SJBookShop

Saba & Jabin

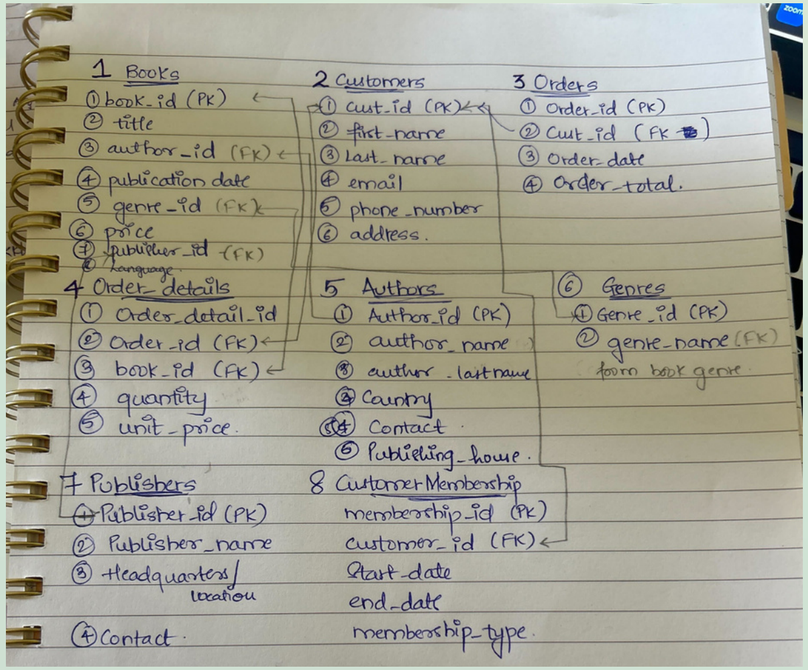
CFG PROJECT
-SABA & JABIN

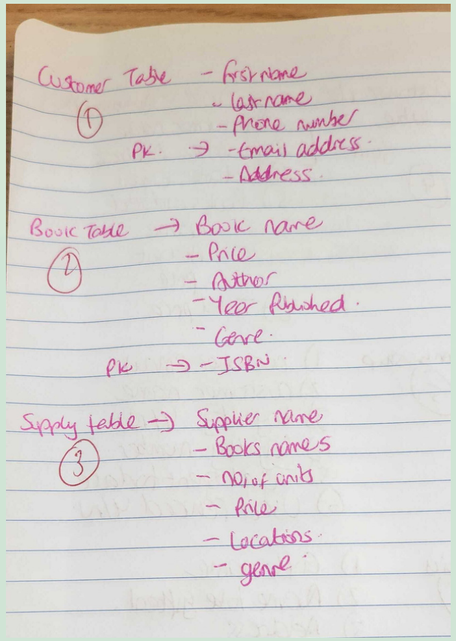
Purpose:

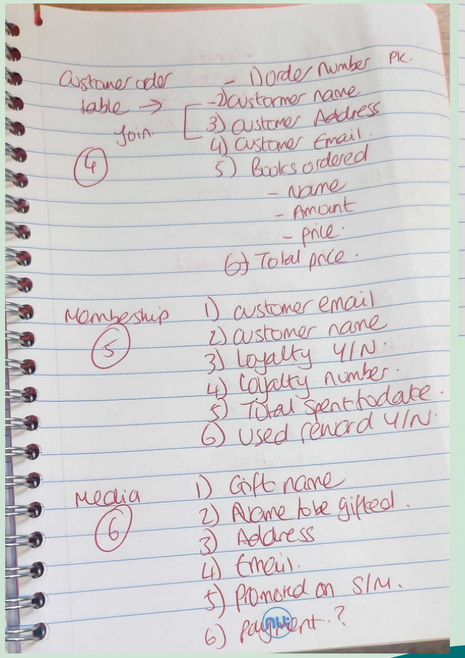
We have developed a MySQL database for a bookshop, serving as a digital organizer. This database will enable us to manage inventory, including books sold and books remaining in stock. It also track details like the genres of books, their authors from the bookshop. Moreover, we have recorded information about the employees who work there. This system allows us to easily monitor and analyze all of this data.

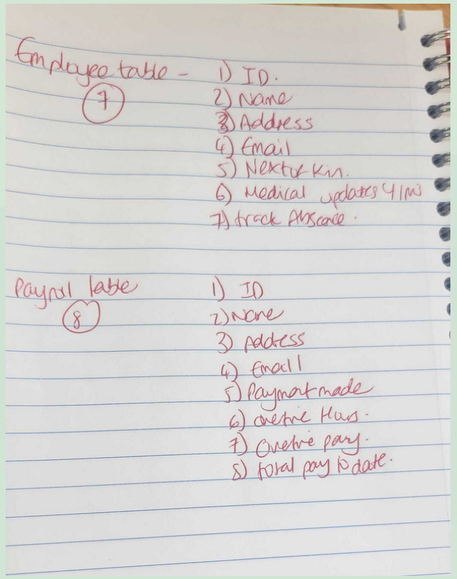
Planning Stage:

When we both agreed with the concept, We Individually came up with the tables required in dataset, and how it will be linked using Primary Key and Foreign Key constraint.





