





Agenda

01

Reglas de la
clase

02

Modo de
evaluación

03

Bases de datos
relacionales

04

Ventajas y
desventajas

05

Normalización

06

Modelamiento
OLTP

07

Laboratorio

Reglas de clases

- Mantener el micrófono apagado en caso no vayan a hablar.
- Preguntar en caso que tengan dudas
- Mantenerse atento a la clase.

Modo de evaluación

Evaluación continua

Ejercicios, challenges y/o test.

Producto de la clase

Creación de una Base de datos
Transaccional (OLTP) con MySQL

Módulo 2

Introducción a Base de Datos

Sesión 2

Modelamiento de Base de Datos Transaccionales

Normalización

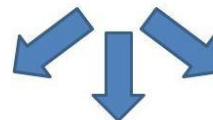
Structured Query Language

La normalización es la transformación de las vistas de usuario complejas y del almacén de datos a un juego de estructuras de datos más pequeñas y estables.

ALUMNOS MATRICULADOS				
rut	nombre	apellido	cod_curso	descripcion
1-9	Pedro	Pérez	AE600	Algoritmos y Estructuras de datos
2-7	Juan	Jara	BD253	Bases de Datos
2-7	Juan	Jara	AE600	Algoritmos y Estructuras de datos
3-5	Diego	Díaz	BD253	Bases de Datos
4-4	Maria	Martinez	BD253	Bases de Datos

La normalización de bases de datos es un proceso que consiste en designar y aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional con objeto de minimizar la redundancia de datos.

ALUMNO		
rut	nombre	apellido
1-9	Pedro	Pérez
2-7	Juan	Jara
3-5	Diego	Díaz
4-4	Maria	Martinez



CURSO	
cod_curso	descripcion
AE600	Algoritmos y Estructuras de datos
BD253	Bases de Datos

MATRICULA	
rut	cod_curso
1-9	AE600
2-7	BD253
2-7	AE600
3-5	BD253
4-4	BD253

Problema de redundancia

matricula	nombre	dirección	telefono	curso	código	carrera
100	juan	ongolmo 340, concepción	78872890	base de datos	bd1	ingeniería civil informatica
100	juan	ongolmo 340, concepción	78872890	estadística	e2	ingeniería civil informatica
100	juan	ongolmo 340, concepción	78872890	analitica	a5	ingeniería civil informatica
200	ana	san martín 840, santiago	78342367	estadística	e2	ingeniería comercial
200	ana	san martín 840, santiago	78342367	analitica	a5	ingeniería comercial

Primera Formal (1NF)

Una tabla debe cumplir con las siguientes condiciones:

- **Valores Atómicos:** Cada columna de la tabla debe contener valores indivisibles (o atómicos), lo que significa que no deben haber listas o conjuntos de valores en una única columna.
- **Unicidad:** Cada columna debe tener un nombre único.
- **Mismo Dominio:** Cada columna debe almacenar datos del mismo tipo.
- **Orden No Significativo:** El orden en que se almacenan los datos no debe afectar la integridad de la base de datos.

matricula	nombre	dirección	telefono	carrera
100	juan	ongolmo 340, concepción	78872890	ingeniería civil informatica
200	ana	san martín 840, santiago	78342367	ingeniería comercial



matricula	curso	código
100	base de datos	bd1
100	estadística	e2
100	analitica	a5
200	estadística	e2
200	analitica	a5

Segunda Formal (2NF)

Para alcanzar la 2NF, una tabla ya debe estar en 1NF y además debe cumplir con:

- **Eliminación de la Dependencia Parcial:** No debe existir dependencia entre las claves no primarias y una parte de la clave primaria en una tabla. Esto aplica solo a las tablas con claves primarias compuestas. Si una clave no primaria depende solo de parte de la clave compuesta, entonces la tabla no está en 2NF.

