Database relationships

When a table references the data from another table, it's called a **relationship.** The key in the referenced table is called the **foreign key**, and the key in the referencing table is called the **primary key**.

But it's a matter of perspective: If table **profile** is linked to table **User** determining which is the primary and foreign key depends on which table you are looking at.

Database relationships

Database relationships are the associations between different tables in a relational database. In a relationship database management system (RDBMS or DBMS), data is organized into tables. The relationships between these tables determine how the data in one table is connected to the data in the other table.

database schema

A database schema is a blueprint that describes how a database is structured and how its data is organized. It's typically represented visually using an entity-relationship diagram database schema defines the structure of the database, including:Tables, Fields, Relationships between tables and fields
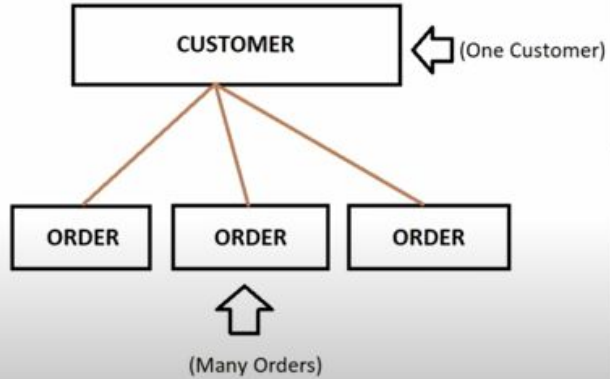
Other data elements Columns

The three main types of relationships:

- ❏ **One-to-one**
- ❏ **One-to-many**
- ❏ **many-to-many**

One to many relationship

A one-to-many relationship exists in a relational database when one row in table A is linked to many rows in table B, but only one row in table B is linked to one row in table A.

In a one-to-many relationship, one record in a table can be associated with one or more records in another table.

For example, each customer can have many Transaction.

Jimmie Q. Bobo

JAHJSAHDW7674ZOI784

Transaction details

Account NUBAN	0007381980
Bank	🟢 GTBank
Date	11 Jun, 2021 11:49PM

Transaction summary

	Total inflows		Total outflow
↙	₦205,732,333.51	↗	₦101,235,891.67

All transactions 1,150 records

🔍 �filter ⬇ Export

	TRANSACTION DATE	AMOUNT	DESCRIPTION	CHANNEL	FETCHED DATE
☐	11/02/2021 8:00am	-₦1,050,209.00	Mc loc pos prch-121210005383--accelerex network li oy oyng-	ATM	03 Aug, 2021
☐	11/02/2021 8:00am	-₦7,090,000.00	Nlp/abn/edison maduabuchukwu obodo/ trffrmedisonmaduabuchukwuobodotoedisonmaduabuch ukwuobodo	POS	03 Aug, 2021
☐	11/02/2021 8:00am	+₦205,732,333.51	sms notification charge jul 2021	Web	03 Aug, 2021
☐	11/02/2021 8:00am	-₦64,500.00	Nip cr/okra technologies limited/abn	POS	03 Aug, 2021
☐	11/02/2021 8:00am	-₦1,050,209.00	Transfer charges	Not Available	03 Aug, 2021
☐	11/02/2021 8:00am	+₦21,100,554.23	mclocposprch-072328799650--2011la040941478 lang-	Not Available	03 Aug, 2021
☐	11/02/2021 8:00am	-₦103,832.89	nip/stack/paystack/okra payment authorization refundokra technologies ltd	Not Available	03 Aug, 2021

Rows Per Page: 10 ⌄

Page 1 of 115 ‹ ›

Configuration Manager

SnapShots | Migration | **Transactions**

Transaction List

* Select Association   [1014Dev-QA   ▼]

Select a Transaction and [RollBack] [View]

