## SECCIÓN 1: AMAZON REDSHIFT

### CONTEXTO:

En Podemos Progresar, utilizamos Redshift como nuestra solución principal de data warehouse. Debido al rápido crecimiento de nuestra cartera de microcréditos grupales en México, el volumen de datos ha crecido exponencialmente, creando desafíos de rendimiento.

### Escenario:

Como Data Engineer con responsabilidades de arquitectura, debes proponer una arquitectura de Redshift escalable para soportar nuestro crecimiento continuo. El data warehouse debe manejar eficientemente datos de nuestros ciclos de préstamos grupales, comportamiento de pago semanal, y métricas de desempeño de grupos y oficiales de crédito.

### Tareas:

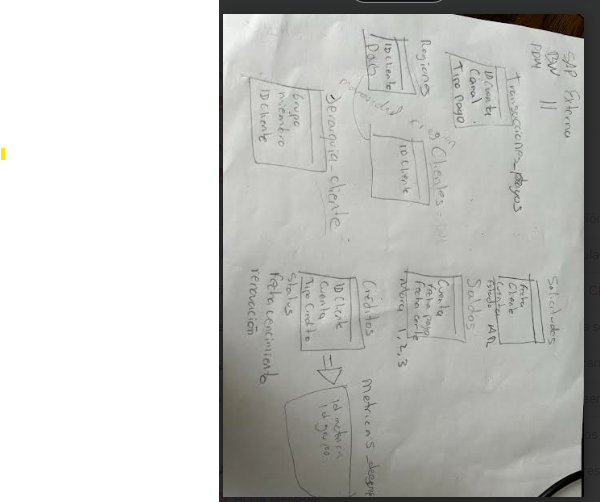
1.1. Arquitectura escalable para microcréditos grupales:

- Diseña una arquitectura de Redshift optimizada para nuestro modelo de negocio de créditos grupales, considerando:

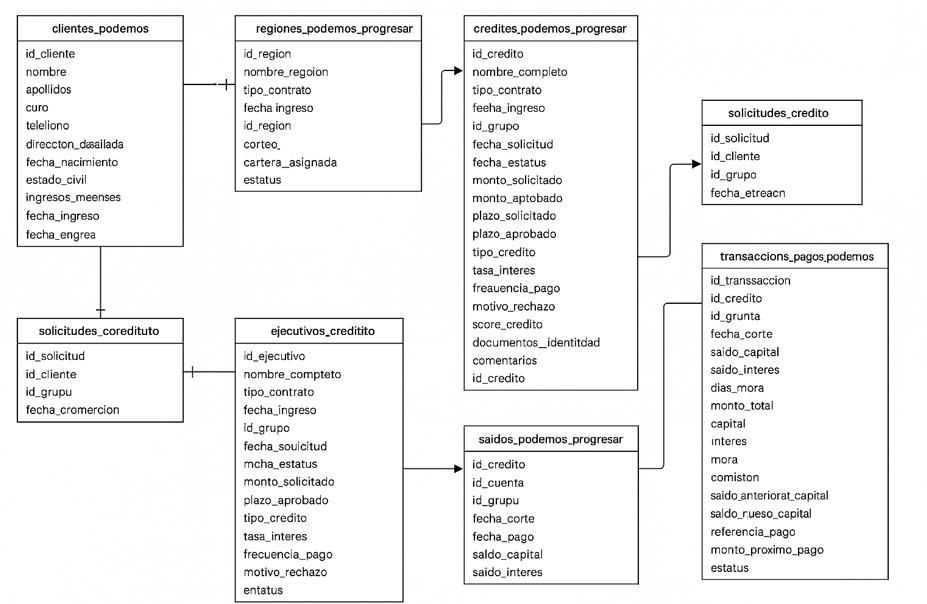
\* Estructura para manejar la naturaleza jerárquica de nuestros datos (grupos > miembros individuales)

