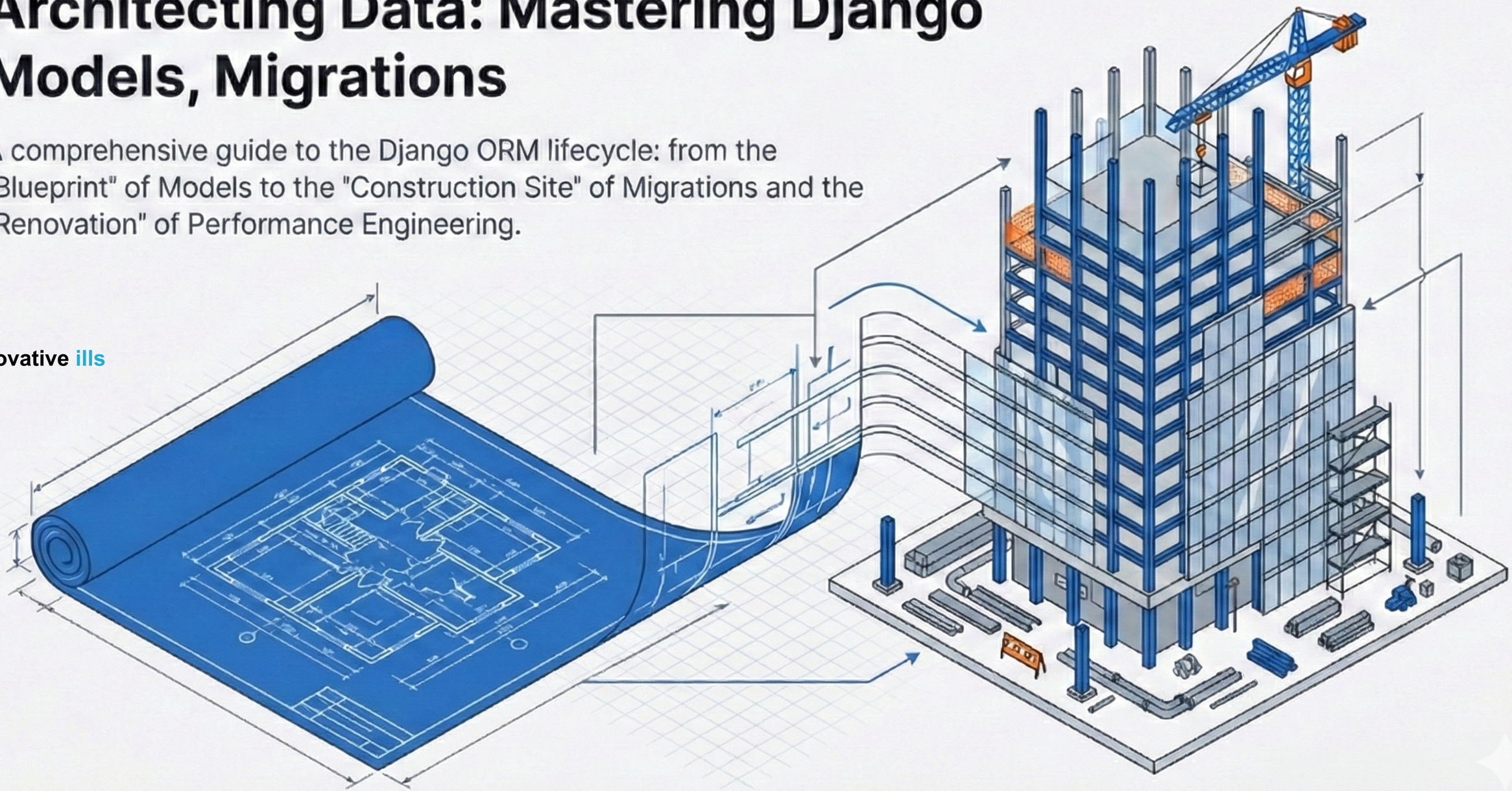
Architecting Data: Mastering Django Models, Migrations

A comprehensive guide to the Django ORM lifecycle: from the "Blueprint" of Models to the "Construction Site" of Migrations and the "Renovation" of Performance Engineering.



innovative ills

Authored by Abdullah Al Mahmud

The Blueprint: Django Models as the Source of Truth

The Model is a Python class that acts as the single, definitive source of truth. It defines the essential fields and behaviors of the data you are storing.

