

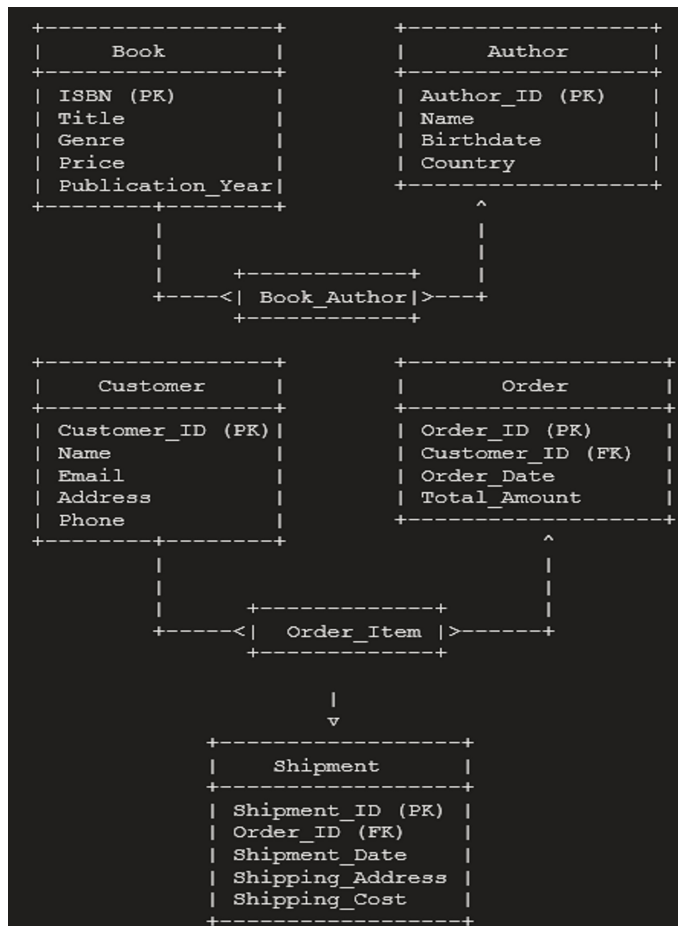
## Day7Assingments

12 May 2024 12:36 PM

**Assignment 1:** Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

### Business Scenario: Online Bookstore

Let's design an ER diagram for an online bookstore system. The system will manage information about books, authors, customers, orders, and shipments. Our goal is to create a well-structured database design that reflects proper normalization.



**Assignment 2:** Design a database schema for a library system, including tables, fields, and constraints like NOT NULL, UNIQUE, and CHECK. Include primary and foreign keys to establish relationships between tables.

create database Librarymgmt;  
use Librarymgmt;

create table Books(  
book\_id int primary key,  
title varchar(225) not null,  
author varchar(225) not null,  
published\_date date,  
isbn varchar(20) unique,

```
avl_copies int,  
total_copies int  
);
```

```
create table Members(  
member_id int primary key,  
name varchar(255) not null,  
email varchar(225) unique,  
phone int  
);
```

```
create table Borrowings(  
borrow_id int primary key,  
book_id int,  
member_id int,  
borrow_date date,  
due_date date,  
foreign key(book_id) references Books(book_id),  
foreign key(member_id) references Members(member_id)  
);
```

```
1 • create database Librarymgmt;  
2 • use Librarymgmt;  
3 • create table Books(  
4     book_id int primary key,  
5     title varchar(225) not null,  
6     author varchar(225) not null,  
7     published_date date,  
8     isbn varchar(20) unique,  
9     avl_copies int,  
10    total_copies int  
11 );  
12 • create table Members(  
13     member_id int primary key,  
14     name varchar(255) not null,  
15     email varchar(225) unique,  
16     phone int  
17 );  
  
18 • create table Borrowings(  
19     borrow_id int primary key,  
20     book_id int,  
21     member_id int,  
22     borrow_date date,  
23     due_date date,  
24     foreign key(book_id) references Books(book_id),  
25     foreign key(member_id) references Members(member_id)  
26 );
```

**Assignment 3:** Explain the ACID properties of a transaction in your own words. Write SQL statements to simulate a transaction that includes locking and demonstrate different isolation levels to show concurrency control.

