



DB Mi Financiera Demo

MTI JESÚS E. ROMERO MORENO

DIVISIÓN DE TECNOLOGÍAS DE LA INFORMACIÓN
Y COMUNICACIÓN

DB Mi Financiera Demo Nivel I

Material de apoyo prácticas de laboratorio

Objetivo y Contexto de Uso de la Base de Datos "mi_financiera_demo"

La base de datos "**mi_financiera_demo**" ha sido diseñada con un enfoque educativo y práctico, destinada a estudiantes universitarios que se especializan en áreas como administración de bases de datos, sistemas de información y finanzas. Este recurso ficticio simula las operaciones de una institución financiera, proporcionando un entorno realista para aprender y practicar habilidades clave en la gestión y optimización de información en empresas del sector financiero.

Objetivo Principal

El objetivo primordial de "**mi_financiera_demo**" es funcionar como un recurso didáctico que permita a los estudiantes:

1. **Comprender y aplicar conceptos esenciales** del diseño y manejo de bases de datos.
2. **Experimentar con procesos complejos**, como la normalización de datos, la creación de relaciones entre tablas, la implementación de índices para optimizar consultas y la gestión de datos específicos al ámbito financiero.
3. **Familiarizarse con el análisis y manejo de grandes volúmenes de datos**, preparándolos para enfrentar escenarios reales en su vida profesional.

Este enfoque no solo permite asimilar conocimientos teóricos, sino también adquirir experiencia práctica, un elemento crucial en su formación académica.

Contexto de Uso

La base de datos está estructurada para simular el flujo de información dentro de una entidad financiera típica, abordando aspectos como:

- Gestión de sucursales, empleados y clientes.
- Administración de productos financieros como préstamos y pagos.
- Seguimiento de cobros y desempeño de la institución.

Este escenario educativo proporciona un **entorno seguro y realista** donde los estudiantes pueden experimentar y probar diversas estrategias de diseño y optimización de bases de datos. Además, el sistema fomenta un aprendizaje activo, permitiendo a los usuarios abordar desafíos reales sin riesgos asociados a errores en ambientes productivos.

Crecimiento y Adaptabilidad

La base de datos "**mi_financiera_demo**" ha sido diseñada con un enfoque en la **escalabilidad y flexibilidad**, características que reflejan las necesidades de una empresa en constante crecimiento. Los estudiantes tienen la oportunidad de observar y experimentar cómo una base de datos:

- Evoluciona ante cambios en los requerimientos del negocio.
- Se adapta al incremento en el volumen de datos y en la demanda de rendimiento.
- Integra nuevas funcionalidades mientras mantiene su eficiencia y consistencia.

Estas cualidades resaltan la importancia de crear estructuras sólidas y flexibles desde el inicio del diseño de una base de datos.

