

# INTRODUCCIÓN A SPRING DATA JPA

# PERSISTENCIA DE DATOS

- Necesidad de almacenar datos.
  - Legal
  - Relación comercial
  - Explotación
- **BASES DE DATOS**

# BASES DE DATOS

## SQL Vs NoSQL: Data Structures

### SQL Database

Relational



Analytical (OLAP)

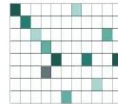


### NoSQL Database

Key - Value



Column - Family



Graph



Document



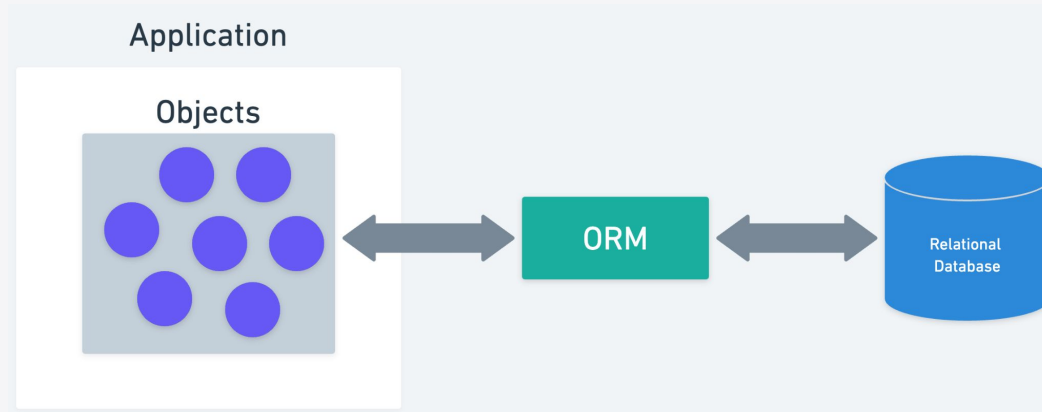
# DESFASE OBJETO-RELACIONAL

```
PreparedStatement ps = con
    .prepareStatement("SELECT * FROM empleados
WHERE id = ?");
ps.setInt(1, id);
ResultSet rs = ps.executeQuery();
Empleado result = null;
if (rs.next()) {
    result = new Empleado(rs.getInt("id"),
        rs.getString("nombre"),
        rs.getString("apellidos")

        rs.getDate("fecha_nacimiento").toLocalDate(),
        rs.getFloat("sueldo"));
}
rs.close();
ps.close();
return result;
```

# SOLUCIÓN DESFASE OBJETO-RELACIONAL

- ORM: *Object-Relational Mapping*



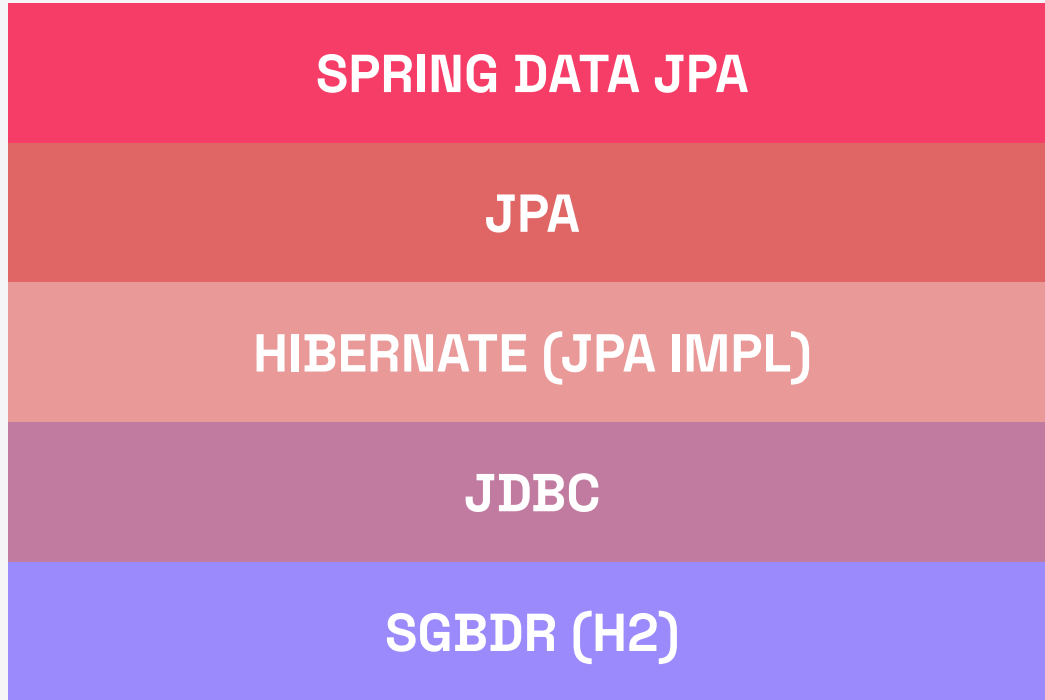
# ORMs PARA JAVA



# LA BATALLA DE LAS SIGLAS

- JDBC: Java Database Connectivity
- JPA: Java Persistence API
- SGBDR: Sistema Gestor de Bases de Datos Relacional
- H2: Base de datos

# STACK





# JPA

- Es una *estandarización* de la persistencia en Java
- Sin implementación concreta
  - Proveedor: Hibernate, EclipseLink, OpenJPA ...

# SPRING DATA

- Proyecto paraguas
- Acceso a orígenes de datos: SQL, NoSQL, ...
- Modelo consistente

# SPRING DATA JPA

- Aprovecha lo mejor de JPA
- Elimina código repetitivo
- API sencilla para acceder a los datos
- Una única dependencia *starter*

# CARACTERÍSTICAS SPRING DATA JPA

- Repositorios basados en interfaces
- Consultas derivadas del nombre de un método
- Consultas JPQL o SQL nativo

# CARACTERÍSTICAS SPRING DATA JPA

- Otras estrategias de consulta
- Paginación y ordenación
- Integración con Spring Boot
- Gestión automática de transacciones

# REPASAMOS NUESTRO STACK

