
Proyecto MySQL

Flavio Ambroggio

CoderHouse | Comisión 34950

Introducción

HOME BANKING

La temática que elegí para la base de datos es de un Homebanking el cual va a contener información sobre clientes con sus cuentas, transacciones, tarjetas, préstamos, etc.

Elegí esta temática para complementar con una página web, mediante la cual se pueda gestionar los servicios ofrecidos por una institución bancaria.

Objetivo

HOME BANKING

El objetivo planteado fue crear una base de datos para administrar toda la información correspondiente a los clientes y sus respectivas cuentas. Como así también todo lo relacionado a los diferentes servicios prestados por la institución bancaria.

Situación Problemática

HOME BANKING

A raíz de la cantidad y sensibilidad de la información y los datos necesarios para llevar a cabo las diferentes prestaciones en el rubro bancario surge la necesidad de crear una base de datos fiable donde almacenar los mismos.

Modelo de Negocio

HOME BANKING

- Los clientes pueden registrar un único domicilio pero un mismo domicilio puede pertenecer a varios clientes.
- Los clientes pueden habilitar muchas cuentas pero estas pueden ser solamente de tipo cuenta corriente o caja de ahorro.
- Los servicios se pueden pagar solamente por medio de tarjetas.
- Las operaciones se ven reflejadas en las transacciones e impactan en las cuentas.

Diagrama Entidad Relación

HOME BANKING

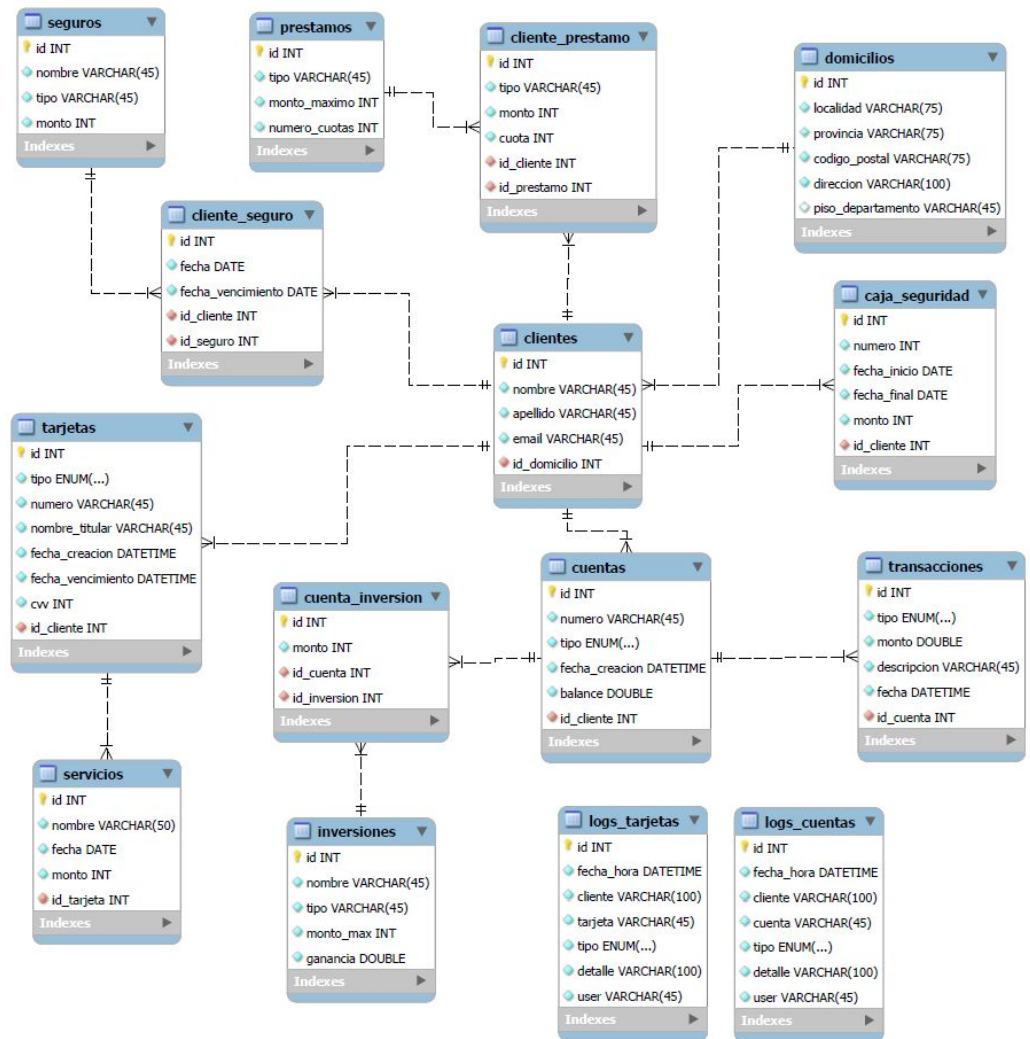


Diagrama Entidad Relación

HOME BANKING

- Se definen dos tablas principales: clientes y cuentas.
- Se definen tres tablas intermedias: cliente_prestamo, cliente_seguro y cliente_inversion.
- Se definen las siguientes tablas estáticas: seguros, prestamos, domicilios, caja_seguridad, transacciones, inversiones, servicios, tarjetas.
- Se definen dos tablas de registros: logs_tarjetas y logs_cuentas.

