

M4_ L4 Ejercicios de aplicación #4 Cassandra

Portafolio: [KarenTapiaLanda/BootCamp-Ing-de-Datos: BootCamp Ingeniería de Datos](#)

1. Diseño de la estructura en Apache Cassandra

Tabla: llamadas_por_cliente

```
CREATE TABLE llamadas_por_cliente ( numero_origen TEXT, fecha DATE, numero_destino TEXT, duracion INT, -- en segundos PRIMARY KEY ((numero_origen), fecha) );
```

Clave de partición: numero_origen → permite distribuir los datos por cliente

Clave de ordenamiento: fecha → permite consultar por rango de fechas

2. Inserción de datos de ejemplo

```
INSERT INTO llamadas_por_cliente (numero_origen, fecha, numero_destino, duracion) VALUES ('+56912345678', '2025-08-01', '+56987654321', 120); INSERT INTO llamadas_por_cliente (numero_origen, fecha, numero_destino, duracion) VALUES ('+56912345678', '2025-08-02', '+56911223344', 300); INSERT INTO llamadas_por_cliente (numero_origen, fecha, numero_destino, duracion) VALUES ('+56998765432', '2025-08-01', '+56912345678', 180);
```

3. Consultas requeridas

a) Todas las llamadas realizadas por un número específico

```
SELECT * FROM llamadas_por_cliente WHERE numero_origen = '+56912345678';
```

b) Llamadas en un rango de fechas

```
SELECT * FROM llamadas_por_cliente WHERE numero_origen = '+56912345678' AND fecha >= '2025-08-01' AND fecha <= '2025-08-31';
```

c) Total de minutos consumidos por un cliente en un mes

```
SELECT SUM(duracion) FROM llamadas_por_cliente WHERE numero_origen = '+56912345678' AND fecha >= '2025-08-01' AND fecha <= '2025-08-31';
```

Cassandra no soporta agregaciones complejas por defecto. Para métricas como esta, se recomienda usar Spark + Cassandra o precalcular totales en una tabla adicional.

4. Comparación: Cassandra vs. Bases Relacionales

Aspecto	Cassandra	Relacional (MySQL, PostgreSQL)
Escalabilidad	Horizontal, sin punto único de fallo	Vertical, más limitada
Rendimiento	Optimizado para escritura masiva	Optimizado para consistencia y joins
Modelado de datos	Basado en consultas (denormalizado)	Basado en relaciones (normalizado)
Consistencia	Eventual (BASE)	Fuerte (ACID)
Costos operativos	Menor en entornos distribuidos	Mayor en escalamiento vertical
Integración	Compatible con Spark, Kafka, etc.	Compatible con BI y SQL tradicional

Recomendación final

Cassandra es la mejor opción para este caso por:

- Su capacidad de manejar millones de registros de llamadas distribuidas
- Su rendimiento en escritura y lectura por clave de partición
- Su tolerancia a fallos y escalabilidad horizontal