Naturaleza Ficticia y Propósito Académico

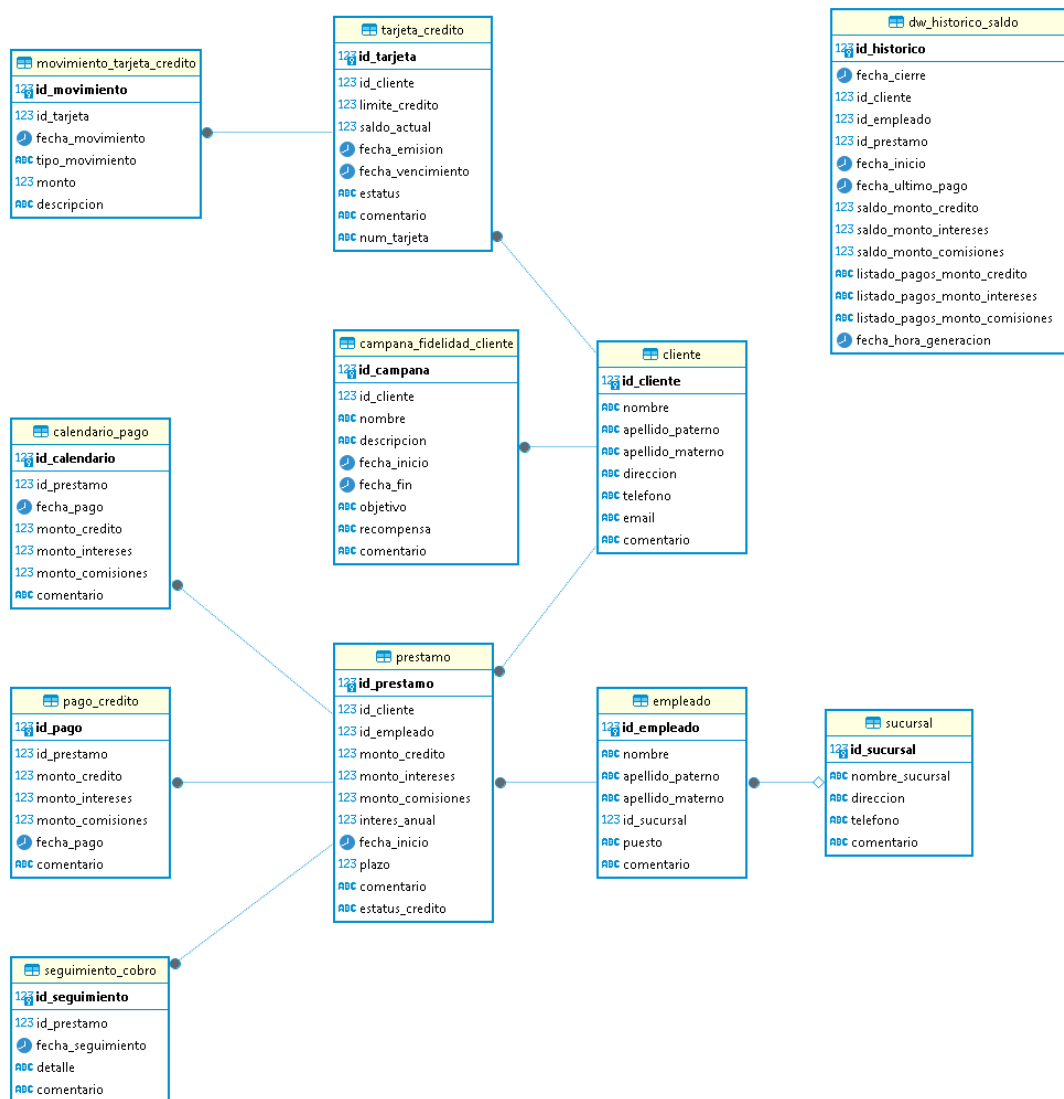
Es fundamental enfatizar que "**mi_financiera_demo**" es un recurso educativo completamente ficticio. No está asociado a ninguna entidad financiera real ni utiliza datos verídicos de clientes o transacciones. Su propósito principal es proporcionar un entorno controlado donde los estudiantes puedan:

- Explorar conceptos teóricos en la práctica.
- Cometer errores y aprender de ellos sin consecuencias graves.
- Desarrollar habilidades técnicas y analíticas en un entorno diseñado específicamente para su crecimiento académico.

En resumen, la base de datos **"mi_financiera_demo"** se presenta como una herramienta esencial para estudiantes interesados en el campo de las bases de datos y sistemas de información. Ofrece un contexto práctico y seguro que complementa la formación teórica, fomentando un aprendizaje activo y aplicado. Su diseño flexible y escalable garantiza una experiencia educativa enriquecedora, adaptada a las necesidades dinámicas del entorno académico y profesional.

Estructura de la Base de Datos

En esta sección, se describen las tablas que componen la base de datos "mi_financiera_demo", incluyendo el objetivo de cada tabla.



1. Tabla sucursal

- **Objetivo:** Almacenar información sobre las sucursales de la institución financiera.
- **id_sucursal** (INT PK autoincremento)
- **nombre_sucursal** (VARCHAR(50))
- **direccion** (VARCHAR(100))
- **telefono** (VARCHAR(15)): Considerando el formato de teléfono en México.
- **comentario** (TEXT)
- **Índices:**
 - Índice en **nombre_sucursal** para búsquedas por nombre.

2. Tabla empleado

- **Objetivo:** Almacenar información sobre los empleados de la institución financiera.
- **id_empleado** (INT PK autoincremento)
- **nombre** (VARCHAR(50))
- **apellido_paterno** (VARCHAR(50))
- **apellido_materno** (VARCHAR(50))
- **id_sucursal** (INT FK a **sucursal.id_sucursal**)
- **puesto** (VARCHAR(50))
- **comentario** (TEXT)
- **Índices:**
 - Índice en **id_sucursal** para mejorar las consultas por sucursal.
 - Índice en **nombre**, **apellido_paterno** y **apellido_materno** para búsquedas por apellidos y nombre.

