Course Module 1: Course Unit 13: Django (Object-Relational Mapping (ORM) Layer)

Week 16

Django (Object-Relational Mapping (ORM) Layer)

Objectives:

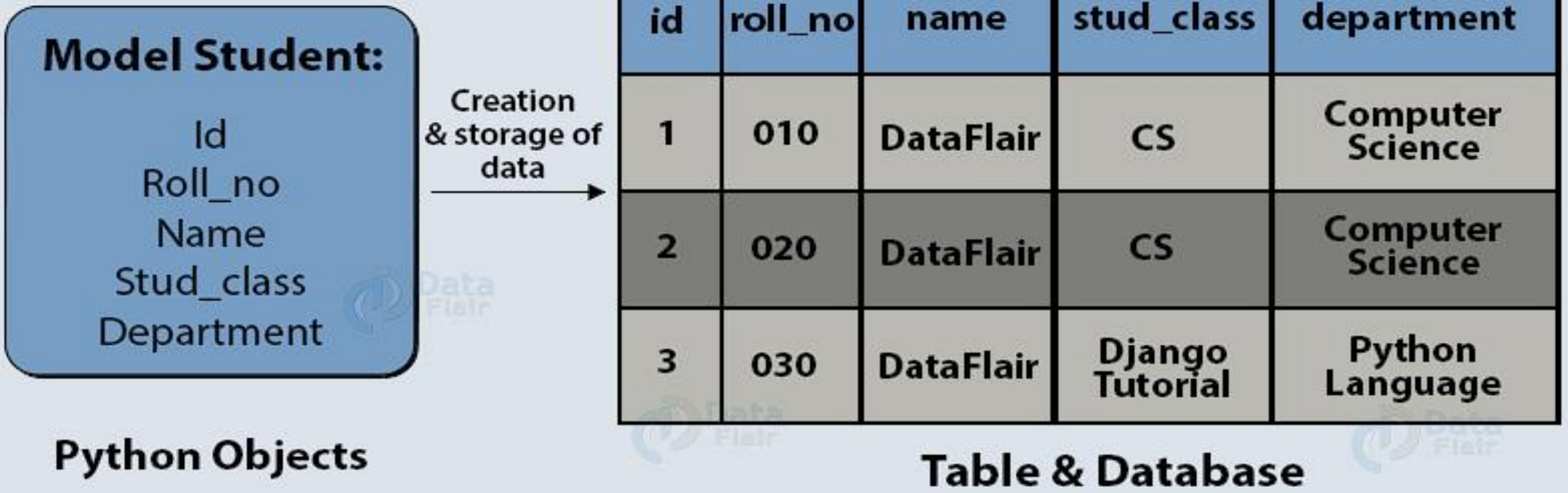
1. Describe Django Object-Relational Mapping (ORM) Layer.
2. Implement Django Object-Relational Mapping (ORM) Layer.
3. Plan a website that will utilize Django.

ORM

ORM

- object-relational mapper
- main goal is to transmit data between a relational database and application model
- automates this transmission, such that the developer need not write any SQL.
- maps objects attributes to respective table fields. It can also retrieve data in that manner.

