



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

AYUDANTÍA 6

SQL: Bases de datos relacionales

Ayudante: Felipe Fuentes Porras

BASES DE DATOS RELACIONALES

Es una colección de datos. La idea es tener estos datos ordenados y almacenados óptimamente para aprovechar el contenido de esto. La información es muy valorada, y es importante saber almacenarla y extraerla.



RELACIÓN E INSTANCIA

- La relación contiene un esquema de la relación e instancias de esta.
- Un ejemplo de esquema seria:
 - Jugador(id INTEGER, nombre VARCHAR(20), vida INTEGER)
- Instancias serian un conjunto de tuplas que cumplen con los criterios del esquema, formando una tabla.

RESTRICCIONES

- Al crear el esquema, uno puede definir ciertas restricciones a los valores. Esto se hace para mantener la integridad de los datos.
- Ejemplos:
 - Uniques
 - Especificar un rango de valores
 - Especificar un tipo de dato

PRIMARY KEY

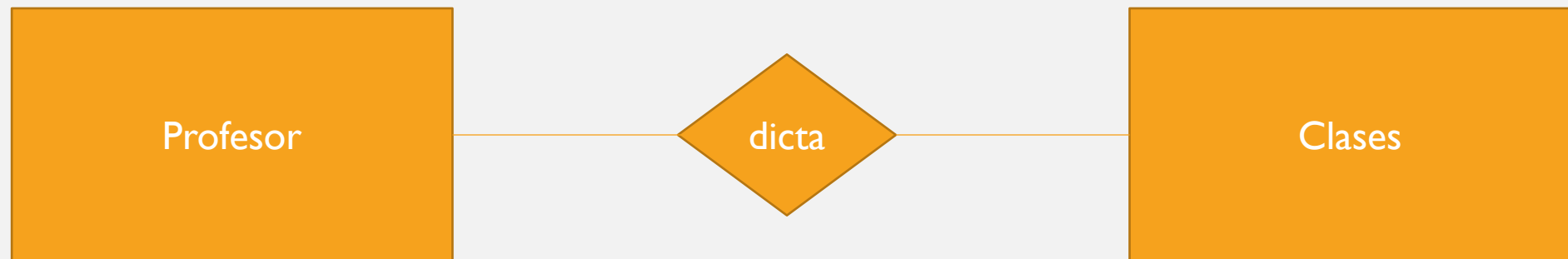
- Las llaves son un identificador único para cada tupla dentro de una tabla.
- En una tabla pueden haber varios candidatos a llaves.
- Las llaves no necesariamente son una única columna, una llave puede representar varias columnas de una tabla
- Un subconjunto de una tupla que contenga a una llave se denomina superllave.

FOREIGN KEY

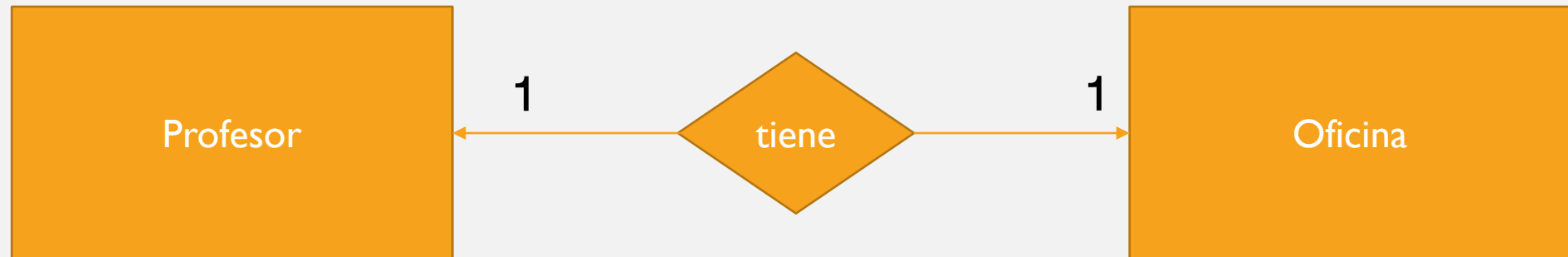
- Muchas veces vamos a querer hacer referencias a otras relaciones (tablas).
- La forma correcta de hacer esto es por medio de foreign keys
- Una foreign key es una referencia a una primary key de otra tabla a la que se le quiere hacer referencia.
- Una llave foránea puede tener valor nulo, no así una llave primaria.