3. Tabla cliente

- **Objetivo:** Almacenar información sobre los clientes de la institución financiera.
- **id_cliente** (INT PK autoincremento)
- **nombre** (VARCHAR(50))
- **apellido_paterno** (VARCHAR(50))
- **apellido_materno** (VARCHAR(50))
- **direccion** (VARCHAR(100))
- **telefono** (VARCHAR(15))
- **email** (VARCHAR(100))
- **comentario** (TEXT)
- **Índices:**
 - Índice en **nombre**, **apellido_paterno** y **apellido_materno**.

4. Tabla tarjeta_credito

- **Objetivo:** Almacenar información sobre las tarjetas de crédito emitidas a los clientes.
- **id_tarjeta** (INT PK autoincremento)
- **id_cliente** (INT FK a **cliente.id_cliente**)
- **limite_credito** (DECIMAL(12, 2))
- **saldo_actual** (DECIMAL(12, 2))
- **fecha_emision** (DATE)
- **fecha_vencimiento** (DATE)
- **estatus** (VARCHAR(20))
- **num_tarjeta** (VARCHAR(16))
- **Índices:**
 - Índice en **id_cliente** para facilitar la búsqueda de tarjetas por cliente.

5. Tabla movimiento_tarjeta_credito

- **Objetivo:** Registrar los movimientos financieros realizados con las tarjetas de crédito.
- **id_movimiento** (INT PK autoincremento)

- **id_tarjeta** (INT FK a **tarjeta_credito.id_tarjeta**)
- **fecha_movimiento** (DATE)
- **tipo_movimiento** (VARCHAR(20))
- **monto** (DECIMAL(12, 2))
- **descripcion** (TEXT)
- **Índices:**
 - Índice en **id_tarjeta** y **fecha_movimiento** para optimizar la búsqueda de movimientos.

6. Tabla **prestamo**

- **Objetivo:** Almacenar información sobre los préstamos otorgados a los clientes.
- **id_prestamo** (INT PK autoincremento)
- **id_cliente** (INT FK a **cliente.id_cliente**)
- **id_empleado** (INT FK a **empleado.id_empleado**)
- **monto_credito** (DECIMAL(12, 2))
- **monto_intereses** (DECIMAL(12, 2))
- **monto_comisiones** (DECIMAL(12, 2))
- **interes_anual** (DECIMAL(5, 2))
- **plazo** (INT)
- **comentario** (TEXT)
- **estatus_credito** (VARCHAR(20))
- **Índices:**
 - Índice en **id_cliente** para facilitar la búsqueda de préstamos por cliente.
 - Índice en **id_empleado** para facilitar la búsqueda de préstamos por empleado.
 - Índice en **fecha_inicio** para consultas basadas en la fecha de inicio del préstamo.

7. Tabla **calendario_pago**

- **Objetivo:** Registrar el calendario de pagos de los préstamos.
- **id_calendario** (INT PK autoincremento)
- **id_prestamo** (INT FK a **prestamo.id_prestamo**)
- **fecha_pago** (DATE)
- **monto_credito** (DECIMAL(12, 2))
- **monto_intereses** (DECIMAL(12, 2))
- **monto_comisiones** (DECIMAL(12, 2))
- **comentario** (TEXT)
- **Índices:**
 - Índice en **id_prestamo** y **fecha_pago** para mejorar la eficiencia en la consulta de pagos programados.

8. Tabla **pago_credito**

- **Objetivo:** Registrar los pagos realizados por los clientes en relación con los préstamos.
- **id_pago** (INT PK autoincremento)
- **id_calendario** (INT FK a **calendario_pago.id_calendario**)
- **monto_credito** (DECIMAL(12, 2))
- **monto_intereses** (DECIMAL(12, 2))
- **monto_comisiones** (DECIMAL(12, 2))
- **fecha_pago** (DATE)
- **comentario** (TEXT)
- **Índices:**
 - Índice en **id_calendario** y **fecha_pago** para optimizar la búsqueda de pagos.

9. Tabla **seguimiento_cobro**

- **Objetivo:** Registrar las acciones de seguimiento de cobro realizadas sobre los préstamos.
- **id_seguimiento** (INT PK autoincremento)
- **id_prestamo** (INT FK a **prestamo.id_prestamo**)
- **fecha_seguimiento** (DATE)
- **detalle** (TEXT)
- **comentario** (TEXT)
- **Índices:**
 - Índice en **id_prestamo** y **fecha_seguimiento** para un seguimiento eficiente.