Vistas

HOME BANKING

- **prestamos_solicitados** - Esta vista devuelve un listado de los clientes que tienen solicitados préstamos, mostrando el tipo de los mismos, el monto solicitado, la cantidad de cuotas como así también el monto por cuota. Su objetivo es mostrar toda la información relevante respecto a los préstamos solicitados por los clientes. Las tablas que la componen son clientes, cliente_prestamo y prestamo.
- **tarjetas_vencidas** - Esta vista devuelve un listado de los clientes que tengan tarjetas vencidas, mostrando el nombre y apellido del cliente, el número de la tarjetas, el tipo y la fecha que venció. Su objetivo es facilitar la información relativa a las tarjetas vencidas. Las tablas que la componen son clientes y tarjetas.
- **transacciones_año_actual** - Esta vista devuelve un listado de los clientes que hayan realizado transacciones durante el año actual. Mostrando el nombre y apellido del cliente, el número de cuenta, tipo de la misma y la cantidad de transacciones. Su objetivo es llevar un registro de la actividad de las cuentas de cada cliente. Las tablas que la componen son clientes, cuentas y transacciones.
- **sumatoria_transacciones** - Esta vista devuelve un listado de los clientes y sus cuentas con movimiento de dinero, y la sumatoria total de los debitos y creditos de dichas cuentas. Su objetivo es llevar un registro de los movimientos de dinero de las cuentas de los clientes. Las tablas que la componen son clientes, cuentas y transacciones.
- **registro_por_cliente** - Esta vista devuelve el listado de los clientes con la cantidad de cuentas, cantidad de préstamos solicitados y la cantidad de tarjetas activas. Su objetivo es mostrar un recuento de la cantidad de servicios brindados a cada cliente. Las tablas que la componen son clientes, cuentas, cliente_prestamo, tarjetas.

Funciones

HOME BANKING

- **get_loans_by_email** - Esta función devuelve la cantidad de préstamos a partir del email del cliente. Su objetivo es agilizar la obtención de esta información a partir del email de cada cliente. Las tablas que manipula son clientes y cliente_prestamo. Por el momento no está implementada
- **amount_per_payments** - Esta función devuelve el monto en pesos por cada cuota de un préstamo. Su objetivo es mostrar de manera rápida el monto correspondiente a cada cuota de algún préstamo. La tabla que manipula es cliente_prestamo. Por el momento no está implementada.
- **cliente_by_cuenta** - Esta función devuelve el nombre y el apellido del cliente a través del número de alguna de sus cuentas. Su objetivo es obtener de manera más rápida el nombre y apellido del cliente mediante el número de cuenta para ser usados en otros procesos. Las tablas que manipula son clientes y cuentas. Está implementada en los triggers.
- **cliente_by_tarjeta** - Esta función devuelve el nombre y el apellido del cliente a través del id de alguna de sus tarjetas. Su objetivo es obtener de manera más rápida el nombre y apellido del cliente mediante el id de alguna de sus tarjetas para ser usados en otros procesos. Las tablas que manipula son clientes y tarjetas. Está implementada en los triggers y stored procedure.

Stored Procedures

HOME BANKING

- **sp_eliminar_tarjetas_vencidas**

Este sp se encarga de eliminar las tarjetas vencidas que tengan los clientes: En el caso de no tener ninguna vencida va a devolver un mensaje de error.

La tabla con la que interactúa es tarjetas.

- **sp_listado_prestamos_ordenados**

Este sp devuelve un listado de todos los préstamos solicitados por clientes y da la posibilidad de ordenarlos por alguna de sus columnas en orden ascendente o descendente.

Las tablas con las que interactúa son clientes y cliente_prestamo.

Triggers

HOME BANKING

- **logs_tarjetas_delete**- Este trigger crea un registro en la tabla logs_tarjetas antes de eliminar una tarjeta de un cliente. Dicho registro consta de un id, la fecha en la que se eliminó la tarjeta, el nombre y apellido del cliente, el número de la tarjeta, el tipo, una breve descripción, y el usuario que la eliminó.
- **logs_tarjetas_insert**- Este trigger crea un registro en la tabla logs_tarjetas después de insertar una nueva tarjeta de un cliente. Dicho registro consta de un id, la fecha en la que se insertó la tarjeta, el nombre y apellido del cliente, el número de la tarjeta, el tipo, una breve descripción, y el usuario que la insertó.
- **logs_cuentas_delete** - Este trigger crea un registro en la tabla logs_cuentas antes de eliminar una cuenta de un cliente. Dicho registro consta de un id, la fecha en la que se eliminó la cuenta, el nombre y apellido del cliente, el número de la cuenta, el tipo, una breve descripción, y el usuario que la eliminó.
- **logs_cuentas_update** - Este trigger crea un registro en la tabla logs_cuentas después de actualizar una cuenta de un cliente. Dicho registro consta de un id, la fecha en la que se actualizó la cuenta, el nombre y apellido del cliente, el número de la cuenta, el tipo, una breve descripción, y el usuario que la actualizó.
- **servicios_insert** - Este trigger crea una nueva transacción después de que se paga un servicio con alguna tarjeta.

Contenido...

Base de datos completa

Backup_Homebanking_Final.sql

Scripts

DB_Homebanking_Final.sql

Vistas_Final.sql

Funciones_Final.sql

Stored_Procedures_Final.sql

Triggers_Final.sql

Extras

DER_Homebanking_Final.pdf