

Manejo de Bases de Datos

Hector Florez

Agenda

1. Bases de Datos Relacionales
2. Diseño de Bases de Datos
3. Bases de Datos y Aplicaciones
4. Almacenamiento de datos, transacciones y bases de datos NoSQL

Bases de Datos Relacionales

Introducción

Introducción

- **Base de datos:**
 - Colección de datos almacenados en archivos que contiene información relevante de un negocio.
 - Permite manejar grandes cantidades de información.
- **Sistema manejador de bases de datos (DBMS):**
 - Colección de programas que permite acceder a los datos almacenados en una base de datos.
- **Aplicaciones transaccionales - Aplicaciones de bases de datos - Sistemas de información.**
 - Aplicaciones en contextos como: comercio, banca, academico, etc.

Introducción

- **Propósitos de las base de datos:**
 - Mantener la información organizada
 - Asegurar las siguientes características:
 - Consistencia
 - Facilidad de acceso
 - Aislamiento
 - Integridad
 - Atomicidad
 - Concurrencia
 - Seguridad

Introducción

- **Niveles de abstracción de datos**
 - Nivel Físico
 - Describe cómo los datos están almacenados
 - Nivel Lógico
 - Describe qué datos están almacenados y qué relaciones existen entre los datos
 - Nivel de visualización
 - Describe las formas de presentar los datos al usuario

Introducción

- **Modelos**

- Modelo Entidad - Relación
 - Usa objetos denominados entidades y relaciones
 - Una entidad es la representación de un concepto de la realidad
- Modelo Relacional
 - Usa tablas para representar entidades y relaciones
 - Cada tabla tiene múltiples columnas con único nombre
 - Es el modelo más utilizado

Introducción

- **Lenguajes**
 - Data Manipulation Language (DML)
 - Permite acceder y manipular datos en una forma organizada
 - Los tipos de acceso son:
 - Creación.
 - Consulta.
 - Actualización
 - Eliminación
 - Data Definition Language (DDL)
 - Especifica la estructura y métodos de acceso de la base de datos

Introducción

- **Bases de datos relacionales**

- Basado en el modelo relacional
- Se compone de tablas (Ej: Instructor, Department)

| <i>ID</i> | <i>name</i> | <i>dept_name</i> | <i>salary</i> |
|-----------|-------------|------------------|---------------|
| 22222 | Einstein | Physics | 95000 |
| 12121 | Wu | Finance | 90000 |
| 32343 | El Said | History | 60000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 98345 | Kim | Elec. Eng. | 80000 |
| 76766 | Crick | Biology | 72000 |
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 58583 | Califieri | History | 62000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 15151 | Mozart | Music | 40000 |
| 33456 | Gold | Physics | 87000 |
| 76543 | Singh | Finance | 80000 |

| <i>dept_name</i> | <i>building</i> | <i>budget</i> |
|------------------|-----------------|---------------|
| Comp. Sci. | Taylor | 100000 |
| Biology | Watson | 90000 |
| Elec. Eng. | Taylor | 85000 |
| Music | Packard | 80000 |
| Finance | Painter | 120000 |
| History | Painter | 50000 |
| Physics | Watson | 70000 |

Introducción

- **Bases de datos relacionales**
 - DML

```
select instructor.name  
from instructor  
where instructor.dept_name = 'History';
```

```
select instructor.ID, department.dept_name  
from instructor, department  
where instructor.dept_name= department.dept_name and  
      department.budget > 95000;
```