MODELO ENTIDAD RELACIÓN

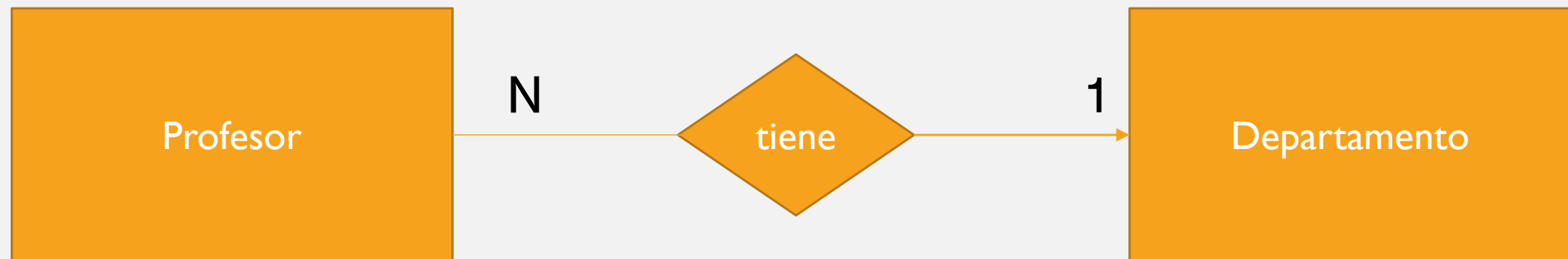
Forma para representar las relaciones entre entidades/tablas



