001	M01	22/2/2022	1040
2002	M01	5/4/2022	225
2003	M02	7/5/2022	1250
2004	M02	12/12/2022	650
2005	M03	3/2/2023	1350

```
85 CREATE TABLE OrderList (  
86     OrderList_ID varchar(10) PRIMARY KEY NOT NULL,  
87     Manager_ID nvarchar (10) FOREIGN KEY REFERENCES Manager(Manager_ID),  
88     Date date,  
89     OrderTotal int  
90 );  
91 INSERT INTO OrderList (OrderList_ID,Manager_ID,Date,OrderTotal)  
92 VALUES  
93 (2001,'M01','22 February 2022',1040),  
94 (2002,'M01','5 April 2022',225),  
95 (2003,'M02','7 May 2022',1250),  
96 (2004,'M02','12 December 2022',650),  
97 (2005,'M03','3/2/2023',1350)
```

Results Messages

	OrderList_ID	Manager_ID	Date	OrderTotal
1	2001	M01	2022-02-22	1040
2	2002	M01	2022-04-05	225
3	2003	M02	2022-05-07	1250
4	2004	M02	2022-12-12	650
5	2005	M03	2023-03-02	1350

3.10 Create and Insert Data for BoolTable

Invoice_ID (PK)	Book_ID (FK)	Publisher_ID (FK)	OrderList_ID (FK)	Quantity	SubTotal
3001	B1	P201	2001	10	540
3002	B2	P202	2001	10	600
3003	B3	P202	2002	5	225
3004	B4	P203	2003	8	520
3005	B5	P203	2003	5	380
3006	B6	P204	2003	5	350
3007	B7	P205	2004	10	650
3008	B8	P206	2005	15	1350

```

101  --ListDetail
102  CREATE TABLE ListDetail(
103      Invoice_ID varchar(10),
104      Book_ID nvarchar(10) FOREIGN KEY REFERENCES Book(Book_ID),
105      Publisher_ID nvarchar(10) FOREIGN KEY REFERENCES Publisher(Publisher_ID),
106      OrderList_ID varchar(10) FOREIGN KEY REFERENCES OrderList(OrderList_ID),
107      Quantity int,
108      SubTotal int
109  );
110  INSERT INTO ListDetail(Invoice_ID,Book_ID,Publisher_ID,OrderList_ID,Quantity,SubTotal)
111  VALUES
112  ('3001','B1','P201','2001',10,540),
113  ('3002','B2','P202','2001',10,600),
114  ('3003','B3','P202','2002',5,225),
115  ('3004','B4','P203','2003',8,520),
116  ('3005','B5','P203','2003',5,380),
117  ('3006','B6','P204','2003',5,350),
118  ('3007','B7','P205','2004',10,650),
119  ('3008','B8','P206','2005',15,1350)

```

Results Messages

	Invoice_ID	Book_ID	Publisher_ID	OrderList_ID	Quantity	SubTotal
1	3001	B1	P201	2001	10	540
2	3002	B2	P202	2001	10	600
3	3003	B3	P202	2002	5	225
4	3004	B4	P203	2003	8	520
5	3005	B5	P203	2003	5	380
6	3006	B6	P204	2003	5	350
7	3007	B7	P205	2004	10	650
8	3008	B8	P206	2005	15	1350

3.11 Create and Insert Data for Rating Table

Rating_ID (PK)	Member_ID (FK)	Book_ID (FK)	Rating	Review
4001	I001	B3	6	Must-Read
4002	I001	B1	4	Clever plot
4003	I002	B2	5	entertaining read
4004	I003	B4	3	Not my cup of tea.
4005	I004	B8	8	Satisfied
4006	I005	B3	5	Decent overall.
4007	I001	B6	6	An imaginative and immersive journey.
4008	I004	B3	7	Nice book

```

123  --Rating
124  CREATE TABLE Rating(
125      Rating_ID varchar(10) PRIMARY KEY NOT NULL,
126      Member_ID nvarchar(10) FOREIGN KEY REFERENCES Member(Member_ID),
127      Book_ID nvarchar(10) FOREIGN KEY REFERENCES Book(Book_ID),
128      Rating int,
129      Review nvarchar(50)
130  );
131  INSERT INTO Rating(Rating_ID,Member_ID,Book_ID,Rating,Review)
132  VALUES
133  ('4001','I001','B3',6,'Must-Read'),
134  ('4002','I001','B1',4,'Clever plot'),
135  ('4003','I002','B2',5,'entertaining read'),
136  ('4004','I003','B4',3,'Not my cup of tea'),
137  ('4005','I004','B8',8,'Satisfied'),
138  ('4006','I005','B3',5,'Decent overall'),
139  ('4007','I001','B6',6,'An imaginative and immersive journey'),
140  ('4008','I004','B3',7,'Nice book')

