**How To Create User Credentials And Grant Privileges In Postgres**

Creating user credentials and grand privileges in Postgres can be tricky. You have to make sure each user has the right access level to the right tables and schemas. Each database works differently even they all are relational databases.

Postgres has its own way of creating users and granting privileges. It took me a few goes to work it out. If you just grant privileges for a user, the access privileges do not work for the table created in the future. The trickiest part is to run the alter default privilege statement so that the user can have the same privileges for the newly created tables, which is stated in the Postgres documentation on [ALTER DEFAULT PRIVILEGES](https://www.postgresql.org/docs/9.3/static/sql-alterdefaultprivileges.html).

Ok, let’s have a look at a scenario and script to make the magic happen!

**Scenarios**

I launched a [Postgres RDS instance](https://docs.aws.amazon.com/AmazonRDS/latest/UserGuide/CHAP_PostgreSQL.html) in AWS and created schemas with root account in the database called mydatahack. If you sign up for AWS, you have 12 months of free RDS access as [Amazone RDS Free Tier](https://aws.amazon.com/rds/free/), which is pretty cool. Of course, you can [download](https://www.postgresql.org/download/) and install it locally.

[cc lang="sql" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

create schema playground;

create schema datamart;

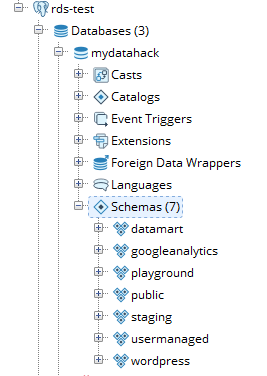
create schema staging;

create schema usermanaged;

create schema wordpress;

create schema googleanalytics;

[/cc]



I need to create 3 different users.

* developer: All access to all tables in all schemas
* readonly: Read only access to all tables in all schemas
* datahack: All access to all tables in the playground and usermanaged schemas and readonly for the rest

**Steps**

1. Create a group with CREATE ROLE statement
2. Grant Usage on Schema with GRANT USAGE ON SCHEMA statement
3. Grant Privileges on Exisiting Tables with [GRANT](https://www.postgresql.org/docs/9.0/static/sql-grant.html) <> On ALL TABLES statement
4. Set set the privileges that will be applied to objects created in the future with [ALTER DEFAULT PRIVILEGES](https://www.postgresql.org/docs/9.3/static/sql-alterdefaultprivileges.html)
5. Great a user with password and add to the user group

**Scripts**

**developer user**

[cc lang="sql" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

**readonly user**

[cc lang="sql" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

**datahack user**

[cc lang="sql" tab\_size="4" lines="-1" theme="mac-classic" line\_numbers="false"]

[/cc]

Yeah, you successfully created the users that meet the requirements.

Cool!

-- (1) Developer

Drop Role If Exists developeraccess;

CREATE ROLE developeraccess;

GRANT USAGE ON SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

To developeraccess;

GRANT ALL ON ALL TABLES IN SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

TO developeraccess;

ALTER DEFAULT PRIVILEGES IN SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

GRANT All ON TABLES TO developeraccess;

Drop User If Exists developer;

CREATE USER developer WITH PASSWORD 'Password1';

GRANT developeraccess TO developer;

-- (2) Readonly

Drop Role If Exists readaccess;

CREATE ROLE readaccess;

GRANT USAGE ON SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

To readaccess;

GRANT SELECT ON ALL TABLES IN SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

To readaccess;

ALTER DEFAULT PRIVILEGES IN SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

GRANT SELECT ON TABLES TO readaccess;

Drop User If Exists readonly;

CREATE USER readonly WITH PASSWORD 'Password1';

GRANT readaccess TO readonly;

-- (3) datahack

Drop Role If Exists datahackaccess;

CREATE ROLE datahackaccess;

GRANT USAGE ON SCHEMA

playground, datamart, staging, usermanaged, wordpress, googleanalytics

To datahackaccess;

GRANT ALL ON ALL TABLES IN SCHEMA

playground, usermanaged

To datahackaccess;

ALTER DEFAULT PRIVILEGES IN SCHEMA

playground, usermanaged

GRANT ALL ON TABLES TO datahackaccess;

GRANT SELECT ON ALL TABLES IN SCHEMA

datamart, staging, wordpress, googleanalytics

To datahackaccess;

ALTER DEFAULT PRIVILEGES IN SCHEMA

datamart, staging, wordpress, googleanalytics

GRANT SELECT ON TABLES TO datahackaccess;

Drop User If Exists datahack;

CREATE USER datahack WITH PASSWORD 'Password1';

GRANT datahackaccess TO datahack;