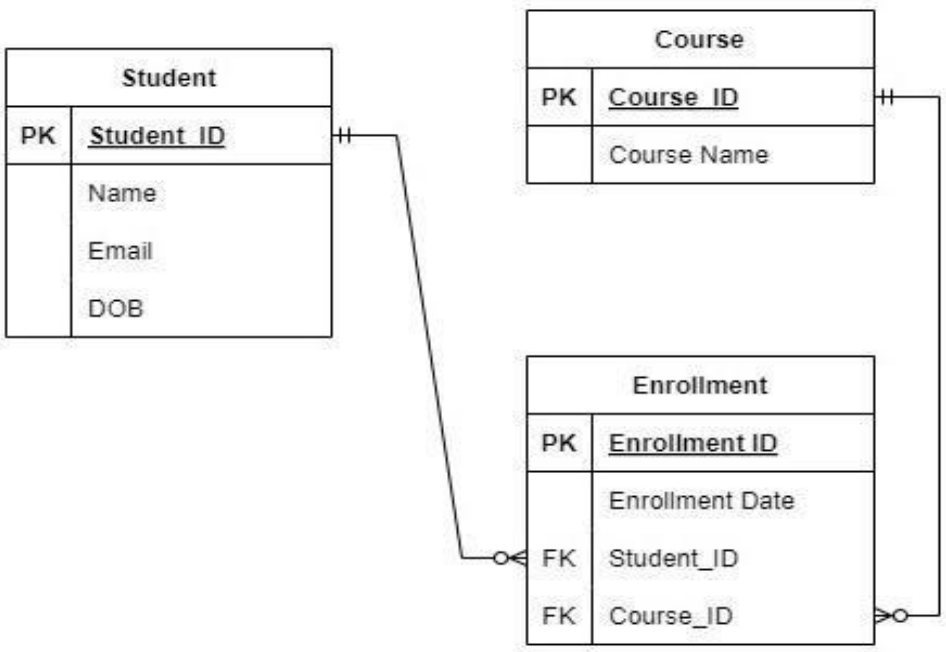
ASSIGNMENT-1

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.

Solution:



Above is an ER Diagram of a University course enrollment system that tracks students, courses and enrollments:

In this scenario, all tables (Student, Course, and Enrollment) adhere to 3NF:

●Student: Student_ID is the primary key, and all attributes are dependent on Student ID.

●Course: Course_ID is the primary key, and all attributes are dependent on Course ID.

●Enrollment: Enrollment_ID is the primary key, and Enrollment Date, Student ID, and Course ID are dependent on Enrollment ID.

ASSIGNMENT-2

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.

Solution:

Design a database schema for a library system. We'll create several tables to represent different aspects of the library.

1.Books Table:

- Fields: book_id, title, author, publication_year, genre, ISBN
- Constraints: NOT NULL for title, author, and ISBN
- Primary Key: book_id

2.Authors Table:

- Fields: author_id, author_name, birth_year, nationality
- Constraints: NOT NULL for author_name
- Primary Key: author_id

3.Members Table:

- Fields: member_id, first_name, last_name, email, phone_number
- Constraints: NOT NULL for first_name, last_name, and email
- Primary Key: member_id

4.Borrowed Books Table:

- Fields: borrow_id, book_id, member_id, borrow_date, return_date
- Constraints: NOT NULL for book_id, member_id, borrow_date
- Primary Key: borrow_id
- Foreign Keys: book_id references Books(book_id), member_id references Members(member_id)

5.Genres Table:

- Fields: genre_id, genre_name
- Constraints: NOT NULL for genre_name
- Primary Key: genre_id

6.Book-Genre Relationship Table:

- Fields: book_id, genre_id
- Constraints: UNIQUE combination of book_id and genre_id
- Foreign Keys: book_id references Books(book_id), genre_id references Genres(genre_id)

ASSIGNMENT-3

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.

Solution:

The ACID properties are the four key characteristics that ensure the reliability and consistency of transactions in a database system:

Atomicity: This property ensures that a transaction is an all-or-nothing operation. It means that either all the operations within a transaction are completed successfully and committed to the database, or none of them are. There is no partial execution or partial commit.

Consistency: This property ensures that the database remains in a consistent state before and after the transaction. It means that the database constraints, such as foreign key constraints

and unique constraints, are not violated by the transaction. The transaction transforms the database from one consistent state to another consistent state.

Isolation: This property ensures that the execution of transactions concurrently does not interfere with each other. Each transaction should appear to execute independently without being affected by the concurrent execution of other transactions. Isolation prevents phenomena like dirty reads, non-repeatable reads, and phantom reads.