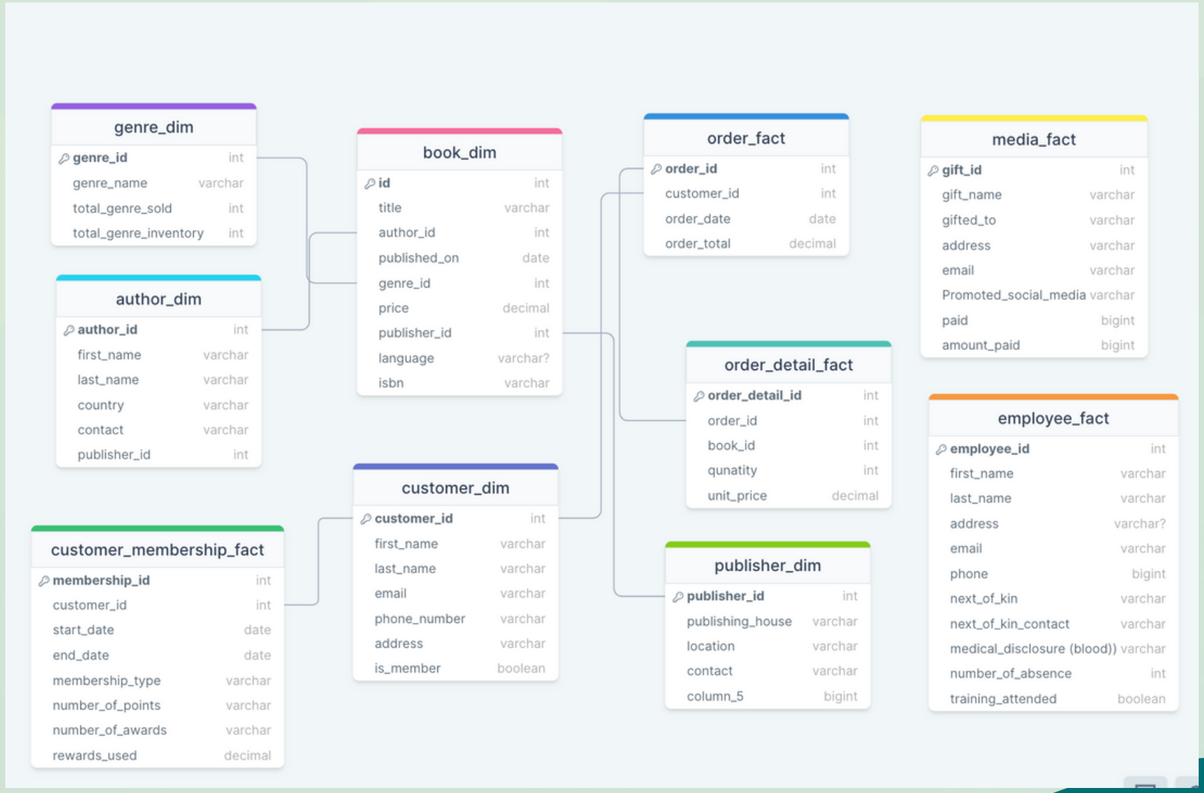




CFG PROJECT
-SABA & JABIN

ER Diagram:

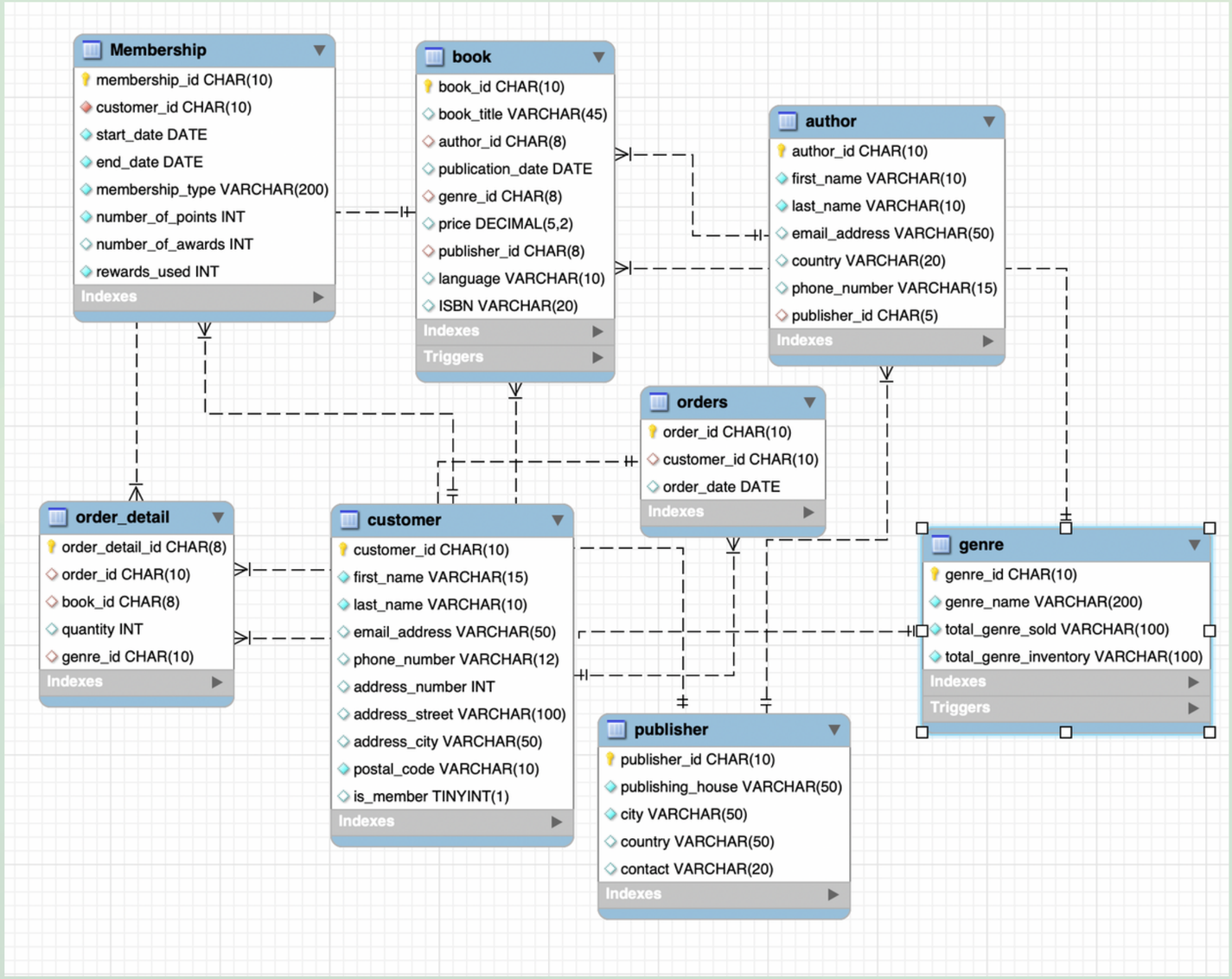
This is the ER diagram at planning stage, to have a look if all tables are connected well.