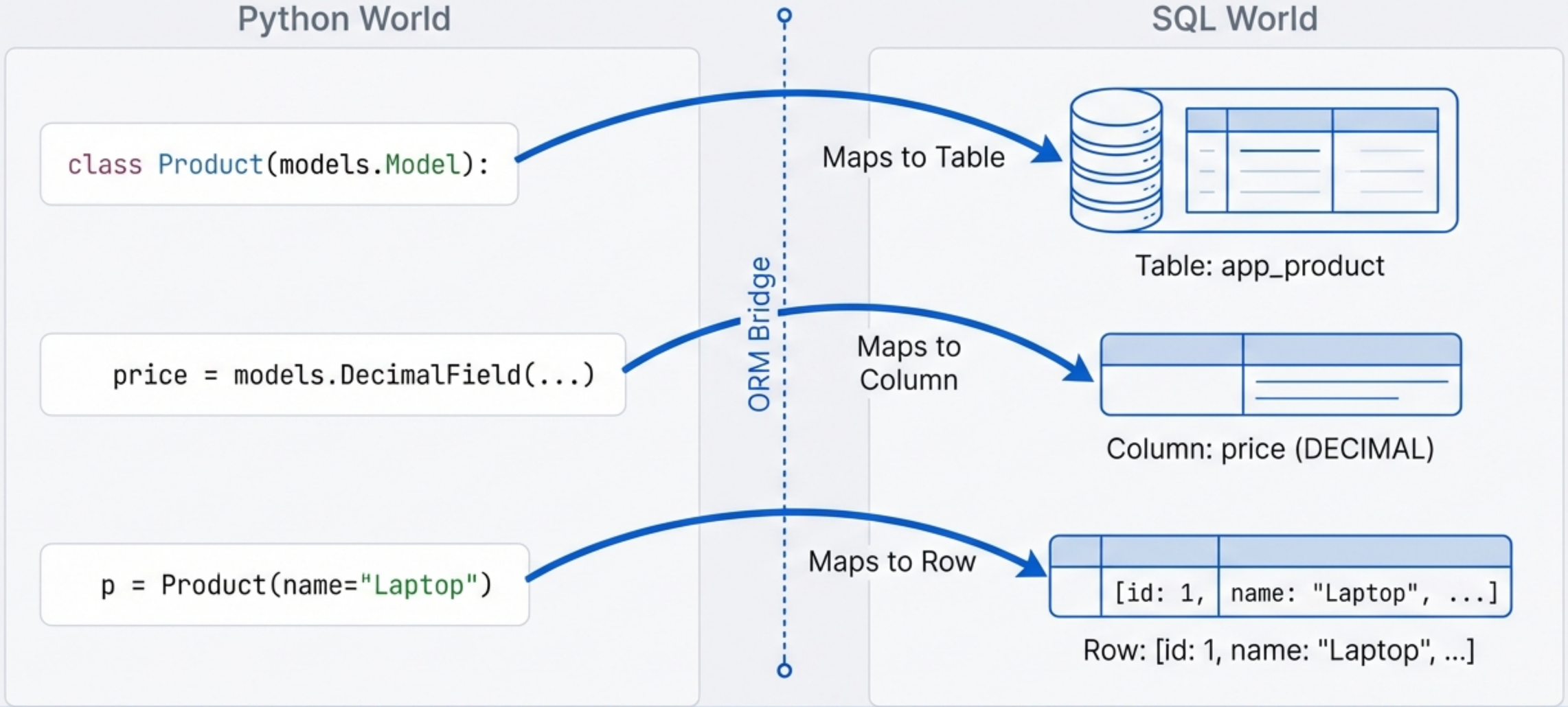
```python
from django.db import models

class Product(models.Model):
    name = models.CharField(max_length=100)
    price = models.DecimalField(max_digits=10, decimal_places=2)
    in_stock = models.BooleanField(default=True)

    def __str__(self):
        return self.name
```

Inheritance (Connects to ORM)

Schema Definitions

Authored by Abdullah Al Mahmud

The Translation Layer: Mapping Python to SQL

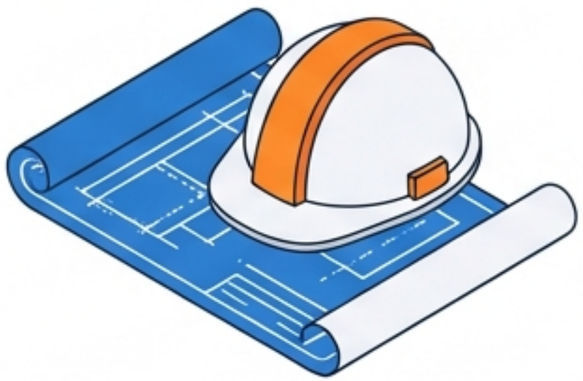


Authored by Abdullah Al Mahmud

Database Evolution: Migrations as Version Control

Just as Git tracks changes to your code, Migrations track changes to your database schema.
It is a two-step process.

