

INSTITUTO KRIETE DE INGENIERIA Y CIENCIAS



Reto 3: DataBase

Estudiante:

Monica Daniela Alvarenga Mejia

Docente:

José Luis Montalvo

Materia:

Física aplicada II

FASE 2: NORMALIZACIÓN

Este documento presenta la transformación del **Diagrama Entidad–Relación (ER)** al **modelo relacional**. Además, se explica el proceso de normalización hasta la **Tercera Forma Normal (3FN)**, con el fin de eliminar redundancias y asegurar la integridad de los datos

Modelo relacional:

| Autor | |
|----------|--------------------------|
| ID_Autor | PK, INT, Autoincremental |
| Nombre | VARCHAR |

| Libro | |
|----------|--------------------------|
| ID_Libro | PK, INT, Autoincremental |
| Titulo | VARCHAR |
| ISBN | VARCHAR, UNIQUE |
| ID_Autor | FK -> Autor.ID_Autor |

| Estudiante | |
|---------------|--------------------------|
| ID_Estudiante | PK, INT, Autoincremental |
| Nombre | VARCHAR |
| Grado | VARCHAR, UNIQUE |

| Prestamo | |
|------------------|-------------------------------|
| ID_Prestamo | PK, INT, Autoincremental |
| Fecha_Prestamo | Date |
| Fecha_Devolucion | date |
| Estado | VARCHAR |
| Id_libro | FK -> Libro.ID.libro |
| Id_estudiante | FK-> Estudiante.ID_estudiante |

| Estudiante | |
|---------------|---------|
| ID_Estudiante | INT |
| nombre | VARCHAR |
| Apellido | VARCHAR |
| carrera | VARCHAR |
| Correo | VARCHAR |

Normalizacion

1FN:

- **Atomicidad:** Cada celda de una tabla debe contener un solo valor, no múltiples valores.
- **Unicidad de columnas:** Cada columna debe almacenar un único tipo de dato.
- **Sin filas ni columnas duplicadas:** Se asegura la unicidad de las filas y columnas.

Evita guardar cosas como "Teléfonos: 2222-1111, 7777-3333" en una sola celda. En lugar de eso, cada teléfono tendría su propio registro.

Eliminar grupos repetidos y asegurar que cada campo tiene un solo valor.

En vez de poner en la tabla **Libro** los nombres de varios autores en una sola celda, creamos la tabla **Autor** y usamos una FK.

2FN:

- La tabla debe estar en 1NF.

Eliminar dependencias parciales (cuando un campo depende solo de parte de la clave primaria compuesta).

Elimina redundancia cuando hay claves compuestas.

En **Préstamo**, si hubiéramos usado clave compuesta (id_libro + id_estudiante), podríamos repetir datos del libro o del estudiante. Al usar **id_prestamo** como PK y poner las FKs, evitamos redundancia.

3FN:

- La tabla debe estar en 2NF.

Eliminar dependencias transitivas (atributos que dependen de otros atributos que no son clave).

Evita datos derivados o repetidos que pueden actualizarse mal.

Si en la tabla **Libro** hubiéramos guardado también nombre_autor, este dependería de id_autor (no de la PK del libro directamente). Al quitarlo y dejarlo solo en la tabla **Autor**, eliminamos redundancia y posibles errores (ej: cambiar un nombre de autor solo en un libro y no en todos).

Si hubiéramos dejado **nombre_autor** dentro de **Libro**, cada vez que ese autor escriba otro libro, tendríamos que volver a escribir su nombre. Esto genera **redundancia** y riesgo de inconsistencia (ej: "Gabriel García Márquez" escrito distinto en cada fila). Con la normalización, solo se guarda una vez en **Autor**.