

- Transactions are the building blocks for DBs, they help define groups of read/write operations
- Have a general awareness for ACID (Atomicity, Consistency, Isolation, and Durability)

what about this statement: I had only used UPDATE and INSERT ones but never the BEGIN TRANSACTION and COMMIT:

BEGIN TRANSACTION;

UPDATE Channels

SET subscribers = subscribers + 1

WHERE channelId = "jayzern"

INSERT Subscribers

VALUES ("you", "jayzern")

COMMIT;

SQL Expert

The **BEGIN TRANSACTION**, **COMMIT**, and sometimes **ROLLBACK** (not mentioned but related) statements are used in SQL to manage transactions. Transactions are crucial for maintaining data integrity and ensuring that multiple operations execute as a single unit of work, meaning either all operations succeed together, or none of them take effect (if there's an error or rollback).

Here's how each component functions within the context of a transaction:

BEGIN TRANSACTION

This statement marks the start of a transaction. It signals the database management system (DBMS) that the subsequent operations (like **UPDATE**, **INSERT**, **DELETE**) should be treated as part of a single logical transaction. This is important for preserving the atomicity of the group of operations.

SQL Operations

Between `BEGIN TRANSACTION` and `COMMIT`, you can perform various SQL operations. These operations are part of the transaction and thus will be committed all at once if they all succeed.

In your example, there are two operations:

`UPDATE Channels SET subscribers = subscribers + 1 WHERE channelId = "jayzern"`: This increases the subscriber count by 1 for the channel with `channelId` equal to "jayzern".

`INSERT INTO Subscribers VALUES ("you", "jayzern")`: This adds a new row to the `Subscribers` table, indicating that the user "you" has subscribed to the channel "jayzern".

COMMIT

This statement is used to end the transaction and make all changes performed in the transaction permanent. If the `COMMIT` is successful, all changes made during the transaction are saved to the database.

If an error occurs during the transaction (for example, if one of the SQL operations fails due to a constraint violation or another issue), you can use the `ROLLBACK` statement to undo all changes made in the transaction, reverting the database to its previous state before the transaction started. This helps ensure the consistency and integrity of your data.

Note on Your Syntax

While your example correctly demonstrates the use of a transaction, be aware that the exact syntax for beginning a transaction can vary between different SQL databases. For instance, some databases might use `BEGIN TRANSACTION;`, others might use `START TRANSACTION;`, and in some cases, just `BEGIN;` might suffice.

Also, in your `INSERT` statement, it's generally good practice to specify the column names for clarity, like this:

sql

Copy code

```
INSERT INTO Subscribers (column1, column2) VALUES ("you", "jayzern");
```

Replace `column1` and `column2` with the actual column names for better clarity and to avoid potential errors.

The use of transactions, especially in operations that involve multiple steps or when dealing with important data, is considered best practice as it helps maintain data accuracy and integrity.