10. Tabla **campana_fidelidad_cliente**

- **Objetivo:** Almacenar información sobre las campañas de fidelidad dirigidas a los clientes.
- **id_campana** (INT PK autoincremento)
- **id_cliente** (INT FK a **cliente.id_cliente**)
- **nombre** (VARCHAR(50))
- **fecha_inicio** (DATE)
- **fecha_fin** (DATE)
- **objetivo** (VARCHAR(100))
- **recompensa** (VARCHAR(100))
- **comentario** (TEXT)
- **Índices:**
 - Índice en **id_cliente** para facilitar la búsqueda de campañas por cliente.

11. Tabla **dw_historico_saldo**

- **Objetivo:** Almacenar el historial de saldos de los clientes, empleados y préstamos.
- **id_historico** (INT PK autoincremento)
- **fecha_cierre** (DATE)
- **id_cliente** (INT FK a **cliente.id_cliente**)
- **id_empleado** (INT FK a **empleado.id_empleado**)
- **id_prestamo** (INT FK a **prestamo.id_prestamo**)
- **fecha_inicio** (DATE)
- **fecha_ultimo_pago** (DATE)
- **saldo_monto_credito** (DECIMAL(12, 2))
- **saldo_monto_intereses** (DECIMAL(12, 2))
- **saldo_monto_comisiones** (DECIMAL(12, 2))
- **listado_pagos_monto_credito** (TEXT)
- **listado_pagos_monto_intereses** (TEXT)
- **listado_pagos_monto_comisiones** (TEXT)
- **fecha_hora_generacion** (TIMESTAMP DEFAULT CURRENT_TIMESTAMP)
- **Índices:**
 - Índice en **id_cliente** para facilitar la búsqueda de historiales por cliente.
 - Índice en **id_empleado** para facilitar la búsqueda de historiales por empleado.
 - Índice en **id_prestamo** para facilitar la búsqueda de historiales por préstamo.

Relaciones entre Tablas

- **Empleado - Sucursal:** Cada empleado está vinculado a una sucursal.
- **Empleado - Prestamo:** Cada empleado tipo asesor de crédito tiene múltiples préstamos otorgados.
- **Cliente - Prestamo:** Un cliente puede tener múltiples préstamos.
- **Prestamo - Pago_credito/Calendario_pago/Seguimiento_cobro:** Cada préstamo tiene asociados pagos, un calendario de pagos y seguimientos de cobro.

- **Cliente - Tarjeta_credito:** Un cliente puede tener múltiples tarjetas de crédito.
- **Tarjeta_credito - Movimiento_tarjeta_credito:** Cada tarjeta de crédito tiene asociados múltiples movimientos.
- **Cliente - Campana_fidelidad_cliente:** Un cliente puede participar en múltiples campañas de fidelidad.
- **Cliente/Empleado/Prestamo - dw_historico_saldo:** Cada cliente, empleado y préstamo tiene su historial de saldos registrado.

Consideraciones por mejorar.

Seguridad de Datos:

Es crucial implementar medidas de seguridad para proteger datos sensibles, especialmente en una base de datos financiera.

Integridad Referencial:

Las relaciones entre las tablas se manejan a través de llaves foráneas, asegurando la integridad de los datos.

Escalabilidad:

El diseño debe ser capaz de manejar el crecimiento en el número de registros y las consultas concurrentes, se requiere un programa de mantenimiento y continuidad operativa.

Nota para Estudiantes:

Existe la oportunidad de Mejora y Optimización en la Base de Datos "mi_financiera_demo"

Estimados Estudiantes,

Quiero recordarles que la base de datos "mi_financiera_demo" utilizada en nuestro curso está abierta a mejoras y modificaciones por parte de ustedes. Esta es una oportunidad única para aplicar sus conocimientos y habilidades en un escenario práctico y dinámico.

Mejoras y Correcciones:

A lo largo del curso, se les anima a identificar y realizar mejoras en el diseño de la base de datos. Esto puede incluir la corrección de errores que descubran, ajustes en la estructura de las tablas o la implementación de nuevos elementos que consideren necesarios. La idea es que puedan experimentar con el diseño y la optimización de bases de datos en un entorno seguro y controlado.

Adaptación a Nuevos Escenarios:

Además, a medida que avancemos en el curso, se introducirán nuevos escenarios y requisitos. Esto les desafiará a adaptar y modificar "mi_financiera_demo" para que se ajuste a estas nuevas necesidades. Cada cambio en el escenario es una oportunidad para practicar la adaptación y escalabilidad de las bases de datos en situaciones que simulan el mundo real.

Feedback y Evaluación:

Sus contribuciones y modificaciones serán parte integral de su **aprendizaje y evaluación** en el curso. Se alienta la discusión abierta sobre los cambios realizados, permitiendo que el intercambio de ideas y el feedback constructivo sean parte del proceso educativo.

