

Assignment -7 (SQL)

Prakash Manikanta Irrinki
prakashnaidu9494@gmail.com

Assignment 1:

Analyse 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.

Entities:

1. Patient.
2. Doctor.
3. Test.

Relationships:

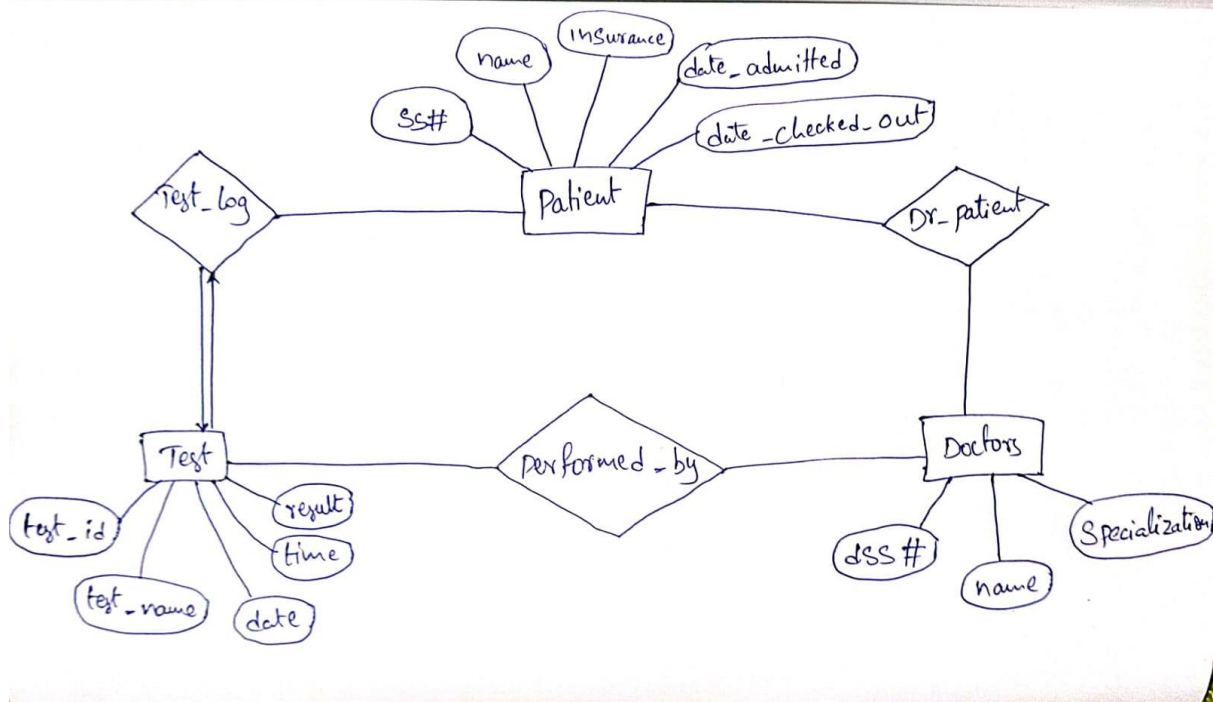
1. Patient visits Doctor.
2. Doctor treats Patient.
3. Doctor works in Department.
4. Patient has Appointment.
5. Patient undergoes Test.

Attributes:

1. Patient: SS#, name, insurance, date_admitted, date_checked_out.
2. Doctor: DSS, Name, Specialization .
3. Test: test_ID, test_name, Date, time, result.