```

Results Messages

	Rating_ID	Member_ID	Book_ID	Rating	Review
1	4001	I001	B3	6	Must-Read
2	4002	I001	B1	4	Clever plot
3	4003	I002	B2	5	entertaining read
4	4004	I003	B4	3	Not my cup of tea
5	4005	I004	B8	8	Satisfied
6	4006	I005	B3	5	Decent overall
7	4007	I001	B6	6	An imaginative and immersive journey
8	4008	I004	B3	7	Nice book

4.0 SQL-Data Manipulation Language (DML)

Data manipulation language or DML, will allow the E-Bookstore to perform various operations such as querying, requesting filtered or sorted information from the database, updating, inserting and deleting records (Rouse, 2019).

Function	Description	Code
SELECT	Retrieves data from one or more tables.	SELECT P.Publisher_ID, P.Name AS PublisherName, COUNT(B.Book_ID) AS NumBooksPublished FROM Publisher P LEFT JOIN Book B ON P.Publisher_ID = B.Publisher_ID GROUP BY P.Publisher_ID, P.Name;
JOIN	Combines rows from two or more tables based on a related column.	JOIN CartDetail CD ON CD.Cart_ID = C.Cart_ID
GROUP BY	Groups identical data based on specified columns.	GROUP BY P.Publisher_ID, P.Name
WHERE	Filters rows based on specified conditions.	WHERE NOT EXISTS (SELECT 1 FROM MemberReceipt MR JOIN ReceiptDetail RD ON MR.Receipt_ID = RD.Receipt_ID WHERE MR.Member_ID = M.Member_ID AND RD.Book_ID = B.Book_ID);
SUM	Calculates the sum of a set of values.	SUM (MemberReceipt.TotalCost)
MAX	Retrieves the maximum value from a set of values.	MAX (Stock)
COUNT	Counts the number of rows or occurrences of a value.	COUNT (R.Rating_ID)
AVG	Calculates the average value of a set of numeric values.	AVG (R.Rating)

ORDER BY	Arranges the result set in a specified order.	ORDER BY Total_Sales DESC;
FOREIGN KEY	Establishes relationships between tables.	Publisher_ID nvarchar(10) FOREIGN KEY REFERENCES Publisher (Publisher_ID)
LEFT JOIN	Returns all rows from the left table and matching rows from the right table.	LEFT JOIN ListDetail LD ON O.OrderList_ID = LD.OrderList_ID
HAVING	Filters results based on aggregate functions in combination with GROUP BY .	HAVING COUNT (DISTINCT MR.Receipt_ID) >= 2;
TOP 1 WITH TIES	Filter all rows tied for the top value are included in the results by not just showing the single top row.	SELECT TOP 1 WITH TIES P.Publisher_ID, P.Name AS Publisher_Name, COUNT(B.Book_ID) AS Total_Books_Published

4.1 Finding the total number of books published by each publisher

```

222 --i) To find the total number of books published by each publisher
223 SELECT P.Publisher_ID, P.Name AS PublisherName, COUNT(B.Book_ID) AS NumBooksPublished
224 FROM Publisher P
225 LEFT JOIN Book B ON P.Publisher_ID = B.Publisher_ID
226 GROUP BY P.Publisher_ID, P.Name;

```

Results Messages

	Publisher_ID	PublisherName	NumBooksPublished
1	P201	ABJ Company	1
2	P202	JZ Company	2
3	P203	JANT Company	2
4	P204	QRET Company	1
5	P205	SAKU Company	1
6	P206	APD Company	1

4.2 Finding the books in the shopping cart which has not been checked out

```
228 --ii) books in the shopping cart for each member which have not been check-out
229
230 SELECT CD.Cart_ID, M.MemberName, B.Book_ID, B.Name AS BookName, CD.Quantity, CD.TotalSummary
231 FROM CartDetail CD
232 INNER JOIN Cart C ON CD.Cart_ID = C.Cart_ID
233 INNER JOIN Member M ON C.Member_ID = M.Member_ID
234 INNER JOIN Book B ON CD.Book_ID = B.Book_ID
235 WHERE NOT EXISTS (
236     SELECT 1
237     FROM MemberReceipt MR
238     JOIN ReceiptDetail RD ON MR.Receipt_ID = RD.Receipt_ID
239     WHERE MR.Member_ID = M.Member_ID AND RD.Book_ID = B.Book_ID
240 );
241
```