Desarrollo de Habilidades:

Este modelo de enseñanza está diseñado para desarrollar habilidades críticas en el campo de las bases de datos, incluyendo el pensamiento analítico, la resolución de problemas y la adaptabilidad a situaciones cambiantes. Al final del curso, esperamos que hayan adquirido no solo conocimientos técnicos, sino también una comprensión más profunda de cómo las bases de datos pueden evolucionar y adaptarse en un entorno empresarial.

Archivos Adicionales:

Para facilitar la implementación de la base de datos "mi_financiera_demo", se proporcionan los siguientes archivos:

- **Script SQL:** Un archivo "mi_financiera_demo.sql" que permite crear la estructura de la base de datos en PostgreSQL desde consola o desde Dbeaver.
- **Script de Sembrado:** Un archivo "sembrado_mi_financiera_demo_postgresql.js" en Node.js para el llenado de datos ficticios usando la librería Faker.

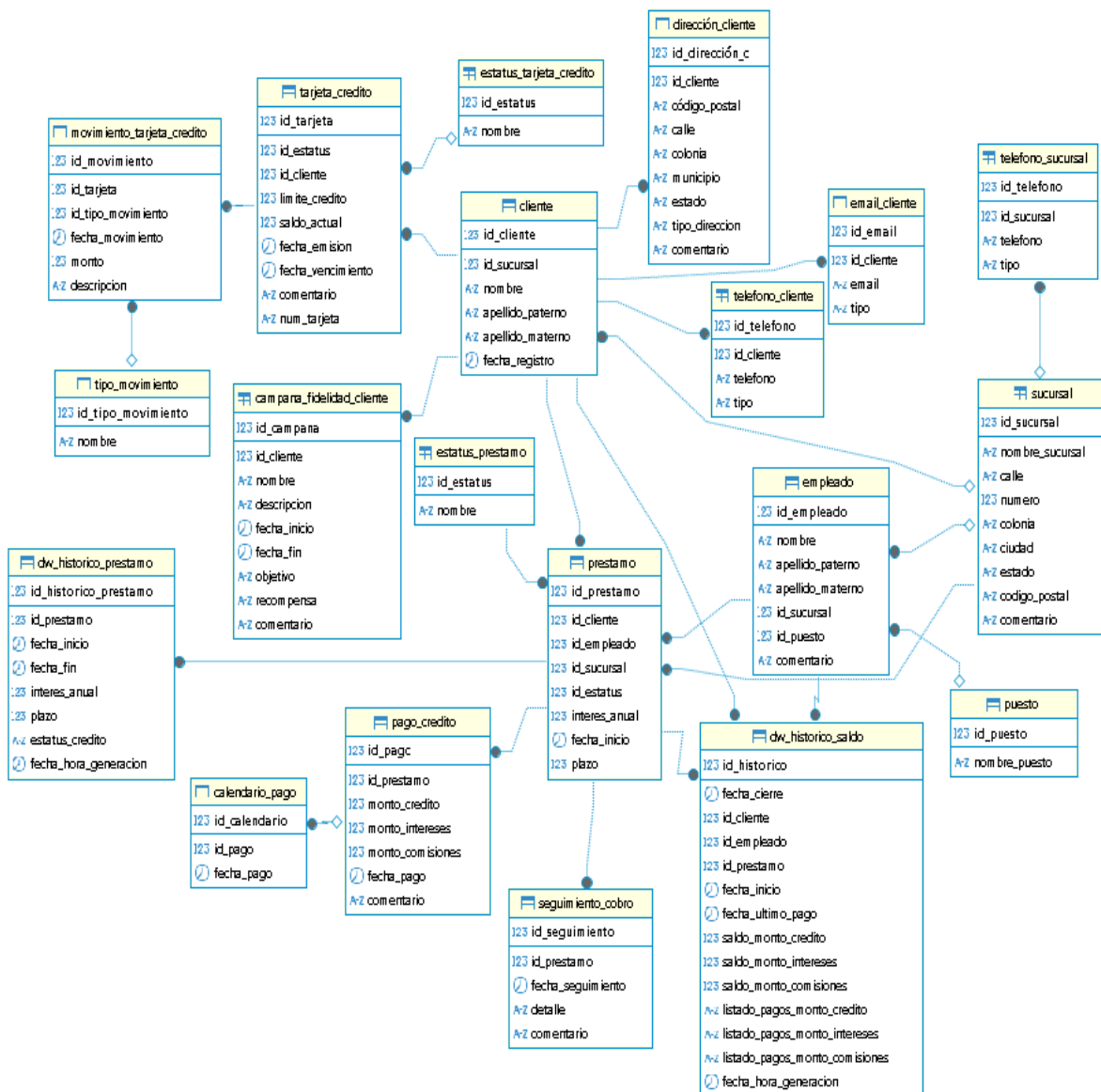
Propuesta de Mejora de la BD Mi Financiera

