



GROUP 3

**BINARYFIELD
BOOLEANFIELD
NULLBOOLEANFIELD**

By : Widaad, Afiq Ilham



WHAT IS BINARY FIELD

- A SPECIAL FIELD TO STORE RAW BINARY DATA.
- IT CAN BE ASSIGNED BYTES, BYTEARRAY, OR MEMORYVIEW
- DATA TYPE : BINARY DATA (EXAMPLE : FILES, IMAGES, ETC).
- EXAMPLE OF DATA TYPE : TRUE, FALSE OR NULL
- BY DEFAULT, BINARYFIELD SETS EDITABLE TO FALSE, THAT IS IT CAN'T BE INCLUDED IN A MODELFORM.
- IT CAN NOT BE MANUALLY ENTERED AND NEEDS TO BE ASSIGNED THROUGH A VIEW OR DJANGO SHELL DUE TO BINARYFIELD STORES RAW DATA

Database Storage Impact:

This field stores raw binary data directly in the database. In databases like PostgreSQL, it maps to a BYTEA type, while in MySQL, it maps to a BLOB type. The size and storage requirements depend on the size of the binary data being stored.

Example of the Usage of Binary Field

```
# Generated by Django 2.2.5 on 2019-09-25 06:00

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='GeeksModel',
            fields=[
                ('id',
                 models.AutoField(auto_created = True,
                                   primary_key = True,
                                   serialize = False,
                                   verbose_name = 'ID'),
                ('geeks_field', models.BinaryField()),
            ],
        ),
    ]
```

2. BOOLEAN FIELD

- **DATA TYPE** : Boolean (True/False).
- **CODING IN DJANGO** : Used for storing True or False values.
- **EXAMPLE OF DATA** : True, False
- **DATABASE STORAGE IMPACT** :
 - This field stores boolean values. In most databases, it maps to a BOOLEAN or TINYINT type (e.g., 1 for True and 0 for False in MySQL). The storage impact is minimal as it typically requires only 1 byte.

- **USAGE EXAMPLE** :

```
PYTHON
```

```
FROM DJANGO.DB IMPORT MODELS
```

```
CLASS MYMODEL(MODELS.MODEL):
```

```
    IS_ACTIVE = MODELS.BOOLEANFIELD(DEFAULT=TRUE)
```

CODING IN DJANGO

```
# Generated by Django 2.2.5 on 2019-09-25 06:00

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='GeeksModel',
            fields=[
                ('id',
                 models.AutoField(auto_created = True,
                                   primary_key = True,
                                   serialize = False,
                                   verbose_name = 'ID'
                                   )),
                ('geeks_field', models.BooleanField()),
            ],
        ),
    ]
```

WHAT IS NULLBOOLEANFIELD ??

A FIELD IN DJANGO MODELS THAT WAS USED TO REPRESENT A BOOLEAN FIELD THAT CAN BE SET TO TRUE, FALSE, OR NONE (NULL).

DATA TYPE : NULLBOOLEANFIELD (TRUE / FALSE / NONE)

CODING IN DJANGO : USED FOR STORING TRUE , FALSE , OR NULL

EXAMPLE OF DATA : TRUE / FALSE / NONE

DATABASE STORAGE IMPACT : THIS FIELD STORES BOOLEAN VALUES WITH AN ADDITIONAL NULL OPTION, WHICH ALLOWS FOR AN INDETERMINATE STATE. IN MOST DATABASES, THIS IS STORED SIMILARLY TO BOOLEAN OR TINYINT, WITH AN ADDITIONAL POSSIBILITY OF NULL. HOWEVER, SINCE IT IS DEPRECATED, THE RECOMMENDED APPROACH IS TO USE BOOLEANFIELD(NULL=TRUE) INSTEAD.

**USAGE
EXAMPLE** →

```
FROM DJANGO.DB IMPORT MODELS
```

```
CLASS MYMODEL(MODELS.MODEL):
```

```
    MY_FIELD = MODELS.BOOLEANFIELD(NULL=TRUE)
```

CODING IN DJANGO MODEL


```
from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [
    ]

    operations = [
        migrations.CreateModel(
            name='GeeksModel',
            fields=[
                ('id', models.AutoField(auto_created=True, primary_key=True, serialize=False,
                ('geeks_field', models.BooleanField(null=True))),
            ],
        ),
    ]
```







THANK YOU

#FREEPALESTINE 

By : Widaad, Afiq , Ilham