Results Messages

	Cart_ID	MemberName	Book_ID	BookName	Quantity	TotalSummary
1	1	Audrey Lim	B8	The Hating Game	1	225
2	1	Audrey Lim	B4	The Goldfinch	1	225
3	2	Mary Kho	B5	A Little Life	1	76
4	3	Issabelle Crocker	B6	Life of Pi	1	70

4.3 Finding the books which has the highest average ratings

```
243 --iii) the books which has the highest average rating
244 SELECT B.Book_ID, B.Name AS Title, AVG(R.Rating) AS HighestAverageRating
245 FROM Rating R INNER JOIN Book b ON R.Book_ID = b.Book_ID GROUP BY B.Book_ID, B.Name
246 ORDER BY HighestAverageRating DESC;
```

Results Messages

	Book_ID	Title	HighestAverageRating
1	B8	The Hating Game	8
2	B6	Life of Pi	6
3	B3	Call Me By Your Name	6
4	B2	In The Name of Love	5
5	B1	The Girl With No Name	4
6	B4	The Goldfinch	3

4.4 Finding the total number of feedback from each member

```
249  --iv) the total number of feedbacks per member
250
251  SELECT M.Member_ID, M.MemberName, COUNT(R.Rating_ID) AS TotalFeedbacks
252  FROM Member M
253  LEFT JOIN Rating R ON M.Member_ID = R.Member_ID
254  GROUP BY M.Member_ID, M.MemberName;
---
```

Results Messages

	Member_ID	MemberName	TotalFeedbacks
1	I001	Audrey Lim	3
2	I002	Mary Kho	1
3	I003	Issabelle Crocker	1
4	I004	Sia Jin Yang	2
5	I005	Mishya Ali	1
6	I006	Mohd Izzat	0
7	I007	Rose Bong	0

4.5 Finding the publisher who published the most books

The below shows two ways in which the data has been found, either in the order of the total book published in descending order (highest on top, least at the bottom) or only the book which has the highest total books published.

```
257 SELECT P.Publisher_ID, P.Name AS Publisher_Name,  
258 COUNT(B.Book_ID) AS Total_Books_Published  
259 FROM Publisher P  
260 INNER JOIN Book B ON P.Publisher_ID = B.Publisher_ID  
261 GROUP BY P.Publisher_ID, P.Name  
262 ORDER BY Total_Books_Published DESC;  
263
```

	Publisher_ID	Publisher_Name	Total_Books_Published
1	P202	JZ Company	2
2	P203	JANT Company	2
3	P204	QRET Company	1
4	P205	SAKU Company	1
5	P206	APD Company	1
6	P201	ABJ Company	1

By using the query as shown below, we can find out the publisher(s) who had **published the most books**.

```
256 --v) the publisher who published the most number of books  
257 SELECT TOP 1 WITH TIES P.Publisher_ID, P.Name AS Publisher_Name,  
258 COUNT(B.Book_ID) AS Total_Books_Published  
259 FROM Publisher P  
260 INNER JOIN Book B ON P.Publisher_ID = B.Publisher_ID  
261 GROUP BY P.Publisher_ID, P.Name  
262 ORDER BY Total_Books_Published DESC;
```

	Publisher_ID	Publisher_Name	Total_Books_Published
1	P202	JZ Company	2
2	P203	JANT Company	2

4.6 Finding the total number of books ordered by the store manager from every publisher

```
265  --vi) total number of books ordered by store manager from each publisher
266  SELECT O.Manager_ID, P.Publisher_ID, P.Name AS PublisherName, COUNT(DISTINCT LD.Book_ID) AS NumBooksOrdered
267  FROM OrderList O
268  JOIN ListDetail LD ON O.OrderList_ID = LD.OrderList_ID
269  JOIN Publisher P ON LD.Publisher_ID = P.Publisher_ID
270  GROUP BY O.Manager_ID, P.Publisher_ID, P.Name;
```

Results Messages

	Manager_ID	Publisher_ID	PublisherName	NumBooksOrdered
1	M01	P201	ABJ Company	1
2	M01	P202	JZ Company	2
3	M02	P203	JANT Company	2
4	M02	P204	QRET Company	1
5	M02	P205	SAKU Company	1
6	M03	P206	APD Company	1

4.7 Finding the genre of book which is most in stock

```
272 --vii) the genres of the book which has the most number of book quantity in stock
273 SELECT B.Genre, SUM(B.Stock) AS TotalStock
274 FROM Book B
275 WHERE B.Stock = (SELECT MAX(Stock) FROM Book)
276 GROUP BY B.Genre;
```

Results Messages

	Genre	TotalStock
1	Mystery	40

4.8 Finding the bestselling book

```
279 --viii) the bestselling book
280 SELECT B.Book_ID,B.Name AS BestSelling_Book,
281 SUM(RD.Quantity) AS TotalSold
282 FROM ReceiptDetail RD INNER JOIN Book B ON RD.Book_ID=B.Book_ID
283 GROUP BY B.Book_ID,B.Name ORDER BY TotalSold desc;
```

