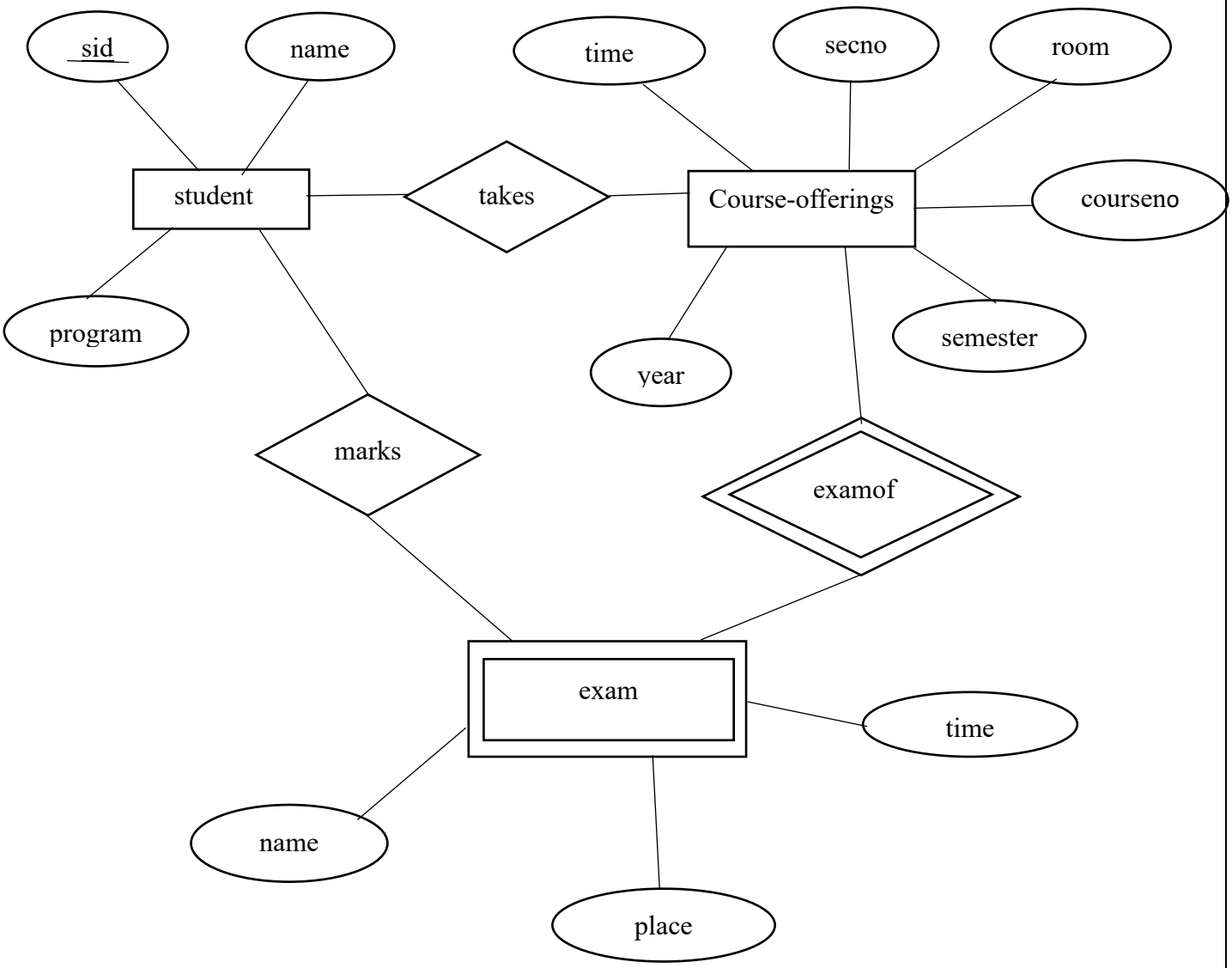
Day 1 Assignments

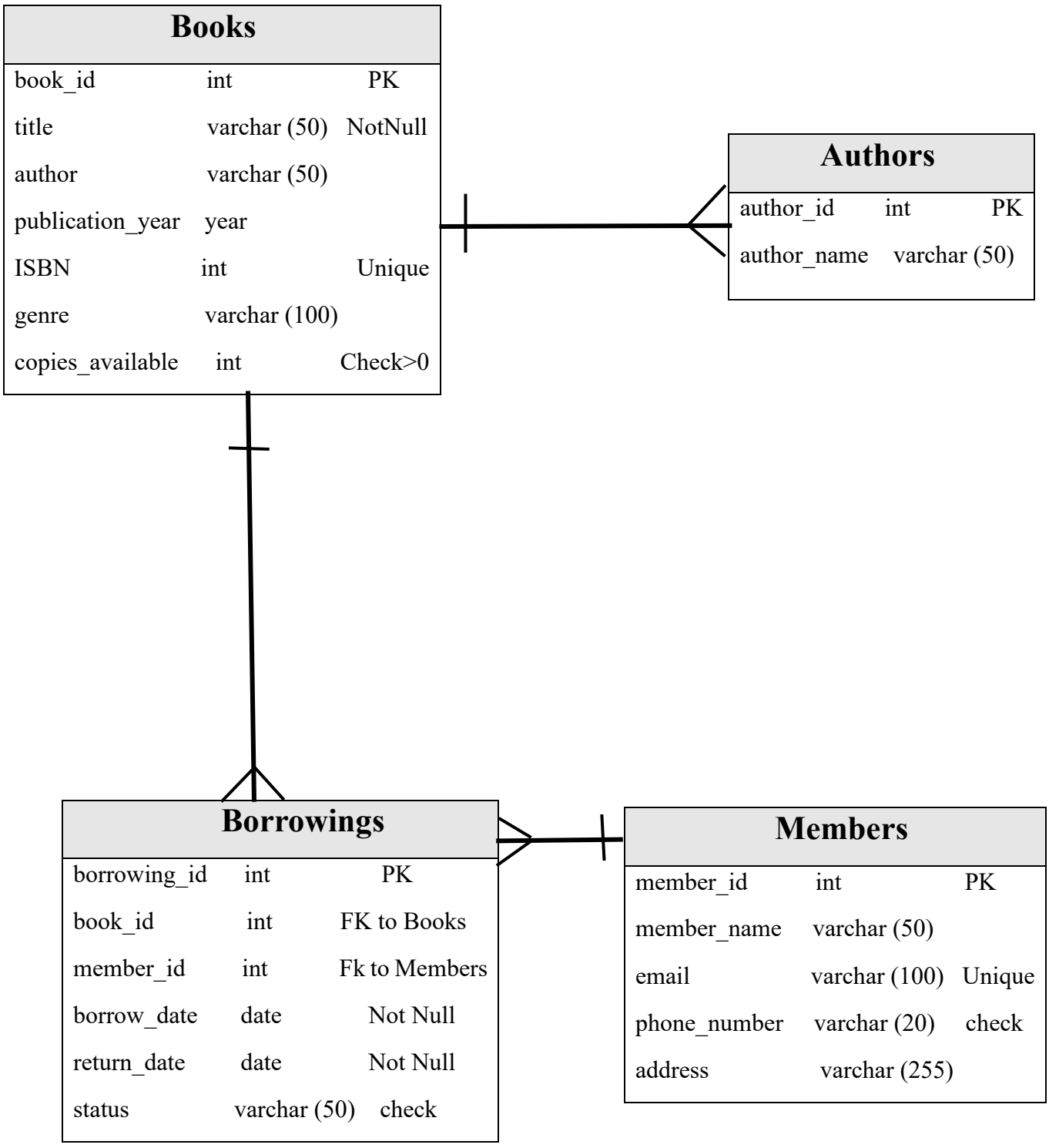
Assignment 1: Analyze a given business scenario and create an ER diagram that includes entities, relationships, attributes, and cardinality. Ensure that the diagram reflects proper normalization up to the third normal form.

E-R diagram that uses only a binary relationship between students and course-offerings. Make sure that only one relationship exists between a particular student and offering pair. Yet you can represent the marks that a student gets in different exams of a course offering.

