

Formas de Normalización en Base de Datos

1. Primera Forma (1FN)

Una tabla está en Primera forma si todos los atributos contienen valores atómicos (es decir, indivisibles) y no hay columnas con listas o múltiples valores.

Ejemplo

Antes de aplicarla:

Usuario (idUsuario, nombre, correos)

1, "Laura", "laura1@gmail.com, laura2@gmail.com"

Después de aplicarla:

Usuario (idUsuario, nombre)

Correos (idCorreos, correo)

1, "Laura"

1, "laura1@gmail.com"

1, "laura2@gmail.com"

2. Segunda Forma (2FN)

Una tabla está en Segunda forma si está en 1FN y todos los atributos no clave dependen completamente de la clave primaria (no solo de una parte si la clave es compuesta).

Ejemplo

Antes de Aplicarla:

Calificaciones (idCancion, idUsuario, título, calificacion)

El título depende solo de idCancion, no de la combinación completa.

Después de Aplicarla:

Calificaciones (idCancion, idUsuario, calificacion)

Canciones (idCancion, título)

3. Tercera Forma (3FN)

Una tabla está en Tercera Forma si está en 2FN y no existen dependencias Transitivas, es decir, ningún atributo no clave depende de otro atributo no clave.

Ejemplo

Antes de Aplicarla:

Canciones (idCancion, título, idInterprete, nombreInterprete)
nombreInterprete depende de idInterprete no directamente de idCancion.

Después de Aplicarla:

Canciones (idCancion, título, idInterprete)
Interpretes (idInterprete, nombreInterprete)

1. 1FN: Valores atómicos, sin listas.

Ejemplo: Dividir correos múltiples en tabla separada.

2FN: Todos los atributos deben depender de toda la clave primaria.

Ejemplo: Título de canción se va a otra tabla.

3FN: Sin dependencias Transitivas.

Ejemplo: Nombre del intérprete se va a una tabla nueva con su ID.

2. 1FN: Se elimina la multivaluación.

Ejemplo: Tabla correos separada de Usuarios.

2FN: Separaremos los datos que no dependen de toda la clave compuesta.

Ejemplo: Canciones con su título en tabla aparte.

3FN: Eliminaremos la dependencia entre atributos no clave.

Ejemplo: Nombre del intérprete separado en tabla intérpretes.

1FN: Todos los valores deben ser simples (atómicos).

Ejemplo: Separar cada correo en filas distintas.

2FN: Atributos no clave deben depender de la clave completa.

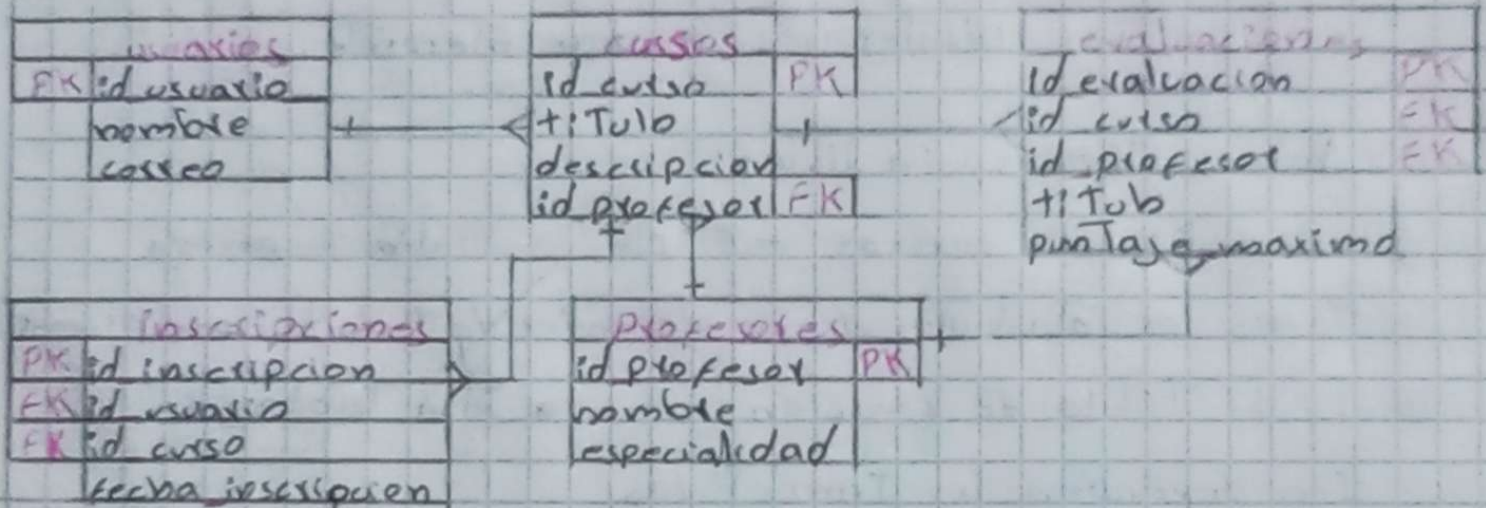
Ejemplo: Separar el título porque depende solo del ID de la canción.

3FN: Un atributo no clave no puede depender de otro no clave.

Ejemplo: Nombre del intérprete va en otra tabla porque depende del ID del intérprete.