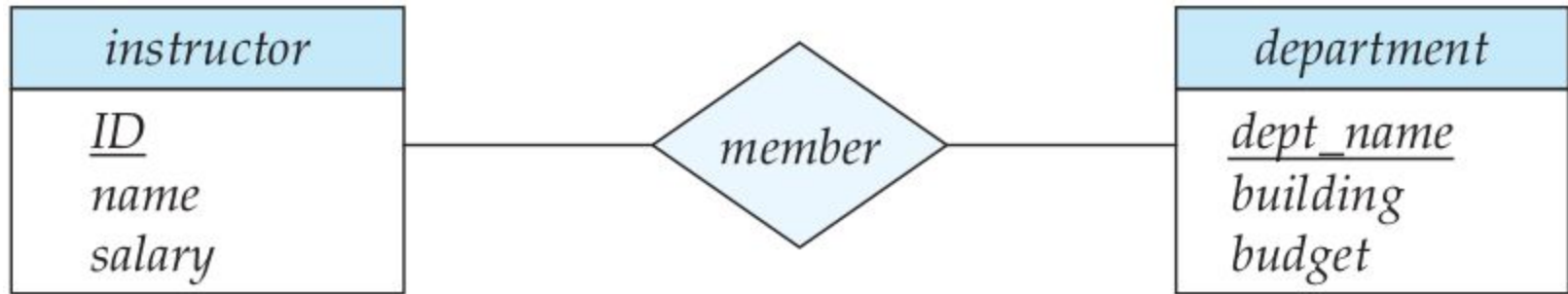
Introducción

- **Bases de datos relacionales**
 - DDL

```
create table department  
  (dept_name    char (20),  
   building     char (15),  
   budget       numeric (12,2));
```

Introducción

- Modelo Entidad Relación



Introducción

- **Normalización**
 - Proceso de diseño
 - Objetivo
 - Generar relaciones que permiten almacenar información sin redundancia innecesaria
 - Facilitar la consulta de información

Introducción

- **Modelo Relacional**
 - Caso de Estudio: Universidad

| <i>ID</i> | <i>name</i> | <i>dept_name</i> | <i>salary</i> |
|-----------|-------------|------------------|---------------|
| 10101 | Srinivasan | Comp. Sci. | 65000 |
| 12121 | Wu | Finance | 90000 |
| 15151 | Mozart | Music | 40000 |
| 22222 | Einstein | Physics | 95000 |
| 32343 | El Said | History | 60000 |
| 33456 | Gold | Physics | 87000 |
| 45565 | Katz | Comp. Sci. | 75000 |
| 58583 | Califieri | History | 62000 |
| 76543 | Singh | Finance | 80000 |
| 76766 | Crick | Biology | 72000 |
| 83821 | Brandt | Comp. Sci. | 92000 |
| 98345 | Kim | Elec. Eng. | 80000 |

| <i>course_id</i> | <i>title</i> | <i>dept_name</i> | <i>credits</i> |
|------------------|----------------------------|------------------|----------------|
| BIO-101 | Intro. to Biology | Biology | 4 |
| BIO-301 | Genetics | Biology | 4 |
| BIO-399 | Computational Biology | Biology | 3 |
| CS-101 | Intro. to Computer Science | Comp. Sci. | 4 |
| CS-190 | Game Design | Comp. Sci. | 4 |
| CS-315 | Robotics | Comp. Sci. | 3 |
| CS-319 | Image Processing | Comp. Sci. | 3 |
| CS-347 | Database System Concepts | Comp. Sci. | 3 |
| EE-181 | Intro. to Digital Systems | Elec. Eng. | 3 |
| FIN-201 | Investment Banking | Finance | 3 |
| HIS-351 | World History | History | 3 |
| MU-199 | Music Video Production | Music | 3 |
| PHY-101 | Physical Principles | Physics | 4 |

Introducción

- **Modelo Relacional**
 - Caso de Estudio: Universidad

| <i>course_id</i> | <i>prereq_id</i> |
|------------------|------------------|
| BIO-301 | BIO-101 |
| BIO-399 | BIO-101 |
| CS-190 | CS-101 |
| CS-315 | CS-101 |
| CS-319 | CS-101 |
| CS-347 | CS-101 |
| EE-181 | PHY-101 |

| <i>dept_name</i> | <i>building</i> | <i>budget</i> |
|------------------|-----------------|---------------|
| Biology | Watson | 90000 |
| Comp. Sci. | Taylor | 100000 |
| Elec. Eng. | Taylor | 85000 |
| Finance | Painter | 120000 |
| History | Painter | 50000 |
| Music | Packard | 80000 |
| Physics | Watson | 70000 |

