Managing ACLs for the Enterprise Lakehouse :-

- When data is mounted to DBFS without passwords being configured, all users having access to the workspace, gains access to the data

Without properly configured ACLs, users can access the entire data.

Unity Catalog helps easy configurations across all the workspaces in an enterprise account.

Recommendations for configuring access to data in multi-workspace environments :-

- Create multiple workspaces, to split and isolate workloads.

- It is not safe to allow users on table ACLs and non-table ACL clusters within the same workspace.

- Interactive queries should never be run on production workspace. Most codes here should be executed as scheduled jobs.

- External tools like Power BI, tableau, will use APIs Tokens to access the Databricks SQL Workspace (Analytics Workspace)

- Table ACL grants are defined in Databricks SQL. based primarily on groups important for SCIM (System for Cross-domain Identity Management specification)
  Ex→ Sales, Finance, HR

- Allow Derivative Datasets to be built in a secured fashion.

- Derivative Datasets should be audited regularly, to reduce unnecessary storage costs and ensure that PII rules are enforced.

Grant Access to Production Datasets

Assumptions

- End-users need read-only access
- Datasets organized by database

```
GRANT USAGE, SELECT, READ_METADATA ON DATABASE hr TO `HR`;
```

Alternative, grant access on specific tables:

```
GRANT USAGE ON DATABASE hr TO `HR`;
GRANT SELECT, READ_METADATA ON TABLE employees TO `HR`;
GRANT SELECT, READ_METADATA ON TABLE addresses TO `HR`;
```

name of group that is to be granted access

Enable Secure Data Sharing

Assumptions

- Teams/depts need a private area to collaborate in
- Datasets must not be shared outside team/dept
- New tables are not automatically shared to other members

```
GRANT USAGE, CREATE ON DATABASE project_data TO `Data Analysts`;
```

Alternatively, members automatically see all new tables:

```
GRANT USAGE, SELECT, READ_METADATA, CREATE ON DATABASE project_data TO `Data Analysts`;
```

⇒ Service Principal is an API-only identity, and do not have access to databricks UI and CLI.

≥ Databricks Admin User can create and manage service principals using the SCIM Api

≥ Ownership can be switched from individual to another individual, as follow :-

```
ALTER DATABASE <db_name> OWNER TO 'HR Admins';
```

≥ Creating DYNAMIC VIEWS that displays PII data, if member of a specific group :-

≥ Suppose, we have this users Tbl table.

(id) & (name) is PII.

id	name	gender
1	aa	m
2	bb	F
3	cc	m

② We have a users group named 'Architect', and we want that only members of this group can see the PII, and no one else.

We secure the table to no access, and then create the following view, and will give access to the view, to everyone/every employee :-

```
%sql
CREATE OR REPLACE users_vw AS
    SELECT
        CASE
            WHEN is_member('architect')
            THEN id
            ELSE "hidden-data"
        END AS <column_alias>,

        CASE
            WHEN is_member('architect')
            THEN name
            ELSE "hidden_data"
        END AS <column, alias>,

        gender
    FROM usersTbl
```

name of the group whose members can see the data in plain text.

if condition do not match, values will be replaced by the text "hidden-data"

column name. This means original data of the column will be displayed if condition match

is_member() Takes group name as parameter, an returns True if the user executing the view is a member of the group.
Else, returns False.

≥ is_member() use in WHERE clause :-

```
%sql
CREATE OR REPLACE VIEW users_la_vw AS
SELECT * FROM users_vw
WHERE
  CASE
    WHEN is_member('ade_demo') THEN TRUE
    ELSE city = "Los Angeles" AND updated > "2019-12-12"
  END
```

this means, no condition is set when user belongs to ade_demo., and thus, can see entire data.

when the user is not a member of ade_demo group, this condition will be set and user can only see records that meet this criteria.