Part 1: Build the DatabaseManager class

Design the DatabaseManager class:

The class should initialize the connection to the PostgreSQL database using psycopg2.

The class should have methods for:

- Connecting to the database (with connection pooling support)
- Executing queries (e.g., SELECT, INSERT, UPDATE, DELETE)
- Managing transactions (begin, commit, and rollback)

Implement proper error handling and ensure the connection is closed when no longer needed.

Design the Model class:

Create a base class for mapping Python objects to database tables.

Implement CRUD operations:

- Create: Insert new records into the database
- Read: Fetch records from the database based on given conditions
- Update: Modify existing records in the database
- Delete: Remove records from the database based on given conditions

Ensure that the Model class can be easily extended for specific table representations.

Part 2: Use the ORM and create a sample application

Design and implement a sample application:

Choose a simple but meaningful application (e.g., a blog with posts, comments, and users).

Create Python classes for each table, inheriting from the Model class.

Define necessary columns and relationships between tables.

Implement CRUD operations using the custom ORM for each class.

Provide examples of how to use the custom ORM for basic CRUD operations and relationships between tables.