django ORM Tutorial

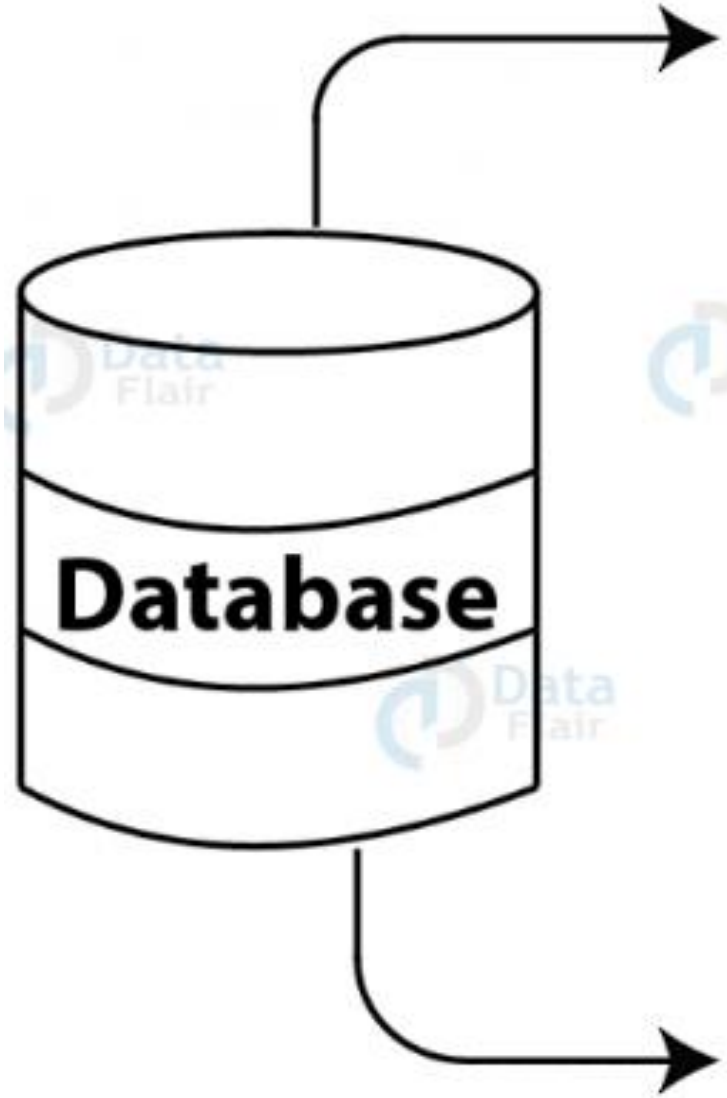


Python objects and a table with corresponding fields. The object's attributes are stored in corresponding fields automatically. An ORM will automatically create and store your object data in the database. You don't have to write any SQL for the same.

Advantages of ORM

- The main advantage ORMs provide is rapid development.
- ORMs make project more portable.
- It is easier to change the database if we use ORMs.

In the past, web developers needed to have **knowledge of databases** too. A database has been an important component from the start. The **programming languages** used for web development use **classes** and **object** for *data-interpretation*. The class is used to a defined data structure in web-applications. Then the same database schema is created in the database. This task requires skill and knowledge of SQL.



SQL

```
Create Table
django tutorials(
Title varchar(50),
Author char(20),
DateofUpload date()
)
```

Django ORM

```
class djangotutorials:
  title = CharField
  author = CharField
  dateofupload =
                DateField
```

ORM

Knowing SQL is also not enough since SQL implementations slightly differ from one another in different databases. This became a difficult and time-consuming task. So, to resolve this, the **concept of ORM** was introduced in web frameworks. **ORMs** automatically create a database schema from defined classes/ models. They generate SQL from Python code for a particular database. ORMs let the developer build the project in one language that means Python.

Web Framework	Flask	Pyramid	Django
ORM	SQLAlchemy	SQLAlchemy	Django ORM

Querysets in Django

- We all use queries to retrieve data from the database.
- Querysets are Django's way to retrieve data from the database.
- The Django ORM lets us use Querysets.

A **Queryset** is a list of objects of a model.

- We use Querysets to filter and arrange our data.
- These make our work as a Python developer easier.

Different Relationships between Fields

- Django ORM provides a level of **abstraction** (used to hide background details or any unnecessary implementation about the data so that users only see the required information) which makes it easy to work with objects. ORM will automatically relate the object's attributes to corresponding table fields.

