VERTICA
by opentext™

# Database Users and Roles
# Tables with access Priveleges

# Database users and privileges

dbadmin

**Privileges for superuser:**

- **SELECT** on *<all tables>*
- **INSERT** on *<all tables>*
- **DELETE** on *<all tables>*

# Database users and privileges

user1

**Individual Privileges:**

**SELECT** on table1
**SELECT** on table2
**DELETE** on table2
**INSERT** on table3
**DELETE** on table3

**Roles:**

**SELECT** on table1
**DELETE** on table2

**SELECT** on table4
**INSERT** on table4
**DELETE** on table4

**INSERT** on table3
**DELETE** on table2

# Authentication vs. authorization

**Authentication:**
**Who Are You?**

- Name?
- Password?
- Access point?

**Authorization:**
**What Can You Access?**

- Schemas?
- Tables?
- Read only?
- Change data?

# Authentication: Creating a user

```
CREATE USER username IDENTIFIED BY password PROFILE password_profile PASSWORD EXPIRE;
```

Daniel Solo

```
CREATE USER dsolo
    IDENTIFIED BY Welcome123
    PROFILE contractor
    PASSWORD EXPIRE;
```

# Authentication: Password profiles

## PASSWORD PROFILE SETTINGS

Min/max length?

#Uppercase? Lowercase?
#Symbols? Digits?

Days before expiration?
Grace period?

Reuse policy?