Durability: This property ensures that once a transaction is committed, its changes are permanently saved in the database and survive system failures such as power outages or crashes. Durability guarantees that the committed transactions remain intact even in the event of a system failure.

SQL Statements for Locking and Isolation Levels:

```
CREATE TABLE BankAccounts (AccountID INT PRIMARY KEY,Balance  DECIMAL(10, 2));

INSERT INTO BankAccounts (AccountID, Balance) VALUES (1, 500), (2, 500);

BEGIN TRANSACTION;

UPDATE BankAccounts SET Balance = Balance – 100 WHERE AccountID = 1;

UPDATE BankAccounts SET Balance = Balance + 100 WHERE AccountID = 2;

COMMIT;
```

Here are the different isolation levels and their effects:

- Read Uncommitted: Allows dirty reads (reading uncommitted changes). No locks are held during read operations.
- Read Committed: Prevents dirty reads. Reads only committed data. Locks are acquired during read operations.
- Repeatable Read: Ensures that data read during a transaction remains consistent. Locks are held until the end of the transaction.
- Serializable: Highest level of isolation. Ensures serial execution of transactions. Locks are held for the entire transaction.

ASSIGNMENT-4

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.

Solution:

```
CREATE DATABASE LibraryDB;
```

```
-- Create BOOK table
CREATE TABLE BOOK (
   BOOK_ID NUMBER PRIMARY KEY,
   TITLE VARCHAR2(55),
   PUBLISHER VARCHAR2(40)
);

-- Create AUTHOR table
CREATE TABLE AUTHOR (
   AUTHOR_ID NUMBER PRIMARY KEY,
   AUTHOR_NAME VARCHAR2(100)
);

-- Create WROTE table (many-to-many relationship)
CREATE TABLE WROTE (
   BOOK_ID NUMBER,
   AUTHOR_ID NUMBER,
   PRIMARY KEY (BOOK_ID, AUTHOR_ID),
   FOREIGN KEY (BOOK_ID) REFERENCES BOOK(BOOK_ID),
   FOREIGN KEY (AUTHOR_ID) REFERENCES AUTHOR(AUTHOR_ID)
);
#Modifying Table Structures
ALTER TABLE BOOK
ADD COLUMN PUBLICATION_YEAR INT;

DROP TABLE BOOK;
```

ASSIGNMENT-5

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.

Solution:

Creating an Index:

- An index is a database structure that improves data retrieval speed. It acts like a roadmap, allowing the database system to quickly locate specific records.
- When you create an index on one or more columns, the index entries serve as pointers to the corresponding rows in the table.
- For example, if you have a large table with 100,000 rows, creating an index on a specific column can significantly speed up queries that involve that column.

-- Create an index on the author_id column in the Books table

CREATE INDEX idx_author_id ON Books (author_id);

Index improves query performance:

1. Faster Data Retrieval: When you execute a query that involves filtering, sorting, or joining based on the author_id, the database engine can use the index to quickly locate the relevant rows in the table without having to perform a full table scan. This leads to faster data retrieval.
2. Reduced Disk I/O: Indexes store a sorted copy of the indexed column's values, making it easier for the database engine to locate specific rows. This reduces the amount of disk I/O required to retrieve data, especially when dealing with large tables.
3. Improved Query Execution Plans: With the index in place, the database optimizer can choose more efficient query execution plans, resulting in faster query processing times.

Removing the Index:

DROP INDEX idx_author_id ON Books;

After dropping the index:

Slower Query Performance: Queries that heavily relied on the index for efficient data retrieval may experience slower performance, especially those involving filtering, sorting, or joining based on the author_id column.

Increased Disk I/O: Without the index, the database engine may need to perform full table scans to locate the desired rows, leading to increased disk I/O and potentially slower query execution times.

Suboptimal Query Execution Plans: The database optimizer may choose less efficient query execution plans in the absence of the index, resulting in suboptimal performance for certain queries.

Assignment-6

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.

Solution:

Creating a database user:

Create ‘user1’ identified by ‘user@1’;

GRANT SELECT, INSERT, UPDATE, DELETE ON testdb.* TO

'user1';

REVOKE UPDATE, DELETE ON testdb.* FROM 'user1'; DROP USER ‘user1’;

Assignment-7

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.

Solution:

Inserting new records into the Library table

INSERT:

Insert into Library values(1011,’Atomic habits’,’Ganesh’,600);

Insert into Library values(1891,’Be your own sunshine’,’Candy’,300);

UPDATE:

Update table Library where book_num=1891;

DELETE:

Delete from Library where book_Name=’Atomic Habits’;

Inserting BULK records into the table:

Insert into Library values((1088,’Rich dad poor dad’, ’Williams’, 450),(8988, ’The woman in me’ ,’Tracy martin’, 780));