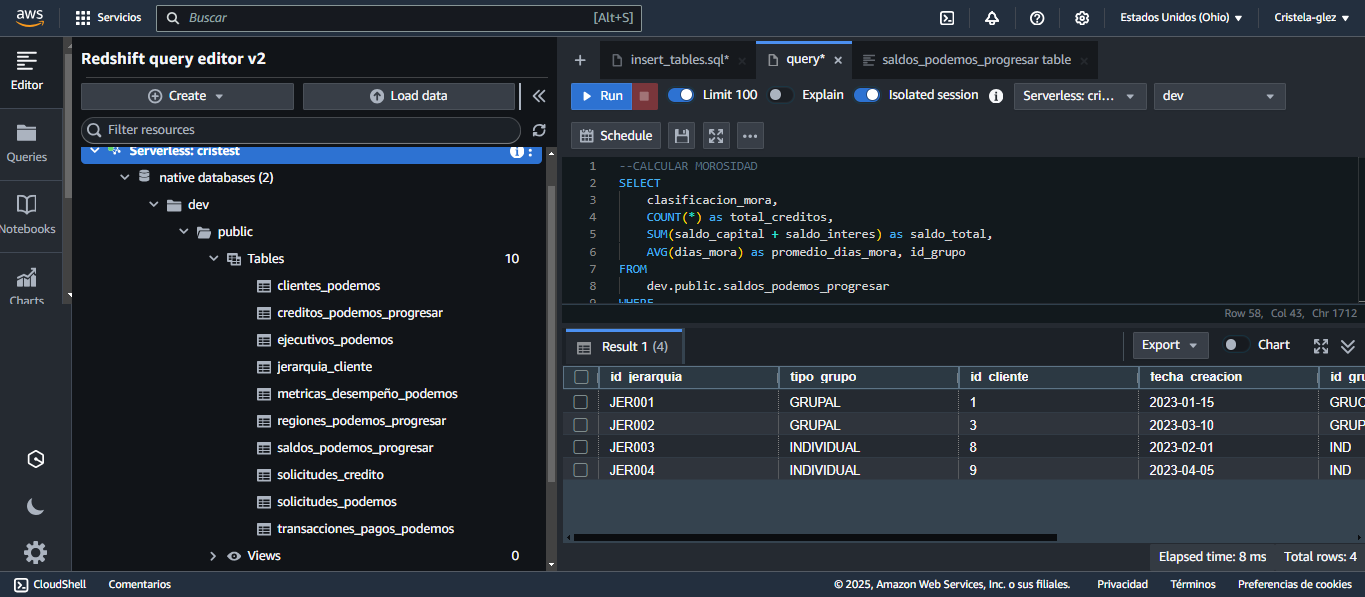
**R**. Diseñé un modelo de datos basándome en un diagrama entidad relación, definiendo cada una de las entidades con campos y llaves que creí conveniente definir para contestar cada uno de los escenarios que se mencionan. Adicional hice una cuenta gratuita en Amazon AWS cargando datos de prueba en formato CSV, donde cree un workspace, posteriormente un rol y un bucket para poder crear dichas tablas.

Borrador del diagrama E-R



El diagrama entidad relación es el siguiente:



Estructura en redshift:

Se creo una tabla para visualizar la clasificación de los grupos en relación con el cliente y región a la que pertenece el grupo:

Tabla:



CREATE TABLE public.jerarquia\_cliente (

id\_jerarquia varchar(20) NOT NULL,

tipo\_grupo varchar(50),

id\_cliente varchar(10) NOT NULL ,

fecha\_creacion date,

id\_grupo varchar(20) NOT NULL,

id\_region integer,

estatus varchar(150),

PRIMARY KEY (id\_jerarquia)

)

DISTSTYLE

DISTKEY (id\_grupo)

SORTKEY (id\_region, fecha\_creacion);

\* Optimización para patrones de consulta específicos de microcréditos (análisis de morosidad grupal, tasas de renovación de ciclos)

Para el análisis de la morosidad se creó la tabla saldos\_podemos\_progresar con la siguiente estructura;