… and more …

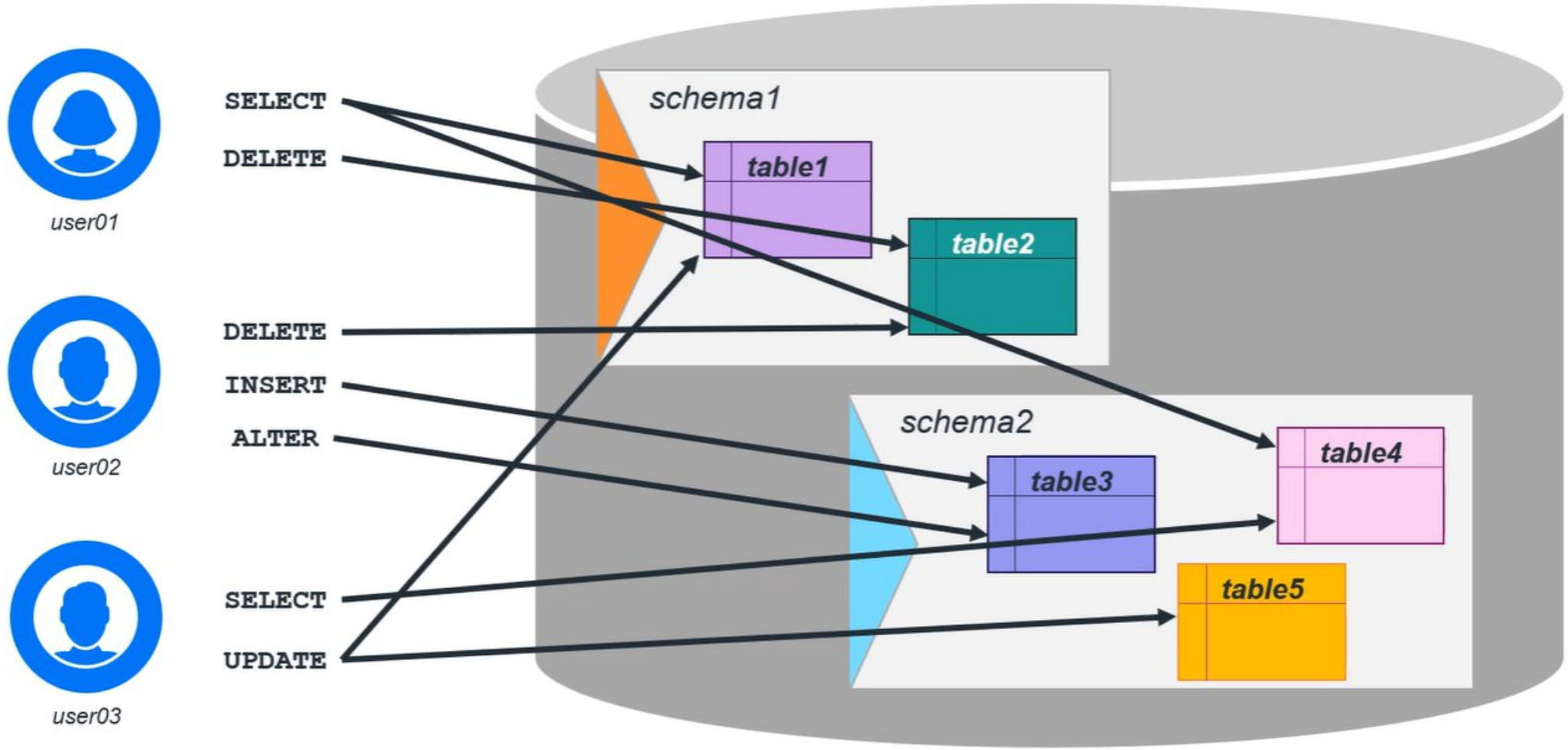

**Default Profile:**
- password lifetime: unlimited
- failed login attempts: unlimited
- minimum length: unlimited
- minimum upper case: unlimited
- minimum symbols: unlimited
- minimum digits: unlimited
- …


**Profile 1:**
- password lifetime: 30 days
