Día	Mes	Año	Hora	Institución		
Alumno: Karen Daniela Holguin Cruz					Código	Materia
Curso	Bimestre	Semestre	Salón	Hoja No ____ de ____		CALIFICACIÓN
Profesor						

1. Sistema de Gestión Escolar:

Entidades: Estudiante, Profesor, Curso, Grado, Matricula, Nota, Horario

• Relacion Pivotes (por entidad):

Entidad	FK1	FK2
Estudiante	id_grado	id_Curso
Profesor	id_grado	id_Curso
Curso	id_Profesor	id_grado
Grado		
Matricula	id_estudiante	id_curso
Nota	id_estudiante	id_Curso
Horario	id_Profesor	id_Curso

• Diccionario de Datos:

Entidad	Atributos (clave)
Estudiante	id_estudiante(PK), nombre, id_grado (FK), id_Curso (FK)
Profesor	id_profesor(PK), nombre, id_grado (FK), id_Curso (FK)
Curso	id_Curso(PK), nombre, id_Profesor, id_grado (FK)
Grado	id_grado (PK), nombre
Matricula	id_Matricula(PK), id_estudiante(FK) id_Curso (FK)
Nota	id_nota (PK), id_estudiante (FK), id_Curso (FK) nota
Horario	id_horario(PK), dia, hora, id_Profesor (FK), id_Curso (FK)

• funciones:

```
INSERT INTO Estudiantes VALUES (1, 'Ana lopez', 101 201);
SELECT nombre FROM curso WHERE id_Curso IN (
SELECT id_Curso FROM Matricula WHERE id_estudiante = 1
);

UPDATE Nota SET nota = 4.5 WHERE id_nota = 1;

DELECT FROM Matricula WHERE id_matricula = 1;

CREATE TABLE Grado (
id_grado INT PRIMARY KEY,
nombre VARCHAR (50)
);

CREATE TABLE profesor (
id_Profesor INT PRIMARY KEY,
nombre VARCHAR (50),
id_grado INT,
id_Curso INT
);
```

```
CREATE TABLE Curso(
id_Curso INT PRIMARY KEY,
nombre VARCHAR (50),
id_Profesor INT,
id_grado INT
);

CREATE TABLE Estudiante(
id_estudiante INT PRIMARY KEY,
nombre VARCHAR (50),
id_grado INT,
id_Curso INT
);

CREATE TABLE Matricula(
id_matricula INT PRIMARY KEY,
id_estudiante INT,
id_Curso INT
);

CREATE TABLE Nota(
id_nota INT PRIMARY KEY,
id_estudiante INT,
id_Curso INT,
nota FLOAT
);

CREATE TABLE Horario(
id_horario INT PRIMARY KEY,
dia VARCHAR (20)
hora TIME,
id_profesor INT,
id_curso INT
);

ALTER TABLE Estudiante
ADD FOREIGN KEY (id_grado) REFERENCES Grado (id_grado)
ADD FOREIGN KEY (id_curso) REFERENCES Curso (id_Curso)
```

2. Clinica/Hospital:

Entidades: Paciente, Médico, cita, Especialidad, Historia Clinica, Diagnóstico, Medicamento

Entidad	FK1	FK2
Paciente	id_cita	id_medico
Médico	id_especialidad	id_cita
Cita	id_paciente	id_medico
Especialidad		
Historia Clinica	id_paciente	id_diagnostico
Diagnóstico	id_medico	id_paciente
Medicamento	id_diagnóstico	id_paciente

