

---

# Databases



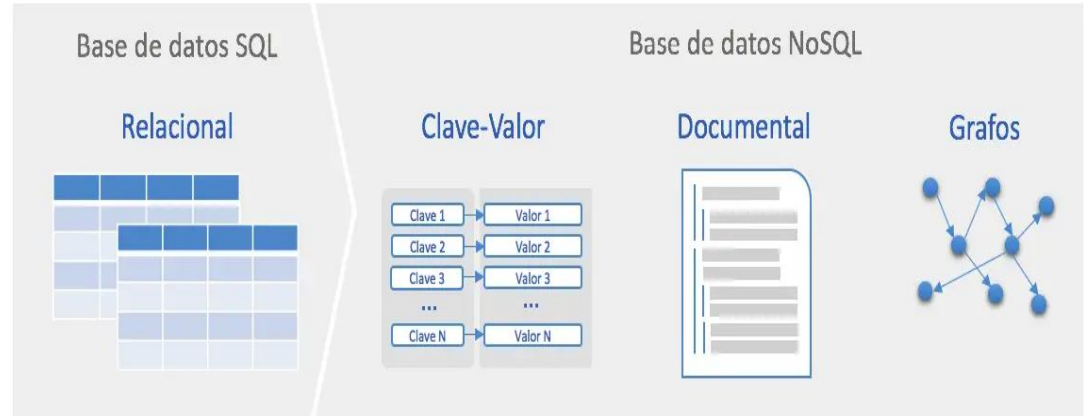
# Objectives

- Use a local SQL instance
- Normalize a database
- Create a database schema
- Articulate the purpose of foreign keys and primary keys

# Databases - Introduction

## Conceptos básicos

- Bases de datos
- Bases de datos relaciones (SQL)
  - MySQL
  - PostgreSQL
  - Oracle
  - SQL Server
- Bases de datos no relacionales (noSQL)
  - Redis
  - Cassandra
  - MongoDB
  - Neo4J



# Relational table

## Relation (Table)

Row/Tuple/Record

Column/Attribute/Field

name	birth	gpa	grad
Anderson	1987-10-22	3.9	2009
Jones	1990-4-16	2.4	2012
Hernandez	1989-8-12	3.1	2011
Chen	1990-2-4	3.2	2011

Column Types

VARCHAR(30)      DATE      FLOAT      INT

# Prerequisites

## Setup

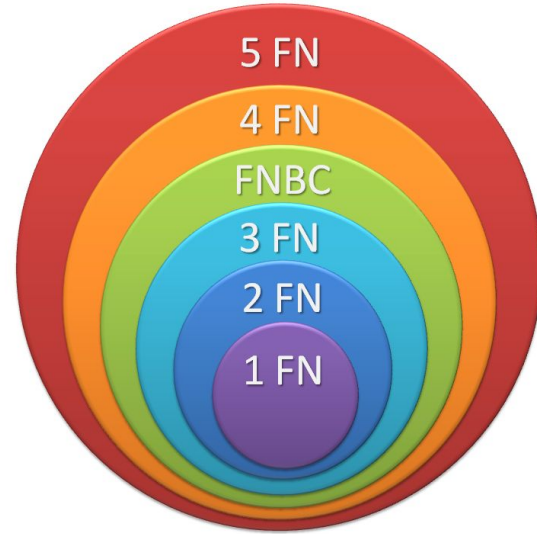
- MySQL server installed
- Workbench installed



# Normalization

La **normalización** es una técnica de diseño de base de datos que organiza las tablas de una manera que reduce la redundancia y la dependencia de los datos.

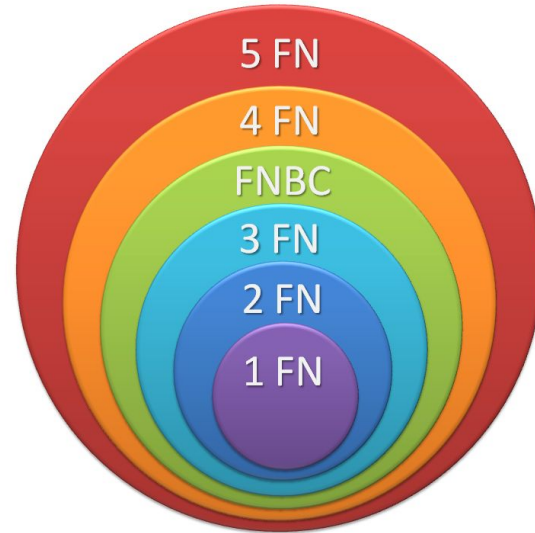
Divide las tablas más grandes en tablas más pequeñas y las vincula usando relaciones.



# Database Normalization

## Highlights

- No existen entradas duplicadas
- Debe haber una clave primaria
- Ningún campo debe depender de una clave no primaria



## Normalization - Examples

Student Name	Course Code	Course Name	Room Number	Professor	Grade
Maya Charlotte	CS101	Intro to Java	1802	Balderez	98
James Fields	CS101	Intro to Java	1802	Balderez	82
Michael Alcocer	CS101	Intro to Java	1650	Su	65
Maya Charlotte	CS103	Databases	1200	Rojas	89
Michael Alcocer	CS101	Intro to Java	1650	Su	65
James Fields	CS101	Intro to Java	1802	Balderez	82



## Normalization - Eliminar entradas repetidas

Student Name	Course Code	Course Name	Room Number	Professor	Grade
Maya Charlotte	CS101	Intro to Java	1802	Balderez	98
James Fields	CS101	Intro to Java	1802	Balderez	82
Michael Alcocer	CS101	Intro to Java	1650	Su	65
Maya Charlotte	CS103	Databases	1200	Rojas	89