- failed login attempts: 2
- minimum length: 8
- minimum upper case: 1
- minimum symbols: 1
- minimum digits: 3
- …

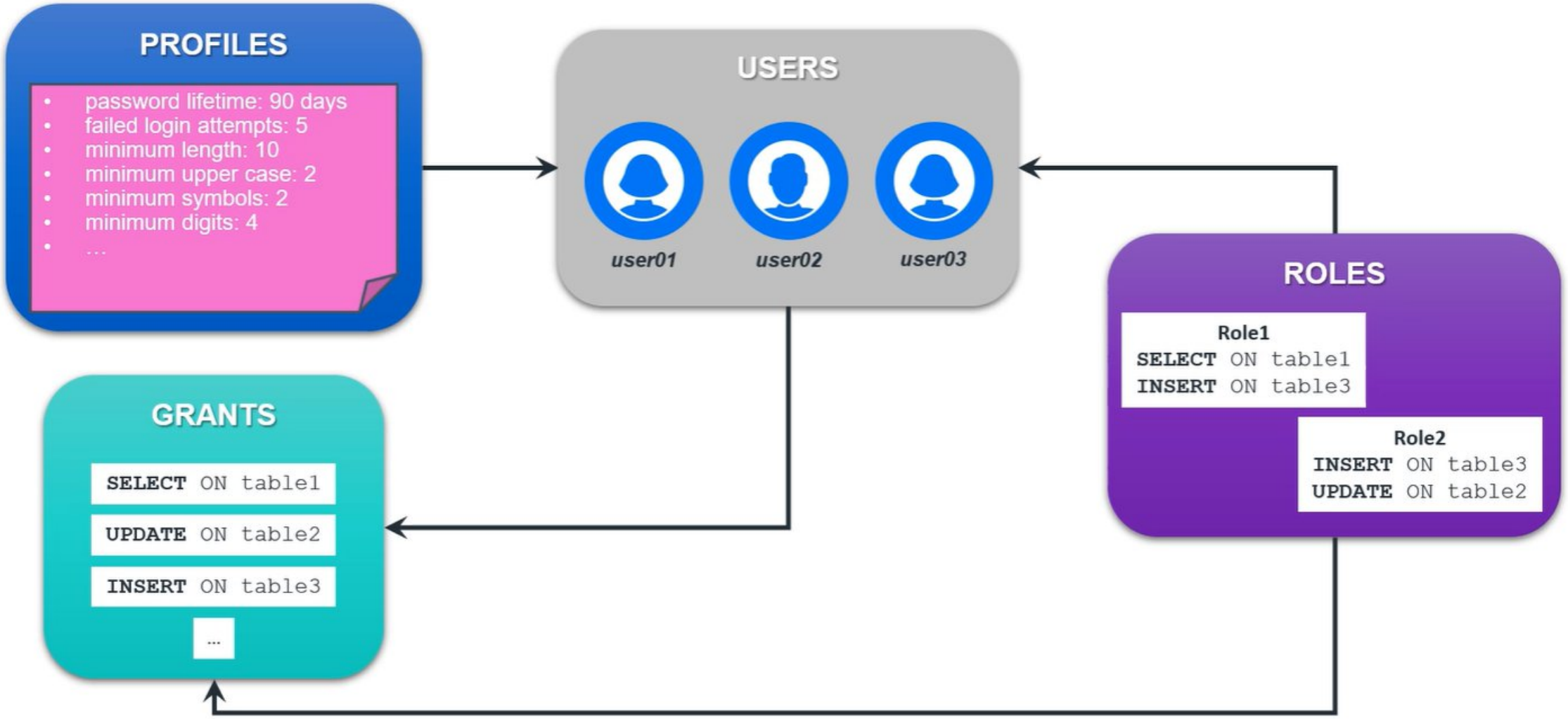

**Profile 2:**
- password lifetime: 90 days
- failed login attempts: 5
- minimum length: 10
- minimum upper case: 2
- minimum symbols: 2
- minimum digits: 4
- …