E-R diagram for marks database



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.



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.

A transaction is a single logical unit of work that accesses and possibly modifies the contents of a database. Transactions access data using read and write operations. In order to maintain consistency in a database, before and after the transaction, certain properties are followed. These are called ACID properties.

ACID Properties:

• **Atomicity:** A transaction is either fully executed or not executed at all. It follows the "all or nothing" rule.

• **Consistency:** After a transaction, the database remains in a consistent state. Integrity constraints are maintained.

• **Isolation:** Concurrent transactions do not interfere with each other. Changes made by one transaction are not visible to others until committed.

• **Durability:** Once a transaction is committed, its changes are permanent even in case of system failure.

SQL Statements for Transaction Simulation:

Let's consider a simple example where we transfer money from one account to another.

```
1. -- Assume we have two accounts: Account X and Account Y
2. BEGIN TRANSACTION;
3. -- Deduct 100 from Account X
4. UPDATE Accounts SET Balance = Balance - 100 WHERE AccountNumber = 'X';
5.  -- Add 100 to Account Y
6. UPDATE Accounts SET Balance = Balance + 100 WHERE AccountNumber = 'Y';
7. COMMIT; -- If successful, commit changes; otherwise, rollback
```

Day 2 - 6th Assignment

Assignment 6: Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Report on the Use of Transaction Logs for Data Recovery

Introduction

Transaction logs are essential components of database management systems (DBMS). They record all changes made to the database, providing a reliable means to recover data in case of unexpected failures, such as system crashes or power outages. By maintaining a sequential record of all transactions, these logs ensure data integrity and consistency, facilitating both point-in-time recovery and rollback of transactions.

Functionality of Transaction Logs

1.Record Keeping: Transaction logs capture every change to the database, including insertions, updates, deletions, and the state of transactions (committed or uncommitted).

2.Atomicity and Durability: Ensuring the atomicity and durability of transactions, transaction logs guarantee that all operations within a transaction are completed successfully or none at all.

3.Recovery Mechanism: In the event of a failure, the DBMS uses the transaction log to identify the state of the database at the time of the crash and to replay or rollback transactions to restore the database to a consistent state.

Hypothetical Scenario: Data Recovery Using Transaction Logs

Scenario Overview

Imagine a financial services company, FinTech Corp, managing a critical database that stores transaction records for millions of users. This database processes high volumes of financial transactions daily, requiring robust mechanisms for data integrity and availability.

Incident Description

On a busy Monday morning, FinTech Corp experiences an unexpected power outage due to a severe thunderstorm. The outage lasts for several hours, and during this period, the database server shuts down abruptly, interrupting multiple ongoing transactions.

Recovery Using Transaction Logs

1.Initial Analysis:

- After power is restored, the IT team assesses the impact and discovers that several transactions were in progress at the time of the shutdown.
- The database did not get a chance to complete these transactions or ensure they were committed to the disk.

2.Transaction Log Utilization:

- The database management system (DBMS) automatically initiates the recovery process upon startup.
- The DBMS reads the transaction log to determine the last consistent state before the crash.
- The transaction log entries indicate which transactions were committed and which were still in progress when the power outage occurred.

3.Redo and Undo Operations:

- **Redo:** For transactions marked as committed in the transaction log but not yet written to the database files, the DBMS re-applies these changes to ensure all committed transactions are reflected in the database.
- **Undo:** For transactions that were in progress but not committed, the DBMS rolls back these changes to maintain consistency. This involves reverting any partial updates made by these transactions.

4.Verification and Consistency Check:

- After applying the redo and undo operations, the DBMS performs a consistency check to ensure the integrity of the database.
- The database is brought back to a consistent state, with all committed transactions applied and uncommitted transactions rolled back.

Conclusion

Transaction logs are indispensable for database recovery in the event of unexpected failures. By meticulously recording each transaction and maintaining the order of operations, these logs enable DBMSs to restore databases to a consistent state, ensuring data reliability and continuity of operations even under adverse conditions.