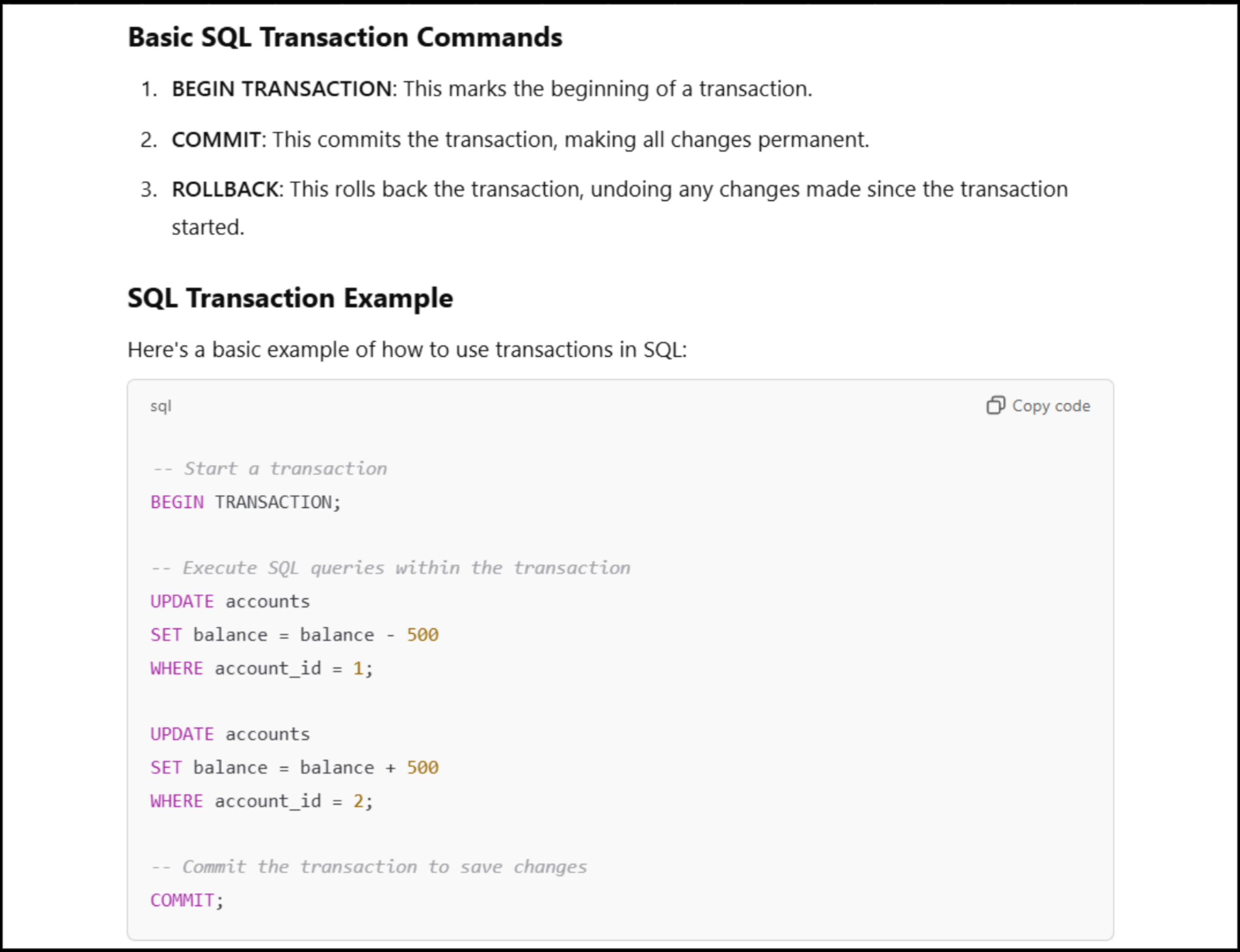
[Implenting Transactions]



Basic SQL Transaction Commands

1. **BEGIN TRANSACTION**: This marks the beginning of a transaction.
2. **COMMIT**: This commits the transaction, making all changes permanent.
3. **ROLLBACK**: This rolls back the transaction, undoing any changes made since the transaction started.

