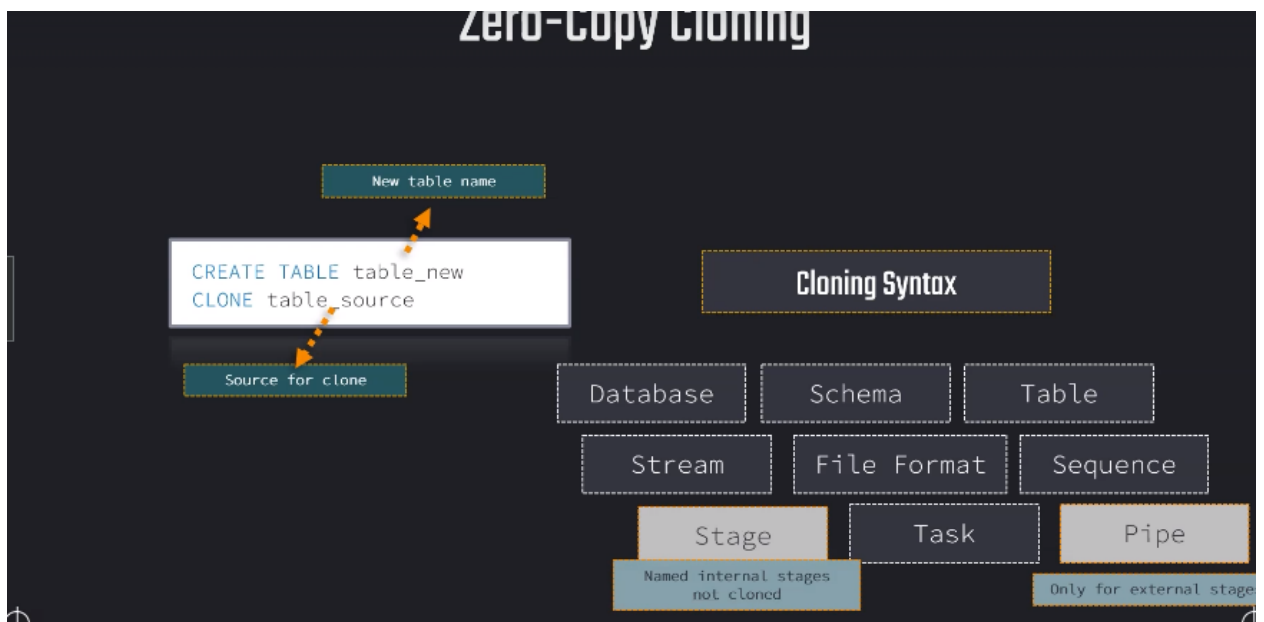


## Zero-Copy Cloning

Zero copy cloning in Snowflake is a feature that allows you to create a copy of a database, schema, or table without copying the underlying data. Instead, Snowflake creates a reference to the original data, which means that the cloned object and the original object share the same physical storage.

Zero copy cloning is a fast and efficient way to create copies of data in Snowflake. It is also a cost-effective way to create copies of data, because you only pay for the storage of the original data.