Se presenta la propuesta de mejora de la base de datos Mi Financiera Demo brindada por el profesor, identificando las oportunidades para optimizar su estructura enfatizando en la normalización y mejoras de las tablas. El enfoque principal fue:

1. Normalizar estructuras repetitivas o variables
2. Mejorar la gestión de relaciones entre entidades
3. Facilitar la generación de reportes estratégicos
4. Garantizar la integridad referencial de los datos

Estructura de la Base de Datos Mejorada

A continuación, se describen las mejoras de las tablas que componen la base de datos, incluyendo el objetivo y desglose de estas. La siguiente imagen muestra el diagrama ER de la base de datos resultante.



1. Normalización de Estructuras Variables

Primeramente, se agregaron tablas normalizadas para aquellos campos que pueden variar con el paso del tiempo. Dando como resultado:

- **Tabla puesto**
 - **Objetivo:** Gestionar los cambios de los diferentes puestos que un empleado puede tener dentro de la empresa.
 - **id_puesto** SERIAL PRIMARY KEY
 - **nombre_puesto** VARCHAR(50) NOT NULL
- **Tabla estatus_tarjeta_credito**
 - **Objetivo:** Facilitar el seguimiento de los posibles estados de una tarjeta (Activa, Bloqueada, Cancelada, etc).
 - **id_estatus** SERIAL PRIMARY KEY
 - **nombre** VARCHAR(50) NOT NULL
- **Tabla estatus_prestamo**
 - **Objetivo:** Facilitar el seguimiento del estado de los préstamos.
 - **id_estatus** SERIAL PRIMARY KEY
 - **nombre** VARCHAR(50) NOT NULL DEFAULT 'Activo'
- **Tabla tipo_movimiento**
 - **Objetivo:** Clasificar los movimientos de la tarjeta (Compra, Pago, Reembolso, etc.).
 - **id_tipo_movimiento** SERIAL PRIMARY KEY
 - **nombre** VARCHAR(50) NOT NULL

2. Gestión Mejorada de Información de Contacto

- **Tabla telefono_sucursal**
 - **Objetivo:** Permitir que una sucursal pueda tener distintos números de teléfono clasificados por tipo
 - **id_telefono** SERIAL PRIMARY KEY
 - **id_sucursal** INT
 - **telefono** VARCHAR(15) NOT NULL
 - **tipo** VARCHAR(50) NOT NULL
- **Tabla email_cliente**
 - **Objetivo:** Permitir que un cliente pueda tener distintos correos electrónicos.
 - **id_email** SERIAL PRIMARY KEY
 - **id_cliente** INT
 - **email** VARCHAR(100) NOT NULL UNIQUE
 - **tipo** VARCHAR(50) NOT NULL
- **Tabla telefono_cliente**
 - **Objetivo:** Permitir que un cliente pueda tener distintos números telefónicos.
 - **id_telefono** SERIAL PRIMARY KEY
 - **id_cliente** INT
 - **telefono** VARCHAR(15) NOT NULL UNIQUE
 - **tipo** VARCHAR(50) NOT NULL

- **Tabla dirección_cliente**

- **Objetivo:** Almacenar las diferentes direcciones en las que puede radicar un cliente, estas pueden cambiar con el paso del tiempo por eso la necesidad de crear otra tabla.
- **id_dirección_c** SERIAL PRIMARY KEY,
- **id_cliente** INT NOT NULL,
- **código_postal** VARCHAR(10),
- **calle** VARCHAR(100),
- **colonia** VARCHAR(50),
- **municipio** VARCHAR(50),
- **estado** VARCHAR(50),
- **tipo_direccion** VARCHAR(20)

3. Mejoras en Estructuras Principales

- **Tabla sucursal (modificada)**

- **Objetivo:** Atomizar el campo de la dirección para facilitar la creación de reportes por municipio, o estado, etc.
- **id_sucursal** SERIAL PRIMARY KEY,
- **nombre_sucursal** VARCHAR(50) NOT NULL,
- **calle** VARCHAR(100) NOT NULL,
- **numero** INT NOT NULL,
- **colonia** VARCHAR(50) NOT NULL,
- **ciudad** VARCHAR(50) NOT NULL,
- **estado** VARCHAR(50) NOT NULL,
- **codigo_postal** VARCHAR(10) NOT NULL