## Normalization - Agregar clave primaria

id	Student Name	Course Code	Course Name	Room Number	Professor	Grade
0	Maya Charlotte	CS101	Intro to Java	1802	Balderez	98
1	James Fields	CS101	Intro to Java	1802	Balderez	82
2	Michael Alcocer	CS101	Intro to Java	1650	Su	65
3	Maya Charlotte	CS103	Databases	1200	Rojas	89

## Normalization - Mover relaciones no primarias

id	Student Name	Course Code	Room Number	Professor	Grade
0	Maya Charlotte	CS101	1802	Balderez	98
1	James Fields	CS101	1802	Balderez	82
2	Michael Alcocer	CS101	1650	Su	65
3	Maya Charlotte	CS103	1200	Rojas	89

Course Code	Course Name
CS101	Intro to Java
CS103	Databases

# Normalization - Mover relaciones no primarias

id	Student Name	Course id	Grade
0	Maya Charlotte	CS101-A	98
1	James Fields	CS101-A	82
2	Michael Alcocer	CS101-B	65
3	Maya Charlotte	CS103-A	89

id	Course Code	Room Number	Professor
CS101-A	CS101	1802	Balderez
CS101-B	CS101	1650	Su
CS103-A	CS103	1200	Rojas

Course Code	Course Name
CS101	Intro to Java
CS103	Databases

## Normalization - Exercise

Customer Name	Address	email	product purchased	price	seller	quantity ordered
Agustine Riviera	123 SunnyDale	<a href="mailto:arivi5@sample.com">arivi5@sample.com</a>	Electric Toothbrush	15.15	Pitz	5
Agustine Riviera	123 SunnyDale	<a href="mailto:arivi5@sample.com">arivi5@sample.com</a>	Pencil	3.12	Pitz	13
Agustine Riviera	422 Cruz Verde	<a href="mailto:agus.tino.riviera@example.com">agus.tino.riviera@example.com</a>	Umbrella	20.17	Hardy	1
Rebecca Tonna	5 Real Ortega	<a href="mailto:happyrio@example.com">happyrio@example.com</a>	Pencil	3.12	Pitz	20
Sonia Riviera	123 SunnyDale	<a href="mailto:sonia.riviera@sample.com">sonia.riviera@sample.com</a>	Cellphone	512.00	Orange	1
Rebecca Tonna	5 Real Ortega	<a href="mailto:happyrio@example.com">happyrio@example.com</a>	Cellphone	512.00	Orange	1
Jaime Torres	15 Sunnydale	<a href="mailto:prof.torres@sample.edu">prof.torres@sample.edu</a>	Camera	327.99	Orange	1



# Aggregate functions

## Highlights

- COUNT
- AVG
- SUM
- MIN/MAX
- GROUP BY
- HAVING

# Aggregate functions - Examples

## Examples

- Cantidad de notas por sección
- Promedio de notas por sección
- Promedio de notas por sección entre las secciones cuyo promedio sea sobresaliente (promedio > 80)
- Suma de las notas, cantidad de notas y promedio de las notas entre los alumnos de la sección CS101-A



# Intermediate Querying

## Highlights

- `LIKE`
- `BETWEEN`
- `AND/OR`
- `ORDER BY`
- `DISTINCT`

# Intermediate Querying - Examples

## Examples

- `Notas de secciones que comiencen por CS101`
- `Notas con puntajes entre 85 y 95`
- `Notas de estudiantes cuyos nombres comiencen por M o que su puntaje sea mayor a 95`
- `Notas ordenadas por score de manera descendente`
- `Nombres de los estudiantes`

# Exercises

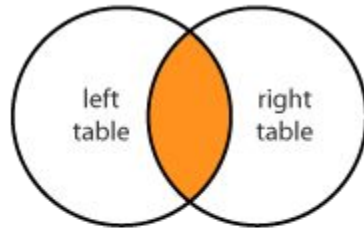
## Exercises

- Promedio de notas por estudiante
- Total de cursos tomado por cada estudiante
- La nota más baja por sección
- La nota más alta por sección
- Nombre y nota de todos los estudiantes de la sección CS103 ordenados por nota de mayor a menor
- Lista alfabética de los distintos estudiantes cuyos nombres caigan alfabéticamente desde la L hasta la R

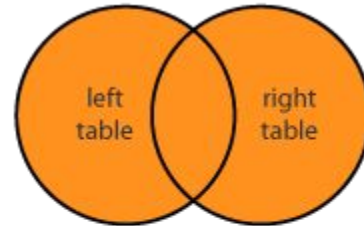


# Joins

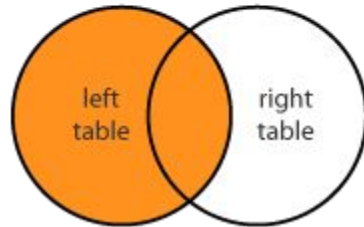
INNER JOIN



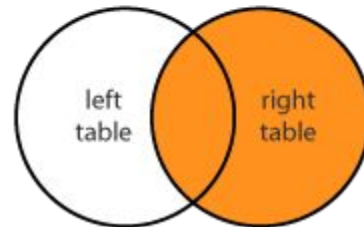
FULL JOIN



LEFT JOIN



RIGHT JOIN



# Joins

employee

id	name	department_id
1	Rafferty	31
2	Jones	33
3	Heisenberg	33
4	Robinson	34
5	Smith	34
6	Williams	NULL

department

id	name
31	Sales
33	Engineering
34	Clerical
35	Marketing

# Join - Exercises

## Exercises

- Nombre, nota y nombre del instructor de todos los estudiantes
- Nombre, nota y nombre del curso de todos los estudiantes que hayan estudiado en un curso llamado "Databases"