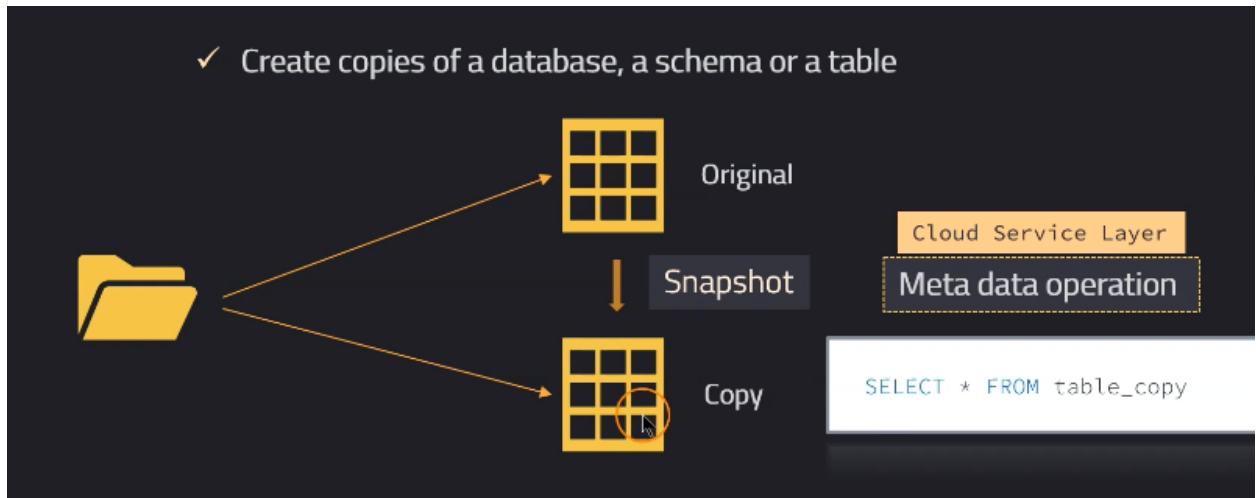
- 1.
2. We can also clone this



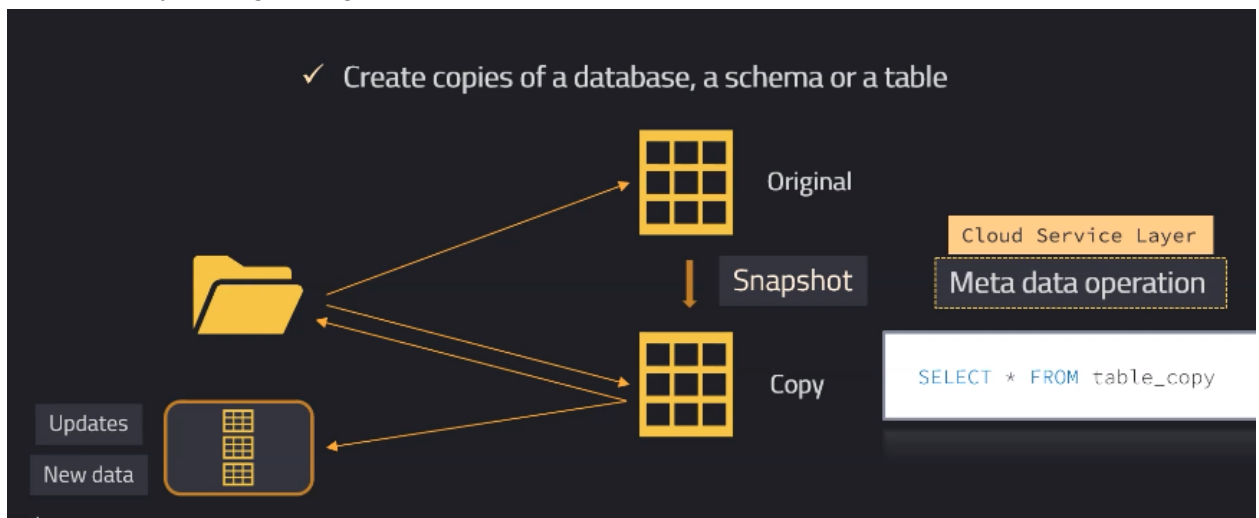
3. But in Pipe cloning is only for external stage and in stages...no cloning for named internal stages

Cloning a database or schema will  
clone all contained objects

- 4.



