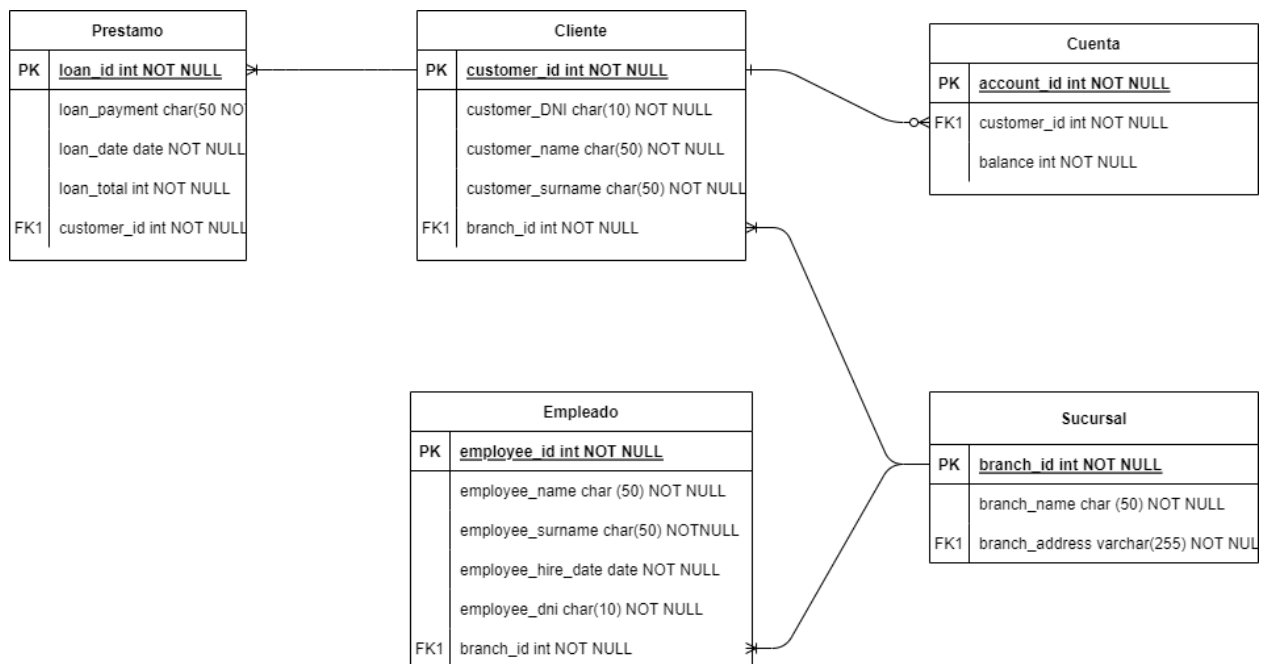


## Introducción

ITBANK tiene un modelo de datos definido para la operación de su negocio. El modelo de datos permite conocer:

- Los tipos de los datos que hay en la base y la forma en que se relacionan.
- Las restricciones de integridad
- Entender cómo llevar adelante operaciones de manipulación de los datos

Se adjunta el siguiente extracto con las principales entidades:



## Primera problemática

Dada la base de datos ITBANK, se solicita ampliar el modelo de datos conocido con las nuevas entidades generadas por el home banking.

1. Crear en la base de datos los tipos de cliente, de cuenta y marcas de tarjeta. Insertar los valores según la información provista en el Sprint
2. Agregar la entidad tarjeta teniendo en cuenta los atributos necesarios para la operación del home banking. Se sugieren los siguientes campos: Numero (único e irrepitable, con una restricción ante cada inserción que no debe superar 20 números/espacios), CVV, Fecha de otorgamiento, Fecha Expiración. Almacenar si es una tarjeta de credito o debito
3. Relacionar las tarjetas con la tabla donde se guardan las marcas de tarjeta
4. Relacionar las tarjetas con el cliente al que pertenecen
5. Insertar 500 tarjetas de crédito con sus respectivos datos ([www.generatedata.com](http://www.generatedata.com)) asociándolas a los clientes de forma aleatoria
6. Agregar la entidad direcciones, que puede ser usada por los clientes, empleados y sucursales con los campos utilizados en el SPRINT 5
7. Insertar 500 direcciones, asignando del lote inicial a clientes, empleados, o sucursal, de forma aleatoria. Teniendo en cuenta que un cliente o empleado puede tener múltiples direcciones, pero la sucursal, solo una.
8. Ampliar el alcance de la entidad cuenta para que identifique el tipo de misma
9. Asignar un tipo de cuenta a cada registro de cuenta de forma aleatoria
10. Corregir el campo employee\_hire\_date de la tabla empleado con la fecha en formato YYYY-MM-DD

Para tal fin se recomienda usar las sentencias CREATE, ALTER, INSERT y UPDATE

## Segunda problemática

Una vez realizado los cambios en la base de datos, se pide entregar un archivo SQL con los siguientes requerimientos de consulta frecuente.