The Architect

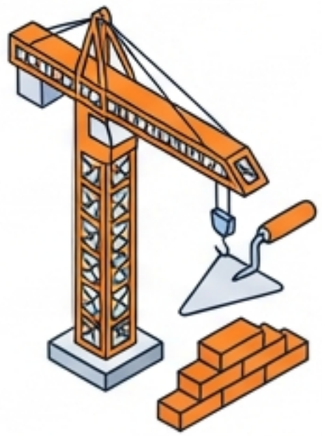


`makemigrations`

Plans the changes.
Writes the instructions.

Flow of Evolution

The Construction Crew



`migrate`

Executes the build.
Modifies the database.

Authored by Abdullah Al Mahmud

The Architect: Planning the Changes

```
python manage.py makemigrations
```

File Visualization.

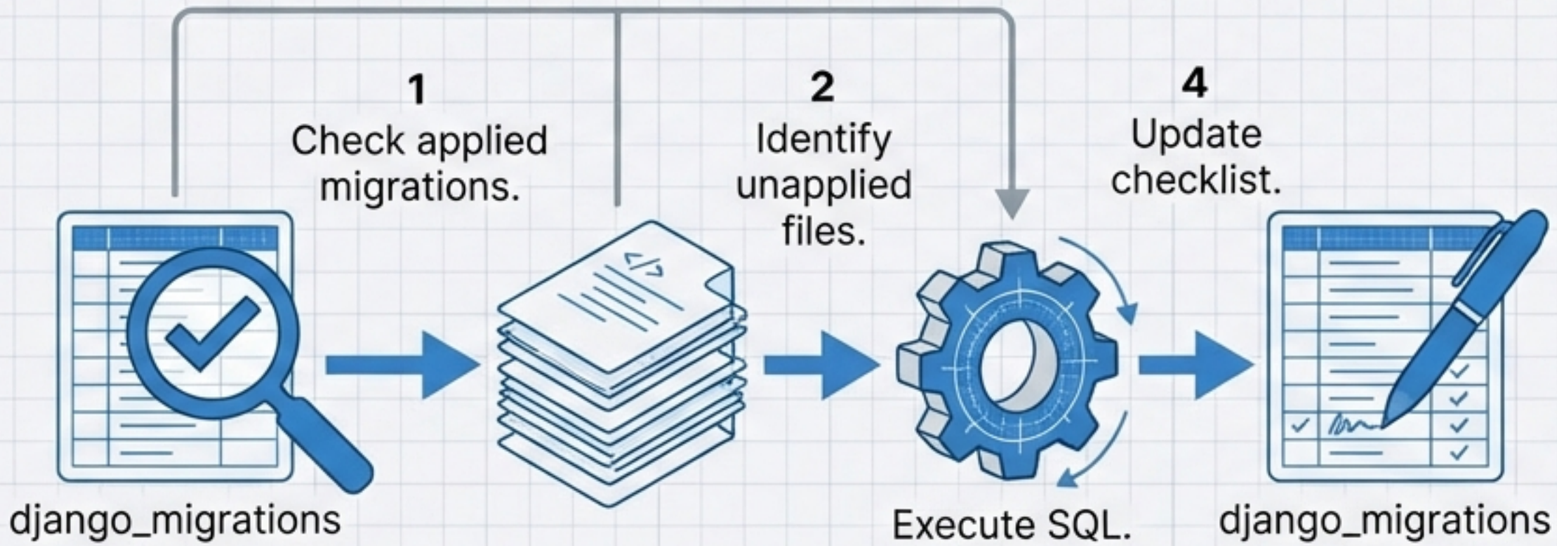
```
operations = [
    migrations.CreateModel(
        name='Product',
        fields=[...],
    ),
]
```

Authored by Abdullah Al Mahmud

The Construction Crew: Executing the Build

```
python manage.py migrate
```

Internal Process Diagram (The Checklist)



`django_migrations`

app	name	applied
auth	0001_initial	2023-10-27 10:00:00
contenttypes	0001_initial	2023-10-27 10:00:01
my_app	0001_initial	2023-11-15 14:30:00