# Authorization: User privileges

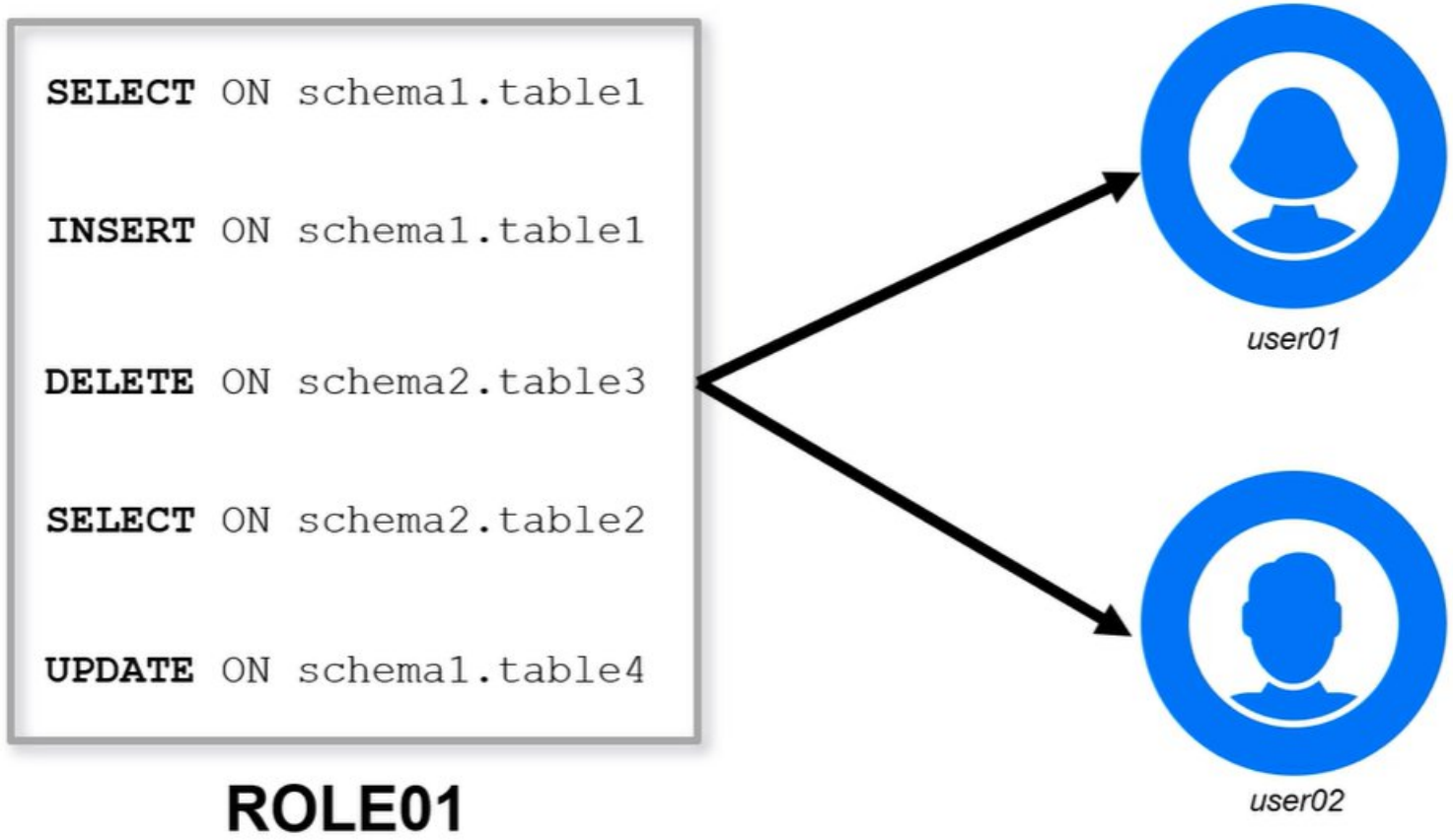
```
GRANT privilege ON table TO grantee;
```



