Topic: Django Base Type of 4 fields in models.

Simple Django Project, Create class base models.

Use to 4 type of django models field.

In Django's models.py**, the line: artist = models.ForeignKey(Musician, on_delete=models.CASCADE)**

Note : creates a **foreign key relationship** between the `Musician` model and the model in which this line is used.

Breaking It Down:

1. **`models.ForeignKey(Musician, ...)`**
   - This establishes a **one-to-many** relationship.
   - It means each record in the current model is associated with **one** `Musician`, but a `Musician` can have **multiple** related records.

2. **`on_delete=models.CASCADE`**
   - This specifies what happens when a referenced `Musician` is deleted.
   - `CASCADE` means **if a Musician is deleted, all related records in this model will also be deleted.**
   - It's useful to maintain **referential integrity** and avoid orphaned records.

How It Works:

- If a **Musician** (e.g., "John Doe") has multiple albums linked to them in the `Album` model.
- If "John Doe" is deleted, all their albums will also be removed from the database.
- **Alternative `on_delete` Options:**

Option	**Behavior**
`models.CASCADE`	Deletes related records.
`models.PROTECT`	Prevents deletion if related records exist
`models.SET_NULL`	Sets the foreign key to `NULL` instead of deleting.
`models.SET_DEFAULT`	Sets the foreign key to a default value.
`models.DO_NOTHING`	Does nothing (may cause integrity issues).