Select	Transaction ID	Description	Performed By	Date	Status
○	1372	No Description	DemoUser	Sat Dec 16 05:52:57 GMT+05:30 2006	Done
○	1390	No Description	DemoUser	Sat Dec 16 07:00:18 GMT+05:30 2006	Done
○	1430	Rollback of Transaction 1372	DemoUser	Wed Dec 20 06:06:07 GMT+05:30 2006	Done
○	1431	Rollback of Transaction 1372	DemoUser	Wed Dec 20 06:33:07 GMT+05:30 2006	Done
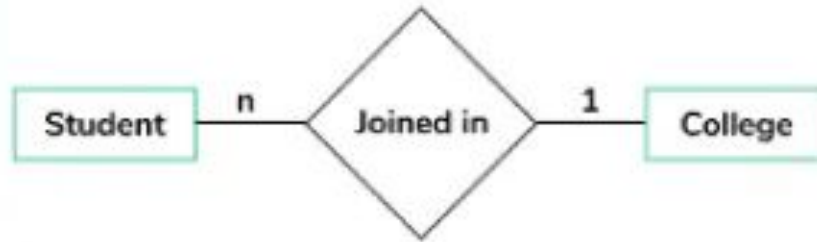
Many to one relationship

Example Scenario:

Cohort: Cohort 15

Students: Pharez, Daguy, Hilson

Cohort: Cohort 12

Students: Chidinma, NkemGod

In this example:

Cohort 15 can have multiple students **(Pharez, Daguy, Hilson).**

Cohort 12 can have multiple students (Chidinma, NkemGod).

Each **student** is associated with exactly **one cohort**,

A many-to-one relationship in database design refers to a scenario where many column in database table are associated with a single column of another table

Example Using Cohort and Student:

let's define a many-to-one relationship between a cohort (group of students)

Cohort:

Represents a group of students enrolled in the same course or program.

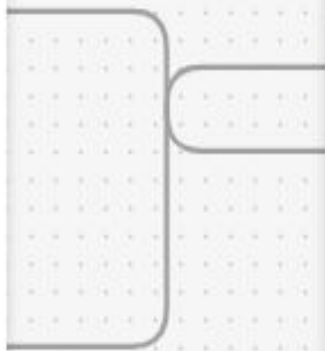**Student**

Represents an individual enrolled in a cohort.

One to one relationship

One-to-one relationship

HUSBAND | WIFE
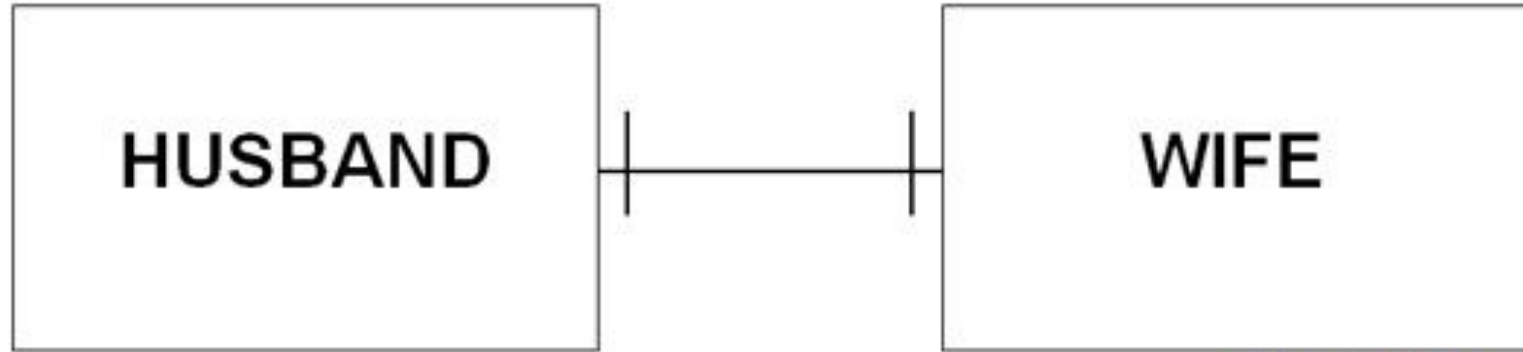
A one-to-one relationship in database design is a type of association between two tables in which each record in one table is linked to exactly one record in another table, and vice versa. This relationship is established by linking the primary key of one table to the primary key of another table using a foreign key constraint.

