WEEK 3

THE DATA ARCHIVES

Relational Database Fundamentals

PRESENTED BY **FILSAN MUSA & FADUMO DIRIYE**

TABLE OF CONTENT

01 INSIDE A RELATIONAL DATABASE

02 TABULAR DATA STORAGE

03 RELATIONSHIPS IN SQL

04 PRIMARY AND FOREIGN KEYS

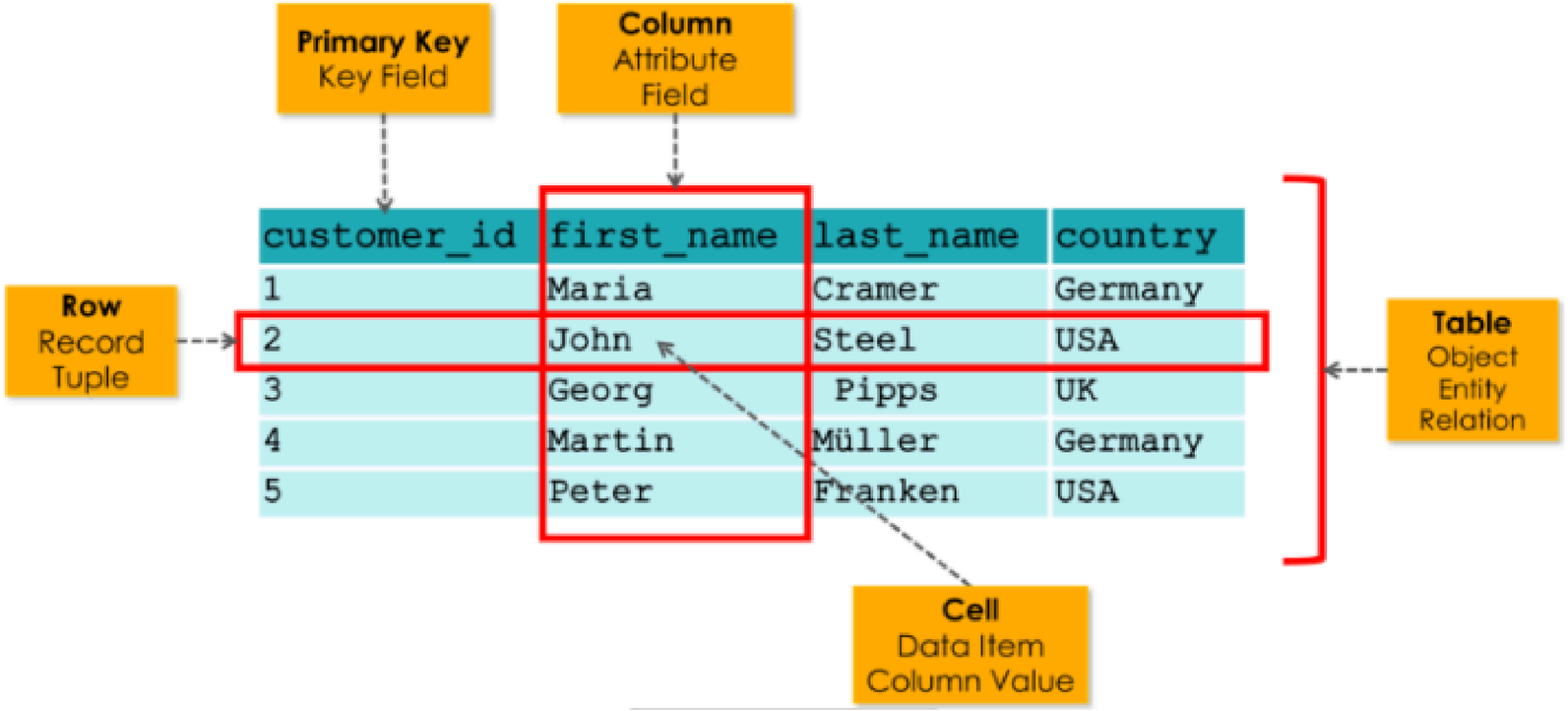
05 UNDERSTANDING SCHEMAS AND ERDS

Inside a Relational Database

A relational database is typically composed of multiple tables, each storing information about a unique entity. To understand what an entity is, let’s imagine that we are building a relational database for an e-commerce website, we might want a table to store product information, and another to store customer information, and perhaps another to store order details and so on. In this instance, each of these tables would be considered a separate entity, as they each store information about different facets of the e-commerce business.