- 5.
6. After we create a clone a table....we can query from that
7. **Task** check whether...if we change the data in original data..will this change in the cloned copy ? ..check while doing hands on(did hands on and found that..new data will not reflected in cloned table)
8. Now if add any new data to the cloned copy...then snowflake only store this additional data..thereby saving storage costs



9. For all the other data which are not updated...we will just refer the original data

# Zero-Copy Cloning

- ✓ Create copies of a database, a schema or a table



- ✓ Cloned object is independent from original table
- ✓ Easy to copy all meta data & improved storage management
- ✓ Creating backups for development purposes
- ✓ Typically combined with time travel

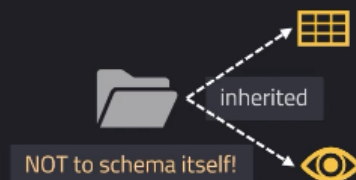
10.

## How about privileges?



NOT to database itself!

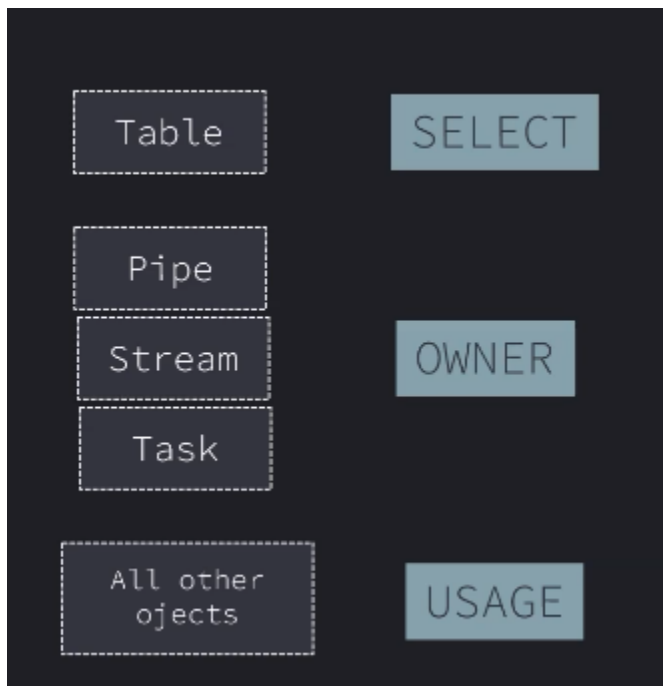
Privileges will always only be inherited to child objects never to source object itself



NOT to schema itself!

11.

12. For using zero copy on table..we need **select** privilege..for cloning pipe,stream,task we need **owner** privilege and for all the other objects we need **usage**.



13. In the cloning ..load history meta data is not copied...which means..we cannot add same data into original table ..bcuz it has some meta data...but in cloned table...we can add the same data..as it does not contain metadata

✓ Load history meta data is not copied  
*Loaded data can be loaded again*

```
CREATE TABLE table_new
CLONE table_source
BEFORE (TIMESTAMP -> 'timestamp')
```

Cloning from specific point in time is possible.