- **Tabla prestamo (modificada)**

- **Objetivo:** Incorporar id_sucursal e id_estatus para mejorar la trazabilidad.
- **id_prestamo** SERIAL PRIMARY KEY,
- **id_cliente** INT NOT NULL,
- **id_empleado** INT NOT NULL,
- **id_sucursal** INT NOT NULL,
- **id_estatus** INT NOT NULL,
- **interes_anual** DECIMAL(5,2) NOT NULL,
- **fecha_inicio** DATE NOT NULL,
- **plazo** INT NOT NULL

4. Data Warehouse Mejorado

- **Tabla dw_historico_prestamo**

- **Objetivo:** Rastrear cambios históricos de los préstamos y no solo de los saldos.
- **id_historico_prestamo** serial4 NOT NULL,
- **id_prestamo** int4 NOT NULL,
- **fecha_inicio** date NOT NULL,
- **fecha_fin** date NOT NULL,
- **interes_anual** numeric(5, 2) NOT NULL,
- **plazo** int4 NOT NULL,
- **estatus_credito** varchar(15) NOT NULL DEFAULT 'Activo'

- **Tabla dw_historico_saldo**

- Se incorporaron las relaciones de las llaves foráneas, ya que la base original no las tenía.

CIERRE DE TEMA**DB MI FINANCIERA DEMO**

Por Martínez de la Cruz José Julián

Fecha: 01/06/2025

Palabras clave: Normalización, Data Warehouse, DBeaver, Optimización de rendimiento, Eficiencia de consultas, Encriptación de tarjetas, Seguridad de datos.

1. Objetivo del Tema

El propósito de este tema es aprender los conceptos básicos del funcionamiento de DBeaver mediante el rediseño de una base de datos ficticia con propósitos académicos para mejorar su estructura y normalización con el fin de brindar una gestión robusta de clientes, sucursales, préstamos y tarjetas de crédito.

2. Resumen

La base de datos Mi Financiera Demo fue optimizada mediante las siguientes mejoras:

Normalización de Estructuras

- 1. Tablas de catálogo:** Se agregaron puesto, estatus_tarjeta_credito, estatus_prestamo y tipo_movimiento para centralizar atributos que pueden cambiar con el tiempo (ejemplo: "Activo", "Cancelado"). Para eliminar las redundancias y facilitar actualizaciones.
- 2. Atomización de datos:** La dirección en sucursal se dividió en campos específicos (calle, número, colonia, ciudad, estado, código postal). Para permitir la generación de reportes por municipio, etc.

Gestión de Contactos Mejorada

- **Se crearon tablas especializadas para:**
 - Teléfonos: telefono_cliente y telefono_sucursal
 - Emails: email_cliente.
 - Direcciones: dirección_cliente.

Relaciones y Data Warehouse

Finalmente, se agregaron claves foráneas faltantes (ejemplo: id_sucursal en prestamo). Además del desarrollo de un dw_historico_prestamo para gestionar los cambios en plazos, tasas de interés y estatus de los préstamos.

3. Conclusión

Tras realizar la actividad, pude conocer como influye la normalización en una base de datos para volverla más eficiente. Con las mejoras implementadas no solo se puede optimizar el rendimiento, sino que también facilita las consultas ahorrando recursos.