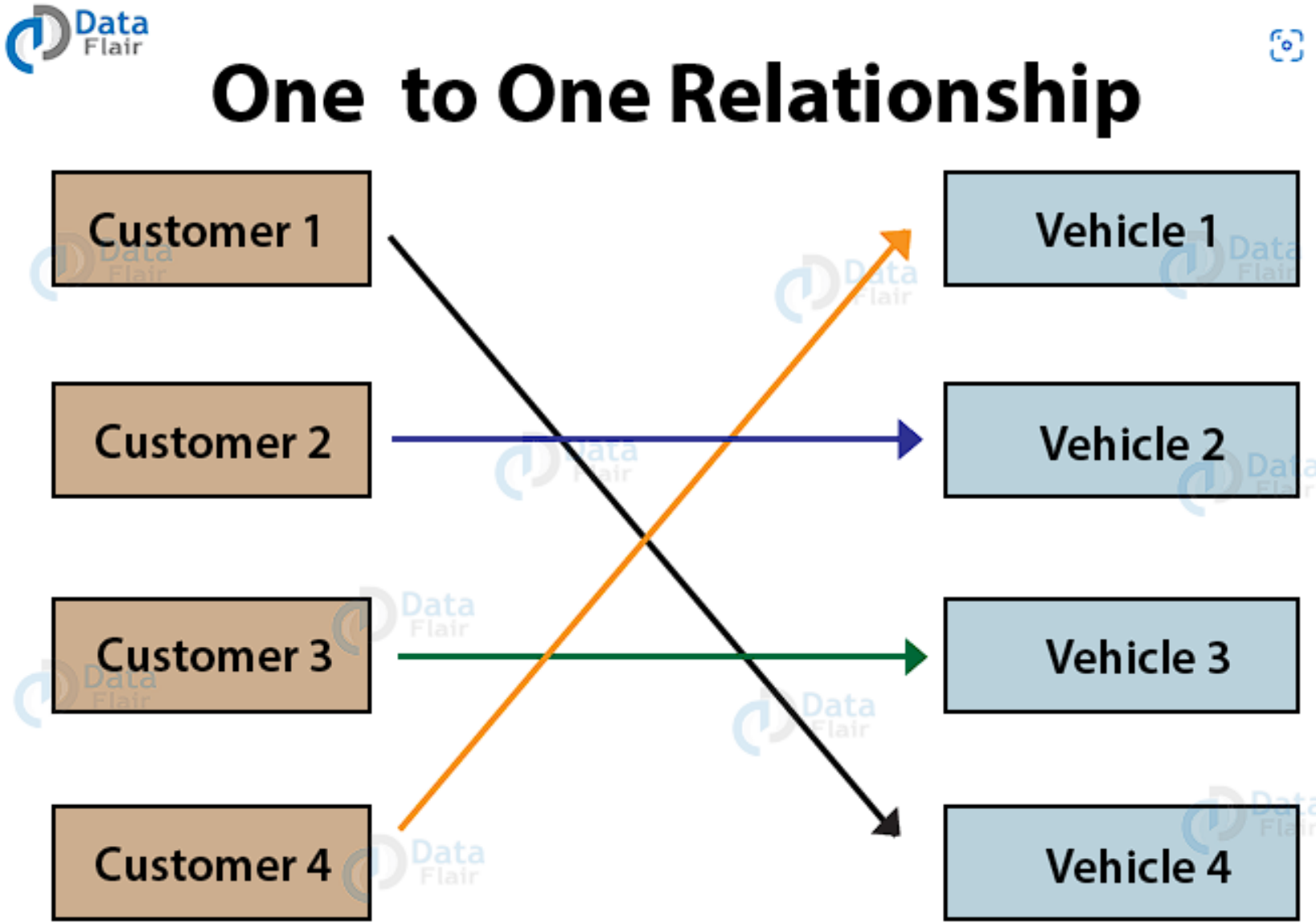
Field relationships:

- One to One
- One to Many
- Many to Many

Different Relationships between Fields

- **One to One**

A one-to-one relationship exists between two tables. For each row in table1, there shall be a row/ entity in table2.