Entidad	Atributos
Paciente	id_Paciente(pk), nombre, id_cita(fk), id_medicofk
Medico	id_medico (pk) nombre, id_especialidad fk
cita	id_cita (pk) fecha, id_paciente (fk), id_medico
Especialidad	id_Especialidad (pk), hombre
Historia Clinica	id_historia(pk), observación, id_diagnostico
Diagnóstico	id_diagnostico pk, descripcion, id_medico
Medicamento	id_medicamento pk nombre, id_diagnostico
	id_

```
CREATE TABLE Especialidad (
id_especialidad INT PRIMARY KEY,
nombre VARCHAR (50)
);

CREATE TABLE Medico (
id_medico INT PRIMARY KEY
nombre VARCHAR (50),
id_especialidad INT,
id_cita INT
);

CREATE TABLE Paciente (
id_Paciente INT PRIMARY KEY,
nombre VARCHAR (50),
id_cita INT
id_Medico INT
);

CREATE TABLE cita(
id_cita INT PRIMARY KEY,
fecha DATE
id_Paciente INT,
id_Medico INT
);

CREATE TABLE Diagnostico(
id_diagnostico INT PRIMARY KEY,
descripción VARCHAR (100),
id_medico INT,
id_Paciente INT
);

CREATE TABLE HistoriaClinica(
id_historia INT PRIMARY KEY,
observaciones TEXT,
id_diagnostico INT
);

CREATE TABLE Medicamento(
id_medicamento INT PRIMARY KEY,
nombre VARCHAR (50),
id_diagnostico INT,
id_Paciente INT
);
```

3. Sistema de restaurante

• Cliente, Pedido, Producto, Mesa, Mesero, Datalle pedido, factura

Entidad	FK1	FK2
Cliente	id_Pedido	
Pedido	id_Cliente	id_mesa
Producto	id_Pedido	id_mesero
Mesa	id_Cliente	id_detalle
Mesero	id_Pedido	id_mesero
Detalle Pedido	id_Pedido	id_mesa
factura	id_Pedido	id_Prodructo

Entidad	Atributos
Cliente	id_ClientePK, nombre, id_Pedido FK, id_mesa FK
Pedido	id_PedidoPK, fecha, id_cliente, FK, id_mesero FK
Producto	id_ProductoPK, nombre, precio, id_pedido
Mesa	id_MesaPK, numero, id_cliente FK id_mesero
Mesero	id_MeseroPK, nombre, id_pedido FK id_mesa
Detalle pedido	id_DetallePK, id_PedidoFK, id producto
factura	id_facturapK, total, id_pedido FK

```
CREATE TABLE cliente (
id_cliente INT PRIMARY KEY,
nombre VARCHAR (50),
id_pedido INT
id_Mesa INT
);

CREATE TABLE Pedido (
id_pedido INT PRIMARY KEY,
fecha DATE,
id_cliente INT,
id_mesero INT
);

CREATE TABLE Producto (
id_producto INT PRIMARY KEY,
nombre VARCHAR (50),
precio FLOAT,
id_Pedido,
id_detalle
);

CREATE TABLE Mesa (
id_mesa INT PRIMARY KEY,
numero INT,
id_Cliente INT,
id_mesero INT
);
```

Día	Mes	Año	Hora	Institución

Alumno	Código	Materia
Karen Daniela Holguín Cruz		

Curso	Bimestre	Semestre	Salón	Hoja No ____ de ____	CALIFICACIÓN

Profesor

```
CREATE TABLE DetallePedido(
id_detalle INT PRIMARY KEY,
id_Pedido INT,
id_producto INT,
Cantidad INT,
);

CREATE TABLE factura(
id_factura INT PRIMARY KEY,
total FLOAT,
id_pedido INT,
id_cliente INT

INSERT INTO cliente (id_Cliente, nombre, id_pedido)
VALUES (1, 'Mario Ruiz' 101, 5)

SELECT p.nombre, p.precio.
FROM DetallePedido
JOIN producto p. id_producto= p.id_producto
WHERE id_pedido= 101;

UPDATE DetallePedido
SET Cantidad= 3
WHERE id_factura =1001;
```

4. Tienda Online

- Cliente, Producto, Carrito, Pedido, DetallePedido, Categoria, Pago

Entidad	FK1	FK2
Cliente	id_Carrito	id_pedido
Producto	id_Categoria	id_Carrito
Carrito	id_Cliente	id_producto
Pedido	id_Cliente	id_Pago
DetallePedido	id_Pedido	id_Producto
Categoria		
Pago	id_Pedido	id_cliente