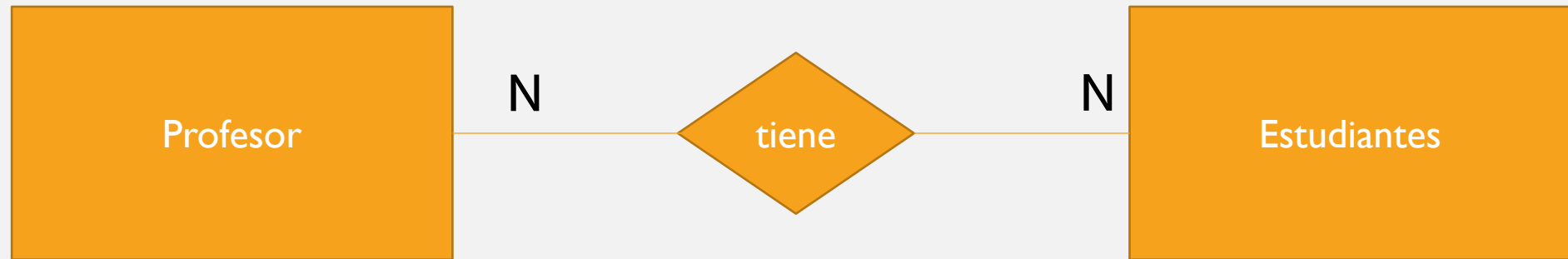
CARDINALIDAD: 1-1



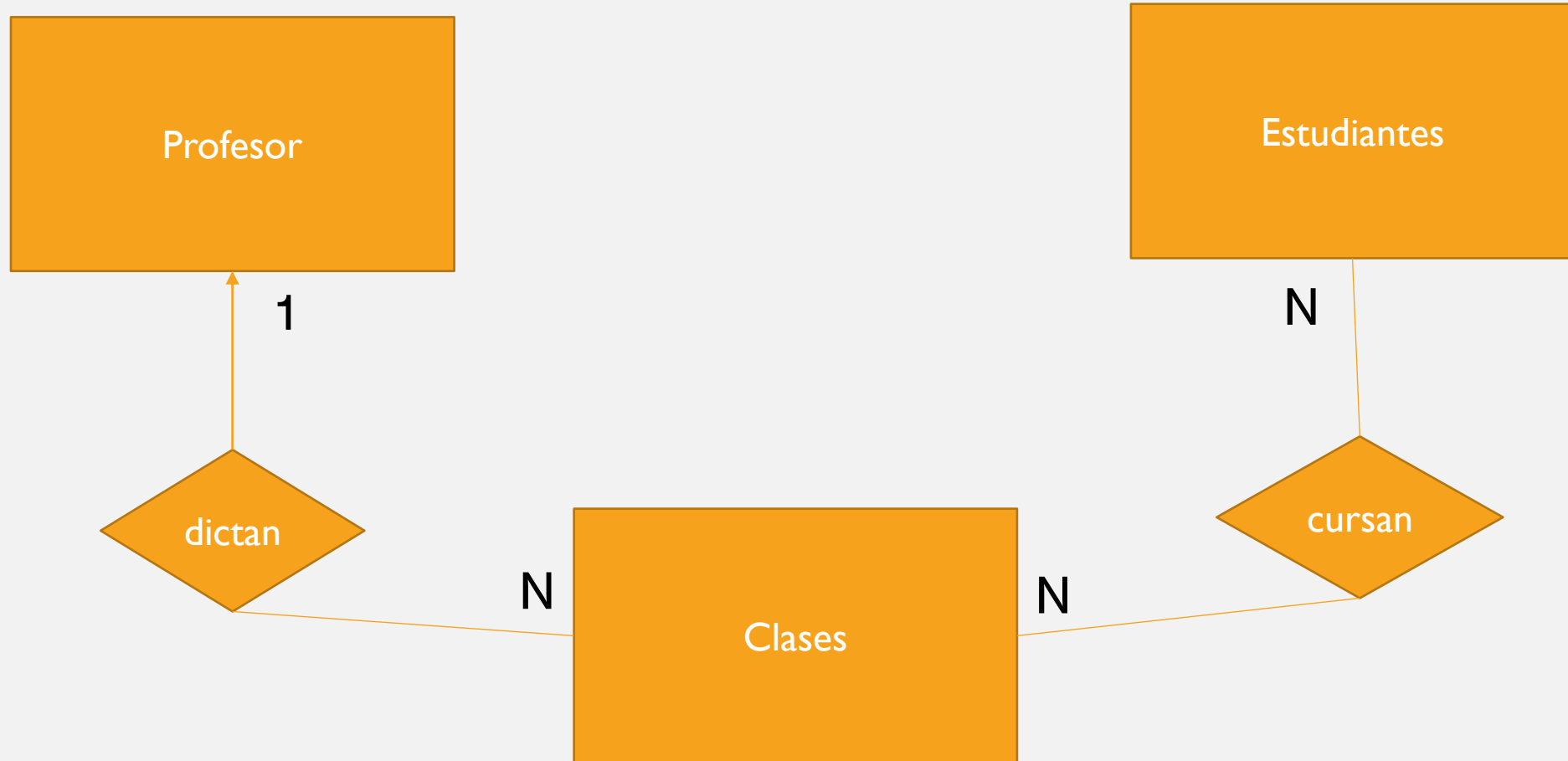
CARDINALIDAD: 1-N



CARDINALIDAD: 1-N



EJEMPLO:



SQLITE

- Nos conectamos a la base de datos.

```
import sqlite3  # importamos el módulo de sqlite  
connection = sqlite3.connect('futbol.db')
```

- Tenemos que inicializar un cursor, el cual nos va a permitir acceder a las tablas

```
cursor = connection.cursor()
```

EJECUTAR ACCIONES EN LA BASE DE DATOS

- Para interactuar con la base de datos se debe utilizar execute y describir la acción a realizar:

```
cursor.execute("CREATE TABLE Equipos(eid INTEGER PRIMARY KEY, nombre TEXT UNIQUE)")
```

- Commit(): Es lo que hará los cambios en la base de datos (ejecuta las acciones)
- Close(): Cerramos la conexión con la base de datos.

```
connection.commit()  
connection.close()
```

CREACIÓN Y MODIFICACIÓN DE RELACIONES

- Create: “CREATE TABLE Equipos(eid INTEGER PRIMARY KEY, nombre TEXT UNIQUE)”
- Drop: “DROP TABLE Equipos”
- Alter: “ALTER TABLE Estudiantes ADD COLUMN carrera TEXT”
- Update: “UPDATE Estudiantes E SET E.edad = E.edad + 1, E.promedio = E.promedio - 1 WHERE E.id = 53688”
- Insert: “INSERT INTO Estudiantes (id, nombre, login, edad, promedio) VALUES (53688, 'Hernán Valdivieso', 'Hfvaldivieso', 20, 5.5)”
- Delete: “DELETE FROM Estudiantes E WHERE E.nombre = 'Hernán’”

TIPOS DE DATOS

Revisen la documentación de los tipos de datos de sqlite:

- TEXT
- INTEGER
- REAL
- DATE

LLAVES

- Llave primaria:

```
cursor.execute("CREATE TABLE Equipos(eid INTEGER PRIMARY KEY, nombre TEXT UNIQUE)")
```

- Llave secundaria:

```
FOREIGN KEY (person_id) REFERENCES Persons
```