CREATE TABLE public.saldos\_podemos\_progresar (

id\_credito VARCHAR(256) NOT NULL,

id\_cuenta INTEGER NOT NULL,

id\_grupo VARCHAR(256) NOT NULL,

fecha\_corte DATE NOT NULL,

fecha\_pago DATE,

saldo\_capital FLOAT4 NOT NULL,

saldo\_interes FLOAT4 NOT NULL,

dias\_mora INTEGER,

clasificacion\_mora VARCHAR(50),

estatus\_credito VARCHAR(50),

id\_ejecutivo VARCHAR(50),

fecha\_proximo\_pago DATE,

monto\_proximo\_pago FLOAT4,

fecha\_update TIMESTAMP DEFAULT GETDATE(),

PRIMARY KEY (id\_credito)

)

DISTSTYLE KEY

DISTKEY (id\_grupo)

SORTKEY (fecha\_corte, id\_credito);

Con este query valide la morosidad;

SELECT

    clasificacion\_mora,

    COUNT(\*) as total\_creditos,

    SUM(saldo\_capital + saldo\_interes) as saldo\_total,

    AVG(dias\_mora) as promedio\_dias\_mora, id\_grupo

FROM

    dev.public.saldos\_podemos\_progresar

WHERE

    fecha\_corte = (SELECT MAX(fecha\_corte) FROM dev.public.saldos\_podemos\_progresar)

GROUP BY

    clasificacion\_mora,id\_grupo

ORDER BY

    CASE clasificacion\_mora

        WHEN 'AL DÍA' THEN 1

        WHEN 'VENCIDO' THEN 2

        WHEN 'MORA TEMPRANA' THEN 3

        WHEN 'MORA AVANZADA' THEN 4

        WHEN 'MORA GRAVE' THEN 5

        WHEN 'INCUMPLIMIENTO' THEN 6

        ELSE 7

    END;

Resultado de la consulta:



----RENOVACION CICLOS: Se podría validar que clientes han tenido más de un crédito los cuales su estatus debe ser pagado, descartando los clientes con un estatus VIGENTE,MOROSO para analizar que clientes son aptos para renovar su crédito

Query:

SELECT

    cr.id\_cliente,

    cl.nombre || ' ' || cl.apellidos AS nombre\_completo,

    j.id\_grupo,

    COUNT(cr.id\_credito) AS creditos\_previos,

    SUM(CASE WHEN cr.estatus = 'PAGADO' THEN 1 ELSE 0 END) AS creditos\_pagados,

    MAX(cr.fecha\_vencimiento) AS ultimo\_vencimiento,

    MAX(cr.monto\_aprobado) AS mayor\_monto\_aprobado

FROM

    public.creditos\_podemos\_progresar cr

JOIN

    public.clientes\_podemos cl ON cr.id\_cliente = cl.id\_cliente

JOIN

    public.jerarquia\_cliente j ON cr.id\_cliente = j.id\_cliente

WHERE

    j.tipo\_grupo = 'GRUPAL'

    AND cr.estatus = 'PAGADO'

    AND NOT EXISTS (

        SELECT 1

        FROM public.creditos\_podemos\_progresar vig

        WHERE vig.id\_cliente = cr.id\_cliente

        AND vig.estatus IN ('VIGENTE', 'MOROSO')

    )

GROUP BY

    cr.id\_cliente, cl.nombre, cl.apellidos, j.id\_grupo

HAVING

    COUNT(cr.id\_credito) >= 1

    AND SUM(CASE WHEN cr.estatus = 'MOROSO' THEN 1 ELSE 0 END) = 0

ORDER BY

    creditos\_pagados DESC,

    ultimo\_vencimiento DESC;

\* Estrategia de distribución de datos para consultas frecuentes sobre grupos y regiones