4. Gráficos o Ejemplos

Diagrama Entidad Relación Original de la Base de Datos

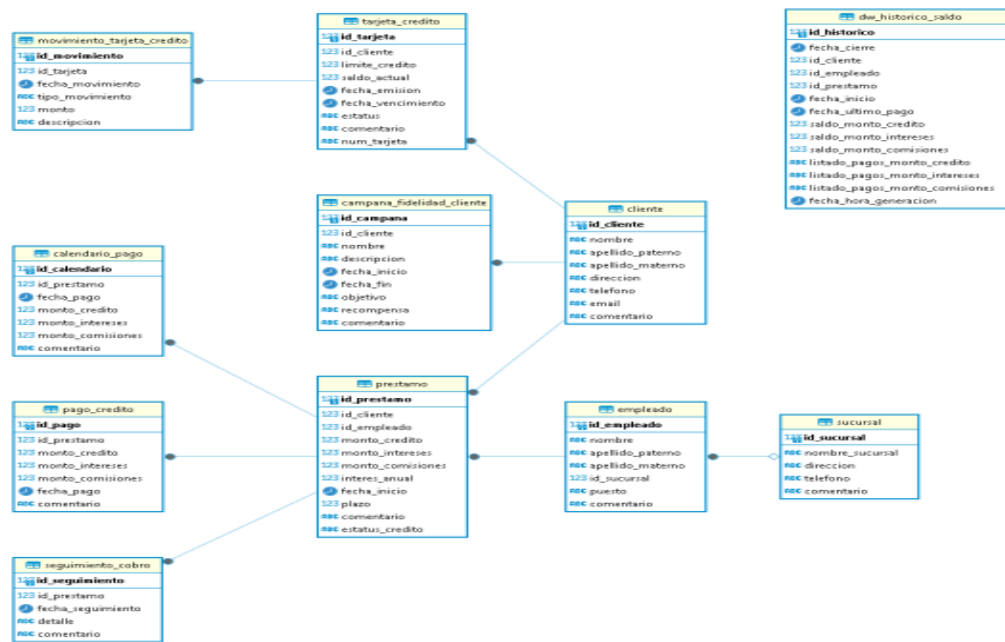
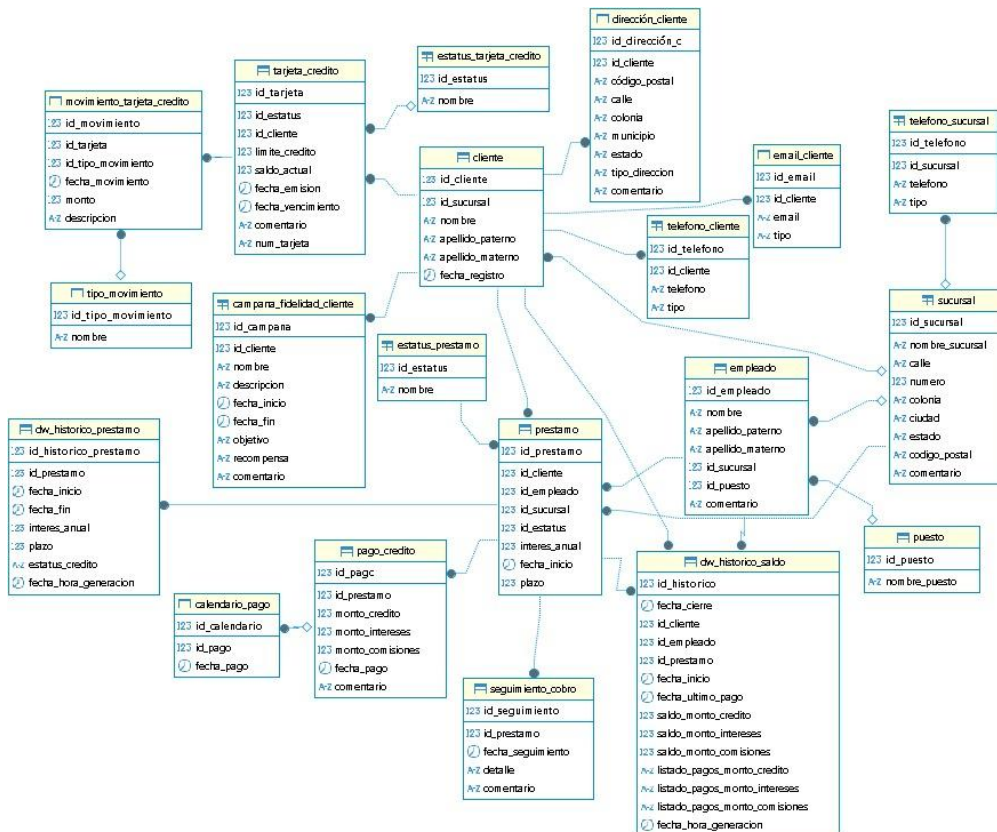


Diagrama Entidad Relación Mejorado de la Base de Datos



5. Referencias

- Presentación: *DB Mi Financiera Demo*, MTI Jesús E. Romero, UTTEC.
- Clase: *Introducción a DBeaver*, MTI Jesús E. Romero, UTTEC.

6. Reflexiones Finales

1. **Habilidades técnicas reforzadas:** Aprendí de mejor manera en qué casos se debe aplicar normalización. Ya que no sabía que si un atributo variaba con el paso del tiempo era necesario crear una tabla entidad para el atributo en cuestión.
2. **Aseguramiento de tarjetas de crédito:** Las tarjetas de los clientes siempre deben viajar encriptadas en todo momento por la base de datos para asegurar que nadie tenga acceso a la misma. Y al realizar operaciones con ella, comparar ambos valores encriptados.