Introducción

- **Modelo Relacional**
 - Caso de Estudio: Universidad

| <i>course_id</i> | <i>sec_id</i> | <i>semester</i> | <i>year</i> | <i>building</i> | <i>room_number</i> | <i>time_slot_id</i> |
|------------------|---------------|-----------------|-------------|-----------------|--------------------|---------------------|
| BIO-101 | 1 | Summer | 2009 | Painter | 514 | B |
| BIO-301 | 1 | Summer | 2010 | Painter | 514 | A |
| CS-101 | 1 | Fall | 2009 | Packard | 101 | H |
| CS-101 | 1 | Spring | 2010 | Packard | 101 | F |
| CS-190 | 1 | Spring | 2009 | Taylor | 3128 | E |
| CS-190 | 2 | Spring | 2009 | Taylor | 3128 | A |
| CS-315 | 1 | Spring | 2010 | Watson | 120 | D |
| CS-319 | 1 | Spring | 2010 | Watson | 100 | B |
| CS-319 | 2 | Spring | 2010 | Taylor | 3128 | C |
| CS-347 | 1 | Fall | 2009 | Taylor | 3128 | A |
| EE-181 | 1 | Spring | 2009 | Taylor | 3128 | C |
| FIN-201 | 1 | Spring | 2010 | Packard | 101 | B |
| HIS-351 | 1 | Spring | 2010 | Painter | 514 | C |
| MU-199 | 1 | Spring | 2010 | Packard | 101 | D |
| PHY-101 | 1 | Fall | 2009 | Watson | 100 | A |