alumnos

matricula	nombre	dirección	telefono	carrera
100	juan	ongolmo 340, concepción	78872890	ingeniería civil informatica
200	ana	san martín 840, santiago	78342367	ingeniería comercial

alumno_curso

matricula	código
100	bd1
100	e2
100	a5
200	e2
200	a5

cursos

curso	código
base de datos	bd1
estadística	e2
analitica	a5

Tercera Formal (3NF)

Para estar en 3NF, una tabla ya debe cumplir con 1NF y 2NF, y además:

- **Eliminación de las Dependencias Transitivas:** Ningún atributo no clave debe depender de otro atributo no clave. Cada atributo no clave debe depender solo de la clave primaria.

alumnos

matricula	nombre	dirección	telefono	id_carrera
100	juan	ongolmo 340, concepción	78872890	C2100
200	ana	san martín 840, santiago	78342367	C2020

alumno_curso

matricula	código
100	bd1
100	e2
100	a5
200	e2
200	a5

cursos

curso	código
base de datos	bd1
estadística	e2
analitica	a5

carreras

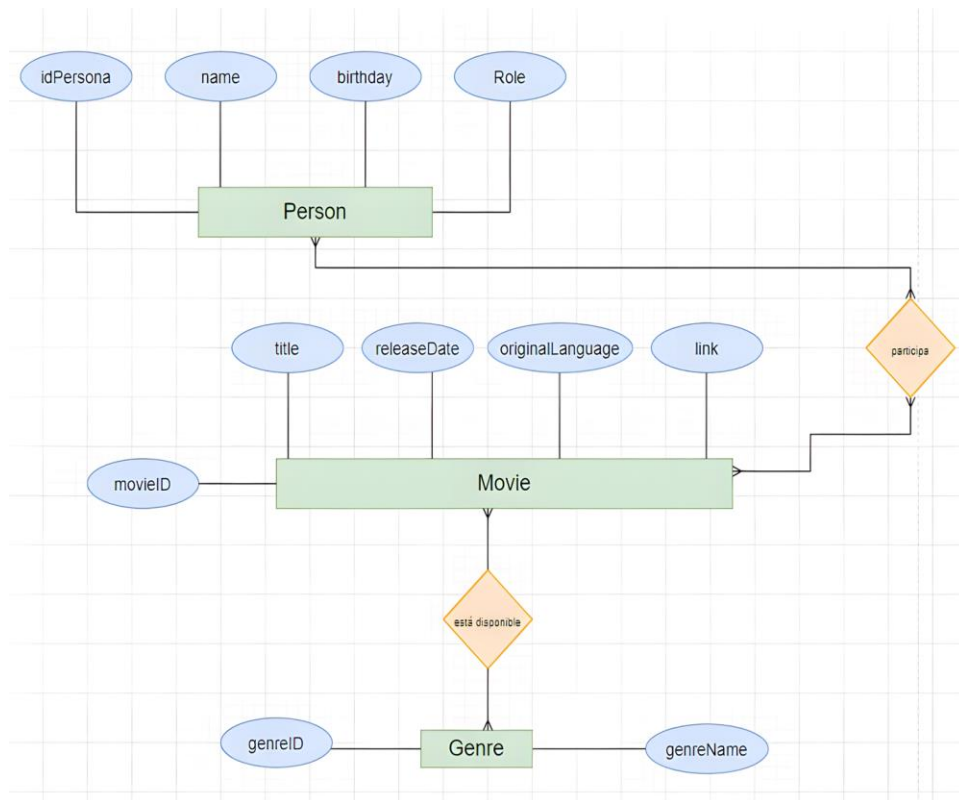
id_carrera	carrera
C2100	ingeniería civil informatica
C2020	ingeniería comercial

Entidad Relación

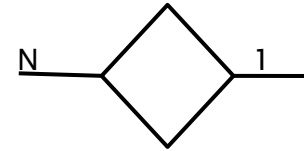
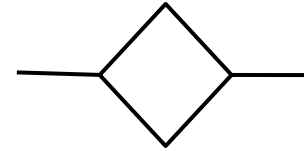
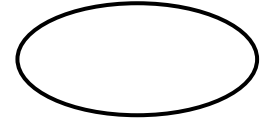
Los diagramas entidad-relación (ER) son una herramienta fundamental en el diseño de bases de datos.

