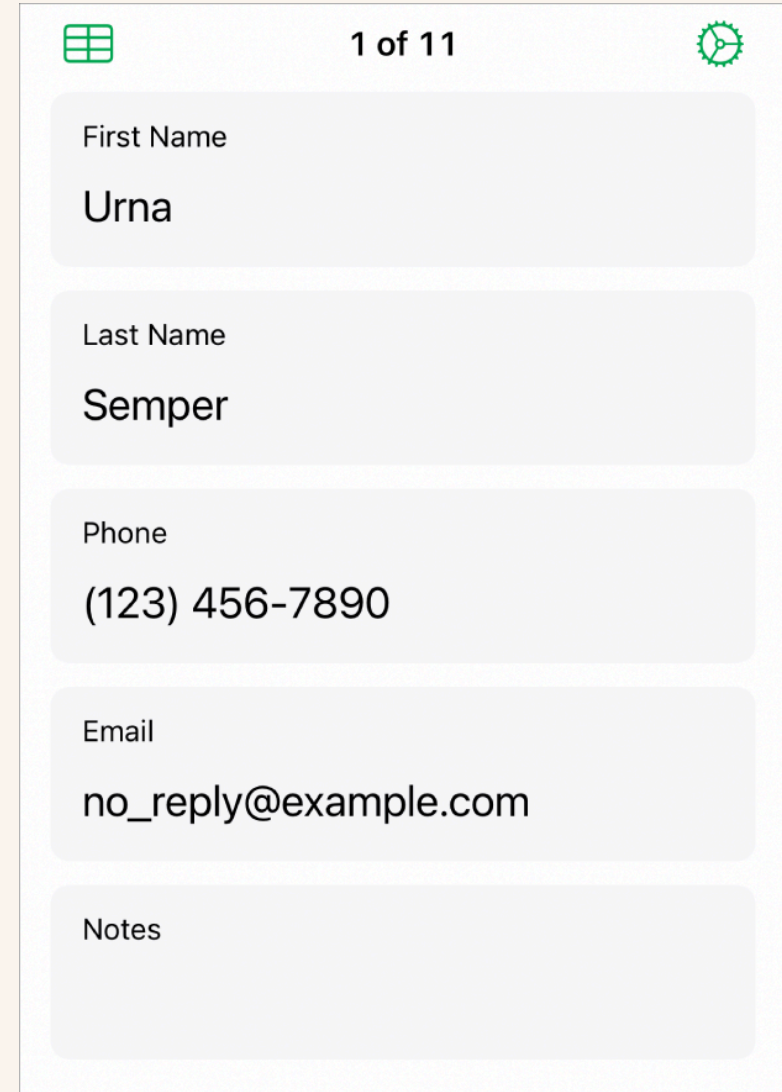


Intro to Database and SQL

What Is a Database?

A structured collection of data that's easy to store, update, and search.



First Name	Urna
Last Name	Semper
Phone	(123) 456-7890
Email	no_reply@example.com
Notes	

Why Do We Need Databases?

Scenario: Managing Sales Data in a Growing Company

- You're running a small business with 5 product lines.
- Each month, every branch sends you its sales report as an **Excel file**.
- After one year, you now have **60 spreadsheets** from 12 branches.

"Which customer bought Product A on January?"

To answer this, you'd need to:

- Open and merge dozens of spreadsheets manually.
- Check for inconsistent column names and duplicate records.
- Risk errors whenever someone updates an old file.

Databases to the Rescue

- Store all sales records in **one structured place**.
- Query data instantly using SQL:

```
SELECT customer_name  
FROM sales  
WHERE product = 'A' AND month = 'January';
```

Why Do We Need Databases?

Database =

 **organized data** +  **software that manages it (DBMS)**

- **Grows with you (Scalable):**
Start small on a laptop → scale to millions of users.
- **Works for everyone (Shared access):**
Multiple people or apps can use the same data safely.
- **Keeps data clean and accurate:**
Built-in rules prevent errors and duplicates.
- **Fast answers:**
Query huge data sets in seconds instead of hours.