Entidad	Atributos
Cliente	id_cliente (PK) nombre id. carrito fk. id. pedido fk
Producto	id_Producto (PK), nombre, id_categoria. id_carrito
Carrito	id_Carrito PK id_cliente. id_producto
Pedido	id_pedido (PK) fecha, id_cliente. id_pago fk
DetallePedido	id_detalle, id_pedido. producto fk. Cantidad
Categoria	id_Categoria PK. nombre
Pago	id_pago PK. metodo. id_pedido fk. id_cliente fk

```
CREATE TABLE Categoria(
id_categoria INT PRIMARY KEY,
nombre VARCHAR (50)
);

CREATE TABLE Producto(
id_producto INT PRIMARY KEY,
nombre VARCHAR (60);
id_Categoria INT,
id_Carrito INT
);

CREATE TABLE cliente(
id_Cliente INT PRIMARY KEY,
nombre VARCHAR (60);
id_carrito INT,
id_Pedido INT
);

CREATE TABLE Carrito(
id_Carrito INT PRIMARY KEY,
id_cliente INT,
id_producto INT,
);

CREATE TABLE DetallePedido(
id_detalle INT PRIMARY KEY,
id_pedido INT,
id_producto INT,
);

CREATE TABLE Pago(
id_pago INT PRIMARY KEY,
metodo VARCHAR (30),
id_Pedido INT
);

INSERT INTO cliente (id_cliente,nombre,)
VALUES (1, 'Laura Torres', 101, 201);

SELECT p.nombre
FROM Producto p
JOIN carrito c ON id_producto= c.id_producto
WHERE c.id_cliente = 1;

UPDATE Pago
SET metodo= 'transferencia'
WHERE id_pago=301

DELETE FROM Carrito
WHERE id_carrito=101 AND id_producto=501;
```

5. Parqueadero:

- Vehiculo, Cliente, Espacio, Ingreso, Salida, Tarifa, Empleado

Entidad	FK1	FK2
Vehiculo	id_cliente	
cliente	id_vehiculo	id_espacio
Espacio	id_vehiculo	id_tarifa
		id_empleado

Ingreso	id_Vehiculo	id_espacio
Salida	id_Vehiculo	id_ingreso
Tarifa	id_Cliente	id_vehiculo
Empleado	id_espacio	id_ingreso

Entidad	Atributos
Vehiculo	id_vehiculo(pk), placa, id_cliente fk, id_espacio fk
Cliente	id_Cliente(pk) nombre, id_Vehiculo fk, id_tarifa
Espacio	id_Espacio(pk) numero, id_vehiculo fk, id_empleado fk
Ingreso	id_Ingreso(pk) hora, id_vehiculo fk, id_espacio
Salida	id_Salida (pk) hora, id_ id_vehiculo, id_ingreso fk
Tarifa	id_Tarifa(pk), monto, id_Cliente, id_vehiculo
Empleado	id_Empleado(pk) nombre, id_espacio fk

```
CREATE TABLE vehiculo(
id_vehiculo INT PRIMARY KEY,
placa VARCHAR (10)
id-Cliente INT
id-espacio INT
);

CREATE TABLE cliente(
id_cliente INT PRIMARY KEY,
nombre VARCHAR (50),
id_vehiculo INT,
id_tarifa INT
);

CREATE TABLE Espacio(
id_espacio INT PRIMARY KEY,
numero INT
id_vehiculo INT,
id_empleado INT
);

CREATE TABLE Ingreso(
id_ingreso INT PRIMARY KEY,
hora TIME
id_vehiculo INT,
id_Espacio INT
);

CREATE TABLE Salida(
id-salida INT PRIMARY KEY,
hora TIME,
id_vehiculo INT,
id_Ingreso INT
);

CREATE TABLE Empleado(
id_empleado INT PRIMARY KEY,
nombre VARCHAR (50),
id_espacio INT,
id_ingreso INT
);

INSERT INTO Ingreso (id_ingreso, hora, id_vehiculo, id_espacio)
VALUES (1, '08:30:00', 101, 3);
```

```
SELECT v.placa
FROM vehiculo v
JOIN Espacio e ON v.id_espacio=e.id_espacio
WHERE e.numero=3;

UPDATE tarifa
SET monto= 8500
WHERE id_tarifa=2;

DELETE FROM Salida
WHERE id_salida =1;
```

6. Cine:

- Pelicula, Sala, función, Boleto, cliente, Asiento, Empleado

Entidad	FK1	FK2
Pelicula	id_Sala	id_función
Sala	id_empleado	id_función
función	id_Pelicula	id_Sala
Boleto	id_Cliente	id_función
Cliente	id_asiento	id_boleto
Asiento	id_Sala	id_función
Empleado	id_Sala	id_función