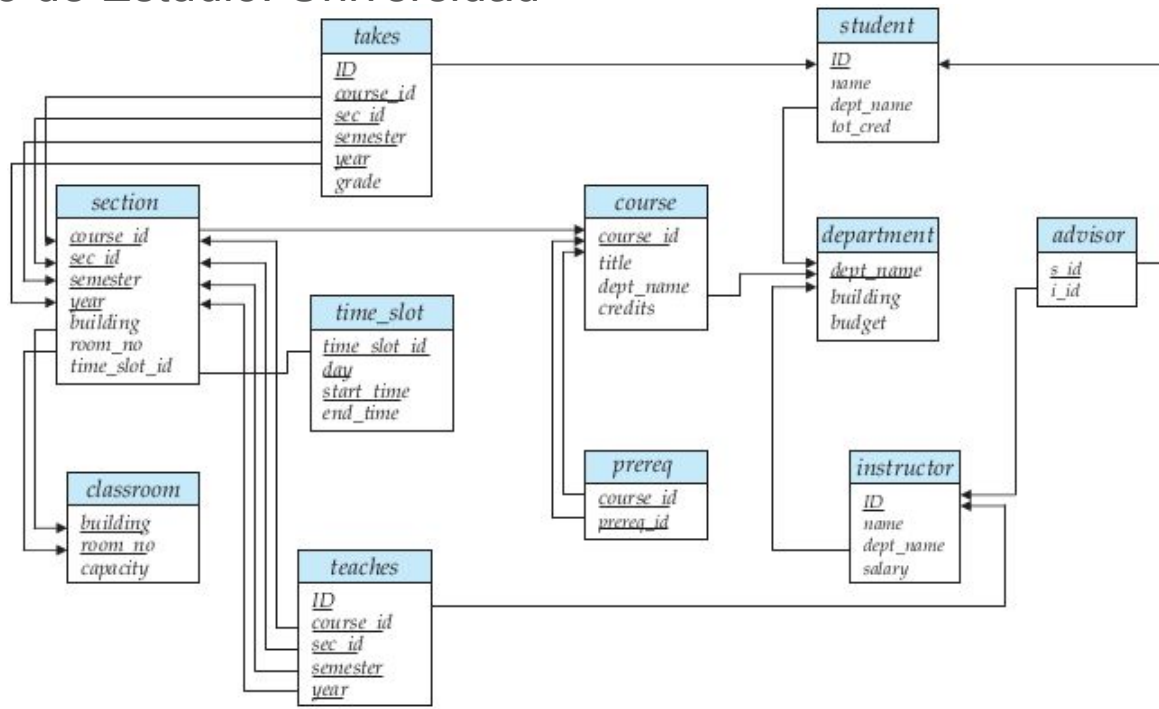
Introducción

- **Modelo Relacional**
 - Caso de Estudio: Universidad

| <i>ID</i> | <i>course_id</i> | <i>sec_id</i> | <i>semester</i> | <i>year</i> |
|-----------|------------------|---------------|-----------------|-------------|
| 10101 | CS-101 | 1 | Fall | 2009 |
| 10101 | CS-315 | 1 | Spring | 2010 |
| 10101 | CS-347 | 1 | Fall | 2009 |
| 12121 | FIN-201 | 1 | Spring | 2010 |
| 15151 | MU-199 | 1 | Spring | 2010 |
| 22222 | PHY-101 | 1 | Fall | 2009 |
| 32343 | HIS-351 | 1 | Spring | 2010 |
| 45565 | CS-101 | 1 | Spring | 2010 |
| 45565 | CS-319 | 1 | Spring | 2010 |
| 76766 | BIO-101 | 1 | Summer | 2009 |
| 76766 | BIO-301 | 1 | Summer | 2010 |
| 83821 | CS-190 | 1 | Spring | 2009 |
| 83821 | CS-190 | 2 | Spring | 2009 |
| 83821 | CS-319 | 2 | Spring | 2010 |
| 98345 | EE-181 | 1 | Spring | 2009 |

Introducción

- **Modelo Relacional**
 - Caso de Estudio: Universidad



Bases de Datos Relacionales

Lenguajes Formales de Consulta

Lenguajes Formales de Consulta

- **Algebra Relacional**

- Es un lenguaje de consulta
- Permite realizar operaciones que toma entidades y relaciones y produce una nueva relación

- **Operaciones**

- Select

$$\sigma_{dept_name = \text{"Physics"}} (instructor)$$

$$\sigma_{salary > 90000} (instructor)$$

$$\sigma_{dept_name = \text{"Physics"} \wedge salary > 90000} (instructor)$$

$$\sigma_{dept_name = building} (department)$$

Lenguajes Formales de Consulta

- **Operaciones**

- Project

$$\Pi_{ID, name, salary}(instructor)$$

- Composition

$$\Pi_{name} (\sigma_{dept_name = \text{“Physics”}} (instructor))$$

- Union

$$\Pi_{course_id} (\sigma_{semester = \text{“Fall”} \wedge year = 2009} (section)) \cup \\ \Pi_{course_id} (\sigma_{semester = \text{“Spring”} \wedge year = 2010} (section))$$

Lenguajes Formales de Consulta

- **Operaciones**

- Set-Difference

$$\Pi_{course_id} (\sigma_{semester = \text{"Fall"} \wedge year = 2009} (section)) - \Pi_{course_id} (\sigma_{semester = \text{"Spring"} \wedge year = 2010} (section))$$

- Cartesian-Product

$$\sigma_{dept_name = \text{"Physics"}}(instructor \times teaches)$$

- Set-Intersection

$$\Pi_{course_id} (\sigma_{semester = \text{"Fall"} \wedge year = 2009} (section)) \cap \Pi_{course_id} (\sigma_{semester = \text{"Spring"} \wedge year = 2010} (section))$$

Lenguajes Formales de Consulta

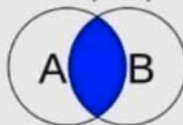
- **Operaciones**

- Natural-Join

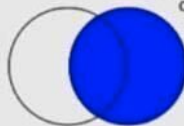
$$\Pi_{name, course_id} (instructor \bowtie teaches)$$

- Full Outer join
- Left Outer join
- Right Outer join

SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key

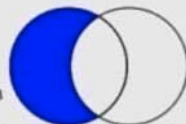


SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key

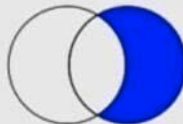


SQL JOINS

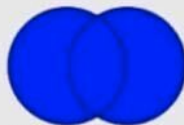
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL



SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE a.key IS NULL



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL



Lenguajes Formales de Consulta

- **Operaciones**

- Aggregate Functions

- Sum

$$\mathcal{G}_{\text{sum}(\text{salary})}(\text{instructor})$$

- Count-distinct

- Average

$$\mathcal{G}_{\text{average}(\text{salary})}(\text{instructor})$$