La metodología ER se centra en identificar las entidades relevantes, sus atributos, y las relaciones entre estas entidades.

Facilita el entendimiento de cómo se relacionan entre sí los diferentes elementos de una base de datos.

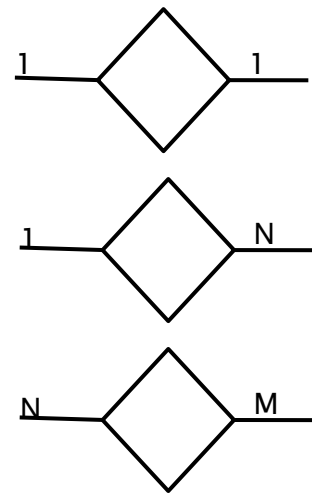


01	Entidades	<ul style="list-style-type: none">Una entidad representa un objeto o concepto del mundo real que es distinguible de otros objetos y conceptos. Las entidades tienen atributos que representan propiedades o características del objeto.
02	Atributos	<ul style="list-style-type: none">Los atributos son las características o propiedades que describen a una entidad. Por ejemplo, una entidad "Persona" podría tener atributos como nombre, dirección y fecha de nacimiento..
03	Relaciones	<ul style="list-style-type: none">Las relaciones describen cómo se asocian las entidades entre sí. Por ejemplo, una relación "trabaja en" podría conectar entidades "Empleado" y "Departamento" para indicar que un empleado trabaja en un departamento específico.
04	Cardinalidad	<ul style="list-style-type: none">La cardinalidad especifica el número de instancias de una entidad que pueden asociarse con cada instancia de otra entidad a través de una relación.