Entidad	Atributos
Pelicula	id_pelicula (pk), titulo, id_Sala fk, id_función
Sala	id_Sala (pk) numero, id_empleado fk
función	id_función pk, hora, id_pelicula fk, id_Sala
Boleto	id_boleto pk, precio, id_cliente fk
Cliente	id_Cliente pk, nombre, id_asiento fk
Asiento	id_Asiento (pk) numero, id_Sala fk, id_función fk
Empleado	id_empleado pk nombre, id_Sala

```
CREATE TABLE Pelicula (
id_pelicula INT PRIMARY KEY,
titulo VARCHAR (50),
id_Sala INT
id_función INT
);

CREATE TABLE Sala(
id_sala INT PRIMARY KEY,
numero INT
id_empleado INT
id_función INT
);

CREATE TABLE función(
id_función INT PRIMARY KEY,
hora TIME,
id_pelicula INT,
id_sala INT
);
```

Observaciones

Día	Mes	Año	Hora	Institución		
					Código	Materia
Alumno: Karen Daniela Holguín Cruz						CALIFICACIÓN
Curso	Bimestre	Semestre	Salón	Hoja No ____ de ____		
Profesor						

```
CREATE TABLE cliente (
id_cliente INT PRIMARY KEY,
nombre VARCHAR (50)
id_asiento INT.
id_boleto INT
);

CREATE TABLE Boleto(
id_boleto INT PRIMARY KEY,
Precio FLOAT,
id_cliente INT,
id_funcion INT
);

CREATE TABLE Empleado (
id_empleado INT PRIMARY KEY.
nombre VARCHAR (50),
id_sala INT
id_funcion INT
);

INSERT INTO funcion (id_funcion, hora, id_pelicula)
VALUES (1, '10:00:00', 10,5)

SELECT b.id_boleto, p.titulo, f.hora
FROM boleto b
JOIN funcion f ON id_funcion = f. id_función
JOIN Pelicula p ON id_pelicula = p. id_pelicula
WHERE b.id_cliente=1;

UPDATE cliente
SET id_asiento=205
WHERE id_cliente=1;
)

DELETE FROM funcion
WHERE id_funcion=1;
```

7. Gimnasio

- Usuario, Entrenador, Rutina, Clase, Horario, Pago, Plan

Entidad	FK1	FK2
Usuario	id_rutina	id_plan
Entrenador	id_Clase	id_Horario
Rutina	id_usuario	id_Entrenador
Clase	id_usuario	id_Entrenador
Horario	id_Clase	id_Entrenador
Pago	id_usuario	id_Plan
Plan	id_usuario	id_pago

Entidad	Atributos
Usuario	id_usuario pk, nombre, id_rutina fk id_plan
Entrenador	id_entrenador pk, nombre, id_Clase fk
Rutina	id_rutina pk descripción, id_usuario_entrenador
Clase	id_Clase pk nombre, id_usuario fk, id_entrenador (fk)
Horario	id_horario pk día_hora, id_clase, id_entrenador
Pago	id_pago, monto, id_usuario, id_plan fk
Plan	id_plan, tipo, id_usuario fk, id_pago

```
CREATE TABLE usuario(
id_usuario INT PRIMARY KEY,
nombre (VARCHAR (50))
id_rutina INT
id_plan INT
);

CREATE TABLE Entrenador(
id_entrenador INT PRIMARY KEY,
nombre VARCHAR (50),
id_Clase INT
id_horario INT
);

CREATE TABLE Rutina(
id_rutina INT PRIMARY KEY,
descripción TEXT
id_usuario INT,
id_entrenador INT
);

CREATE TABLE Clase(
id_Clase INT PRIMARY KEY,
nombre VARCHAR (50),
id_usuario INT
id_entrenador INT.
);

CREATE TABLE Plan(
id_plan INT PRIMARY KEY,
tipo VARCHAR (30),
id_usuario INT
id_pago INT
);

CREATE TABLE Pago(
id_Pago INT PRIMARY KEY,
monto FLOAT
id_usuario INT
id_plan INT
);

INSERT INTO usuario (id_usuario, nombre, id_rutina)
VALUES (1, 'Carlos Mendoza', 10, 20);

SELECT c.nombre
FROM clase c
WHERE c.id_entrenador =5;

UPDATE plan
SET tipo= 'Premium'
```

```
DELETE FROM Horario
WHERE id_horario=15;
```

8. Empresa de Transporte:

- Vehículo, Conductor, Ruta, pasajero, Tiquete, Terminal, Horario.