ACID stands for Atomicity, Consistency, Isolation, and Durability.

Atomicity: This property ensures that a transaction is treated as a single unit of work. Either all of its operations are successfully completed, or none of them are. If any part of the transaction fails, the entire transaction is rolled back to its initial state.

Consistency: After a transaction is successfully completed, the database remains in a consistent state. This means that the data satisfies all integrity constraints, such as foreign key relationships, uniqueness constraints, etc.

Isolation: This property ensures that the intermediate state of a transaction is not visible to other transactions running concurrently. Each transaction appears to be the only transaction executing on the database, even though multiple transactions may be executing concurrently.

Durability: Once a transaction is committed, its changes are permanent and will not be lost, even in the event of a system failure. The changes are stored in non-volatile memory (disk), ensuring data persistence.

We start a transaction using `BEGIN TRANSACTION;` to ensure atomicity.

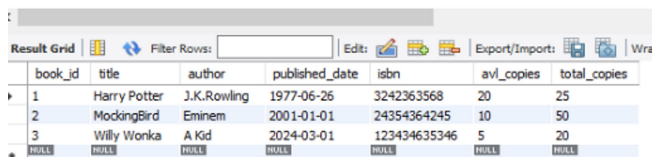
We set a specific isolation level (`READ UNCOMMITTED`, `READ COMMITTED`, `REPEATABLE READ`, `SERIALIZABLE`) to control how transactions interact with each other concurrently.

We update the `available_copies` of the book and insert a new transaction record atomically within the transaction.

Finally, we commit the transaction using `COMMIT;` to make the changes permanent in the database.

The `SELECT` statement at the end retrieves and displays the updated available copies of the book.

```
45 • update Books set avl_copies = avl_copies-1
46   where title='MockingBird';
47
```



The screenshot shows a database management tool interface. At the top, there's a 'Result Grid' tab. Below it, a table with 7 columns: book\_id, title, author, published\_date, isbn, avl\_copies, and total\_copies. The table contains three rows of data. The first row is for 'Harry Potter' by J.K. Rowling, published on 1977-06-26, with ISBN 3242363568, 20 available copies, and 25 total copies. The second row is for 'MockingBird' by Eminem, published on 2001-01-01, with ISBN 24354364245, 10 available copies, and 50 total copies. The third row is for 'Willy Wonka' by A Kid, published on 2024-03-01, with ISBN 123434635346, 5 available copies, and 20 total copies. The table is displayed with alternating light blue and white rows.

book_id	title	author	published_date	isbn	avl_copies	total_copies
1	Harry Potter	J.K.Rowling	1977-06-26	3242363568	20	25
2	MockingBird	Eminem	2001-01-01	24354364245	10	50
3	Willy Wonka	A Kid	2024-03-01	123434635346	5	20

**Assignment 4:** Write SQL statements to `CREATE` a new database and tables that reflect the library schema you designed earlier. Use `ALTER` statements to modify the table structures and `DROP` statements to remove a redundant table.

```

1 • create database LibraryDB;
2 • use LibraryDB;
3
4 • create table Books(
5     book_id int primary key,
6     titel varchar(225) not null
7 );
8 • create table Members(
9     member_id int primary key,
10    name varchar(225) not null
11 );
12 • create table Borrowings(
13     b_id int primary key,
14     book_id int,
15     member_id int
16 );

17
18 • alter table Members
19     add address varchar(225);
20
21 • alter table Books
22     modify column book_id varchar(225);
23
24 • drop table Books;
25

```

**Assignment 5:** Demonstrate the creation of an index on a table and discuss how it improves query performance. Use a DROP INDEX statement to remove the index and analyze the impact on query execution.

```

23
24 • select * from Books;
25

```

	book_id	title	author	published_date	isbn	avl_copies	total_copies
▶	1	Harry Potter	J.K.Rowling	1977-06-26	3242363568	20	25
	2	MockingBird	Eminem	2001-01-01	24354364245	10	50
	3	Willy Wonka	A Kid	2024-03-01	123434635346	5	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

```

• create index idx_id on Books(avl_copies);

• explain analyze
  select * from Books where avl_copies = 5;

30
31 • drop index idx_id on Books;
32