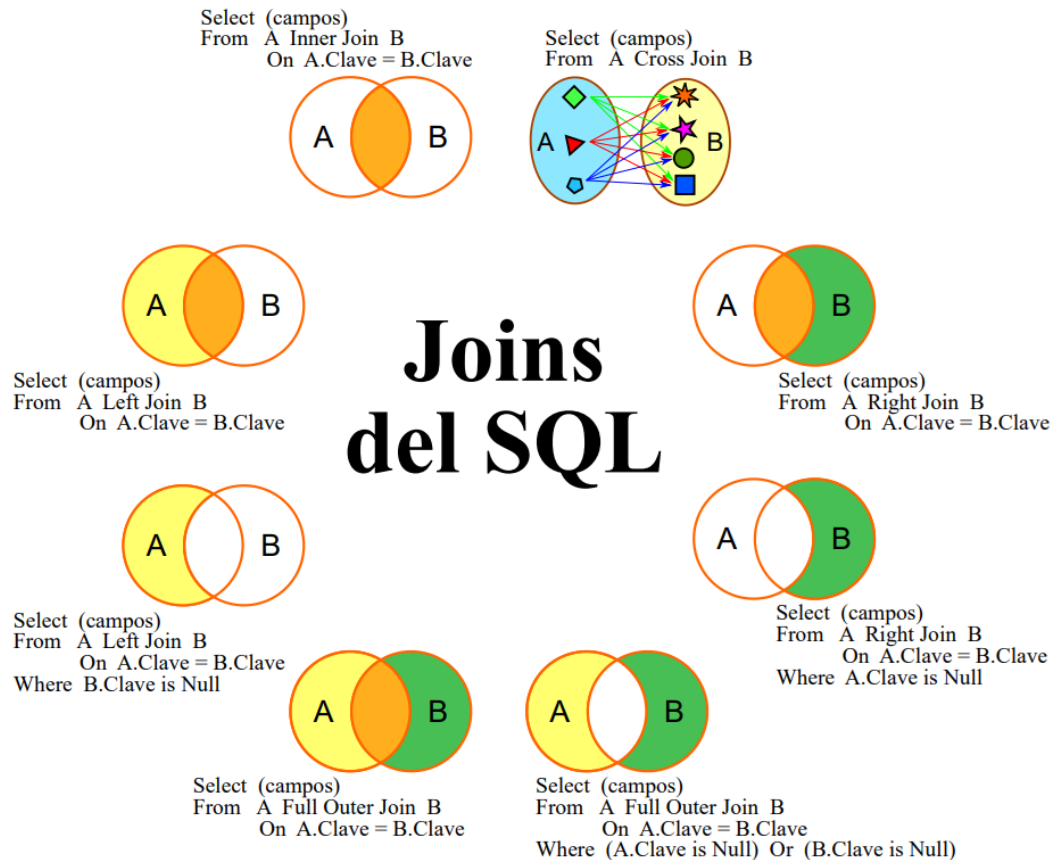
Tipo de Relaciones

01	Uno a Uno (1:1)	<ul style="list-style-type: none">Una instancia de una entidad A se relaciona con una única instancia de una entidad B, y viceversa.
02	Uno a Muchos (1:M)	<ul style="list-style-type: none">Una instancia de una entidad A se puede relacionar con muchas instancias de una entidad B, pero una instancia de B solo se relaciona con una instancia
03	Muchos a Muchos (N:M)	<ul style="list-style-type: none">Muchas instancias de una entidad A pueden relacionarse con muchas instancias de una entidad B.



Uniones

Práctica: JOINS



Modelamiento OLTP

Requisitos y Análisis

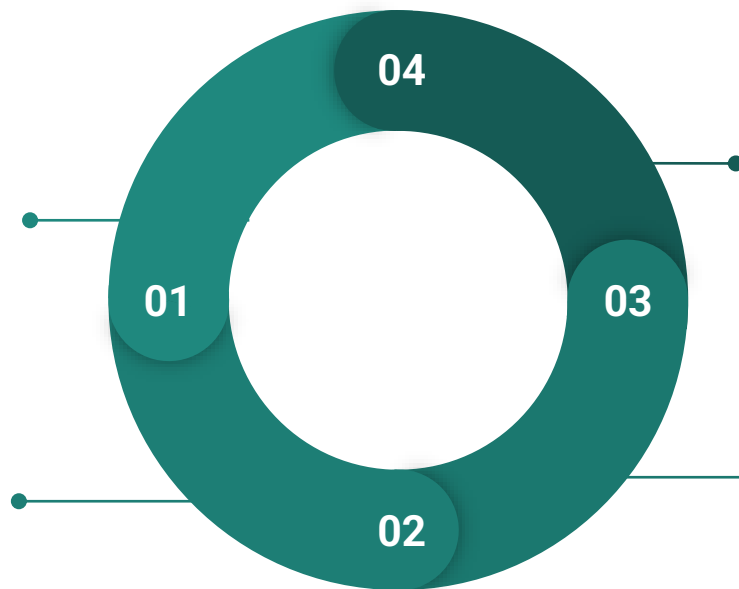
Entender las necesidades del negocio y los requisitos específicos de la base de datos.

Diseño Conceptual

Modelo Entidad-Relación (ER):

Definir entidades, atributos y relaciones.

Normalización: Minimizar la redundancia de datos



Diseño Lógico

Transformación del modelo ER a un modelo relacional: Convertir el diagrama ER en tablas.

Definición de claves primarias y foráneas: Establecer claves que ayudarán en la integridad referencial.

Diseño Físico

Selección del SGBD: Elegir el sistema de gestión de bases de datos

Definición de índices: Decidir qué campos deben ser indexados para optimizar las consultas.