# User management system tables



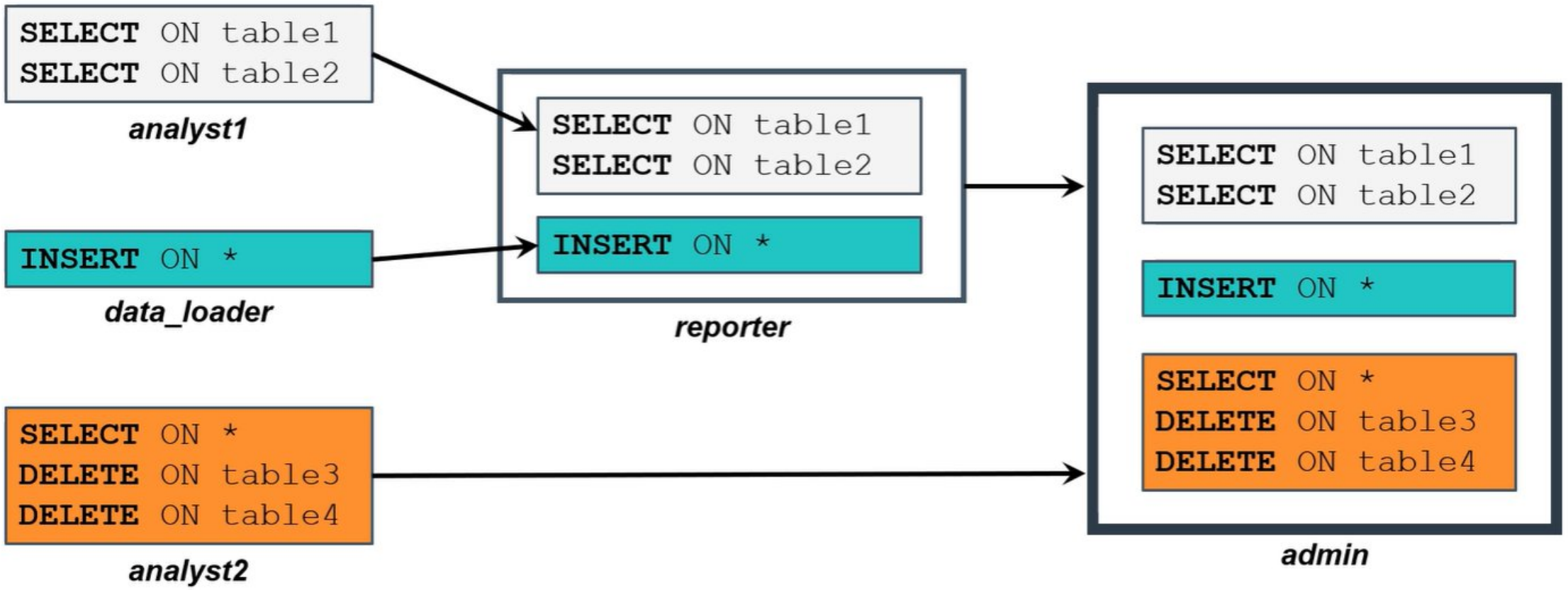
# Authentication: About roles



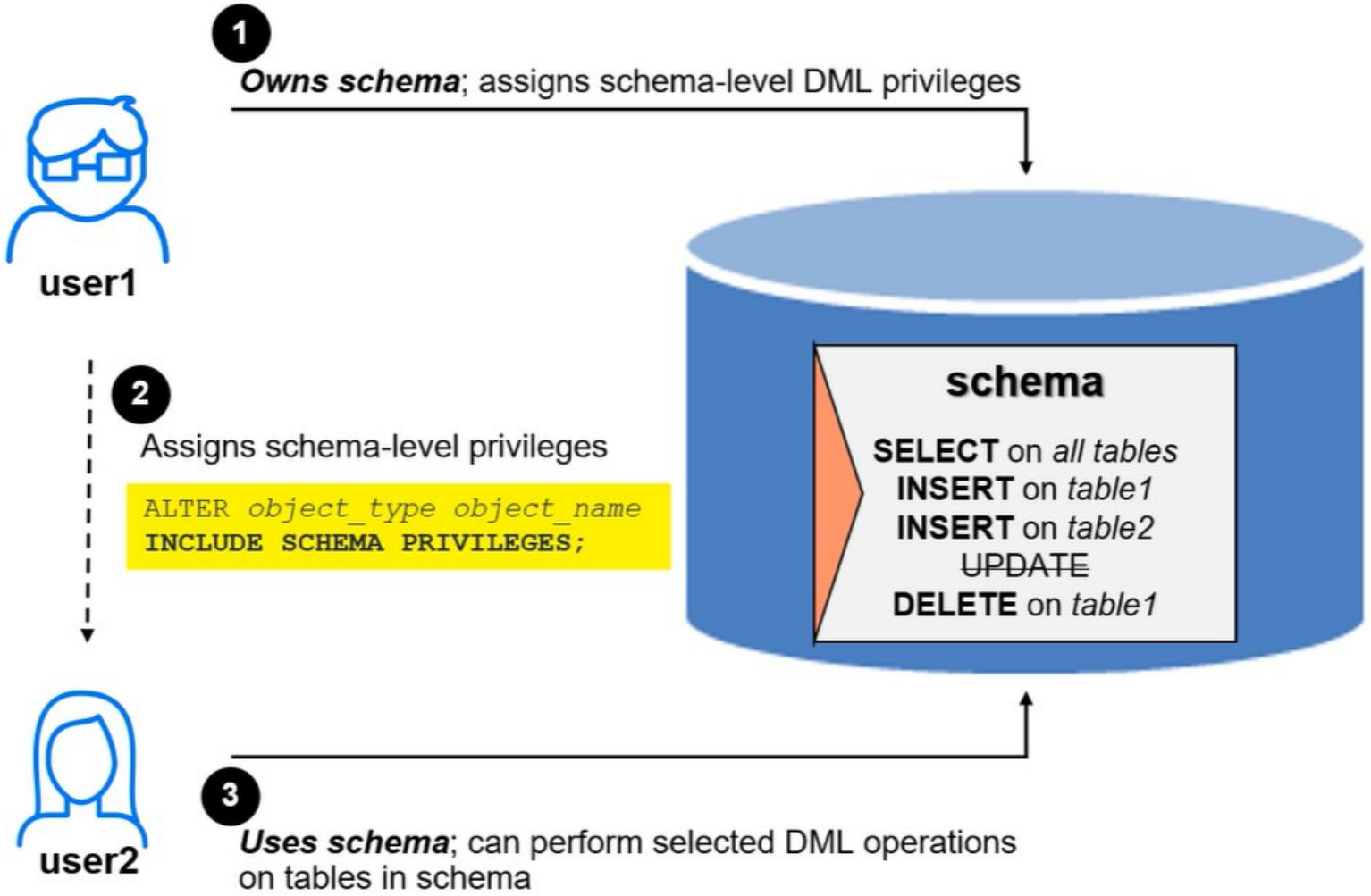
# Default database roles

| Role | Description |
|---|---|
| **PUBLIC** | Default role, no privileges assigned |
| **SYSMONITOR** | Allows for delegation of limited administrative tasks without compromising security or exposing sensitive information |
| **DBDUSER** | • Run DBD from the command line |
| **DBADMIN** | • Manage users, schemas, and roles<br>• View system tables<br>• View and terminate user sessions<br>• Access to all user-created data |
| **PSEUDOSUPERUSER** | • Create schemas<br>• Manage user passwords<br>• Manage user privileges<br>• Manage external procedures, UDF libraries, and UDF function |
| **UDXDEVELOPER** | Allows for creation and management of user-defined library functions |
| **MLSUPERVISOR** | Allows management of all machine learning models |