** Going forward with create table, we have made few changes to final dataset, as per requirement.

CFG PROJECT
-SABA & JABIN

FINAL ER



CFG PROJECT
-SABA & JABIN

Create Table and Insert(o/p)

1.Publisher

```sql
create database SJBookShop;

use SJBookShop;

CREATE TABLE publisher (
publisher_id char(10) NOT NULL PRIMARY KEY,
publishing_house Varchar (50) NOT NULL,
city Varchar (50) NOT NULL,
country Varchar (50),
contact Varchar (20)
);
```

Result Grid | Filter Rows: Search | Edit: | Export/Import:

Publisher_ID	Publishing_House	City	Country	Contact
PID1	Harper _and_Stone_Publishing	New York	USA	7012345678
PID10	Aurora_Press	San Francisco	USA	7912345687
PID11	Silver_Birch_Publications	New York	USA	7019876543
PID12	Pinnacle_Editions	London	UK	7128765432
PID2	Serendipity_Press	London	UK	7123456789
PID3	Bluebird_Books	Paris	France	7234567890
PID4	Sunflower_Publications	Tokyo	Japan	7345678901
PID5	Quill_and _Scroll_Publishing	Sydney	Australia	7456789012
PID6	Moonlit_Publishing	Los Angeles	USA	7567890123
PID7	Crimson_Crest_Books	Berlin	Germany	7678901234
PID8	Starling_House	Rome	Italy	7789012345
PID9	Riverbend_Publishing	Toronto	Canada	7890123456
NULL	NULL	NULL	NULL	NULL

Create Table and Insert(o/p)

2.Author

```sql
create table author(
author_id char(10) NOT NULL PRIMARY KEY,
first_name varchar(10) NOT NULL,
last_name varchar(10) NOT NULL,
email_address varchar(50),
country varchar(20),
phone_number varchar(15),
publisher_id char(5),
CONSTRAINT FK_auth_publisher_id FOREIGN KEY(publisher_id)
REFERENCES publisher(publisher_id)
);
```

Result Grid Filter Rows: Search Edit: Export/Import:

author_id	first_name	last_name	email_address	country	phone_number	publisher_id
AID1	Samuel	Williams	samuel.williams@example.com	United States	+447589214360	PID2
AID10	Zoe	Parker	zoe.parker@example.com	China	+443769854120	PID5
AID11	Ethan	Bennett	ethan.bennett@example.com	Italy	+448149267035	PID7
AID12	Audrey	Lewis	audrey.lewis@example.com	Mexico	+447058397614	PID5
AID13	Owen	Bennett	owen.bennett@example.com	Russia	+447134982705	PID10
AID14	Mia	Stewart	mia.stewart@example.com	Canada	+447396708125	PID12
AID15	Caleb	Wood	caleb.wood@example.com	Brazil	+447265410983	PID11
AID16	Scarlett	Green	scarlett.green@example.com	United States	+447801694735	PID4
AID17	Henry	Foster	henry.foster@example.com	United Kingdom	+447987430562	PID3
AID2	Rachel	Phillips	rachel.phillips@example.com	Canada	+447265437890	PID1
AID3	Oliver	Jenkins	oliver.jenkins@example.com	Brazil	+449812375064	PID3
AID4	Lily	Coleman	lily.coleman@example.com	United Kingdom	+446958132407	PID4
AID5	Daniel	Mitchell	daniel.mitchell@example.com	Australia	+443487652190	PID7
AID6	Stella	Adams	stella.adams@example.com	France	+442390875164	PID9
AID7	Isaac	Carter	isaac.carter@example.com	Germany	+447614098235	PID4
AID8	Grace	Turner	grace.turner@example.com	Japan	+447201986735	PID2
AID9	Benjamin	Hughes	benjamin.hughes@example.c...	India	+447324679810	PID6

Create Table and Insert(o/p)

3.Genre

```sql
CREATE TABLE genre (
genre_id char (10) NOT NULL PRIMARY KEY,
genre_name Varchar (200) NOT NULL,
total_genre_sold Varchar (100) NOT NULL,
total_genre_inventory Varchar (100) NOT NULL
);
```

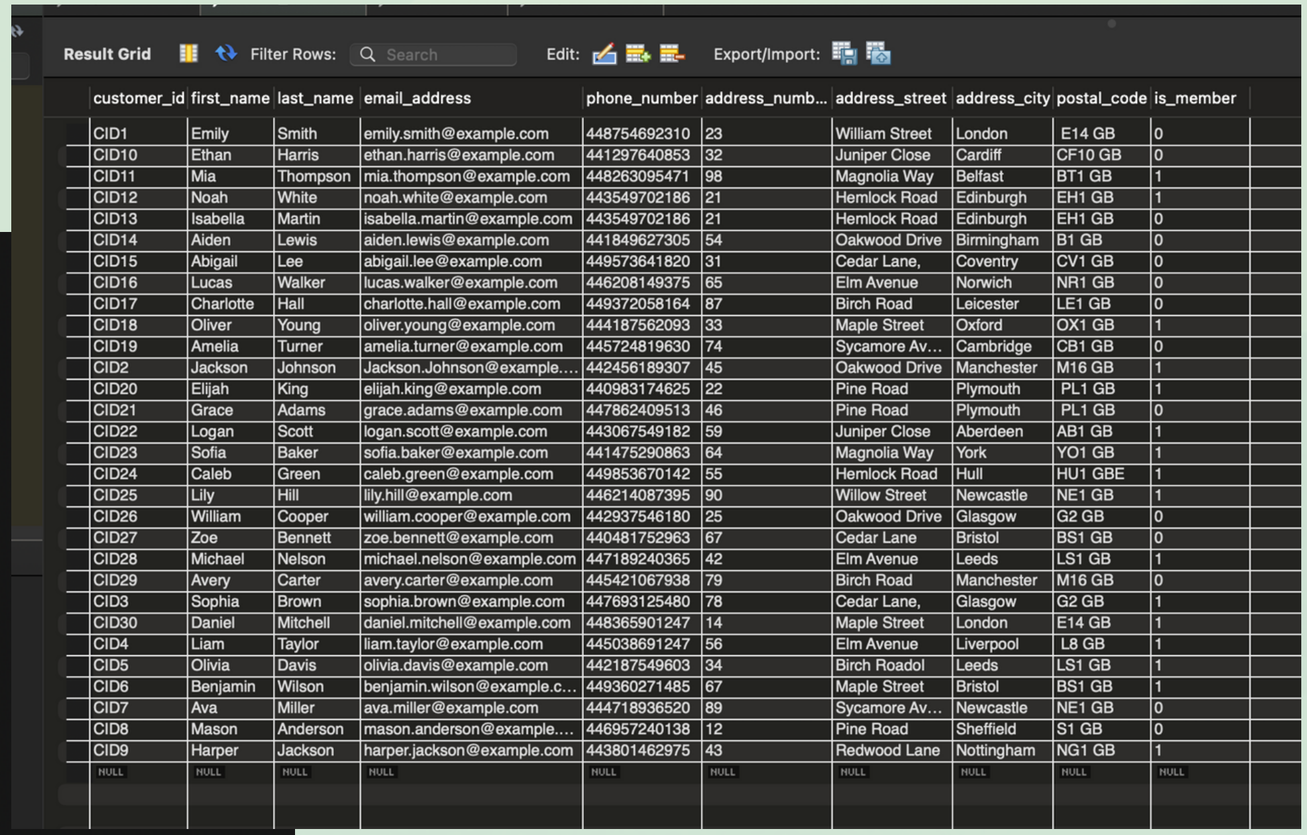
Result Grid Filter Rows: Search Edit:

genre_id	genre_name	total_genre_s...	total_genre_invent...
GID1	Mystery	21	41
GID10	Biography	30	50
GID11	Self-Help	31	51
GID12	Comedy	32	52
GID13	Drama	33	53
GID14	Poetry	34	54
GID15	Crime Fiction	35	55
GID2	Fantasy	22	42
GID3	Science Fiction	23	43
GID4	Romance	24	44
GID5	Thriller	25	45
GID6	Historical Ficti...	26	46
GID7	Horror	27	47
GID8	Adventure	28	48
GID9	Non-Fiction	29	49

Create Table and Insert(o/p)

4.Customer

```
create table customer(
customer_id char(10) NOT NULL PRIMARY KEY,
first_name varchar(15) NOT NULL,
last_name varchar(10) NOT NULL,
email_address varchar(50),
phone_number varchar(12),
address_number INT,
address_street varchar(100),
address_city varchar(50),
postal_code varchar(10) NOT NULL,
is_member boolean
);
```



Result Grid | Filter Rows: Search | Edit: | Export/Import:

customer_id	first_name	last_name	email_address	phone_number	address_numb...	address_street	address_city	postal_code	is_member
CID1	Emily	Smith	emily.smith@example.com	448754692310	23	William Street	London	E14 GB	0
CID10	Ethan	Harris	ethan.harris@example.com	441297640853	32	Juniper Close	Cardiff	CF10 GB	0
CID11	Mia	Thompson	mia.thompson@example.com	448263095471	98	Magnolia Way	Belfast	BT1 GB	1
CID12	Noah	White	noah.white@example.com	443549702186	21	Hemlock Road	Edinburgh	EH1 GB	1
CID13	Isabella	Martin	isabella.martin@example.com	443549702186	21	Hemlock Road	Edinburgh	EH1 GB	0
CID14	Aiden	Lewis	aiden.lewis@example.com	441849627305	54	Oakwood Drive	Birmingham	B1 GB	0
CID15	Abigail	Lee	abigail.lee@example.com	449573641820	31	Cedar Lane,	Coventry	CV1 GB	0
CID16	Lucas	Walker	lucas.walker@example.com	446208149375	65	Elm Avenue	Norwich	NR1 GB	0
CID17	Charlotte	Hall	charlotte.hall@example.com	449372058164	87	Birch Road	Leicester	LE1 GB	0
CID18	Oliver	Young	oliver.young@example.com	444187562093	33	Maple Street	Oxford	OX1 GB	1
CID19	Amelia	Turner	amelia.turner@example.com	445724819630	74	Sycamore Av...	Cambridge	CB1 GB	0
CID2	Jackson	Johnson	Jackson.Johnson@example....	442456189307	45	Oakwood Drive	Manchester	M16 GB	1
CID20	Elijah	King	elijah.king@example.com	440983174625	22	Pine Road	Plymouth	PL1 GB	1
CID21	Grace	Adams	grace.adams@example.com	447862409513	46	Pine Road	Plymouth	PL1 GB	0
CID22	Logan	Scott	logan.scott@example.com	443067549182	59	Juniper Close	Aberdeen	AB1 GB	1
CID23	Sofia	Baker	sofia.baker@example.com	441475290863	64	Magnolia Way	York	YO1 GB	1
CID24	Caleb	Green	caleb.green@example.com	449853670142	55	Hemlock Road	Hull	HU1 GBE	1
CID25	Lily	Hill	lily.hill@example.com	446214087395	90	Willow Street	Newcastle	NE1 GB	1
CID26	William	Cooper	william.cooper@example.com	442937546180	25	Oakwood Drive	Glasgow	G2 GB	0
CID27	Zoe	Bennett	zoe.bennett@example.com	440481752963	67	Cedar Lane	Bristol	BS1 GB	0
CID28	Michael	Nelson	michael.nelson@example.com	447189240365	42	Elm Avenue	Leeds	LS1 GB	1
CID29	Avery	Carter	avery.carter@example.com	445421067938	79	Birch Road	Manchester	M16 GB	0
CID3	Sophia	Brown	sophia.brown@example.com	447693125480	78	Cedar Lane,	Glasgow	G2 GB	1
CID30	Daniel	Mitchell	daniel.mitchell@example.com	448365901247	14	Maple Street	London	E14 GB	1
CID4	Liam	Taylor	liam.taylor@example.com	445038691247	56	Elm Avenue	Liverpool	L8 GB	1
CID5	Olivia	Davis	olivia.davis@example.com	442187549603	34	Birch Roadol	Leeds	LS1 GB	1
CID6	Benjamin	Wilson	benjamin.wilson@example.c...	449360271485	67	Maple Street	Bristol	BS1 GB	1
CID7	Ava	Miller	ava.miller@example.com	444718936520	89	Sycamore Av...	Newcastle	NE1 GB	0
CID8	Mason	Anderson	mason.anderson@example....	446957240138	12	Pine Road	Sheffield	S1 GB	0
CID9	Harper	Jackson	harper.jackson@example.com	443801462975	43	Redwood Lane	Nottingham	NG1 GB	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Create Table and Insert(o/p)

