

# Introducción a SQL

Laboratorio de Base de datos

Cristian E. Sánchez

## Clases virtuales:

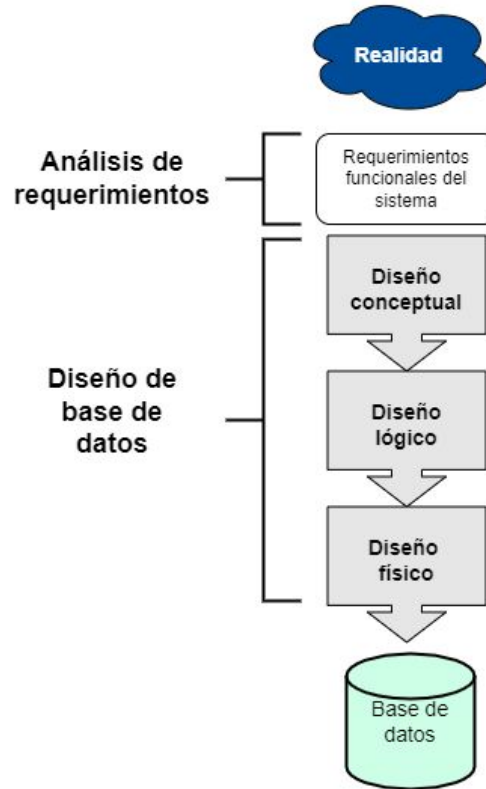
- Semana 12 – Fecha Abril 18
- Semana 13 – Fecha Abril 25
- Semana 14– Fecha Mayo 2
- Semana 15– Fecha Mayo 9

# Agenda

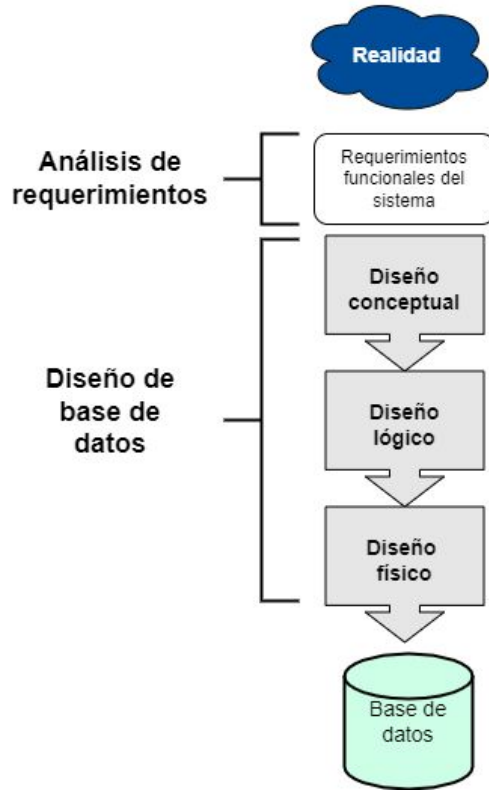
- Diseño lógico
- Lenguaje SQL
- SQL como lenguaje de definición de datos (DDL)

# Diseño lógico

- Encontrar un esquema que sea una representación del mundo real.
- Resuelve el problema de cómo ordenar los datos por medio de tablas



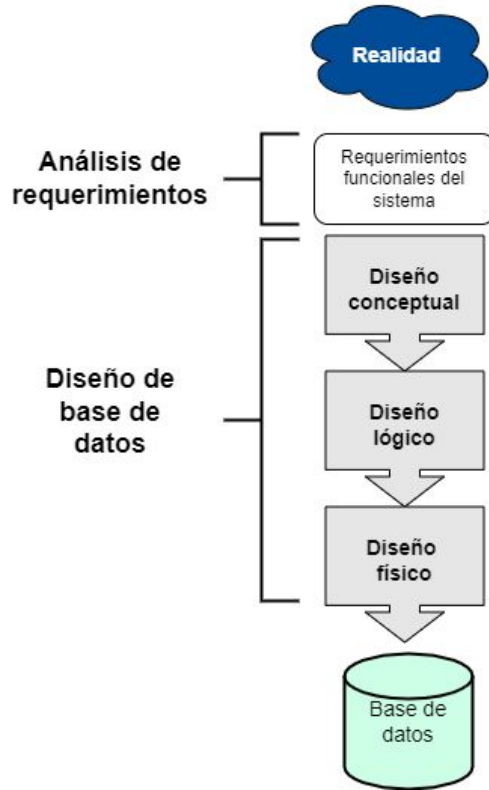
# Diseño lógico



**Diseño conceptual:** Consiste en una descripción de alto nivel de la estructura de la base de datos.

- Es independiente del SGBD
- Se diseña mediante el ***Modelo Entidad-Relación***