```

```

33 • explain analyze
34 select * from Books where avl_copies = 5;

```

11 17:33:09 drop index idx\_id on Books 0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0 0.031 sec  
 12 17:33:41 explain analyze select \* from Books where avl\_copies = 5 1 row(s) returned 0.000 sec / 0.000 sec

Result Grid		Filter Rows:		Edit:	Export/Import:		Wrap Cell Content:
	book_id	title	author	published_date	isbn	avl_copies	total_copies
▶	3	Willy Wonka	A Kid	2024-03-01	123434635346	5	20
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

**Assignment 7:** Prepare a series of SQL statements to INSERT new records into the library tables, UPDATE existing records with new information, and DELETE records based on specific criteria. Include BULK INSERT operations to load data from an external source.

```

1 • create database Librarymgmt;
2 • use Librarymgmt;
3 • create table Books(
4     book_id int primary key,
5     title varchar(225) not null,
6     author varchar(225) not null,
7     published_date date,
8     isbn varchar(20) unique,
9     avl_copies int,
10    total_copies int
11 );
12 • create table Members(
13     member_id int primary key,
14     name varchar(255) not null,
15     email varchar(225) unique,
16     phone int
17 );
18 • create table Members(
19     member_id int primary key,
20     name varchar(255) not null,
21     email varchar(225) unique,
22     phone int
23 );
24 • create table Borrowings(
25     borrow_id int primary key,
26     book_id int,
27     member_id int,
28     borrow_date date,
29     due_date date,
30     foreign key(book_id) references Books(book_id),
31     foreign key(member_id) references Members(member_id)
32 );

```

```

28 • insert into Books(book_id,title,author, published_date, isbn, avl_copies, total_copies)
29 values (1,'Harry Potter','J.K.Rowling','1977-06-26','3242363568',20,25),
30 (2,'MockingBird','Eminem','2001-01-01','24354364245',11,50),
31 (3,'Willy Wonka','A Kid','2024-03-01','123434635346',5,20);
32
33 • insert into Members(member_id,name,email,phone)
34 values (1,'Siva','abcd123@db.in',123456789),
35 (2,'mani','abcd1234@db.in',123456780),
36 (3,'raja','abcd12345@db.in',123456781);
37 • insert into Borrowings(borrow_id,book_id,member_id,borrow_date,due_date)
38 values (1,2,1,'2024-01-01','2024-02-23'),
39 (2,3,1,'2024-02-01','2024-02-23'),
40 (3,1,3,'2024-03-01','2024-03-23');

```

Inserting new data into all the tables:

```

1 • insert into Members(member_id,name,email,phone)
2 values (4,'naga','abcd0123@db.in',123456710),
3 (5,'venkat','abcd10234@db.in',123456730);
4
5 • insert into Borrowings(borrow_id,book_id,member_id,borrow_date,due_date)
6 values (4,3,1,'2024-01-01','2024-02-23'),
7 (5,2,2,'2024-02-01','2024-02-23');
8
9 • insert into Books(book_id,title,author, published_date, isbn, avl_copies, total_copies)
10 values (4,'Businessr','Eminem','1977-06-26','3142363568',20,25),
11 (5,'Overdose','Natori','2023-01-01','24364364245',12,50);
12

```

Updating records:

```

14
15 • update Books set avl_copies = avl_copies+3
16 where title='Overdose';
17
18 • update Members set member_id = member_id+1
19 where name='venkat';
20
21 • update Borrowings set due_date = '2024-12-01'
22 where member_id=1;

```

deleting records:

```

26 • delete from Books
27 where author = 'J.K.Rowling';
28
29 • delete from members
30 where name= 'naga';
31
32 • delete from Borrowings
33 where borrow_id=5;

```

Bulk insert:

```
LOAD DATA INFILE '/path/to/books.csv'  
INTO TABLE Books  
FIELDS TERMINATED BY ','  
LINES TERMINATED BY '\n'  
IGNORE 1 ROWS  
(book_id, title, author, published_date, genre);
```