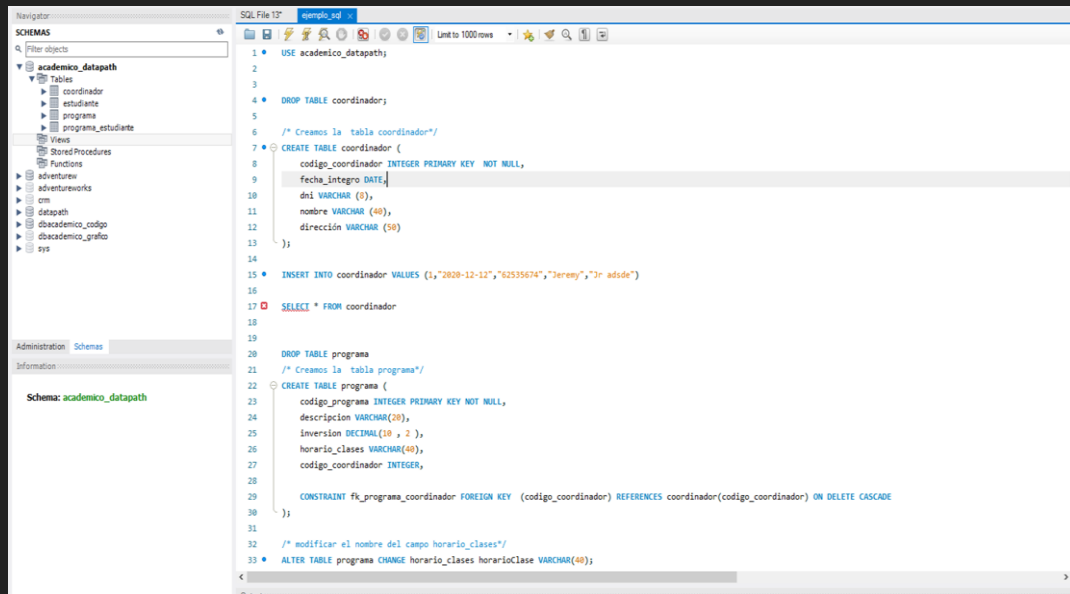
El modelado de datos transaccionales es fundamental en el diseño de bases de datos orientadas a sistemas de procesamiento de transacciones en línea (OLTP), se centra en la eficiencia y rapidez en la ejecución de transacciones.

Los principios del modelado de datos transaccionales tienen como objetivo optimizar el rendimiento, garantizar la integridad de los datos, y facilitar la escalabilidad y el mantenimiento.

Laboratorio: Crear una BD OLTP

Draw IO:

<https://drive.google.com/file/d/12eRJb9w19l8RaYbxiNMpz0jomiO2r4IK/view?usp=sharing>



```
1 USE academico_datapath;
2
3
4 DROP TABLE coordinador;
5
6 /* Creamos la tabla coordinador */
7 CREATE TABLE coordinador (
8     codigo_coordinador INTEGER PRIMARY KEY NOT NULL,
9     fecha_integro DATE,
10     dni VARCHAR (1),
11     nombre VARCHAR (40),
12     direccion VARCHAR (50)
13 );
14
15 INSERT INTO coordinador VALUES (1,"2020-12-12","62535674","jeremy","r adside")
16
17 SELECT * FROM coordinador
18
19
20 DROP TABLE programa
21 /* Creamos la tabla programa */
22 CREATE TABLE programa (
23     codigo_programa INTEGER PRIMARY KEY NOT NULL,
24     descripcion VARCHAR(20),
25     inversion DECIMAL(10, 2),
26     horario_clases VARCHAR(40),
27     codigo_coordinador INTEGER,
28
29     CONSTRAINT fk_programa_coordinador FOREIGN KEY (codigo_coordinador) REFERENCES coordinador(codigo_coordinador) ON DELETE CASCADE
30 );
31
32 /* modificar el nombre del campo horario_clases */
33 ALTER TABLE programa CHANGE horario_clases horarioClase VARCHAR(40);
```



¡Gracias!

Aprende, aplica y crece