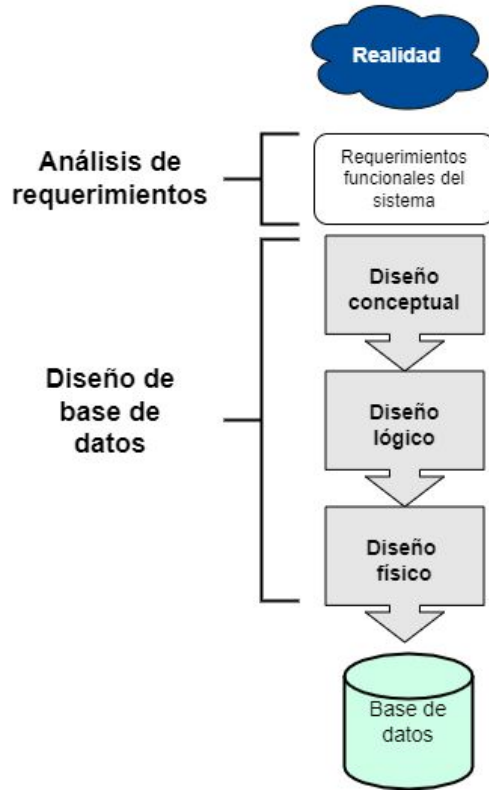
# Diseño lógico



**Diseño Lógico:** Permite construir un esquema de la información que se va a utilizar en la base de datos.

- Se transforma el esquema conceptual en un esquema lógico como el ***Modelo Relacional***.
- Se aplica la normalización para asegurar que las relaciones obtenidas (tablas) no tengan datos redundantes.

# Diseño lógico



**Diseño Físico:** Consiste en la implementación de la base de datos en memoria secundaria. Se definen las estructuras de almacenamiento y los métodos de acceso a la información.

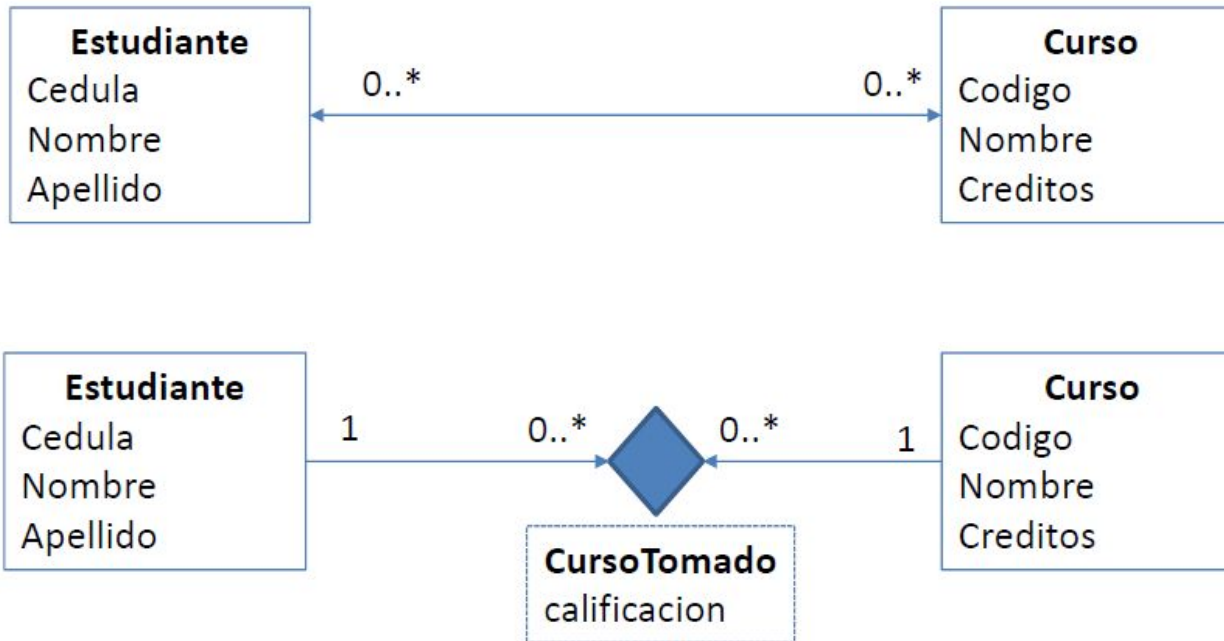
# Diseño lógico – Ejercicio

- Realice el diseño lógico de una base de datos para registrar cursos, estudiantes y sus respectivas calificaciones.
  - Un curso tiene un código, un nombre y un número de créditos.
  - Un estudiante tiene número de cédula, nombre y apellido.
  - Cuando un estudiante pasa un curso se registra su calificación.