- 1 • Crear una vista con las columnas id, numero sucursal, nombre, apellido, DNI y edad de la tabla cliente calculada a partir de la fecha de nacimiento
  - 1.a o Mostrar las columnas de los clientes, ordenadas por el DNI de menor a mayor y cuya edad sea superior a 40 años
  - 1.b o Mostrar todos los clientes que se llaman "Anne" o "Tyler" ordenados por edad de menor a mayor

JSON EN LA PAGINA SIGUIENTE...

- 2 • Actualizar 5 clientes recientemente agregados en la base de datos dado que hubo un error en el JSON que traía la información, la sucursal de todos es la 10
- 3 • Eliminar el registro correspondiente a "Noel David" realizando la selección por el nombre y apellido Si no existe, crearlo
- 4 • Consultar sobre cuál es el tipo de préstamo de mayor importe

- Dado el siguiente JSON. Insertar 5 nuevos clientes en la base de datos y verificar que se haya realizado con éxito la inserción

```
[
  {
    "customer_name": "Lois",
    "customer_surname": "Stout",
    "customer_DNI": 47730534,
    "branch_id": 80,
    "customer_dob": "1984-07-07"
  },
  {
    "customer_name": "Hall",
    "customer_surname": "Mcconnell",
    "customer_DNI": 52055464,
    "branch_id": 45,
    "customer_dob": "1968-04-30"
  },
  {
    "customer_name": "Hilel",
    "customer_surname": "Mclean",
    "customer_DNI": 43625213,
    "branch_id": 77,
    "customer_dob": "1993-03-28"
  },
  {
    "customer_name": "Jin",
    "customer_surname": "Cooley",
    "customer_DNI": 21207908,
    "branch_id": 96,
    "customer_dob": "1959-08-24"
  },
  {
    "customer_name": "Gabriel",
    "customer_surname": "Harmon",
    "customer_DNI": 57063950,
    "branch_id": 27,
    "customer_dob": "1976-04-01"
  }
]
```

## Tercera problemática

- 1 • Seleccionar las cuentas con saldo negativo **si no existen, crearlas**
- 2 • Seleccionar el nombre, apellido y edad de los clientes que tengan en el apellido la letra Z **... 'LIKE '%%z'**
- 3 • Seleccionar el **(nombre, apellido, edad y nombre de sucursal)** de las personas cuyo nombre sea "Brendan" y el resultado ordenarlo por nombre de sucursal
- 4 • Seleccionar de la tabla de préstamos, los préstamos con un importe mayor a \$80.000 y los préstamos prendarios utilizando la unión de tablas/consultas (recordar que en las bases de datos la moneda se guarda como integer, en este caso con 2 centavos) **inner / left join tabla1 -- tabla2 where tabla 1...**
- 5 • Seleccionar los prestamos cuyo importe sea mayor que el importe medio de todos los prestamos **select \* from prestamos where prestamos > select avg(todos los prestamos)**
- 6 • Contar la cantidad de clientes menores a 50 años **count(select ... where edad\_cliente < 50)**
- 7 • Seleccionar las primeras 5 cuentas con saldo mayor a 8.000\$ **where ID between 1 and 5**
- 8 • Seleccionar los préstamos que tengan fecha en abril, junio y agosto, ordenándolos por importe **order by fecha, importe**
- 9 • Obtener el importe total de los prestamos agrupados por tipo de préstamos. Por cada tipo de préstamo de la tabla préstamo, calcular la suma de sus importes. Renombrar la columna como loan\_total\_accu

**select sum(columna) from ... group by ...**

## Cuarta problemática

- 1 • Listar la cantidad de clientes por nombre de sucursal ordenando de mayor a menor **count(clientes) group by sucursal order by count(clientes) desc**
- 2 • Obtener la cantidad de empleados por cliente por sucursal en un número real **count(empleados) / count(clientes) as relacion empleados/clientes group by sucursal**
- 3 • Obtener la cantidad de tarjetas de crédito por tipo por sucursal
- 4 • Obtener el promedio de créditos otorgado por sucursal
- 5 • La información de las cuentas resulta crítica para la compañía, por eso es necesario crear una tabla denominada "*auditoria\_cuenta*" para guardar los datos movimientos, con los siguientes campos: *old\_id, new\_id, old\_balance, new\_balance, old\_iban, new\_iban, old\_type, new\_type, user\_action, created\_at*
  - 5.1 ○ Crear un trigger que después de actualizar en la tabla cuentas los campos balance, IBAN o tipo de cuenta registre en la tabla auditoria ?
  - 5.2 ○ Restar \$100 a las cuentas 10,11,12,13,14
- 6 • Mediante índices mejorar la performance la búsqueda de clientes por DNI ?
- 7 • Crear la tabla "movimientos" con los campos de identificación del movimiento, número de cuenta, monto, tipo de operación y hora
  - 7.1 ○ Mediante el uso de transacciones, hacer una transferencia de 1000\$ desde la cuenta 200 a la cuenta 400
  - 7.2 ○ Registrar el movimiento en la tabla movimientos
  - 7.3 ○ En caso de no poder realizar la operación de forma completa, realizar un ROLLBACK