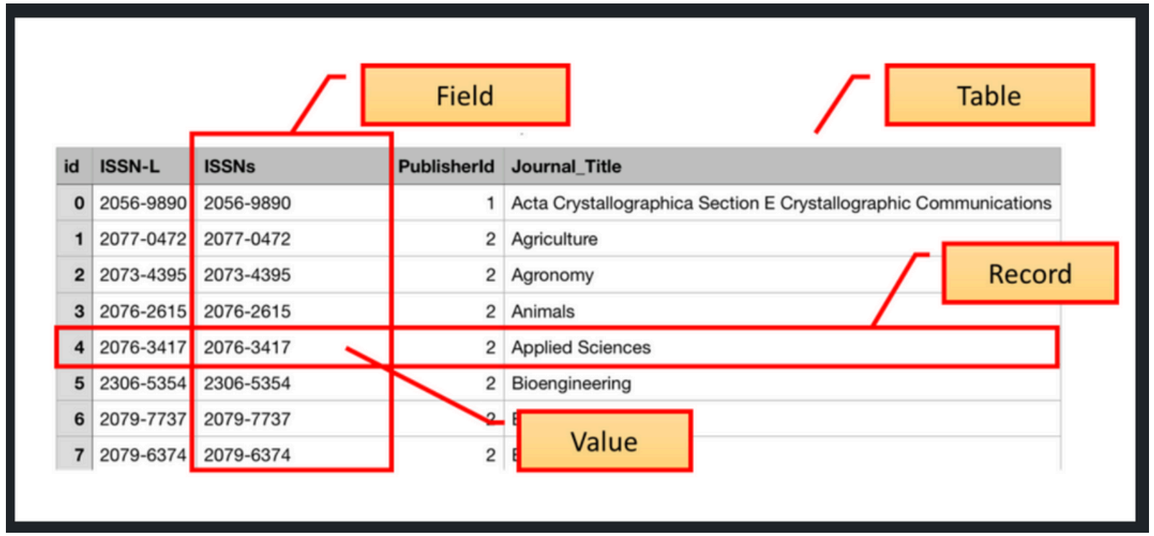
Inside a Relational Database (Cont.)

When thinking about building or even understanding a database, it is helpful to think of a table as an entity. In the case of building a database maintaining this perspective helps prevent data duplication and reduces inefficiencies. A table is also sometimes referred to a relation.



Tabular Data Storage

A table, sometimes referred to as a relation or even an entity, consists of rows, columns, and entries.



id	ISSN-L	ISSNs	PublisherId	Journal_Title
0	2056-9890	2056-9890	1	Acta Crystallographica Section E Crystallographic Communications
1	2077-0472	2077-0472	2	Agriculture
2	2073-4395	2073-4395	2	Agronomy
3	2076-2615	2076-2615	2	Animals
4	2076-3417	2076-3417	2	Applied Sciences
5	2306-5354	2306-5354	2	Bioengineering
6	2079-7737	2079-7737	2	
7	2079-6374	2079-6374	2	