Entidad	FK1	FK2
Vehículo	id_ruta	id_Conductor
Conductor	id_vehículo	id_horario
Ruta	id_terminal	id_horario
Pasajero	id_tiquete	id_ruta
Tiquete	id_pasajero	id_ruta
Terminal	id_ruta	id_vehículo
Horario	id_ruta	id_Conductor
		id_
		id_

Entidad	Atributos
vehículo	id_vehículo PK, placa, id_ruta, Conductor fk
Conductor	id_Conductor pk nombre, id_vehículo fk
Ruta	id_ruta pk, nombre, id_terminal fk
Pasajero	id_pasajero PK, nombre, id_tiquete fk
Tiquete	id_tiquete_precio, id_pasajero, id_ruta
Terminal	id_terminal pk, nombre_id_ruta
Horario	id_horario_hora_salida TIME_id_ruta

```
CREATE TABLE vehículo(
id_vehículo INT PRIMARY KEY,
Placa VARCHAR (10),
id_ruta INT,
id_Conductor INT
);

CREATE TABLE Conductor (
id_Conductor INT PRIMARY KEY,
nombre VARCHAR (50),
id_Vehículo INT,
id_horario INT,
);

CREATE TABLE Ruta(
id_ruta INT PRIMARY KEY,
nombre VARCHAR (50),
id_terminal INT,
id_horario INT
);

CREATE TABLE Pasajero(
id_Pasajero INT PRIMARY KEY,
nombre VARCHAR (50)
id_tiquete INT
id_ruta INT
);

CREATE TABLE Terminal (
id_terminal INT PRIMARY KEY
nombre VARCHAR (50)
id_ruta INT,
id_vehículo INT
);
```

```
CREATE TABLE Horario (
id_horario INT PRIMARY KEY,
hora_Salida TIME,
id_ruta INT,
id_conductor INT
);

INSERT INTO Tiquete (id_tiquete, precio, id_pasajero)
VALUES (1,12000, 101, 5);

SELECT r.nombre
FROM ruta r
JOIN Horario h ON r.id_horario
WHERE h.id_conductor=20;

UPDATE Terminal
SET nombre= Terminal Norte
WHERE id_terminal = 3;

DELETE FROM pasajero WHERE id_pasajero =10;
```

9. Plataforma de cursos virtuales:

- Estudiante, Instructor, Curso, Módulo, clase, Evaluación, Certificado

Entidad	FK1	FK2
Estudiante	id_Curso	id_Certificado
Instructor	id_Curso	id_Clase
Curso	id_instructor	id_Modulo
Módulo	id_Curso	id_Clase
Clase	id_Modulo	id_Instructor
Evaluación	id_Curso	id_Estudiante
Certificado	id_Estudiante	id_Curso

Entidad	Atributos
Estudiante	id_estudiante PK, nombre, id_curso FK
Instructor	id_instructor PK, nombre, id_Curso, id_clase
Curso	id_Curso (PK), nombre, id_curso
Módulo	id_modulo, nombre, id_curso PK
clase	id_Clase PK, titulo, id_modulo, id_instructor
Evaluación	id_evaluación, nota, id_Curso FK
Certificado	id_Certificado, fecha, id_estudiante FK

```
CREATE TABLE Estudiante(
id_estudiante INT PRIMARY KEY,
nombre VARCHAR (50),
id_curso INT,
id_certificado INT
);

CREATE TABLE Instructor(
id_instructor INT PRIMARY KEY,
nombre VARCHAR (50),
);
```