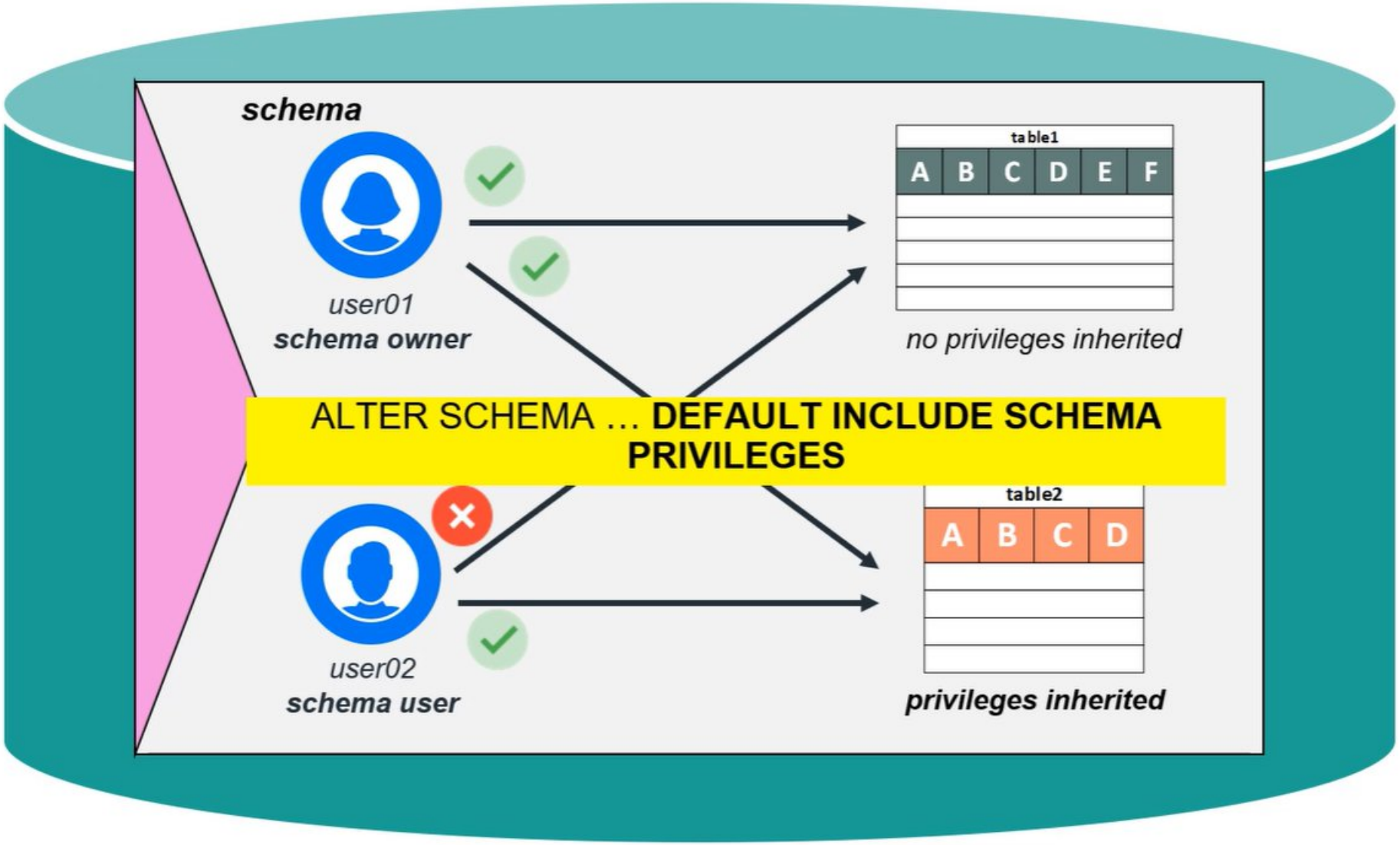
# Role hierarchies



# Inherited access privileges

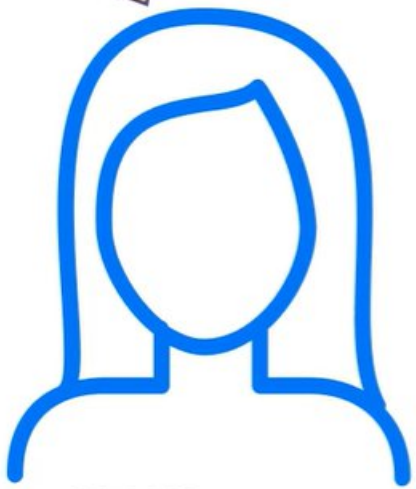


# Inherited privileges and database objects



# Fine-grained data access control

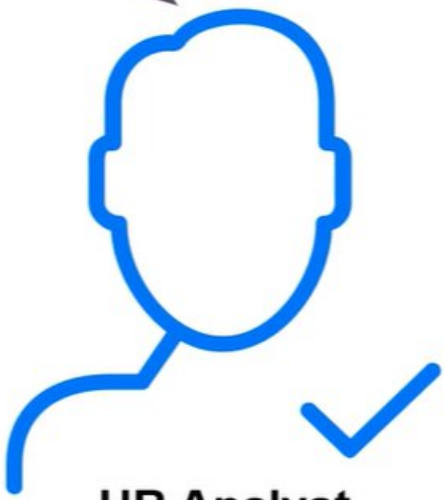
```
SELECT * FROM PEOPLE;
```



**HR Manager**

| Name | SSN |
|---|---|
| Dave | 123-45-6789 |
| Holly | 456-78-9123 |
| Maria | 314-15-9265 |
| Kevin | 667-40-8311 |

```
SELECT * FROM PEOPLE;
```



**HR Analyst**

| Name | SSN |
|---|---|
| Holly | 9123 |
| Dave | 6789 |

```
SELECT * FROM PEOPLE;
```