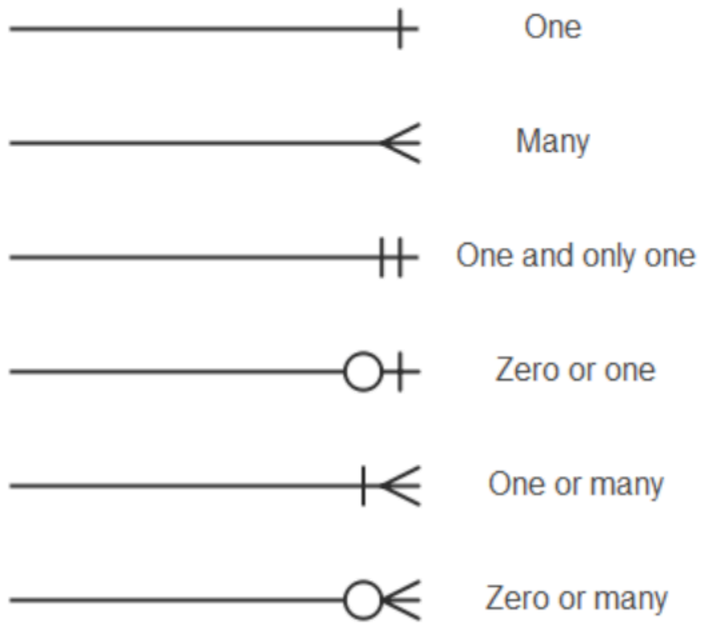
Row (Record): Represents an individual data entry

Column (Attribute/Field): Represents a specific attribute or characteristic, similar to a variable in programming languages

Value (Entry/Field Value): A single piece of information, a datum, found at the intersection of a row and column

Relationships in SQL

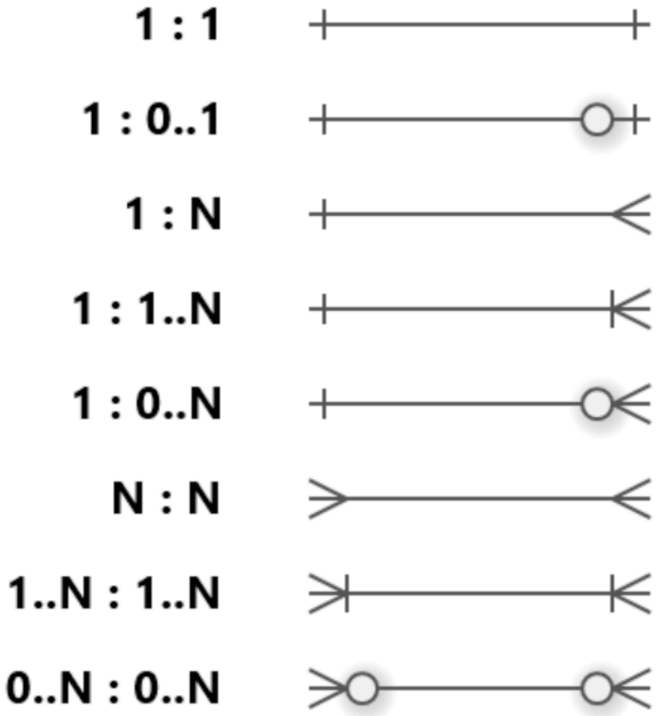
There are many types of relationships that can exist in a relational database. However, the three fundamental ones are: one-to-one (1:1), one-to-many (1:N), and many-to-many (N:N).



1:1 - Each row in table A is related to only one row in table B, and vice versa.

- Examples: Each person has one social insurance number (SIN); Each email can be registered to only one account. (opposite is also true)

Relationships in SQL (Cont.)



1:N - Each row in table A is related to multiple rows in table B, but not the other way around.

- Example: Each customer can make multiple purchases, but each purchase goes back to only one customer