Two Key Concepts to Understand Databases

- **Entity Relationship Diagram (ERD)**
- **Structured Query Language (SQL)**

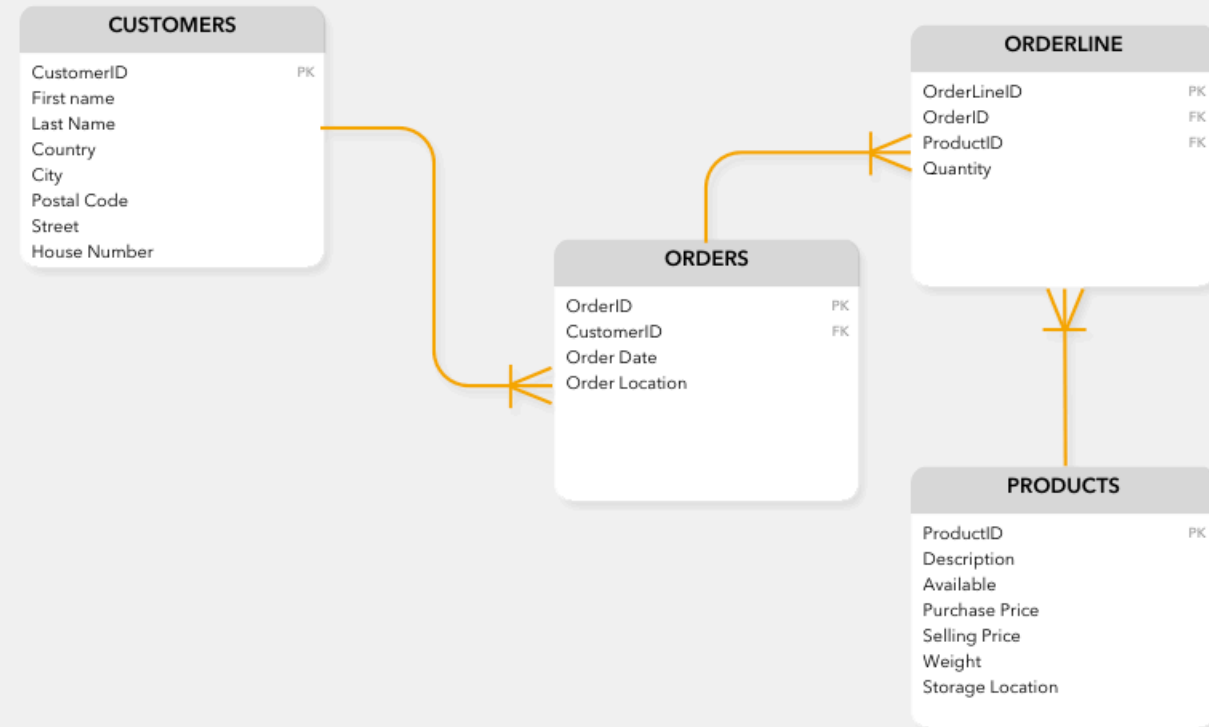
Entity Relationship Diagram (ERD)

A **blueprint** of a database—it shows how data is organized and connected

- **Entity (noun):** Something we store data about
→ e.g., `Student`, `Course`, `Enrollment`
- **Relationship (verb):** How entities are linked
→ e.g., `Student` "enrolls in" `Course`

How Relationships Work in ERD

- Relationships are defined using **keys**
- **Primary Key (PK)**
Unique ID for each record in an entity
- **Foreign Key (FK)**
A field in one entity that links to the primary key of another entity



Structured Query Language (SQL)

To interact with databases



```
SELECT [ALL/DISTINCT] column_list
FROM table_list
[WHERE conditional expression]
[GROUP BY group_by_column_list]
[HAVING conditional expression]
[ORDER BY order_by_column_list]
```

Copyright ©2014 Pearson Education

Essentials of Database Management, 1st ed.,
Pearson.