# Diseño lógico – Ejercicio

- Diseño Conceptual



# Diseño lógico – Ejercicio

Cursos		
PK	código	string
NN	nombre	string
NN	creditos	int

Estudiantes		
PK	cédula	string
NN	nombre	string
NN	apellido	string

CursosTomados		
PK, FK	estudiante	string
PK, FK	curso	string
NN	calificacio n	real

# SQL – Structured Query Language

- SQL es un lenguaje estándar para el acceso y la manipulación de las bases de datos.
- **Permite**
  - Obtener datos de la BD
  - Agregar registros en la BD
  - Eliminar registros de la BD
  - Actualizar registros de la BD
  - Crear nuevas tablas en la BD
  - Crear nuevas Bases de Datos
  - Crear procedimientos almacenados en la BD
  - Ajustar permisos sobre tablas, procedimientos y vistas en la BD

# SQL – Structured Query Language

## Elementos del lenguaje SQL

- **Clausula:** Palabras clave que constituyen las sentencias y consultas sql.
- **Expresiones:** Es la combinación de uno o más valores, operadores y funciones SQL que da como resultado un valor. Se pueden clasificar en las siguientes categorías:
  - Boolean
  - Numeric
  - Date

# SQL – Structured Query Language

## SQL Boolean Expression

Obtiene datos en función de una condición que hace parte de la consulta SQL.

```
SELECT column  
FROM table_name  
WHERE SINGLE_VALUE_MATCHING_EXPRESSION;
```

# SQL – Structured Query Language

## SQL Boolean Expression

Obtiene datos en función de una condición que hace parte de la consulta SQL.

EmpId	EmpName	EmpAge	EmpSalary
1	John	32	2000
2	Smith	25	2500
3	Henry	29	3000

### Query

```
SELECT EmpName FROM Employee WHERE EmpId = 1;
```

**Tarea: ¿En qué consisten  
las expresiones Numeric y  
Date?**

# Crear una tabla

```
CREATE TABLE persons(  
    person_id NUMBER,  
    first_name VARCHAR2(50) NOT NULL,  
    last_name VARCHAR2(50) NOT NULL,  
    PRIMARY KEY(person_id)  
);
```



# Crear una tabla

```
CREATE TABLE supplier_groups(  
    group_id NUMBER ,  
    group_name VARCHAR2(255) NOT NULL,  
    PRIMARY KEY (group_id)  
);
```

```
CREATE TABLE suppliers (  
    supplier_id NUMBER ,  
    supplier_name VARCHAR2(255) NOT NULL,  
    group_id NUMBER NOT NULL,  
    PRIMARY KEY(supplier_id),  
    FOREIGN KEY(group_id) REFERENCES supplier_groups(group_id)  
);
```

# Documentar una tabla

```
COMMENT ON TABLE Cursos IS 'Tabla que contiene los  
cursos';
```

# Documentar una columna en una tabla

COMMENT ON COLUMN Cursos.codigo IS 'Columna para el  
código de los cursos';

# Modificar una tabla

- **Adicionar columnas:**

```
ALTER TABLE table_name
```

```
ADD (column_name col-definition, ...);
```

- **Modificar una columna:**

```
ALTER TABLE table_name
```

```
MODIFY (column_name col-definition, ...);
```

# Modificar una tabla

- **Modificar una tabla**

- **Borrar columnas:**

```
ALTER TABLE table_name
```

```
    DROP COLUMN column_name;
```

- **Renombrar columnas:**

```
ALTER TABLE table_name
```

```
    RENAME COLUMN old_name TO new_name;
```

- **Renombrar tabla:**

```
ALTER TABLE table_name
```

```
    RENAME TO new_table_name;
```

# Modificar una tabla

- **Eliminar una tabla**

Sintaxis:

```
DROP TABLE [schema_name].table_name  
[ CASCADE CONSTRAINTS ]  
[ PURGE ];
```