**Developer**

| Name | SSN |
|---|---|
| Maria | |

# Column access privileges

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a1 | b1 | c1 | d1 |
| a2 | b2 | c2 | d2 |
| a3 | b3 | c3 | d3 |

```
CREATE ACCESS POLICY ON table FOR COLUMN column [ENABLE | DISABLE];
```

*role1*

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a1 | b1 | | d1 |
| a2 | b2 | | d2 |
| a3 | b3 | | d3 |

*or*

*role2*

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a1 | b1 | X | d1 |
| a2 | b2 | X | d2 |
| a3 | b3 | X | d3 |

# Column access privileges

```
CREATE ACCESS POLICY ON table FOR COLUMN column

    CASE
        WHEN ENABLED_ROLE('role1') [AND condition] THEN column_value1
        WHEN ENABLED_ROLE('role2') [AND condition] THEN column_value2
        WHEN ENABLED_ROLE('role3') [AND condition] THEN column_value3
        …
        {ELSE column_valueN}
    END

ENABLE;
```

```
CREATE ACCESS POLICY ON table FOR COLUMN column

    WHERE
        (ENABLED_ROLE('role1') [AND condition]) OR
        (ENABLED_ROLE('role2') [AND condition])
    column_value1

    WHERE
        (ENABLED_ROLE('role3') [AND condition])
        …
    column_value2

ENABLE;
```

# Row access privileges

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a1 | b1 | c1 | d1 |
| a2 | b2 | c2 | d2 |
| a3 | b3 | c3 | d3 |
| a4 | b4 | c4 | d4 |

```
CREATE ACCESS POLICY ON table FOR ROWS [ENABLE | DISABLE];
```

*role1*

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a1 | b1 | c1 | d1 |
| a4 | b4 | c4 | d4 |

*role2*

| table | | | |
|---|---|---|---|
| A | B | C | D |
| a2 | b2 | c2 | d2 |
| a3 | b3 | c3 | d3 |
| a4 | b4 | c4 | d4 |

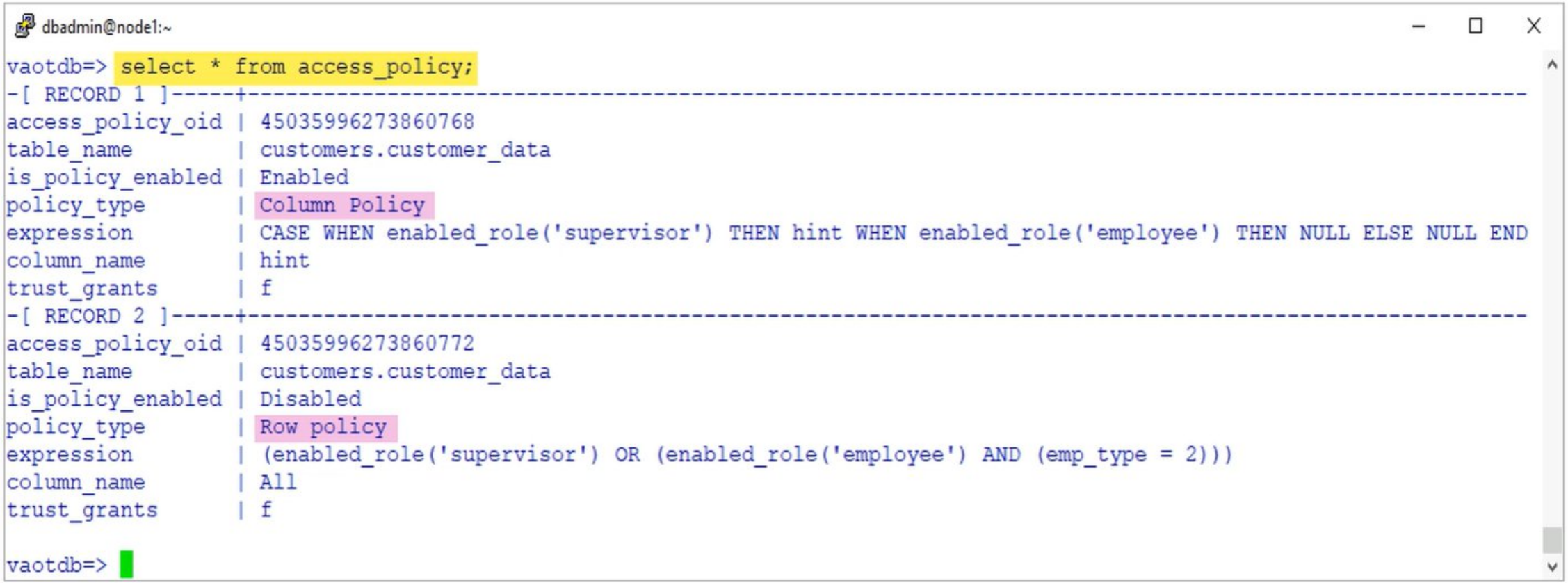
# Row access privileges

```
CREATE ACCESS POLICY ON table FOR ROWS

   WHERE
      (ENABLED_ROLE('role1') [AND condition]) OR
      (ENABLED_ROLE('role2') [AND condition]) OR
      (ENABLED_ROLE('role3') [AND condition]) OR
       ...

ENABLE;
```