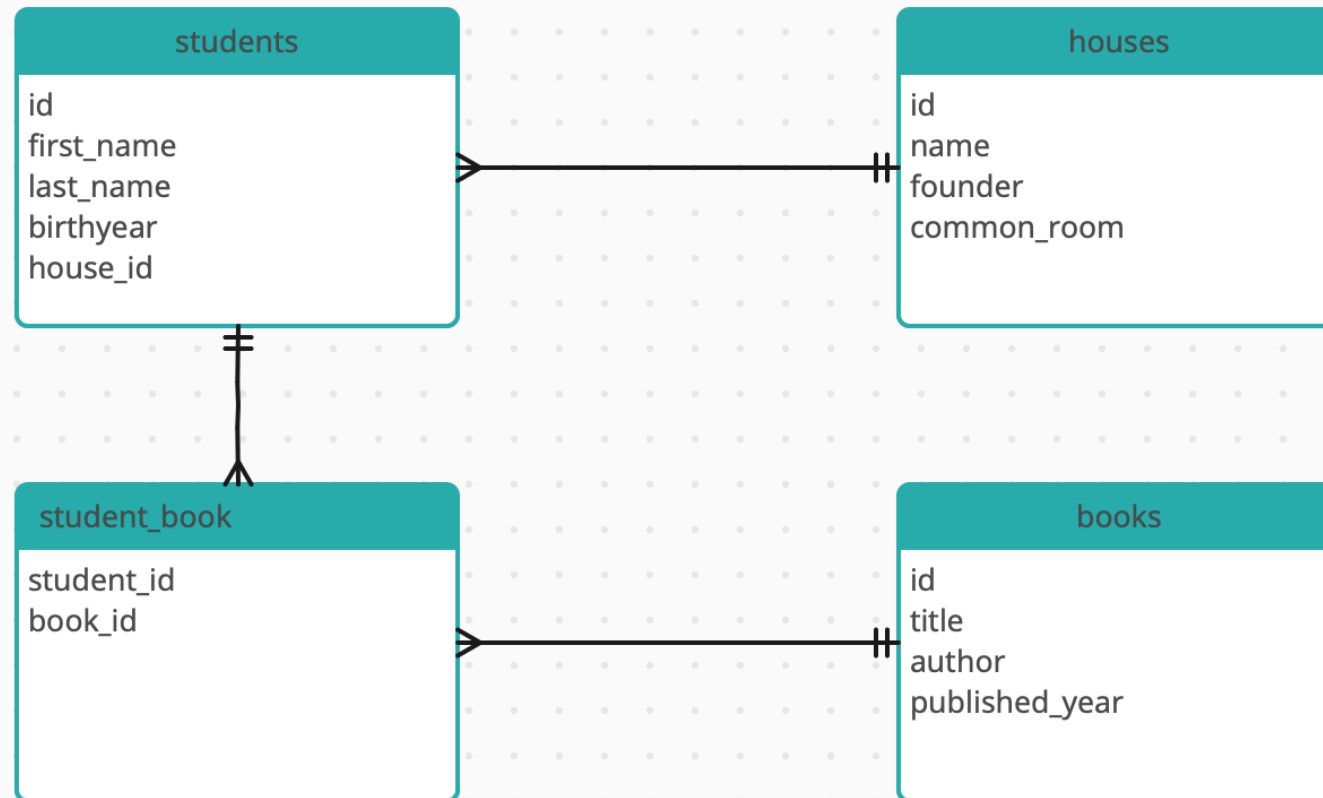
How Do We Build a Database?

1 What do you want to store data about?
(→ Entities)

2 How are they related to each other?
(→ Relationships)



Harry Potter Database - ERD



SELECT columns **FROM** a table

SELECT `column_name` **FROM** `table_name`

```
-- select first_name column from students table  
SELECT first_name FROM students;
```

```
-- select first_name and last_name columns from students table  
SELECT first_name, last_name FROM students;
```

```
-- select all columns from students table  
SELECT * FROM students;
```

LIMIT the number of records returned

```
SELECT column_name FROM table_name LIMIT number
```

```
SELECT * FROM students LIMIT 10;
```

-- is used to add comments in SQL

dash dash space **--**, not *dash dash* **--**

; is used to indicate the end of a SQL statement

Optional if you have only one statement.

SELECT columns **FROM** a table **WHERE** conditions
are true

SELECT `column_name` **FROM** `table_name` **WHERE** `condition`

```
-- select first_name and last_name of all students in house_id = 1  
SELECT first_name, last_name FROM students WHERE house_id = 1;
```

```
-- select first_name and last_name of all students born in 1980  
SELECT first_name, last_name FROM students WHERE birthyear = 1980;
```

```
-- select first_name and last_name of all students born in or after 1980  
SELECT first_name, last_name FROM students WHERE birthyear >= 1980;
```




Write SQL queries to answer the following questions

1. What year was Harry Potter born?
2. What is the name of the student who was born in 1980?
3. Who is the founder of Gryffindor?