

## Demo

✓ Perfect! Let's prepare the same two scripts (remove access except SELECT, and grant full access back) for user `IAM:qwer`, just like we did for `IAM:aplet`.

---

### ✓ Script 1: Remove all access & only allow SELECT for `IAM:qwer` [🔗](#)

```
1 -- Revoke all privileges on schema
2 REVOKE ALL ON SCHEMA public FROM "IAM:qwer";
3
4 -- Revoke all privileges on all existing tables
5 REVOKE ALL PRIVILEGES ON ALL TABLES IN SCHEMA public FROM "IAM:qwer";
6
7 -- Revoke USAGE (cleanup), then grant minimal USAGE back
8 REVOKE USAGE ON SCHEMA public FROM "IAM:qwer";
9 GRANT USAGE ON SCHEMA public TO "IAM:qwer";
10
11 -- Grant SELECT on all existing tables
12 GRANT SELECT ON ALL TABLES IN SCHEMA public TO "IAM:qwer";
13
14 -- Set default privileges for future tables
15 ALTER DEFAULT PRIVILEGES IN SCHEMA public
16 GRANT SELECT ON TABLES TO "IAM:qwer";
17
18 -- Revoke CREATE on schema
19 REVOKE CREATE ON SCHEMA public FROM "IAM:qwer";
20
```

---

### ✓ Script 2: Grant full access back to `IAM:qwer` [🔗](#)

```
1 -- Grant USAGE on schema
2 GRANT USAGE ON SCHEMA public TO "IAM:qwer";
3
4 -- Grant CREATE on schema
5 GRANT CREATE ON SCHEMA public TO "IAM:qwer";
6
7 -- Grant ALL privileges on all existing tables
8 GRANT ALL PRIVILEGES ON ALL TABLES IN SCHEMA public TO "IAM:qwer";
9
10 -- Grant ALL privileges on future tables
11 ALTER DEFAULT PRIVILEGES IN SCHEMA public
12 GRANT ALL PRIVILEGES ON TABLES TO "IAM:qwer";
13
14 -- (Optional) If needed, grant on sequences
15 GRANT USAGE, SELECT, UPDATE ON ALL SEQUENCES IN SCHEMA public TO "IAM:qwer";
16 ALTER DEFAULT PRIVILEGES IN SCHEMA public
17 GRANT USAGE, SELECT, UPDATE ON SEQUENCES TO "IAM:qwer";
18
19 -- (Optional) If needed, grant on functions
20 GRANT EXECUTE ON ALL FUNCTIONS IN SCHEMA public TO "IAM:qwer";
```

```
21 ALTER DEFAULT PRIVILEGES IN SCHEMA public
22 GRANT EXECUTE ON FUNCTIONS TO "IAM:qwer";
23
```

### 💡 Will these affect other users? [🔗](#)

What it affects?	Answer
IAM:qwer only	✅ Yes
Other users	❌ No
PUBLIC role	❌ No

### 💬 Next [🔗](#)

If you'd like, I can also combine both IAM:aplet and IAM:qwer into a single consolidated file or prepare .sql downloads for you directly.

✅ Just say **"Yes, prepare file"** and I'll attach them ready to download!

✅ Perfect! Let's give you **everything in one place**:

### 💣 1 SQL to create the employees table [🔗](#)

```
1 CREATE TABLE public.employees (
2     employee_id integer,
3     first_name character varying(50),
4     last_name character varying(50),
5     department character varying(50),
6     salary integer
7 );
8
```

### 💡 Optional: Insert some sample data [🔗](#)

```
1 INSERT INTO public.employees (employee_id, first_name, last_name, department, salary)
2 VALUES
3 (1001, 'John', 'Doe', 'HR', 50000),
4 (1002, 'Jane', 'Smith', 'Finance', 75000),
5 (1003, 'Bob', 'Brown', 'IT', 90000);
6
```

## 🌟 2 Test script for restricted users (e.g., IAM:aplet or IAM:qwer) [🔗](#)

```
1 -- ✅ SELECT query (should work)
2 SELECT * FROM public.employees LIMIT 10;
3
4 -- ❌ INSERT query (should fail)
5 INSERT INTO public.employees (employee_id, first_name, last_name, department, salary)
6 VALUES (9999, 'Test', 'User', 'Engineering', 100000);
7
8 -- ❌ UPDATE query (should fail)
9 UPDATE public.employees
10 SET salary = 200000
11 WHERE employee_id = 9999;
12
13 -- ❌ DELETE query (should fail)
14 DELETE FROM public.employees
15 WHERE employee_id = 9999;
16
```

### ✅ Expected results [🔗](#)

Query Type	Result
SELECT	✅ Works
INSERT	❌ Permission denied
UPDATE	❌ Permission denied
DELETE	❌ Permission denied

### 🟢 Summary [🔗](#)

- ✓ Create the `employees` table first.
- ✓ Insert sample data (optional, to have data for SELECT).
- ✓ Then run the test script as restricted user.

💬 Let me know if you'd like me to also prepare this as a **ready-to-download .sql file**, or a combined script including grants/revokes too. Just say **"Yes, prepare file!"** 🚀