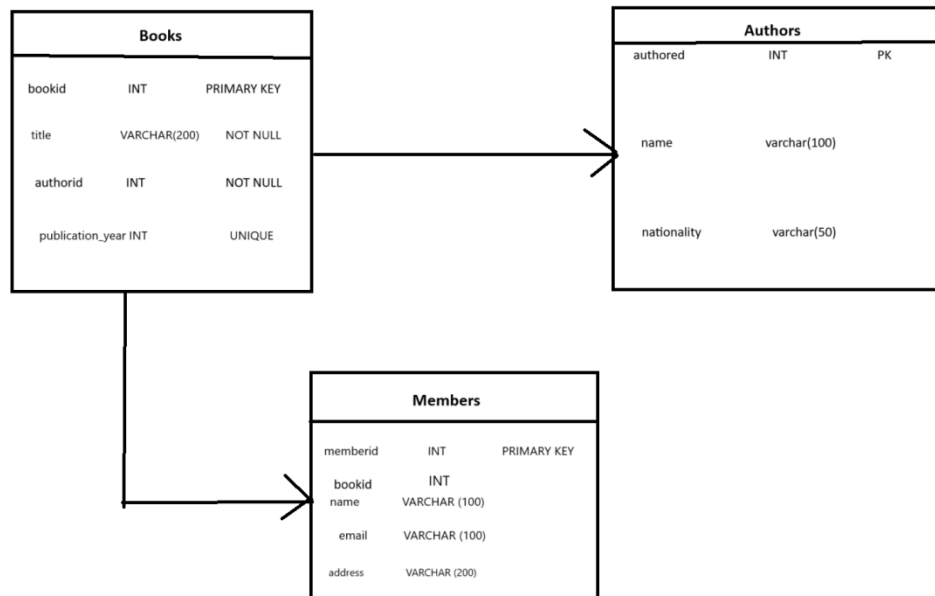
Cardinality:

1. One patient can have multiple appointments.
2. One doctor can have multiple appointments, treat multiple patients, and perform multiple tests.
3. One nurse can assist multiple doctors.
4. One department can have multiple doctors.



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.



SQL Queries:

```
CREATE TABLE Authors ( authorid INT PRIMARY KEY, name VARCHAR(100) NOT NULL,
nationality VARCHAR(50) NOT NULL );
```

Desc authors;

```

+-----+-----+-----+-----+-----+
| Field   | Type   | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| authorid | int    | NO   | PRI | NULL    |      |
| name     | varchar(50) | NO   |     | NULL    |      |
| nationality | varchar(50) | YES  |     | NULL    |      |
+-----+-----+-----+-----+-----+
  
```

```
CREATE TABLE Books (
bookid INT PRIMARY KEY,
title VARCHAR(200) NOT NULL,
authorid INT NOT NULL,
publication_year INT UNIQUE,
FOREIGN KEY (authorid) REFERENCES Authors(authorid) );
```

Desc books;

Field	Type	Null	Key	Default	Extra
bookid	int	NO	PRI	NULL	
title	varchar(50)	NO		NULL	
authorid	int	NO	MUL	NULL	
publish_year	int	YES	UNIQUE	NULL	

CREATE TABLE Members (memberid INT PRIMARY KEY, name VARCHAR (100) NOT NULL, email VARCHAR (100) NOT NULL, address VARCHAR (200) NOT NULL);

Desc members;

Field	Type	Null	Key	Default	Extra
memberid	int	NO	PRI	NULL	
name	varchar(50)	NO		NULL	
email	varchar(50)	NO		NULL	
address	varchar(50)	NO		NULL	

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 Properties:

- ➔ Atomicity: This property ensures that either all the operations within a transaction are successfully completed, or none of them are. If any part of the transaction fails, the entire transaction is rolled back to its original state.
- ➔ Consistency: Consistency ensures that the database remains in a valid state before and after the transaction. All integrity constraints, such as foreign key constraints or uniqueness constraints must be satisfied.
- ➔ Isolation: Isolation ensures that the concurrent execution of transactions results in a state that could be obtained if transactions were executed serially. Isolation levels define the degree to which the operations within one transaction are isolated from the operations of other concurrent transactions.
- ➔ Durability: Durability guarantees that once a transaction has been committed, the changes made by it will persist even in the event of system failure.

```
CREATE TABLE bank_accounts ( account_id INT PRIMARY KEY, balance DECIMAL(10, 2) );
```

```
INSERT INTO bank_accounts (account_id, balance) VALUES (1, 1000.00), (2, 2000.00);
```

```
BEGIN TRANSACTION;
```

```
UPDATE bank_accounts SET balance = balance - 500.00 WHERE account_id = 1;
```

```
UPDATE bank_accounts SET balance = balance + 500.00 WHERE account_id = 2;
```

```
Commit;
```

```
SET TRANSACTION ISOLATION LEVEL READ UNCOMMITTED;
```

```
SELECT balance FROM bank_accounts WHERE account_id = 1;
```

```
SET TRANSACTION ISOLATION LEVEL READ COMMITTED;
```

```
SELECT balance FROM bank_accounts WHERE account_id = 1;
```

```
SET TRANSACTION ISOLATION LEVEL REPEATABLE READ;
```

```
SELECT balance FROM bank_accounts WHERE account_id = 1;
```

```
SET TRANSACTION ISOLATION LEVEL SERIALIZABLE;
```

```
SELECT balance FROM bank_accounts WHERE account_id = 1;
```