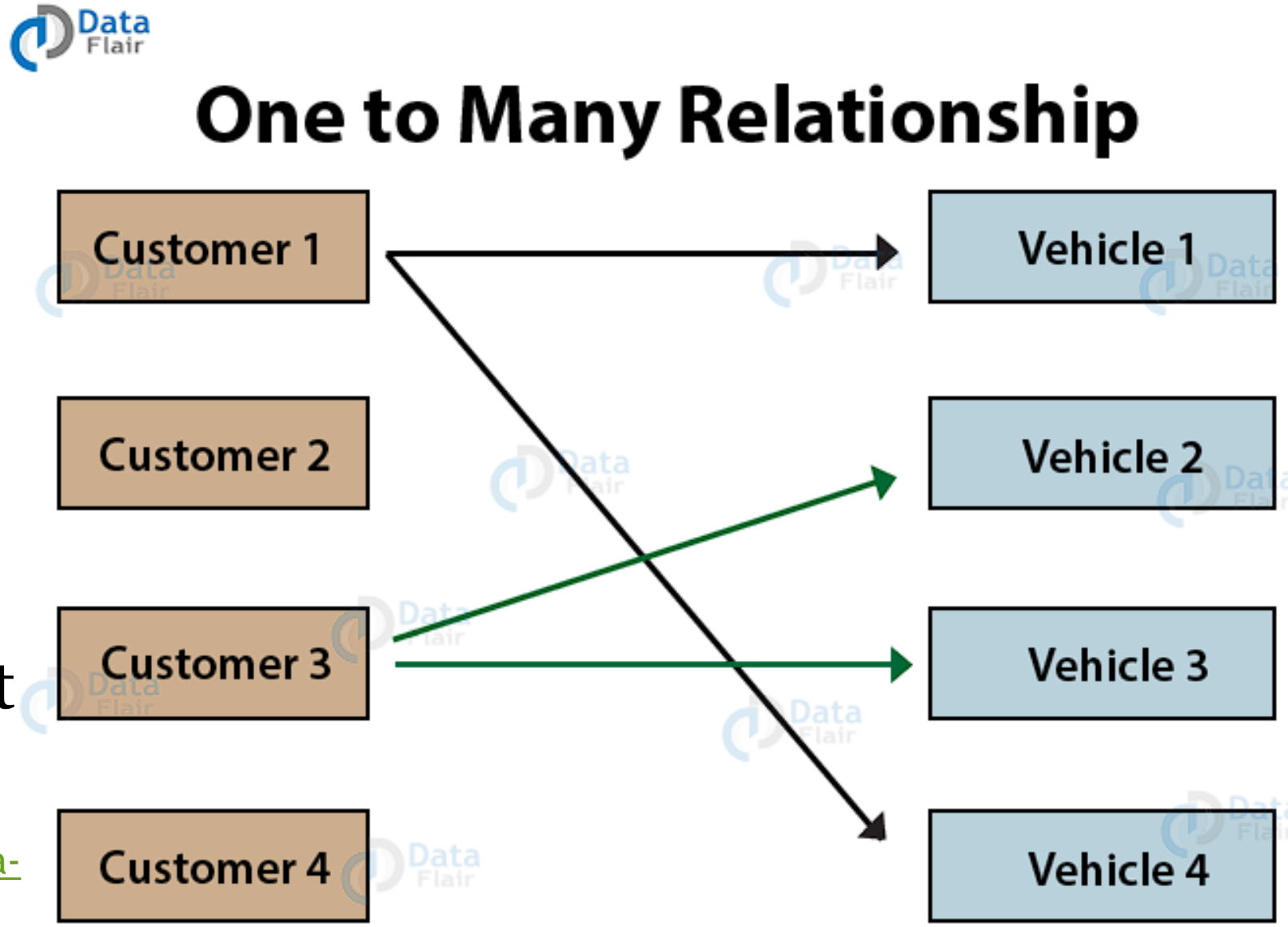


Django ORM Tutorial - The concept to master Django framework - DataFlair (data-flair.training)

Different Relationships between Fields

- **One to Many**

A one to many relationships is where one object from table1 can have multiple relations with entities in table2. Although, table2 objects will have only one relation to the object of table1.