Ejemplos para exposiciones

Gestión de Cursos en Línea



Aquí se mostrara el ejemplo de las 5 Foreign Keys x
Primary Keys.

CREATE TABLE Usuarios (

id_usuario INT PRIMARY KEY,
nombre VARCHAR(100),
correo VARCHAR(100)

CREATE TABLE Profesores

id_profesor INT PRIMARY KEY,
nombre VARCHAR(100),
especialidad VARCHAR(100)

INSERT INTO Cursos VALUES

(101, 'Algebra lineal', 'Curso basico de matrices y vectores', 1),
(102, 'Java desde cero', 'Curso introductorio a Java', 2),
(103, 'SQL Avanzado', 'Consultas complejas y optimizacion', 3),
(104, 'Photoshop basico', 'Diseño y edicion de imagenes', 4);

INSERT INTO Inscripcion VALUES

(1001, 1, 101, '2025-07-01'),
(1002, 2, 102, '2025-07-02'),
(1003, 3, 103, '2025-07-03'),
(1004, 4, 104, '2025-07-04');

INSERT INTO Evaluaciones VALUES

(201, 101, 1, 'Parcial 1', 100),
(202, 102, 2, 'Proyecto Final', 200),
(203, 103, 3, 'Quiz SQL', 50),
(204, 104, 4, 'Diseño Grafico Basico', 150);

Ejemplo de las 10 funciones

SELECT

COUNT(*) AS total_usuarios,

MAX(puntaje_maximo) AS mayor_puntaje,

MIN(puntaje_maximo) AS menor_puntaje,

AVG(puntaje_maximo) AS promedio_puntaje,

SUM(puntaje_maximo) AS suma_total,

UPPER(nombre) AS nombre_maxuscula,

LENGTH(nombre) AS longitud_nombre,

NOW() AS fecha_actual,

SUBSTRING(nombre, 1, 5) AS nombre_corto,

CONCAT(nombre, '-', correo) AS contacto

FROM Usuarios

JOIN Incripciones ON Usuarios.id_usuario = Incripciones.id_usuario

JOIN Cursos ON Incripciones.id_curso = Cursos.id_curso

JOIN Evaluaciones ON Cursos.id_curso = Evaluaciones.id_curso

LIMIT 1;

Ejemplos de las 5 Update

```
UPDATE Cursos SET titulo = 'SQL Intermedio'  
WHERE id_curso = 103;
```

```
UPDATE Usuarios SET correo = 'ana.diaz@gmail.com'  
WHERE nombre = 'Ana Diaz';
```

```
UPDATE Evaluaciones SET puntaje_maximo = puntaje_maximo + 10  
WHERE id_evaluacion = 202;
```

```
UPDATE Profesores SET asignatura = 'Álgebra Avanzada'  
WHERE id_profesor = 7;
```

```
UPDATE Cursos SET id_profesor = 4  
WHERE id_curso = 102;
```

Ejemplos de las 5 Delete

```
DELETE FROM Usuarios  
WHERE id_usuario = 4;
```

```
DELETE FROM Cursos  
WHERE titulo = 'Photoshop básico';
```

```
DELETE FROM Evaluaciones  
WHERE titulo = 'Quiz SQL';
```

```
DELETE FROM Inscripciones  
WHERE id_usuario = 2;
```

```
DELETE FROM Profesores  
WHERE nombre = 'Natalia Ruiz';
```

Ejemplo de las 5 Truncate

```
TRUNCATE TABLE Inscripciones;
```

```
TRUNCATE TABLE Evaluaciones;
```

```
TRUNCATE TABLE Cursos;
```

```
TRUNCATE TABLE Profesores;
```

```
TRUNCATE TABLE Usuarios;
```

CREATE TABLE Cursos (

id_curso INT PRIMARY KEY,

Titulo VARCHAR(100),

descripcion TEXT,

id_profesor INT

FOREIGN KEY (id_profesor) REFERENCES Profesores (id_profesor)

);

CREATE TABLE Inscripciones (

id_inscripcion INT PRIMARY KEY,

id_usuario INT,

id_curso INT,

fecha_inscripcion DATE,

FOREIGN KEY (id_usuario) REFERENCES Usuarios (id_usuario),

FOREIGN KEY (id_curso) REFERENCES Cursos (id_curso)

);

CREATE TABLE Evaluaciones (

id-evaluacion INT PRIMARY KEY,

id_curso INT,

id_profesor INT,

Titulo VARCHAR(100),

puntaje_maximo INT,

FOREIGN KEY (id_curso) REFERENCES Cursos (id_curso),

FOREIGN KEY (id_profesor) REFERENCES Profesores (id_profesor)

);

Ejemplo de los 5 Inserts

INSERT INTO Usuarios VALUES

(1, 'Laura Torres', 'laura@gmail.com'),

(2, 'Carlos Ríos', 'carlos@gmail.com'),

(3, 'Ana Díaz', 'ana@gmail.com'),

(4, 'Julian Gomez', 'julian@gmail.com');

INSERT INTO Profesores VALUES

(1, 'Mario Herrera', 'Matemáticas'),

(2, 'Sofia Mendoza', 'Programación'),

(3, 'David Leon', 'Base de Datos'),

(4, 'Natalia Ruiz', 'Diseño Gráfico');