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.

```
create database assignments;
```

```
use assignments;
```

```
create table bookinfo(book_id int primary key, title varchar (50) not null, author varchar (50),  
publication_year year, ISBN int unique); alter table books add column copies_available int;  
alter table books add column author_id int; alter table books drop column author;
```

```
desc book;
```

Field	Type	Null	Key	Default	Extra
book_id	int	NO	PRI	NULL	
title	varchar(50)	NO		NULL	
publication_year	year	YES		NULL	
ISBN	int	YES	UNI	NULL	

```
| copies_available | int      | YES |  | NULL |  |
| author_id       | int      | YES |  | NULL |  |
+-----+-----+-----+-----+-----+
```

```
create table authorinfo(author_id int primary key, author_name varchar (50));
```

```
desc authorinfo;
```

```
+-----+-----+-----+-----+-----+
| Field   | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| author_id | int      | NO   | PRI | NULL    |       |
| author_name | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

```
create table borrowinginfo(borrowing_id int primary key, book_id int, member_id int,
borrow_date date, return_date date, status varchar (50));
```

```
Desc borrowinginfo;
```

```
+-----+-----+-----+-----+-----+
| Field   | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| borrowing_id | int      | NO   | PRI | NULL    |       |
| book_id     | int      | YES  |     | NULL    |       |
| member_id   | int      | YES  |     | NULL    |       |
| borrow_date | date     | YES  |     | NULL    |       |
| return_date | date     | YES  |     | NULL    |       |
| status      | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

```
create table members (member_id int primary key, member_name varchar (50), email
varchar (100), phone_number varchar(20),address varchar(50));
```

```
desc members;
```

```
+-----+-----+-----+-----+-----+
| Field   | Type     | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+
| memberid | int      | NO   | PRI | NULL    |       |
| name     | varchar(50) | NO   |     | NULL    |       |
| email    | varchar(50) | NO   |     | NULL    |       |
| address  | varchar(50) | NO   |     | NULL    |       |
| phone_number | int      | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+
```

```
create table books2(book_title varchar (20),book_price float);
```

```
drop table books2;
```

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 analyse the impact on query execution.

```
create index index_empname on employee (eid, ename);
```

```
explain select * from employee where ename='max';
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employee	NULL	ref	emo_index	emo_index	83	const	1	100.00	NULL

```
explain select * from employee where eid > 105;
```

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	employee	NULL	range	PRIMARY	PRIMARY	4	NULL	4	100.00	Using where

```
drop index emp_index on employee;
```

```
explain select * from employee where eid > 105;
```

Assignment 6:

Create a new database user with specific privileges using the CREATE USER and GRANT commands. Then, write a script to REVOKE certain privileges and DROP the user.

```
create user prakashs identified by 'system@123';
```

```
select user ();
```

user()
root@localhost

```
select user from mysql.user;
```

user
assignments
mysql.infoschema
mysql.session
mysql.sys
mysys
root

grant all on practicedb.employee to prakashs;
revoke insert on practicedb. employee from prakashs;
revoke update on practicedb. employee from prakashs;
drop user prakashs;

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.

Use assignments;
Desc books;

insert into books values (1241,'Making India Awesome',2005,978156,5748), (1242,'A Bend in the River',2011,978956,5749), (1243,'One indian Girl',2018,978156,5748), (1244,'A Brush with Life',2006,978156,5750);

Select * from books;

alter table books modify column ISBN bigint;
desc authors;

insert into authors values (5748,'Chetan Bhagat'), (5749,'V.S. Naipaul'),(5750,'Satish Gujral'),(5751,'Gita
'> Mehta');

select * from authors;
desc borrowings;

insert into borrowings values(1,1241,451,'2024-02-12','2024-02-21','submitted'),(2,1243,452,'2024-04-23','2024-04-29','submitted'),(3,1244,453,'2024-01-04','2024-01-11','Not submitted');

select * from borrowings;
desc members;

insert into members values (451,'Arun', 'arun@623.com',7483327942,'Hyderabad'), (452,'Rajan', 'rajan@123.com',783539869,'Sripuram'), (453,'Lohith', 'lohit@58.com',7933942983,'Madnpur');

delete from members where member_id=453;