It's a relationship where a record in one table is associated with exactly one record in another entity table.

let's define a many-to-one relationship between a cohort (group of students)

User:

Represents a user on the Facebook platform.

Profile:

Represents additional profile information associated with each user.

Every Facebook user has a unique profile that contains additional information such as bio and location.

The one-to-one relationship ensures that each user's profile details are directly linked and accessible from the user entity.

let's define a many-to-one relationship between a cohort (group of students)

User:

Represents a user on the Facebook platform.

Profile:

Represents additional profile information associated with each user.

Every Facebook user has a unique profile that contains additional information such as bio and location.

The one-to-one relationship ensures that each user's profile details are directly linked and accessible from the user entity.

Students table

Student ID	12345
Last Name	Tang
First Name	Sophie

Contact Info table

Student ID	12345
City	New York
Phone	408-555-3456

One to many relationship

product
id : BigAutoField PK
is_active: BooleanField
slug: SlugField
category: M2M
name : CharField
description : TextField

product_category
id
product_id FK
category_id FK
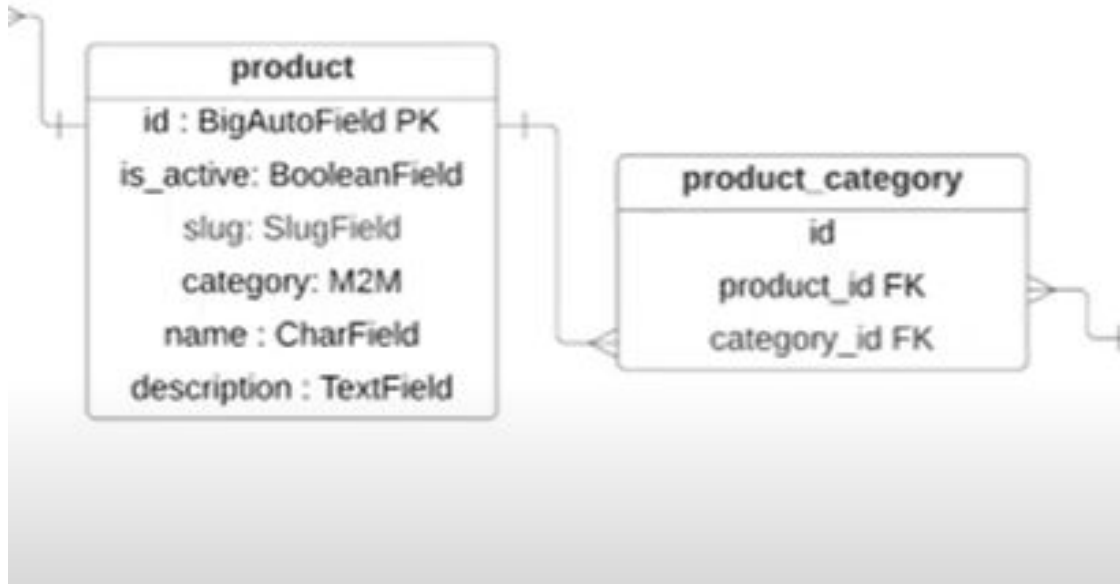
In a many-to-many relationship, records in Table A can have multiple matching records in Table B, a

This type of relationship requires a **junction table** (also known as a bridge or linking table) to manage the relationships.

example of a many-to-many relationship example: student datas and courses. In a school database, a student can enroll in multiple courses, and each course can have multiple students.

Data modeling and schema design,

Referential integrity:

this ensures that the fields in the table from the foreign key correspond to fields in the table from the primary key. If the foreign key does not correspond to an appropriate primary key or the value is null, the referential integrity is violated.

Data modeling and schema design,

Null vs Blank in Django Models

Attribute	Level	True	False (Default)	Common Use Case
null	Database Layer	Allows NULL values in the database	Disallows NULL, uses default or raises IntegrityError	Optional non-string fields (e.g., ForeignKey, DateTimeField)
blank	Form Validation	Field can be left blank in forms	Field is required in forms	Optional fields in forms (e.g., admin or custom forms)

https://fabric.inc/wp-content/uploads/hubspot/table-relationships-ecommerce-database.png