5.Book

```
create table book(
book_id char(10) NOT NULL PRIMARY KEY,
book_title varchar(45),
author_id char(8),
CONSTRAINT FK_author_id FOREIGN KEY (author_id)
REFERENCES author(author_id),
publication_date date,
genre_id char(8),
CONSTRAINT FK_genre_id FOREIGN KEY(genre_id)
REFERENCES genre(genre_id),
price decimal(5,2),
publisher_id char(8),
CONSTRAINT FK_publisher_id FOREIGN KEY(publisher_id)
REFERENCES publisher(publisher_id),
language varchar(10),
ISBN varchar(20)
);
```

Result Grid Filter Rows: Search Edit: Export/Import:

book_id	book_title	author_id	publication_da...	genre_id	price	publisher_id	language	ISBN
BID1	The Crystal Key	AID3	2019-05-12	GID4	12.95	PID1	Spanish	978-1234567890123
BID10	The Lost Chronicles	AID7	2020-08-28	GID8	9.50	PID9	Italian	978-5555555555555
BID11	The Celestial Blade	AID2	2017-04-09	GID3	12.75	PID12	Italian	978-6666666666666
BID12	Riddles of Time	AID9	2019-01-25	GID3	10.99	PID12	Italian	978-7777777777777
BID13	Chronicles of Destiny	AID1	2022-06-14	GID4	13.20	PID12	French	978-8888888888888
BID14	Echoes of Magic	AID1	2019-05-12	GID4	9.25	PID2	English	978-9999999999999
BID15	The Cursed Scroll	AID17	2018-11-15	GID5	13.75	PID2	English	978-1234567890123
BID16	Realm of Secrets	AID8	2016-09-03	GID5	11.20	PID7	French	978-9876543210987
BID17	The Obsidian Amulet	AID8	2021-03-10	GID7	12.50	PID7	Italian	978-1357924680246
BID18	Shadows of the Past	AID5	2020-08-28	GID7	15.80	PID8	English	978-8642098765421
BID19	The Enigma Stone	AID5	2017-04-09	GID15	14.15	PID8	German	978-2345678901234
BID2	Dark Horizon	AID3	2020-08-28	GID5	9.99	PID1	Italian	978-9876543210987
BID20	Echoes from Beyond	AID5	2022-02-22	GID6	10.75	PID6	English	978-8765432109876
BID21	The Forgotten Runes	AID13	2018-11-15	GID6	12.95	PID6	English	978-3141592653589
BID22	Quest for the Lost…	AID13	2019-01-25	GID4	9.99	PID3	English	978-2718281828459
BID23	The Enchanted Whi…	AID12	2016-09-03	GID15	15.50	PID2	French	978-8202818287192
BID24	Secrets of the Astral	AID11	2021-07-18	GID14	8.75	PID6	English	978-7777777777778
BID25	The Eternal Quest	AID9	2020-08-28	GID13	14.25	PID1	English	978-7777777777779
BID3	Forgotten Realms	AID2	2018-11-15	GID4	15.50	PID3	English	978-5555555555555
BID4	Eternal Shadows	AID1	2017-04-09	GID2	8.75	PID3	English	978-8888888888888
BID5	The Silver Serpent	AID6	2022-02-22	GID3	14.25	PID5	German	978-9999999999999
BID6	Whispers of Fate	AID4	2019-05-12	GID6	10.50	PID5	French	978-1111111111111
BID7	The Enchanted Quest	AID8	2018-11-15	GID1	11.99	PID4	French	978-2222222222222
BID8	Mystic Dreams	AID9	2016-09-03	GID8	16.75	PID4	Italian	978-3333333333333
BID9	Secrets Unveiled	AID7	2021-07-18	GID13	13.20	PID9	French	978-4444444444444
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Create Table and Insert(o/p)

6.Orders

```
create table orders(
order_id char(10)  primary key NOT NULL,
customer_id char(10),
CONSTRAINT FK_orders_cust_id FOREIGN KEY(customer_id)
REFERENCES customer(customer_id),
order_date date
);
```

Result Grid Filter Rows: Search

order_id	customer_id	order_date
OID1	CID17	2022-11-24
OID10	CID5	2023-02-14
OID11	CID28	2022-10-15
OID12	CID14	2023-05-31
OID13	CID3	2023-09-22
OID14	CID22	2022-10-05
OID15	CID6	2023-01-29
OID16	CID30	2023-04-02
OID17	CID18	2023-06-24
OID18	CID7	2022-12-03
OID19	CID15	2023-08-07
OID2	CID29	2023-04-15
OID20	CID1	2023-03-17
OID21	CID25	2023-07-17
OID22	CID9	2023-10-02
OID23	CID26	2023-05-12
OID24	CID13	2023-09-03
OID25	CID2	2022-11-08
OID26	CID12	2023-10-10
OID27	CID6	2023-09-25
OID28	CID25	2023-08-14
OID29	CID18	2023-07-03
OID3	CID12	2023-01-07
OID30	CID4	2023-06-22
OID31	CID9	2023-05-11
OID32	CID27	2023-04-05
OID33	CID15	2023-03-20
OID34	CID8	2023-02-08
OID35	CID21	2023-01-17
OID36	CID2	2022-12-04
OID37	CID11	2022-11-19
OID38	CID28	2022-10-27
OID39	CID5	2022-09-12
OID4	CID4	2023-03-28
OID40	CID22	2022-08-03
OID5	CID20	2022-12-11
OID6	CID8	2023-07-05
OID7	CID23	2022-10-27
OID8	CID10	2023-08-19
OID9	CID17	2023-06-10
NULL	NULL	NULL

Create Table and Insert(o/p)

7.Order_details

```
create table order_detail(
order_detail_id char(8) Primary Key NOT NULL,
order_id char(10),
book_id char(8),
quantity int,
genre_id char(10),
FOREIGN KEY (order_id) REFERENCES orders(order_id),
FOREIGN KEY(book_id) REFERENCES book(book_id),
FOREIGN KEY(genre_id) REFERENCES genre(genre_id)
);
```

