MySQL Transactions and AutoCommit

By default, MySQL operates in **AutoCommit** mode. This means that every SQL statement is treated as a **transaction** and is committed automatically. However, for more control over when changes are saved, you can turn **AutoCommit** off and manage transactions manually.

---

1. Disabling AutoCommit

When **AutoCommit** is off, you can explicitly control when to commit or rollback changes.

To disable AutoCommit:

```
SET autocommit = 0;
```

This turns off AutoCommit, meaning that changes you make won't be saved to the database unless you explicitly tell MySQL to commit them.

> **Important**: Until you execute a `COMMIT`, your changes are not permanent.

---

2. COMMIT — Save Changes to the Database

Once you've made changes and you're confident that everything is correct, you can use the `COMMIT` command to save those changes.

To commit a transaction:

```
COMMIT;
```

This saves all the changes made since the last `COMMIT` or `ROLLBACK`. After this point, the changes become permanent.

---

3. ROLLBACK — Revert Changes to the Last Safe Point

If you make an error or decide you don't want to save your changes, you can **rollback** the transaction to its previous state.

To rollback a transaction:

```
ROLLBACK;
```

This undoes all changes since the last `COMMIT` or `ROLLBACK`.

Example Workflow

Here's a simple example of using `COMMIT` and `ROLLBACK` in a transaction:

1. Turn off AutoCommit:

   ```
   SET autocommit = 0;
   ```

2. Make some changes (e.g., updating a salary):

   ```
   UPDATE users SET salary = 80000 WHERE id = 5;
   ```

3. Decide whether to commit or rollback:

   1. If you're happy with the changes, run:

      ```
      COMMIT;
      ```

   2. If you're not happy and want to revert the changes, run:

```
ROLLBACK;
```

---

4. Enabling AutoCommit Again

If you want to turn AutoCommit back on (so that every statement is automatically committed), you can do so with:

```
SET autocommit = 1;
```

Best Practices

- **Use** `COMMIT` when you want to make changes permanent.
- **Use** `ROLLBACK` to discard changes if something goes wrong.
- Consider **disabling AutoCommit** when performing complex updates to avoid saving partial or incorrect data.