◇ CASCADE

Why used:
To automatically apply the same change to related records.

What it does:

- `ON DELETE CASCADE` → deletes related child rows
- `ON UPDATE CASCADE` → updates foreign key values

Why we need it:
✓ Keeps data consistent
✓ Prevents orphan records

Example:
If a **Course** is deleted → all its **Enrollments** are deleted automatically.

◇ SET NULL

Why used:
When you want to **keep the child record**, but remove its link to the parent.

What it does:

- Sets the foreign key column to `NULL`

Why we need it:
✓ Keeps historical data
✓ Indicates "no longer related"

Requirement:

- Foreign key column **must allow NULL**

Example:
If an **Instructor** is deleted → `InstructorID` in **Courses** becomes `NULL`.

◇ SET DEFAULT

Why used:
To replace the foreign key with a **default value** when the parent row changes.

What it does:

- Sets the foreign key to a predefined default value

Why we need it:
✓ Maintains a valid reference
✓ Useful for "Unknown" or "General" records

Requirement:

- Column must have a `DEFAULT` value
- That default value must exist in the parent table

Example:
If a category is deleted → courses are assigned to **"General" category**.

◇ NO ACTION

Why used:
To **prevent changes** that would break relationships.

What it does:

- Rejects DELETE or UPDATE if child rows exist

Why we need it:
✓ Protects important data
✓ Forces manual review

Example:
You cannot delete a **Student** if enrollments still exist.

◇ Quick Comparison

Option	Purpose	Result
CASCADE	Auto apply change	Child rows updated/deleted
SET NULL	Keep child, remove link	FK becomes NULL
SET DEFAULT	Replace with default	FK gets default value
NO ACTION	Block change	Error occurs

◇ Simple memory tip 🧠

- **CASCADE** → *Follow parent*
- **SET NULL** → *Remove relationship*
- **SET DEFAULT** → *Assign backup value*
- **NO ACTION** → *Stop the operation*

◆ خلاصة للحفظ 📝

- CASCADE = التغيير ينتقل
- SET NULL = قطع العلاقة
- SET DEFAULT = قيمة بديلة
- NO ACTION = منع العملية

◇ Why are CASCADE, SET NULL, SET DEFAULT, and NO ACTION important?

1️⃣ Keep data consistent

They prevent **broken relationships** between tables.

Example:
If a parent record is deleted, these rules decide **what happens to related child records**, instead of leaving invalid data.

2️⃣ Prevent orphan records

Without these rules, child records may reference **records that no longer exist**.

Example:
Enrollment points to a CourseID that was deleted → ❌ error data.

3️⃣ Give control over delete/update behavior

Different systems need different behaviors.

- **CASCADE** → delete everything related
- **SET NULL** → keep child data but remove link
- **SET DEFAULT** → assign a safe backup value
- **NO ACTION** → block dangerous changes

This gives database designers **full control**.

4️⃣ Improve data reliability

They ensure that:

- Reports are accurate
- Joins return correct results
- Data reflects real business rules

5️⃣ Reduce manual errors

The database automatically enforces rules instead of relying on users or developers.

✓ Less human mistakes
✓ Safer operations