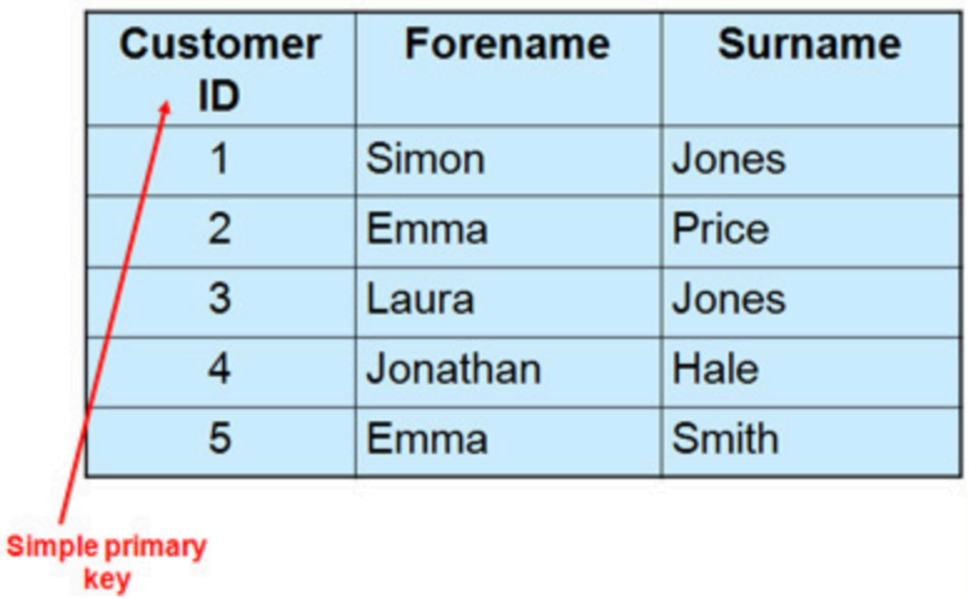
N:N - Each row in table A is related to multiple rows in table B, and each row in table B is a related to multiple rows in table A.

- Example: Users on a social media platform may have multiple followers, and may also be following multiple accounts

Primary & Foreign Keys

What is a Primary Key?

A primary key (PK), whis is typically the first column in a table, is a unique identifier for each record in a table. There are two main types of primary keys: natural keys, and surrogate key.



Natural Key: Derived from existing data

- Examples: SKU, Social Insurance Number, ISBN etc.

Surrogate Key: Artificially generated unique identifiers

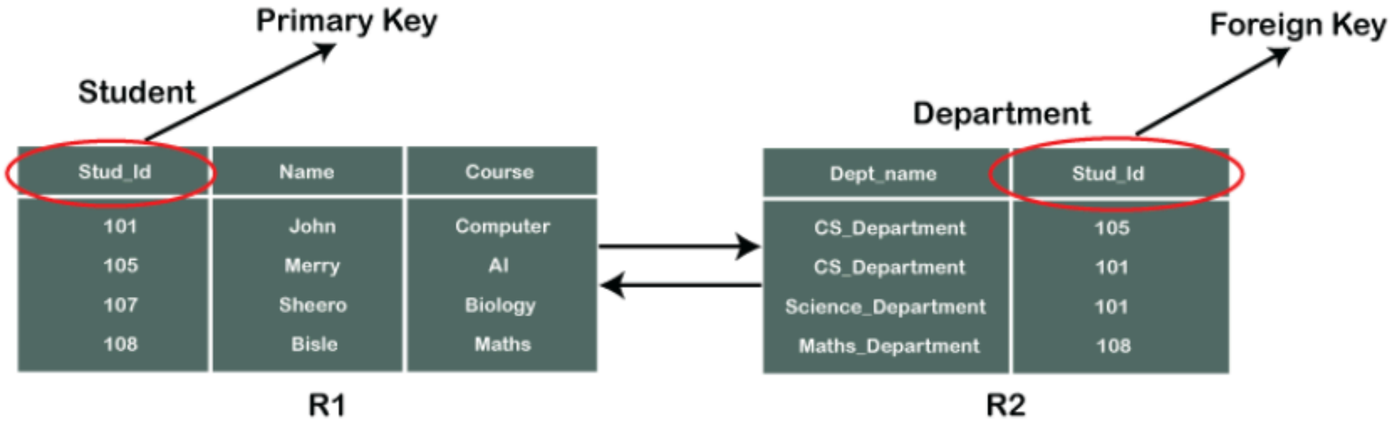
- Examples: Order Number, Employee ID etc.

Primary & Foreign Keys (Cont.)

What is a Foreign Key?