Observaciones

Día	Mes	Año	Hora	Institución		
Alumno Karen Daniela Holguín Cruz					Código	Materia
Curso	Bimestre	Semestre	Salón	Hoja No ____ de ____		CALIFICACIÓN
Profesor						

```
CREATE TABLE Curso(
id_Curso INT PRIMARY KEY,
título VARCHAR(50)
id_instructor INT
id_modulo INT
);

CREATE TABLE Clase(
id_Clase INT PRIMARY KEY,
título VARCHAR (50),
id-modulo INT
id_Instructor INT
);

CREATE TABLE Evaluación(
id_evaluación INT PRIMARY KEY,
nota FLOAT
id_Curso INT,
id_Estudiante INT,
);

INSERT INTO Estudiante VALUES (1,'Luis Peréz),(1001,501);

SELECT título FROM curso WHERE id_Curso IN(
SELECT id_curso FROM estudiante WHERE id_estudiante=1
);

UPDATE Evaluación SET nota=48 WHERE id_evaluación=201;
DELETE FROM Certificado WHERE id-certificado=501;
```

10. Agencia de viajes

- Cliente, viaje, Destino, Reserva, vuelo, Hotel, Guía

Entidad	FK1	FK2
Cliente	id_reserva	id_viaje
viaje	id_Destino	id_guía
Destino	id_vuelo	id_hotel
Reserva	id_cliente	id_viaje
vuelo	id_Reserva	id_destino
Hotel	id_Reserva	id_destino
Guía		

Entidad	Atributos
Cliente	id_clientepk nombre, id_reserva, id_viaje
viaje	id_viaje PK, nombre, id_destino, id_guía FK
Destino	id_destino, lugar, id_vuelo FK
Reserva	id_reserva, fecha, id_clienteFK, id_viaje
vuelo	id_vuelo, aerolina, id_reserva FK
Hotel	id_hotel, nombre id_reserva, id_destino
Guía	id_guía, nombre, id_destino FK

```
CREATE TABLE cliente(
id_cliente INT PRIMARY KEY,
nombre VARCHAR (50)
id_reserva INT
id_viaje INT
);

CREATE TABLE viaje(
id_viaje INT PRIMARY KEY,
nombre VARCHAR (50),
id_destino INT
id_guia INT
);

CREATE TABLE Destino(
id_destino INT PRIMARY KEY,
lugar VARCHAR (50)
id_vuelo INT
id_hotel INT
);

CREATE TABLE vuelo(
id_vuelo INT PRIMARY KEY,
aerolina VARCHAR (50),
id_reserva INT
id_destino INT
);

CREATE TABLE Hotel (
id_hotel INT PRIMARY KEY,
nombre VARCHAR (50),
id_reserva INT
id_Destino INT
);

CREATE TABLE Guia(
id_guia INT PRIMARY KEY,
nombre VARCHAR (50),
id_viaje INT
id_Destino INT
);

INSERT INTO Reserva VALUES (1,'2025,-07-12', 101, 102)

SELECT lugar FROM Destino WHERE id_destino IN(
SELECT id_Destino FROM viaje WHERE id_viaje=(
SELECT id_viaje FROM Cliente WHERE id_cliente=101
)
);

UPDATE Hotel SET nombre='Hotel pacifico WHERE id_hotel=3;
DELETE FROM Guia WHERE id_guia=5;
```

Nombre: Karen Daniela Halguin Cruz

Fecha:

Profesor:

Materia: Base de Datos

Institución:

Curso: Sena

Nota:

1. Concepto de Normalización en Base de datos

• La normalización es un proceso en el diseño de bases de datos relacionales que organiza los datos para reducir la redundancia y mejorar la integridad. Consiste en dividir una base de datos en tablas más pequeñas y establecer relaciones entre ellas siguiendo reglas llamadas formas normales

Ejemplo: Una tabla que contiene información de estudiante y cursos:

Estudiante ID	Nombre	Curso 1	Curso 2
1	Ana	Ingles	Matemáticas

Al normalizar, se separa en dos tablas:

Estudiante ID	Nombre
1	Ana

Estudiante ID	Curso
1	Ingles
1	Matematicas

2. Primera forma Normal (1FN)

• Una tabla está en Primera forma Normal (1FN) si todos los atributos contienen valores atómicos (no repetidos ni agrupados)

Ejemplo: De tabla NO normalizada (violando (1FN)

Estudiante ID	Nombre	Cursos
1	Ana	Inglés-Matemáticas

Tabla en 1FN

Estudiante ID	Nombre	Curso
1	Ana	Inglés
1	Ana	Matemáticas

3. Segunda forma Normal (2FN)

• Una tabla está en Segunda forma Normal 2FN si está en 1FN y todos los atributos no clave dependen completamente de la clave principal (no de una parte de ella si es compuesta)