Result Grid Filter Rows: Search

order_detail...	order_id	book_id	quantity	genre_id
ODID1	OID1	BID1	2	GID4
ODID10	OID5	BID3	3	GID4
ODID11	OID6	BID18	1	GID7
ODID12	OID6	BID4	1	GID2
ODID13	OID8	BID9	2	GID13
ODID14	OID8	BID6	1	GID6
ODID15	OID9	BID11	2	GID3
ODID16	OID7	BID24	1	GID14
ODID17	OID12	BID5	2	GID3
ODID18	OID12	BID21	1	GID6
ODID19	OID9	BID15	1	GID5
ODID2	OID1	BID5	1	GID3
ODID20	OID6	BID3	2	GID4
ODID21	OID5	BID16	1	GID5
ODID22	OID8	BID8	3	GID8
ODID23	OID9	BID14	1	GID4
ODID24	OID30	BID23	1	GID15
ODID25	OID29	BID12	4	GID3
ODID26	OID25	BID10	1	GID8
ODID27	OID27	BID20	2	GID6
ODID28	OID6	BID6	1	GID6
ODID29	OID5	BID17	2	GID7
ODID3	OID1	BID6	1	GID6
ODID30	OID17	BID1	1	GID4
ODID31	OID17	BID22	1	GID4
ODID32	OID11	BID4	2	GID2
ODID33	OID9	BID14	1	GID4
ODID34	OID3	BID7	1	GID1
ODID35	OID1	BID25	2	GID13
ODID36	OID1	BID2	2	GID5
ODID37	OID6	BID19	5	GID15
ODID38	OID3	BID9	1	GID13
ODID39	OID4	BID13	1	GID4
ODID4	OID2	BID19	1	GID15
ODID40	OID8	BID11	3	GID3
ODID5	OID3	BID5	1	GID3
ODID6	OID7	BID12	2	GID3
ODID7	OID3	BID1	1	GID4
ODID8	OID4	BID7	1	GID1
ODID9	OID4	BID8	2	GID8
NULL	NULL	NULL	NULL	NULL

Create Table and Insert(o/p)

8.Membership

```
1
2  CREATE TABLE membership (
3      membership_id char (10) PRIMARY KEY NOT NULL,
4      customer_id char(10) NOT NULL,
5      start_date DATE NOT NULL,
6      end_date DATE NOT NULL,
7      membership_type VARCHAR (200) NOT NULL,
8      number_of_points INT NOT NULL,
9      number_of_awards INT,
0      rewards_used INT NOT NULL
1      );
2
```

```
ALTER TABLE Membership
ADD CONSTRAINT FK_MCustomerID
FOREIGN KEY (customer_ID) REFERENCES customer(customer_id);
```

Result Grid Filter Rows: Search Edit: Export/Import:

membership_id	customer_id	start_date	end_date	membership_type	number_of_poi...	number_of_awards	rewards_used
MID1	CID23	2023-08-12	2031-01-01	Standard Membership	120	5	3
MID10	CID11	2023-05-24	2031-01-01	Senior Citizen Membership	70	6	4
MID11	CID22	2023-12-03	2031-01-01	Senior Citizen Membership	130	10	8
MID12	CID24	2023-02-15	2031-01-01	Standard Membership	80	2	1
MID13	CID28	2023-11-20	2031-01-01	Student Membership	95	10	8
MID14	CID3	2023-09-08	2031-01-01	Student Membership	140	10	10
MID15	CID20	2023-03-31	2031-01-01	Standard Membership	105	1	1
MID16	CID30	2023-05-17	2031-01-01	Standard Membership	190	10	11
MID17	CID27	2023-08-12	2031-01-01	Student Membership	65	16	10
MID18	CID7	2023-05-24	2031-01-01	Senior Citizen Membership	170	10	10
MID2	CID6	2023-05-24	2031-01-01	Standard Membership	75	10	8
MID3	CID25	2023-12-03	2031-01-01	Standard Membership	200	8	4
MID4	CID30	2023-02-15	2031-01-01	Senior Citizen Membership	50	3	2
MID5	CID12	2023-11-20	2031-01-01	Standard Membership	90	10	10
MID6	CID4	2023-09-08	2031-01-01	Student Membership	150	7	6
MID7	CID5	2023-03-31	2031-01-01	Senior Citizen Membership	85	10	10
MID8	CID2	2023-05-17	2031-01-01	Standard Membership	110	4	2
MID9	CID18	2023-08-12	2031-01-01	Standard Membership	160	9	8
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Create Table and Insert(o/p)

7.Membership

```sql

CREATE TABLE membership (
    membership_id char (10) PRIMARY KEY NOT NULL,
    customer_id char(10) NOT NULL,
    start_date DATE NOT NULL,
    end_date DATE NOT NULL,
    membership_type VARCHAR (200) NOT NULL,
    number_of_points INT NOT NULL,
    number_of_awards INT,
    rewards_used INT NOT NULL
    );

```

Result Grid Filter Rows: Search Edit: Export/Import:

membership_id	customer_id	start_date	end_date	membership_type	number_of_poi...	number_of_awards	rewards_used
MID1	CID23	2023-08-12	2031-01-01	Standard Membership	120	5	3
MID10	CID11	2023-05-24	2031-01-01	Senior Citizen Membership	70	6	4
MID11	CID22	2023-12-03	2031-01-01	Senior Citizen Membership	130	10	8
MID12	CID24	2023-02-15	2031-01-01	Standard Membership	80	2	1
MID13	CID28	2023-11-20	2031-01-01	Student Membership	95	10	8
MID14	CID3	2023-09-08	2031-01-01	Student Membership	140	10	10
MID15	CID20	2023-03-31	2031-01-01	Standard Membership	105	1	1
MID16	CID30	2023-05-17	2031-01-01	Standard Membership	190	10	11
MID17	CID27	2023-08-12	2031-01-01	Student Membership	65	16	10
MID18	CID7	2023-05-24	2031-01-01	Senior Citizen Membership	170	10	10
MID2	CID6	2023-05-24	2031-01-01	Standard Membership	75	10	8
MID3	CID25	2023-12-03	2031-01-01	Standard Membership	200	8	4
MID4	CID30	2023-02-15	2031-01-01	Senior Citizen Membership	50	3	2
MID5	CID12	2023-11-20	2031-01-01	Standard Membership	90	10	10
MID6	CID4	2023-09-08	2031-01-01	Student Membership	150	7	6
MID7	CID5	2023-03-31	2031-01-01	Senior Citizen Membership	85	10	10
MID8	CID2	2023-05-17	2031-01-01	Standard Membership	110	4	2
MID9	CID18	2023-08-12	2031-01-01	Standard Membership	160	9	8
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

VIEW

Used joins to combine column from multiple tables.