- 14.
15. For hands on refer online/file

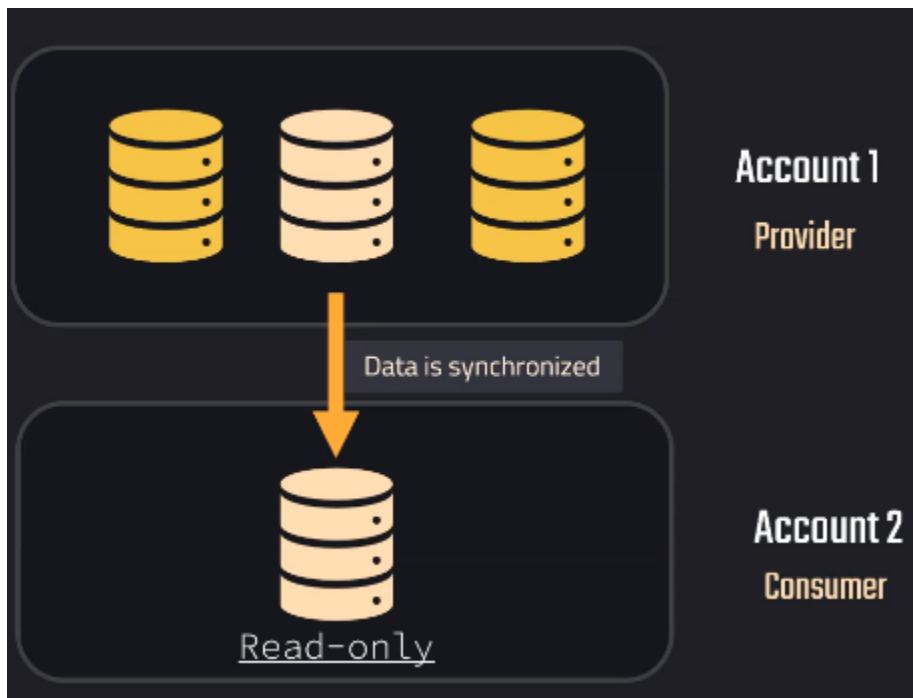
## Data Sharing

1. Usually data sharing is very complicated process...but in snowflake as storage is decoupled from compute...we can share our data easily

Usually this can be also a rather complicated process....

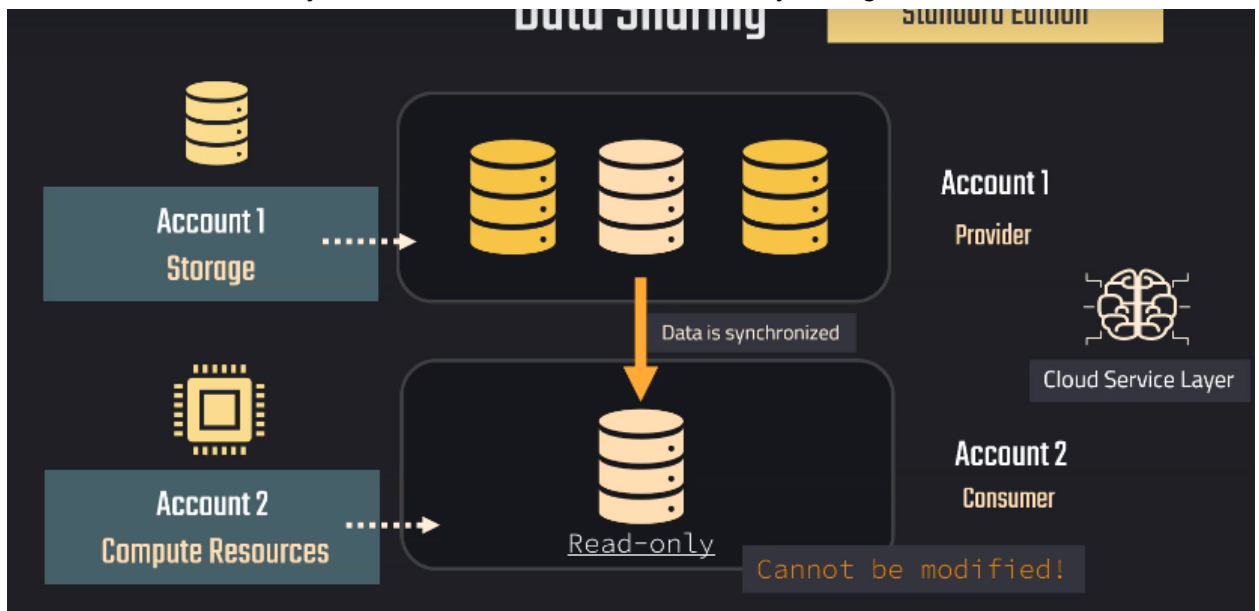
- ✓ Sharing with actually copying data
- ✓ Data is always up-to-date
- ✓ Compute paid by consumer