Ejemplo de violación a 2FN:

Estudiante ID	Curso ID	Nombre Estudiante	Curso Nombre
1	101	Ana	Inglés

Aquí NombreEstudiante depende solo de EstudianteID, no de combinación

Tabla en 2FN (dividida):

EstudianteID	NombreEstudiante
1	Ana

CursoID	CursoNombre
101	Inglés

EstudianteID	CursoID
1	101

4. Tercera forma Normal (3FN)

- Una tabla está en Tercera forma Normal 3FN si esta en 2FN y ningún atributo no clave depende de otro atributo no clave

Ejemplo de violación a 3FN:

EmpleadoID	Nombre	DepartamentoID	DepartamentoNombre
1	Ana	10	Recursos Humanos

Aquí DepartamentoNombre depende DepartamentoID, no directamente de la clave primaria EmpleadoID

Tablas en 3FN:

EmpleadoID	Nombre	DepartamentoID
1	Ana	10

DepartamentoID	DepartamentoNombre
10	Recursos Humanos

- Primera forma Normal

Una tabla esta en primera forma Normal 1FN si todos los atributos Contienen valores atomicos (no repetidos ni agrupados).

Ejemplo De tabla No normalizada (violando 1FN)

EstudianteID	Nombre	Cursos
1	Ana	Ingles - Matematicas

Tabla en 1FN

EstudianteID	Nombre	Curso
1	Ana	Inglés
1	Ana	Matemáticas

• Segunda forma Normal 2fN

Una tabla está en segunda forma Normal 2fN si esta en 1fN y todos los atributos No clave dependen Completamente de la clave principal (no de una parte de ella si es compuesta).

Ejemplo de violación a 2fN

EstudianteID	CursoID	NombreEstudiante	CursoNombre
1	101	Ana	Inglés

Aquí NombreEstudiante depende Solo de EstudianteID, no de convinacion

Tabla en 2fN (dividido):

EstudianteID	NombreEstudiante
1	Ana

CursoID	CursoNombre
101	Ingles

EstudianteID	CursoID
1	101

• Tercera forma Normal 3fN

Una tabla está en Tercera forma Normal 3fN si esta en 2fN y ningún atributo no clave depende de otro atributo no clave

Ejemplo de violacion a 3fN

EmpleadoID	Nombre	DepartamentoID	DepartamentoNombre
1	Ana	10	RecursosHumanos

Aquí DepartamentoNombre depende DepartamentoID, no directamente de la clave primaria EmpleadoID

Tablas en 3fN:

EmpleadoID	Nombre	DepartamentoID
1	Ana	10

DepartamentoID	DepartamentoNombre
10	RecursosHumanos

• Primera forma Normal (1fN)

Una tabla está en primera forma Normal (1fN) si todos los atributos contiene valores atómicos (no repetidos ni agrupados)

Ejemplo: De tabla No normalizada (violando 1fN)

EstudianteID	Nombre	Cursos
1	Ana	Inglés - Matematicas

Tabla en 1FN

EstudianteID	Nombre	Curso
1	Ana	Ingles
1	Ana	Matemáticas

• Segunda forma Normal 2FN

Una tabla está en Segunda forma Normal 2FN si esta en 1FN y todos los atributos no clave dependen completamente de la clave principal (No de una parte de ella si es compuesta)

Ejemplo de violacion a 2FN:

EstudianteID	CursoID	NombreEstudiante	CursoNombre
1	101	Ana	Inglés

Aqui NombreEstudiante depende solo de EstudianteID no de convinacion.

Tabla en 2FN (dividido):

EstudianteID	Ana
1	Ana

CursoID	CursoNombre
101	Ingles

• Tercera forma Normal (3FN)

Una tabla está en trecera forma Normal 3FN si esta en 2FN y ningún atributo no clave depende de otro atributo no clave

EmpleadoID	Nombre	DepartamentoID	DepartamentoNombre
1	Ana	10	Recursos Humanos

Aquí departamentoNombre depende DepartamentoID no directamente de la clave primaria EmpleadoID

Tablas en 3FN:

EmpleadoID	Nombre	DepartamentoID
1	Ana	10

DepartamentoID	DepartamentoNombre
10	Recursos Humanos