A foreign key (FK) establishes relationships between tables by referencing the primary key in the related table.

Examples: In the image below, the “Department” table has “Stud_Id” as a FK which is linked to “Stud_Id” in the “Student” table.



A column or set of columns that links one table to another.

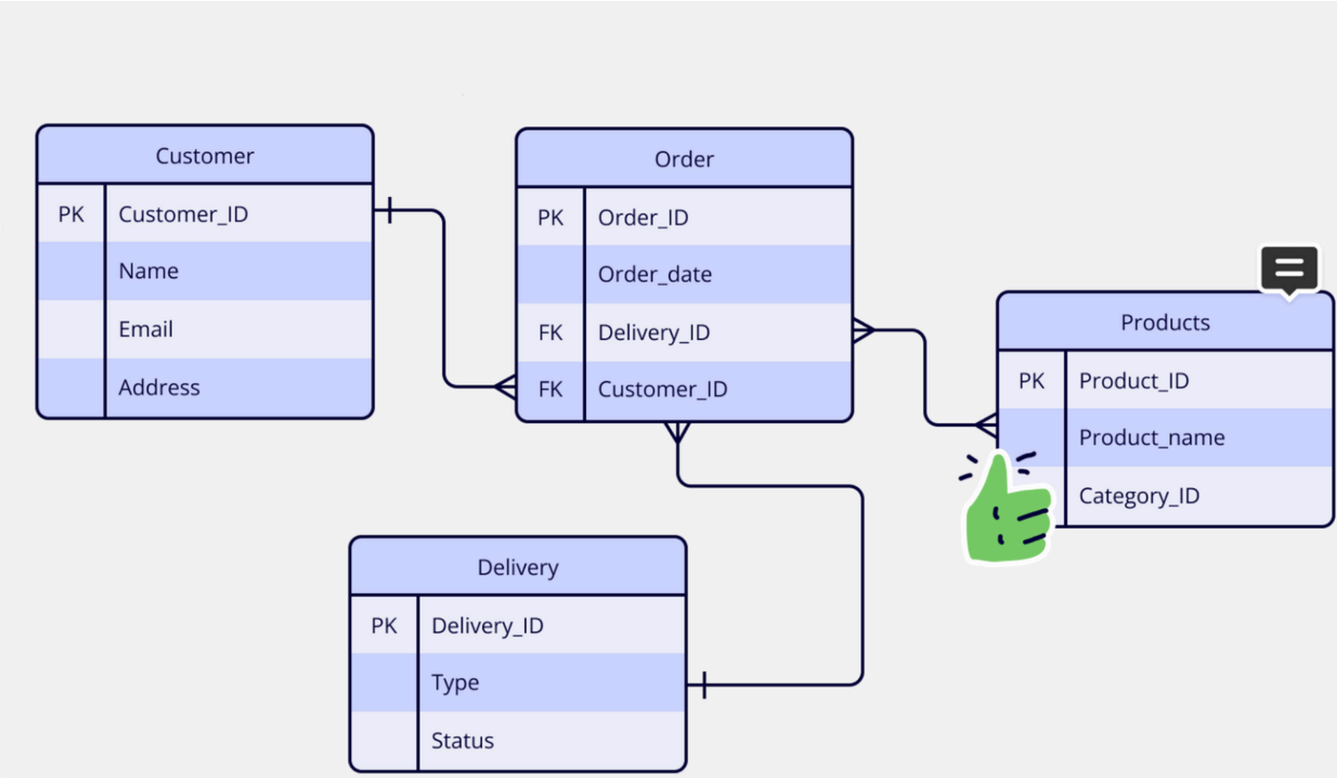
Understanding Schemas & ERD

Schema:

A schema is a blueprint of a relational database and its contents. It captures the relationships between entities (or tables).

Entity-Relationship Diagram (ERD):

An Entity-Relationship Diagram, or an ERD is a visual representation of a schema



Next Steps

Homework:

- Try understanding all the terminology covered
- Complete Week 3 Quiz

Next Topic:

- Defining and Identifying Data Types