2. We just grant the user to the storage layer to the data to be shared
3. The user can use his own compute resources to query and will pay for the compute resources
4. So like this, the data in this share will always be up to date because it's actually the same data....We are just giving access to the user
5. This is available for all snowflake editions

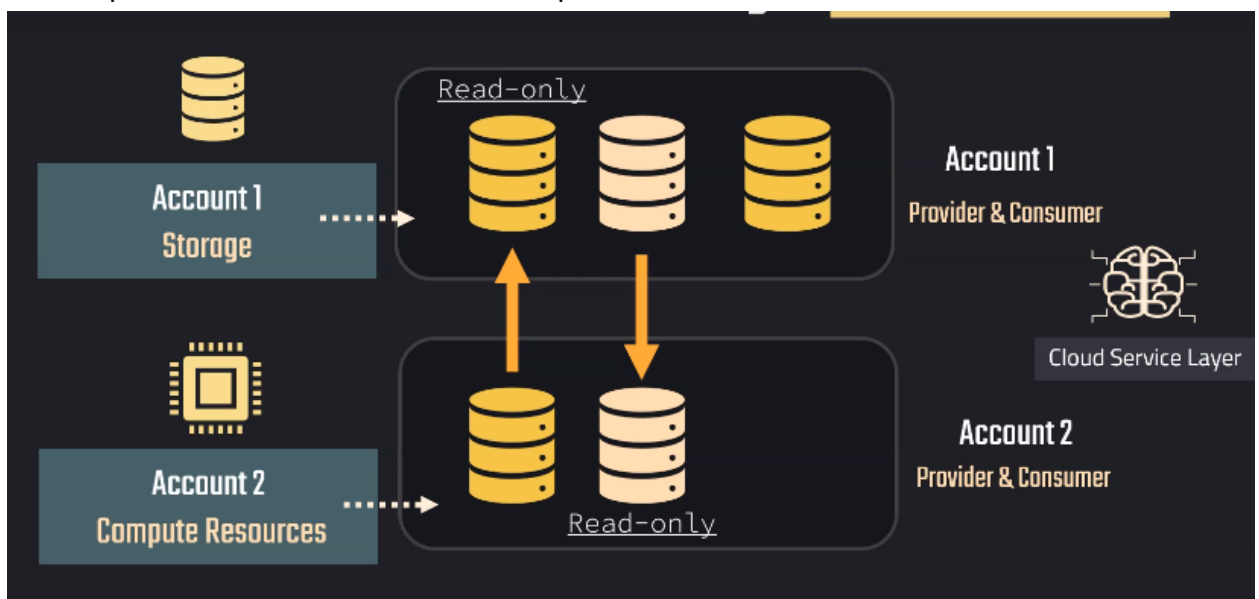


6.

7. Here consumer can only read the data..but cannot make any changes



8. The consumer ..uses his own compute resources ..to perform read only queries  
9. It's also possible for an account to be both provider and consumer



## Setting up share

- Create share** ACCOUNTADMIN role or CREATE SHARE privileges required  

```
CREATE SHARE my_share;
```
- Grant privileges to share**  

```
GRANT USAGE ON DATABASE my_db TO SHARE my_share;  
GRANT USAGE ON SCHEMA my_schema.my_db TO SHARE my_share;  
GRANT SELECT ON TABLE my_table.myschema.my_db TO SHARE my_share;
```
- Add consumer account(s)**  

```
ALTER SHARE my_share ADD ACCOUNT bl67131;
```

10.

- Import share** ACCOUNTADMIN role or IMPORT SHARE / CREATE DATABASE privileges required  

```
CREATE DATABASE my_db FROM SHARE my_share;
```

11.