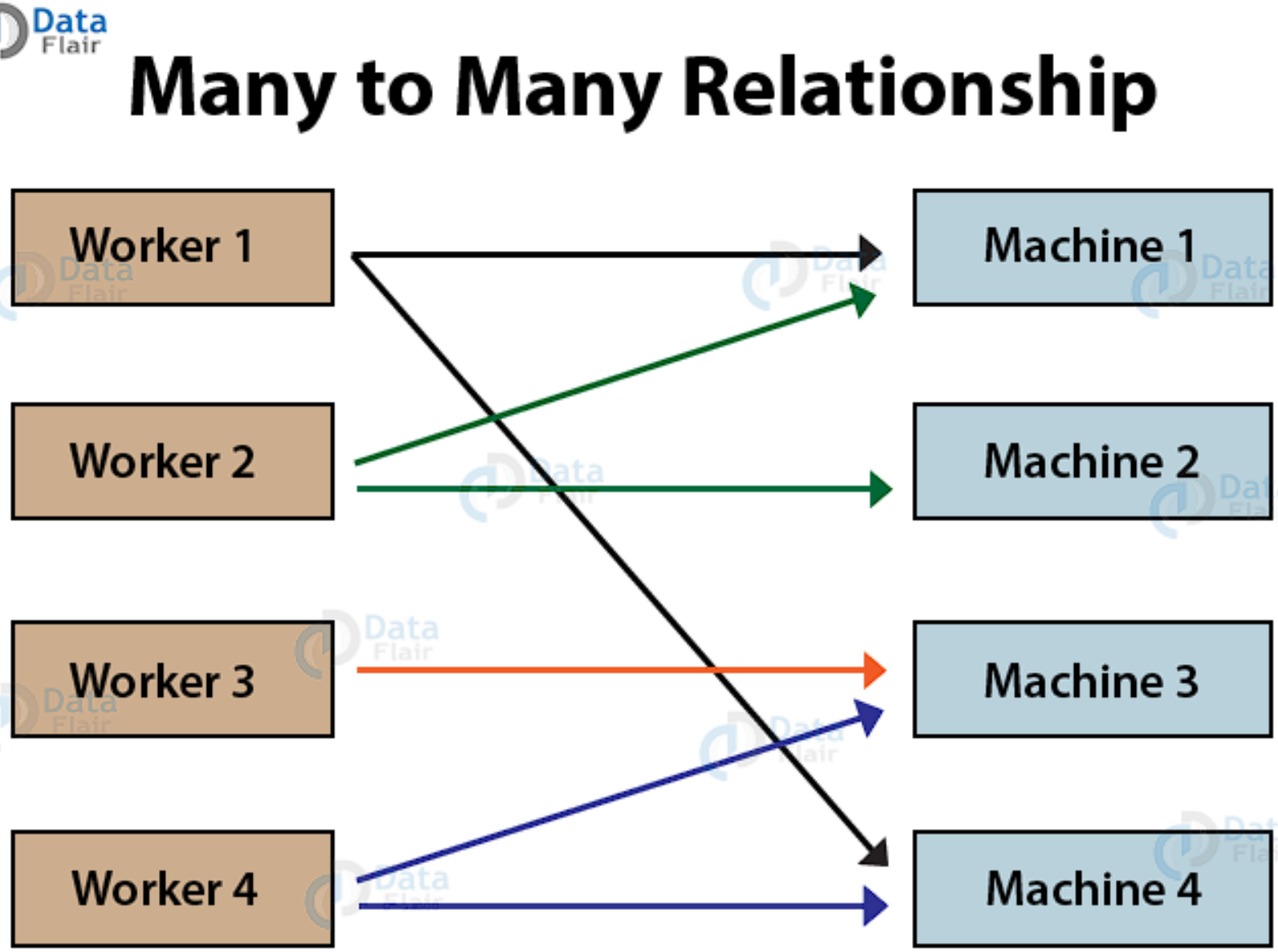


Django ORM Tutorial - The concept to master Django framework - DataFlair (data-flair.training)

Different Relationships between Fields

- **Many to Many**

A worker can be assigned to operate more than one machine. Also, a machine can be operated by multiple workers one at a time.



Django ORM Tutorial - The concept to master Django framework - DataFlair (data-flair.training)

Applications???

Any questions???

REFERENCES:

- Django Tutorial. (2022). https://data-flair.training/blogs/django-tutorial/
- Django Tutorial. (2020). https://www.geeksforgeeks.org/django-tutorial/
- Django Tutorial. (2022). https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction
- Django Tutorial. (n.d.). https://www.tutorialspoint.com/django/index.htm
- Django Tutorial. (n.d.). https://www.w3schools.com/django/index.php
- Getting started with Django. (n.d.). https://www.djangoproject.com/start/