SQL Transaction Example

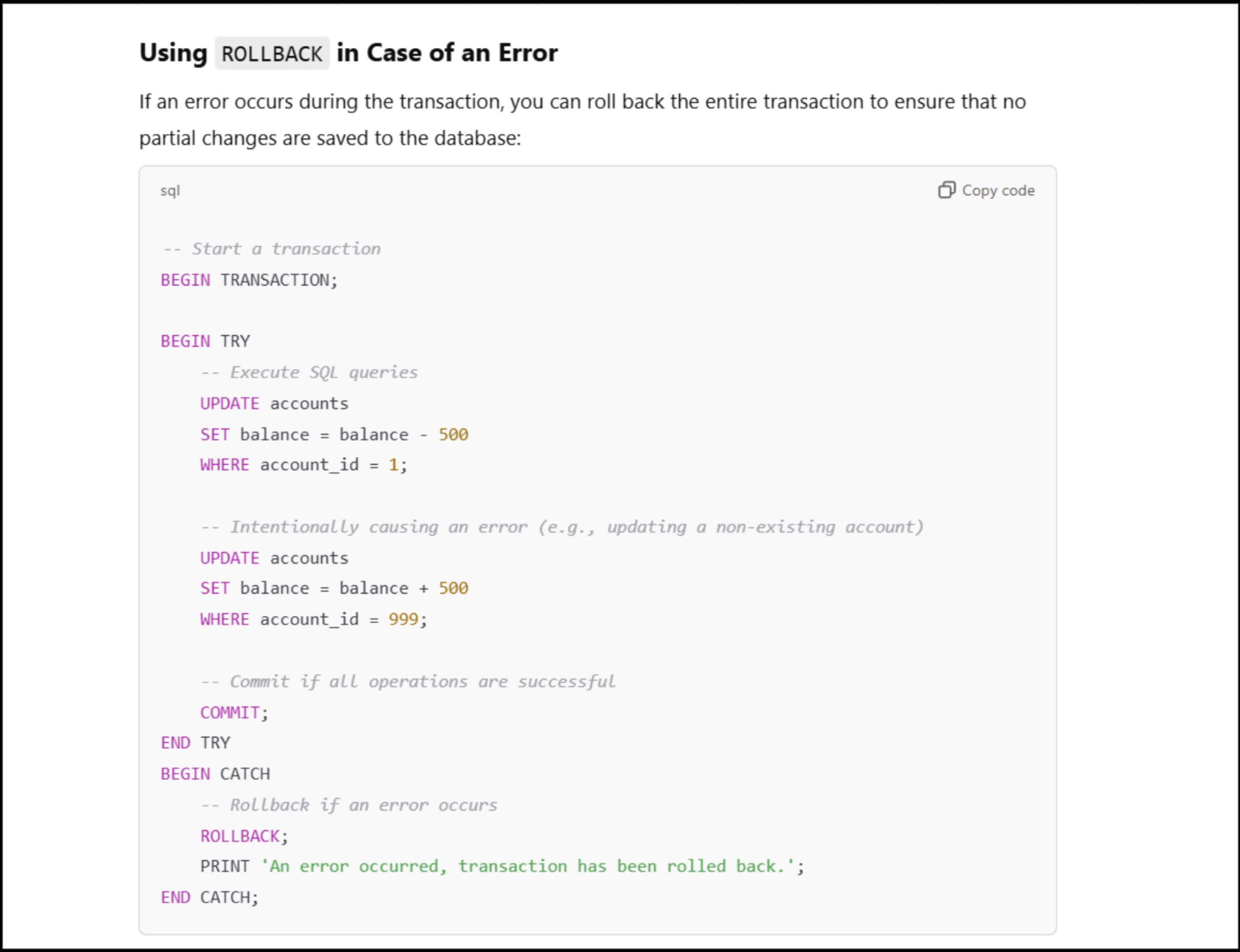
Here's a basic example of how to use transactions in SQL:

```sql
-- Start a transaction
BEGIN TRANSACTION;

-- Execute SQL queries within the transaction
UPDATE accounts
SET balance = balance - 500
WHERE account_id = 1;

UPDATE accounts
SET balance = balance + 500
WHERE account_id = 2;

-- Commit the transaction to save changes
COMMIT;
```



Using `ROLLBACK` in Case of an Error

If an error occurs during the transaction, you can roll back the entire transaction to ensure that no partial changes are saved to the database:

```sql
-- Start a transaction
BEGIN TRANSACTION;

BEGIN TRY
    -- Execute SQL queries
    UPDATE accounts
    SET balance = balance - 500
    WHERE account_id = 1;

    -- Intentionally causing an error (e.g., updating a non-existing account)
    UPDATE accounts
    SET balance = balance + 500
    WHERE account_id = 999;

    -- Commit if all operations are successful
    COMMIT;
END TRY
BEGIN CATCH
    -- Rollback if an error occurs
    ROLLBACK;
    PRINT 'An error occurred, transaction has been rolled back.';
END CATCH;
```