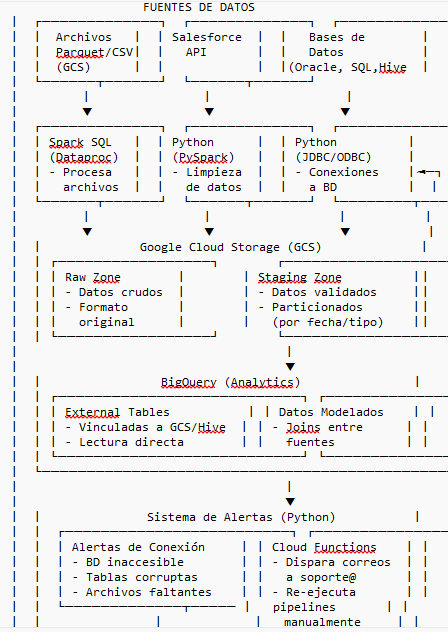
R. Una de las estrategias utilizadas en Bigquery fue partition by y cluster by en mi ejemplo

sobre redshift utilize DISTKEY id\_grupo(créditos grupales o individuales) para mejorar el rendimiento en las consultas y sortkey (id región) ordenado por región a la que pertenece el grupo ya que el enfoque es créditos grupales por región, esto para acelerar el rendimiento a la hora de consultar cualquier métrica con la premisa que el volumen de información día a día va a crecer.

- Comparte un diagrama de arquitectura con justificaciones técnicas específicas basadas en tu experiencia real (no teórica)

**R.** Esta arquitectura utilice en Walmart el cual redujo el SLA esperado del reporte de Ventas Ecom, implementando alertamiento el cual nos daba la capacidad de actuar rápido ante cualquier situación por ejemplo tabla innaccesible, tabla cambio su estructura eh impacto nuestro proceso,password vencido, etc)

Diagrama:



1.2. Optimización basada en patrones de acceso:

- A partir de tu experiencia con sistemas financieros similares:

\* Detalla los patrones de acceso que has observado en organizaciones de microcréditos

**R.** Por ejemplo cuando estuve en Aserta(Financiera) eran consultas transaccionales historial crediticio del cliente, prestamos activos con mora, cartera vencida, Montos prestados, riesgo crediticio

\* Proporciona ejemplos concretos de claves de distribución/ordenación que hayas implementado para datos financieros

**R.** En Aserta la BD era On-Premise Sysbase-Oracle se crearon índices, en Walmart utilice con Bigquery particionamiento y Clustering

\* Comparte al menos dos casos reales donde tuviste que hacer trade-offs entre optimización para diferentes patrones de acceso en datos financieros o de préstamos

**R.** En Walmart se particiono por fecha y por id\_tienda, ya que el volumen de información diario de ventas era de alrededor de 50 millones de transacciones esto redujo a un 45% el tiempo de consulta de reportes.

Aserta consultas on-premise(Oracle) se particiono por fecha y se crearon índices en el caso de consultas OLTP cliente, estado teniendo en cuenta las consultas mas frecuentes, y del lado de reportes se crearon índices para métricas de morosidad, historial crediticio

1.3. Administración avanzada:

- Comparte estrategias específicas que hayas implementado para:

\* Gestionar datos históricos de ciclos de préstamos completados

**R.** En Aserta la tabla de préstamos históricos crecía en un 20% aproximado anual, esto hacia las consultas mas lentas y por ende aumentaba el costo de almacenamiento. Los datos recientes se almacenaban con particiones mensuales mientras los datos de prestamos mayores a 3 años se almacenaban en otra tabla externas con esto se logró consultas históricas más eficientes.

\* Optimizar tablas de transacciones de pagos semanales con alto volumen

En Walmart Cashi la tabla de pagos señales procesaba millones de registros con cuellos de botella en Eventos como el Fin Irresistible o fin de semana la estrategia implementada fue Clustering (sucursal, método pago) y partition by por fecha del lado de Python se utilizaron agregaciones lo cual represento un rendimiento de los procesos batch de 5h a 25 min y las consultas de 3 min a 15 segundos.

\* Un caso donde implementaste una solución innovadora para requisitos de reportería regulatoria financiera

**R.** En Aserta se implementó un reporte en MicroStrategy para cumplir los requisitos de la Comision Nacional de Seguros y Fianzas. El objetivo era generar archivos en formato regulatorio (csv) y validar los saldos y estructura del estado de la situación financiera