A VIEW THAT COMBINES THE **BOOK**, **ORDER_DETAIL**, AND **GENRE** TABLES TO SHOW THE TITLES OF **BOOKS**, THEIR **GENRES**, AND THE **NUMBER OF UNITS SOLD** FOR EACH BOOK.

```sql
CREATE VIEW UnitOfSales AS
SELECT
    b.book_title,
    g.genre_name,
    count(o.book_id) as Unit_Sold
FROM
    book b
JOIN
    order_detail o ON b.book_id = o.book_id
JOIN
    genre g ON b.genre_id = g.genre_id
Group by
    book_title,
    genre_name;

Select * from UnitOfSales;
```

O/p

book_title	genre_name	Unit_Sold
The Crystal Key	Romance	3
The Lost Chronicles	Adventure	1
The Celestial Blade	Science Fiction	2
Riddles of Time	Science Fiction	2
Chronicles of Destiny	Romance	1
Echoes of Magic	Romance	2
The Cursed Scroll	Thriller	1
Realm of Secrets	Thriller	1
The Obsidian Amulet	Horror	1
Shadows of the Past	Horror	1

CFG PROJECT
-SABA & JABIN

VIEW WITH MULTIPLE TABLES

Used joins(5) to combine column from multiple tables.

BELOW IS A VIEW NAMED **BOOKSALES** THAT COMBINES DATA FROM THE **BOOK**, **CUSTOMER**, AND **PUBLISHER** TABLES. THE VIEW INCLUDES INFORMATION ABOUT **BOOK SALES**, INCLUDING THE **BOOK TITLE, CUSTOMER NAMES**, AND THE **PUBLISHER NAME** FOR **EACH BOOK SOLD.**

```sql
CREATE VIEW BOOKSALES AS
SELECT
    b.book_title,
    CONCAT(c.first_name, ' ', c.last_name) AS customer_name,
    p.publishing_house AS publisher_name
FROM
    order_detail od
JOIN
    book b ON b.book_id = od.book_id
JOIN
    publisher p ON p.publisher_id = b.publisher_id
JOIN
    orders o ON od.order_id = o.order_id
JOIN
    customer c ON o.customer_id = c.customer_id;

Select * from BOOKSALES;
```

	book_title	customer_name	publisher_name
	The Crystal Key	Oliver Young	Harper _and_Stone_Publishing
	The Crystal Key	Noah White	Harper _and_Stone_Publishing
	The Lost Chronicles	Jackson Johnson	Riverbend_Publishing
	The Celestial Blade	Charlotte Hall	Pinnacle_Editions
	The Celestial Blade	Ethan Harris	Pinnacle_Editions
	Riddles of Time	Oliver Young	Pinnacle_Editions
	Riddles of Time	Sofia Baker	Pinnacle_Editions
	Chronicles of Destiny	Liam Taylor	Pinnacle_Editions
	Echoes of Magic	Charlotte Hall	Serendipity_Press
	Echoes of Magic	Charlotte Hall	Serendipity_Press
	The Cursed Scroll	Charlotte Hall	Serendipity_Press
	Realm of Secrets	Elijah King	Crimson_Crest_Books
	The Obsidian Amulet	Elijah King	Crimson_Crest_Books

TIMMERMAN INDUSTRIES

SUBQUERY

A SUBQUERY TO **EXTRACT** THE **TOP 5 CUSTOMERS** WITH THE HIGHEST TOTAL ORDER VALUES.

```sql
SELECT c.customer_id,
       c.first_name,
       c.last_name,
       customer_orders.total_order_value
FROM customer c
JOIN (
    SELECT o.customer_id,
    SUM(b.price) AS total_order_value
    FROM order_detail od
    JOIN
        book b ON od.book_id = b.book_id
    JOIN
        orders o ON od.order_id = o.order_id
    GROUP BY
        o.customer_id
    ORDER BY
        total_order_value DESC
    LIMIT 5
) AS customer_orders ON c.customer_id = customer_orders.customer_id;
```

O/p

customer_id	first_name	last_name	total_order_value
CID17	Charlotte	Hall	106.94
CID8	Mason	Anderson	64.70
CID4	Liam	Taylor	57.44
CID10	Ethan	Harris	53.20
CID12	Noah	White	52.39

CFG PROJECT
-SABA & JABIN

Stored Function.

A STORED FUNCTION THAT CALCULATES THE **TOTAL ORDER VALUE** BASED ON **ORDER DETAILS**.

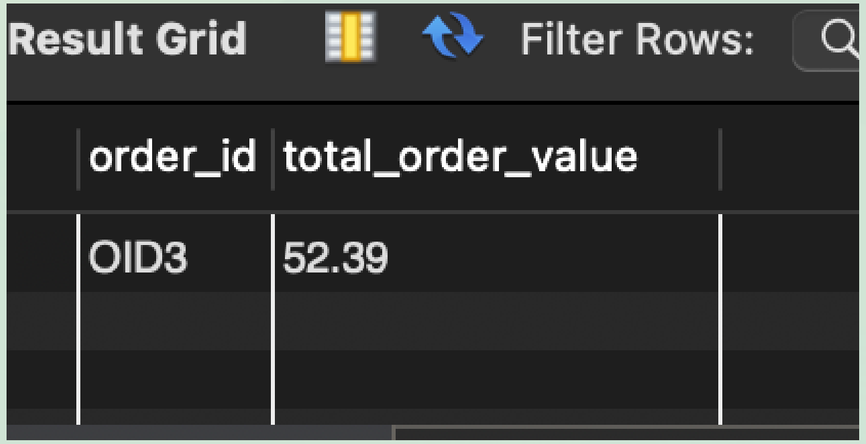
```
DELIMITER //
CREATE FUNCTION TotalOrderValue(order_id char(10))
RETURNS DECIMAL(10, 2)
READS SQL DATA
BEGIN
    DECLARE total_value DECIMAL(10, 2);

    SELECT SUM(b.price) INTO total_value
    FROM book b
    JOIN order_detail od ON od.book_id = b.book_id
    WHERE od.order_id = order_id;

    RETURN total_value;
END;
//
DELIMITER ;
```