## What can be shared

Tables

External Tables

Secure views

Secure materialized views

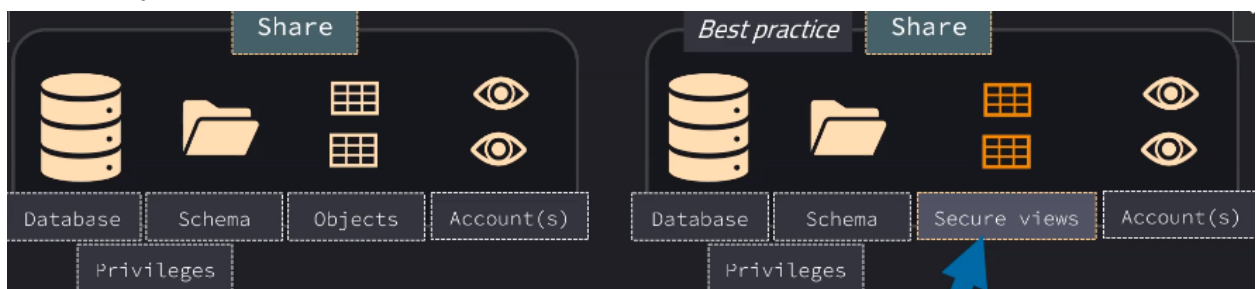
Secure UDFs

12.

13. So as mentioned, the share itself is just the container.

14. And in this share we have one database, then we can grant usage on the schema and then also on all of the objects that we want to grant privileges to.

15. And then the last component is the account .So we can have zero one or multiple accounts. And of course, the privileges to these objects are also part of a share as a best practice.



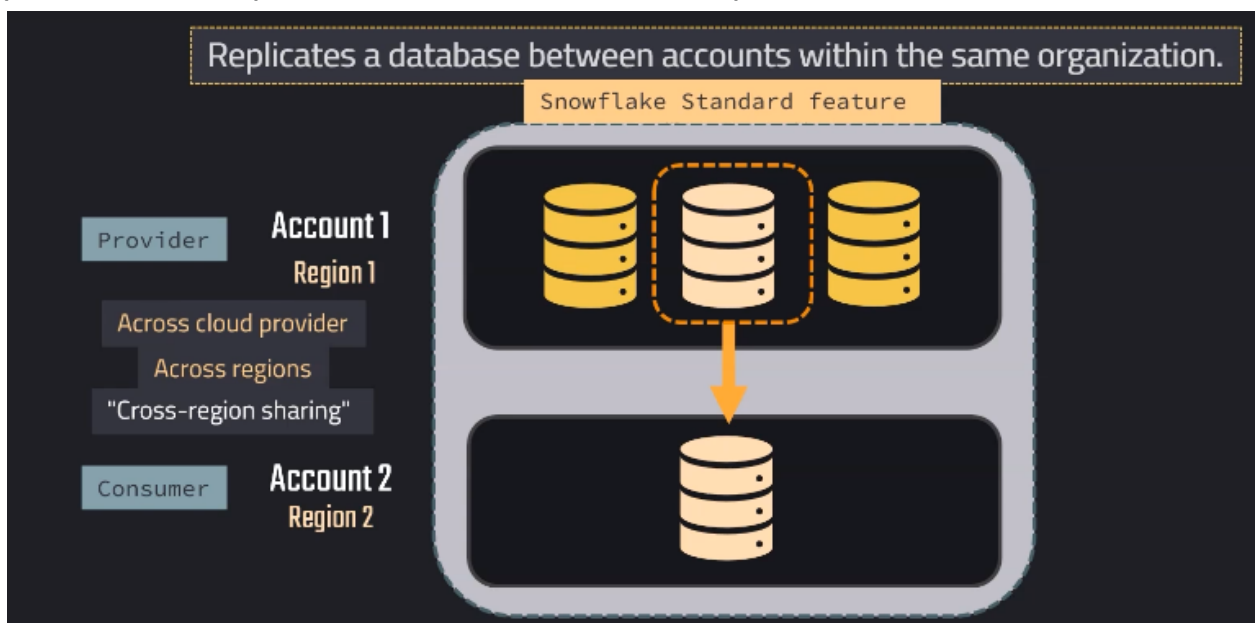
16.