Results Messages

	Book_ID	BestSelling_Book	TotalSold
1	B6	Life of Pi	2
2	B7	The Vanishing Half	1
3	B8	The Hating Game	1
4	B1	The Girl With No Name	1
5	B2	In The Name of Love	1
6	B3	Call Me By Your Name	1
7	B4	The Goldfinch	1

4.9 Finding the members who have spent the most on buying books

```
285 --ix) member(s) who spent most on buying books
286 SELECT M.Member_ID, M.MemberName,
287 SUM(MemberReceipt.TotalCost) AS Total_Spent
288 FROM Member M
289 JOIN MemberReceipt MemberReceipt ON m.Member_ID = MemberReceipt.Member_ID
290 GROUP BY M.Member_ID, M.MemberName
291 ORDER BY Total_Spent DESC;
```

Results Messages

	Member_ID	MemberName	Total_Spent
1	I001	Audrey Lim	239
2	I003	Issabelle Crocker	130
3	I004	Sia Jin Yang	90
4	I002	Mary Kho	60

4.10 Finding the members who have not made any orders

```
293 --x) member(s) who had not make any order
294 SELECT M.Member_ID, M.MemberName
295 FROM Member M
296 LEFT JOIN MemberReceipt MR ON M.Member_ID = MR.Member_ID
297 WHERE MR.Receipt_ID IS NULL;
298
```

Results Messages

	Member_ID	MemberName
1	I005	Mishya Ali
2	I006	Mohd Izzat
3	I007	Rose Bong

4.11 List of purchased books that have not been delivered to the members

```
299 --xi) list of purchased books that have not been delivered to members
300 SELECT RD.Book_ID, B.Name AS Book_Name, MR.Member_ID, M.MemberName
301 FROM MemberReceipt MR
302 JOIN ReceiptDetail RD ON MR.Receipt_ID = RD.Receipt_ID
303 JOIN Book B ON RD.Book_ID = B.Book_ID
304 JOIN Member M ON MR.Member_ID = M.Member_ID
305 WHERE MR.DeliveryStatus = 'Not Delivered';
```

Results Messages

	Book_ID	Book_Name	Member_ID	MemberName
1	B4	The Goldfinch	I003	Issabelle Crocker
2	B7	The Vanishing Half	I003	Issabelle Crocker
3	B8	The Hating Game	I004	Sia Jin Yang

4.12 Members who have more than two orders

```
307 --xii) the members who made more than 2 orders
308 SELECT M.Member_ID, M.MemberName,
309 COUNT(DISTINCT MR.Receipt_ID) AS OrderCount
310 FROM MemberReceipt MR
311 JOIN Member M ON MR.Member_ID = M.Member_ID
312 GROUP BY M.Member_ID, M.MemberName
313 HAVING COUNT(DISTINCT MR.Receipt_ID) > 2;
```

Results Messages

	Member_ID	MemberName	OrderCount
1	I001	Audrey Lim	3

Conclusion

In conclusion, structuring tables for essential entities along with incorporating sample data enables the managers to oversee information such as publishers, members, books, orders, and ratings. The inclusion of foreign key commits to data integrity, ensuring reliable relationships between entities. Through various SQL queries, it facilitates diverse functionalities book information management and customer data. Overall, this database design provides a solid foundation for a well-organized and efficient book management system.

Workload Matrix

AICT005-4-1 Database Systems – Workload Matrix – Part 2

Part	Component	Student 1 Name: Shifra Shuaib Ali	Student2 Name: Leong Ciao Shi	Student 3 Name: Viishnu Sree Ganesh	Total
2	a. Database Schema	30	40	30	100 %
2	b. SQL-Data Definition Language (DDL)	25	50	25	100 %
2	c. SQL-Data Manipulation Language (DML)	40	20	40	100 %

References:

- Rouse, M. (2019). What is Data Manipulation Language (DML)? - Definition from Techopedia. Techopedia.com. <https://www.techopedia.com/definition/1179/data-manipulation-language-dml>
- GeeksforGeeks. (2023, October 31). *SQL Join inner left right and full joins*. <https://www.geeksforgeeks.org/sql-join-set-1-inner-left-right-and-full-joins/?ref=lbp>
- *SQL HAVING Clause*. (n.d.). https://www.w3schools.com/sql/sql_having.asp
- GeeksforGeeks. (2023a, April 18). *SQL HAVING Clause with Examples*. <https://www.geeksforgeeks.org/sql-having-clause-with-examples/>