What is **REVOKE** in PostgreSQL?

In PostgreSQL, the **REVOKE** command is used to **remove permissions (privileges)** that were previously given using `GRANT`.

In simple words: **REVOKE = Take back access**

Why do we use REVOKE?

We use `REVOKE` to:

- Remove user access for security
- Restrict students/teachers when needed
- Stop unauthorized actions (INSERT, UPDATE, DELETE)
- Update role permissions in real systems
- Enforce rules in university or organization databases

Basic Syntax

```
REVOKE privilege(s)
ON object_name
FROM role_name;
```

Common Privileges to Revoke

Privilege	Meaning
SELECT	Read data
INSERT	Add data
UPDATE	Modify data
DELETE	Remove data
ALL	Remove all privileges

Simple Example

Step 1: Assume permission was given

```
GRANT SELECT ON students TO student;
```

Step 2: Now remove it

```
REVOKE SELECT ON students FROM student;
```

👉 Student can no longer read the table.

🏫 Real-World University Example

🎓 Scenario:

- Students were allowed to view marks
- Now exam period is over → access must be restricted

❌ Remove student access to marks

```
REVOKE SELECT ON marks FROM student;
```

👩‍🏫 Restrict teachers from editing marks after submission

```
REVOKE INSERT, UPDATE ON marks FROM teacher;
```

👨‍💼 Remove all permissions from a role

```
REVOKE ALL PRIVILEGES ON students FROM it_staff;
```

🎓 REVOKE LAB: Remove Permissions Step-by-Step (University System)

🎯 Objective

You will learn how to:

- Remove access from users (roles)
- Restrict students, teachers, and staff
- Understand how security is tightened using `REVOKE`

🏁 PRE-SETUP (Assume Already Created)

We already have:

📊 Database:

`university_db`

👥 Roles:

- `student`
- `teacher`
- `admin`

📋 Tables:

- `students`
- `courses`
- `marks`
- `enrollments`

🔐 STEP 1: Check Existing Permissions (Important)

Before revoking, check what a role has:

```
\dp students
```

👉 This shows current privileges on the table.

🚫 STEP 2: Revoke Student Permissions (Full Restriction)

❌ Remove SELECT access from students table

```
REVOKE SELECT ON students FROM student;
```

❌ Remove SELECT from all main tables

```
REVOKE SELECT ON courses FROM student;
REVOKE SELECT ON enrollments FROM student;
REVOKE SELECT ON marks FROM student;
```

👉 Now student cannot read any data.

🧪 TEST AFTER REVOKE (Student Login)

```
SELECT * FROM students;
```

❌ ERROR: permission denied

👩‍🏫 STEP 3: Restrict Teacher Permissions

❌ Remove ability to update marks

```
REVOKE UPDATE ON marks FROM teacher;
```

❌ Remove insert access

```
REVOKE INSERT ON marks FROM teacher;
```

🧪 Test Teacher Access

```
INSERT INTO marks(student_id, course_id, marks)
VALUES (1, 1, 88);
```

❌ Should FAIL

👨‍💼 STEP 4: Revoke ALL Permissions from Admin (Extreme Case)

⚠ Used in system reset or security emergency

```
REVOKE ALL PRIVILEGES ON students FROM admin;
REVOKE ALL PRIVILEGES ON courses FROM admin;
REVOKE ALL PRIVILEGES ON marks FROM admin;
REVOKE ALL PRIVILEGES ON enrollments FROM admin;
```

👉 Admin loses table-level access (NOT superuser privilege if assigned separately)

🔁 STEP 5: Revoke Database Access

❌ Prevent login to database

```
REVOKE CONNECT ON DATABASE university_db FROM student;
```

```
REVOKE CONNECT ON DATABASE university_db FROM teacher;
```

🧪 STEP 6: Verify Connection Block

When student tries to connect:

```
\c university_db
```

❌ ERROR: permission denied

🔓 STEP 7: Advanced REVOKE (Cascade Effect)

If you granted multiple permissions:

```
REVOKE ALL ON marks FROM teacher;
```

👉 Removes:

- SELECT
- INSERT

- UPDATE
- DELETE

🧠 STEP 8: Important Concepts

🔑 GRANT vs REVOKE

Command	Action
GRANT	Give permission
REVOKE	Remove permission

⚠ Key Rules

✔ You can only revoke what was granted ✔ Only owner or admin can revoke permissions ✔ REVOKE does NOT delete user ✔ REVOKE only removes access, not data

---

🏫 REAL-WORLD UNIVERSITY USE CASE

Scenario:

University decides:

- Students should only view marks during exam week
- Teachers should stop editing marks after submission deadline

Solution:

```
REVOKE INSERT, UPDATE ON marks FROM teacher;
```

📊 FINAL SUMMARY

After completing this lab, you learned:

✔ How to remove permissions step-by-step ✔ How to restrict students, teachers, admins ✔ How database security is dynamically controlled ✔ Real-world control of university system access

---