17. The best practice to share the data ..is by secured views...and if we have data coming from multiple databases..then we create one view in a separate database and this will then be the database that we share.

18. If we want to share data to NON snowflake users...then we have to use reader account
19. Practice hands on

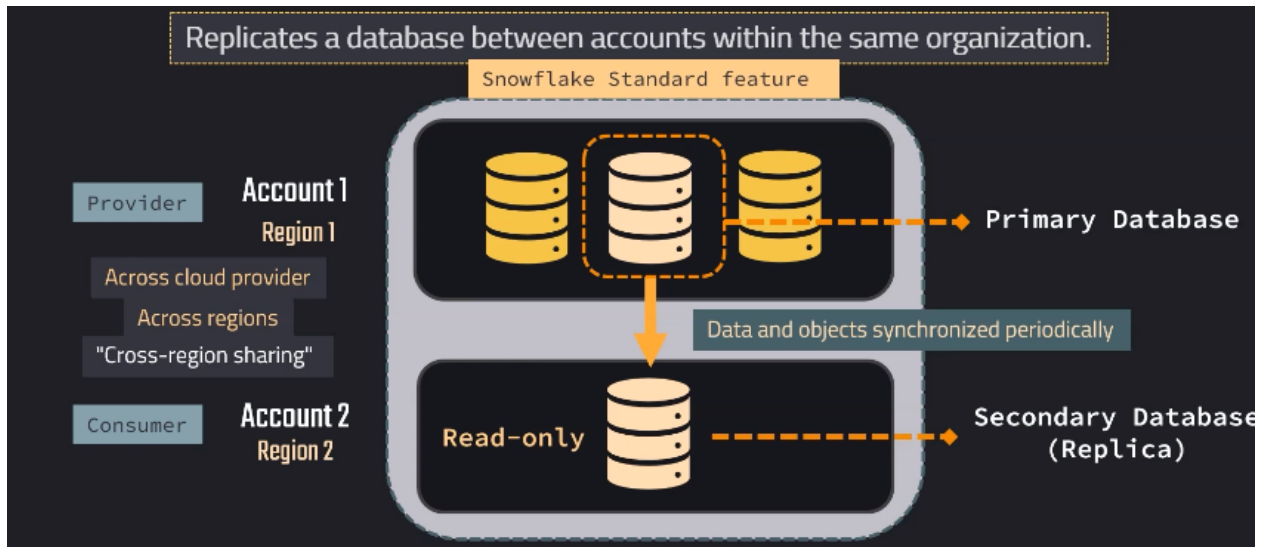
## DataBase Replication

1. So far we have learned about data sharing, which was only possible for sharing data within the same region and within the same cloud provider.
2. And now it is also possible to share data across regions and also across cloud providers. This is possible via so-called database replication.



3. But the big difference is that now the data is actually physically extracted and copied into this other account. So there will be data transfer costs
4. So we have to periodically synchronize the data..to view updated data





5.

1. Enable replication for source account with ORGADMIN role

```
show organization accounts;
```

```
-- Enable replication for each source and target account in your organization
select system$global_account_set_parameter('<organization_name>.<account_name>',
      'ENABLE_ACCOUNT_DATABASE_REPLICATION', 'true');
```

Step 2: Promote a Local Database to Primary Database with ACCOUNTADMIN role

```
ALTER DATABASE my_db1 ENABLE REPLICATION TO ACCOUNTS myorg.account2, myorg.account3;
```

Step 3: Create replica in consumer account

```
CREATE DATABASE my_db1 AS REPLICA OF myorg.account1.my_db1;
```

6.

Step 4: Refresh database

```
ALTER DATABASE my_db1 REFRESH;
```

Ownership privileges are needed

A task can be scheduled with this command

7.

8. And it's also lastly important to know that ownership privileges are needed to refresh this database. we can also set up a task with this command.

