Great question 🔥. The `parentId` field in this schema shows up in **3 tables**:

- `post.parentId`
- `category.parentId`
- `post_comment.parentId`

It's used for **hierarchical (tree-like) relationships** — meaning one record can be a *child* of another record in the same table.

Let me break it down with **examples** 👇

1️⃣ `post.parentId` → Related Posts / Series

This allows posts to be linked together (like chapters in a tutorial series).

Example:

- Post 1: "Python for Beginners" (`parentId = NULL`)
- Post 2: "Python for Beginners – Part 2" (`parentId = 1`)
- Post 3: "Python for Beginners – Part 3" (`parentId = 1`)

Here:

- Post 1 is the **main article (parent)**.
- Posts 2 and 3 are **children posts**, connected via `parentId = 1`.

So you can query: *"Show me all posts that belong to the series 'Python for Beginners'."*

2️⃣ `category.parentId` → Nested Categories

This allows categories to be grouped into hierarchies.

Example:

- Category 1: "Programming" (`parentId = NULL`)
- Category 2: "Python" (`parentId = 1`)
- Category 3: "Machine Learning" (`parentId = 1`)
- Category 4: "Deep Learning" (`parentId = 3`)

Here:

- "Programming" is a **top-level category**.
- "Python" and "Machine Learning" are **subcategories** of "Programming".
- "Deep Learning" is a **sub-subcategory** under "Machine Learning".

So you can build category trees like DataCamp:
Programming → Machine Learning → Deep Learning

3 `post_comment.parentId` → Threaded Comments

This allows **replies** to comments.

Example:

- Comment 1: "Great post!" (`parentId = NULL`)
- Comment 2: "Thanks!" (`parentId = 1`)
- Comment 3: "I also recommend checking out NumPy." (`parentId = 1`)
- Comment 4: "Agree with you." (`parentId = 3`)

Here:

- Comment 1 is a **main comment**.
- Comment 2 and 3 are **direct replies** to Comment 1.
- Comment 4 is a **reply to Comment 3**.

This creates a **threaded discussion** (just like Reddit or Disqus).