# System table: ACCESS_POLICY



# Altering access policies

## Alter expression / change policy status:

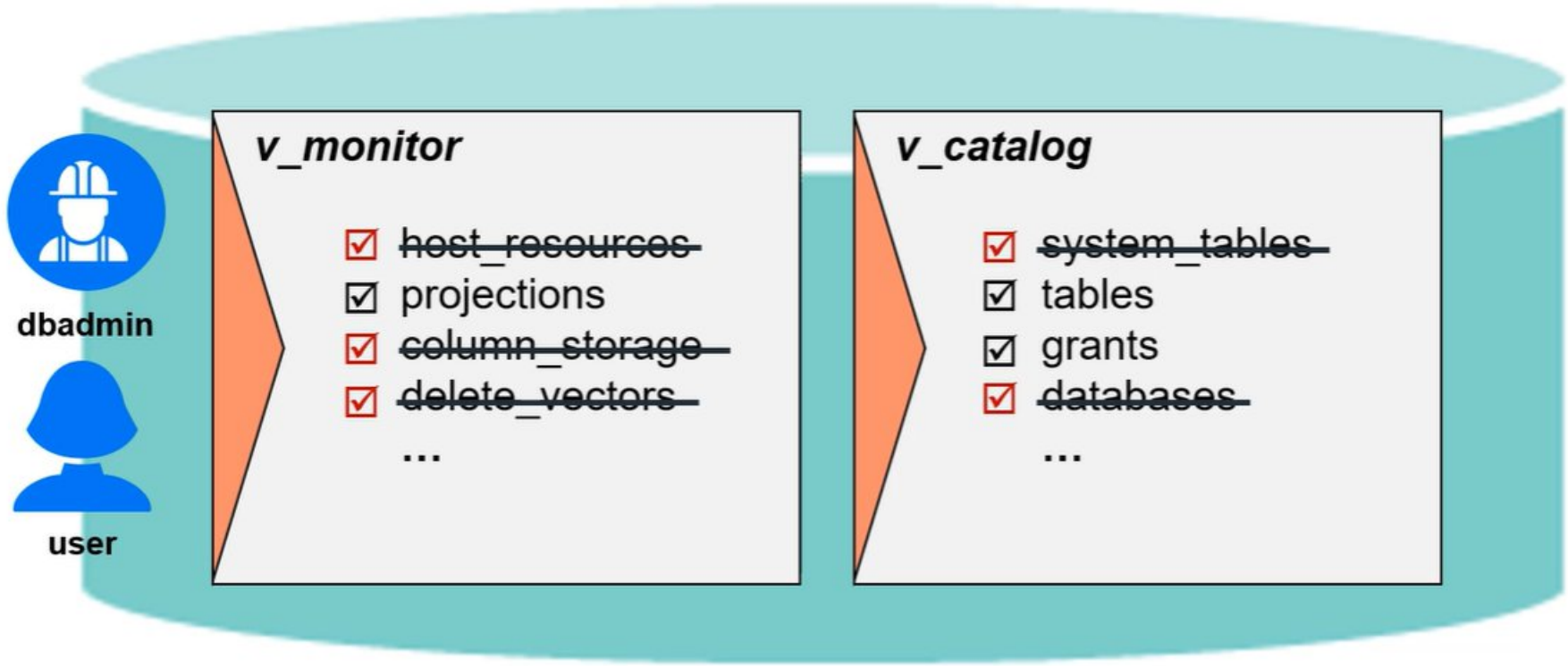
```
ALTER ACCESS POLICY ON table FOR [COLUMN column_name | ROWS]
    [new expression]
[ENABLE | DISABLE];
```

## Permanently remove policy:

```
DROP ACCESS POLICY ON table FOR [COLUMN column_name | ROWS];
```

# Restricting system table access

```
SELECT RESTRICT_SYSTEM_TABLES_ACCESS();
```



opentext™

# Thank you