CALCULATE THE TOTAL ORDER VALUE FOR A SPECIFIC ORDER USING THE STORED FUNCTION.

```
SELECT
    o.order_id,
    TotalOrderValue(o.order_id) AS total_order_value
FROM orders AS o
WHERE o.order_id = 'OID3';
```



CFG PROJECT
-SABA & JABIN

TRIGGER

A TRIGGER THAT LOGS EACH TIME A **NEW BOOK** IS ADDED TO THE **BOOK TABLE**.

```sql
-- inserted records from the genre table.
CREATE TABLE genre_history (
    genre_id CHAR(10),
    genre_name VARCHAR(255),
    total_genre_sold INT,
    total_genre_inventory INT,
    insertion_timestamp TIMESTAMP
);
```

```sql
DELIMITER //
CREATE TRIGGER NewGenre
AFTER INSERT ON genre
FOR EACH ROW
BEGIN
    INSERT INTO genre_history (genre_id, genre_name, total_genre_sold, total_genre_inventory, insertion_timestamp)
    VALUES (NEW.genre_id, NEW.genre_name, NEW.total_genre_sold, NEW.total_genre_inventory, NOW());
END;
//
DELIMITER ;
```

```sql
--  Insert data into the genre table.
INSERT INTO genre (genre_id, genre_name, total_genre_sold, total_genre_inventory)
VALUES ('GID17', 'Science', 20, 30);
```

genre_id	genre_name	total_genre_s...	total_genre_invent...
GID13	Drama	33	53
GID14	Poetry	34	54
GID15	Crime Fiction	35	55
GID16	Data Help	30	50
GID17	Science	20	30

genre_history_table

genre_id	genre_name	total_genre_s...	total_genre_invent...	insertion_timestamp
GID16	Data Help	30	50	2023-10-22 21:07:53
GID17	Science	20	30	2023-10-23 16:30:48

genre_history 21

CFG PROJECT
-SABA & JABIN

Stored Procedure

A STORED PROCEDURE THAT **INSERTS A NEW AUTHOR**,
AND CAN YOU DEMONSTRATE HOW IT RUNS.

author_id	first_name	last_name	email_address	country	phone_number	publisher_id
AID10	Zoe	Parker	zoe.parker@example.com	China	+443769854120	PID5
AID11	Ethan	Bennett	ethan.bennett@example.com	Italy	+448149267035	PID7
AID12	Audrey	Lewis	audrey.lewis@example.com	Mexico	+447058397614	PID5
AID13	Owen	Bennett	owen.bennett@example.com	Russia	+447134982705	PID10
AID14	Mia	Stewart	mia.stewart@example.com	Canada	+447396708125	PID12
AID15	Caleb	Wood	caleb.wood@example.com	Brazil	+447265410983	PID11
AID16	Scarlett	Green	scarlett.green@example.com	United States	+447801694735	PID4
AID17	Henry	Foster	henry.foster@example.com	United Kingdom	+447987430562	PID3
AID18	Saba	Shaikh	something@example.com	london	+44 78798767645	PID5

```
DELIMITER $$
Create procedure InsertNewAuthor(
IN p_author_id char(10),
IN p_first_name varchar(10),
IN p_last_name varchar(10),
IN p_email_address varchar(50),
IN p_country varchar(20),
IN p_phone_number varchar(15),
IN p_publisher_id char(5)
)
BEGIN
INSERT INTO author(author_id, first_name, last_name, email_address, country, phone_number, publisher_id)
Values (p_author_id,p_first_name,p_last_name,p_email_address, p_country, p_phone_number,p_publisher_id);
END $$
DELIMITER ;

CALL InsertNewAuthor('AID18','Saba','Shaikh','something@example.com', 'london','+44 78798767645','PID5');
```

Event

UPDATE MEMBERSHIP TABLE EVERY 30 MINUITS, TO KEEP RECORD OF **POINTS EARNED** BY **EACH MEMBERSHIP CATEGORY.**

```
SET GLOBAL event_scheduler = ON;

Create table membership_points_update(
ID INT AUTO_INCREMENT NOT NULL PRIMARY KEY,
membership_type varchar(50),
points INT,
modifiedTimeStamp TIME
);

DELIMITER $$
CREATE EVENT UpdateMembershipPoints
ON SCHEDULE EVERY 30 Minute
STARTS CURRENT_TIMESTAMP
ON COMPLETION PRESERVE
DO
BEGIN
    truncate table membership_points_update;
    Insert into membership_points_update(membership_type, points, modifiedTimeStamp)
    Select membership_type,
            sum(number_of_points) as points,
            NOW()
            From membership
            group by membership_type;

END;
$$
DELIMITER ;

select * from membership_points_update;
```

ID	membership_type	points	modifiedTimeSta...
1	Standard Membership	1130	19:11:59
2	Senior Citizen Membership	505	19:11:59
3	Student Membership	450	19:11:59
NULL	NULL	NULL	NULL

CFG PROJECT
-SABA & JABIN

Group By and Having:

A QUERY THAT USES THE GROUP BY AND HAVING CLAUSES TO FIND GENRES WITH AN **AVERAGE BOOK PRICE** GREATER THAN **9** POUNDS AND A TOTAL OF **AT LEAST 5 BOOKS SOLD**.

genre_name	average_price	total_sold
Romance	12.393333	13
Drama	13.550000	5
Historical Fiction	11.040000	6
Science Fiction	12.890000	15
Adventure	14.333333	6
Crime Fiction	14.600000	7

```sql
SELECT g.genre_name,
       AVG(b.price) AS average_price,
       SUM(od.quantity) AS total_sold
FROM book b
JOIN genre g ON b.genre_id = g.genre_id
JOIN order_detail od ON od.book_id = b.book_id
GROUP BY g.genre_name
HAVING AVG(b.price) > 9 AND SUM(od.quantity) >= 5;
```

Thank you!

CFG PROJECT
-SABA & JABIN