Granular database access documentation

Exposee

Database choice

After thorough research about databases that offers a granular access model, I landed on one of the databases that I have worked with before. I even remember working with giving different users different access levels, so SQL was the most obvious choice for this assignment.

I mistakenly started making a local database and entered the different values into it. From here I was thinking that I would migrate it into a hosted database on the azure portal. I soon ran into problems with migrating the database with SSMS (sql server management studio). It promted me that I had to make a private endpoint and that ended with a lot of work trying to make a private endpoint with multiple different virtual networks, subnets and private networks but it ended up not working as intended.

I then ended up working with giving the hosted database "Public network access". With this I could just make the required queries in the azure interface, and then access the data from a terminal from anywhere if the hosted server name was known.

I ran the following queries in the azure database query editor interface:

```
CREATE SCHEMA SItest;

CREATE TABLE people (
  data_id INT IDENTITY(1,1) PRIMARY KEY,
  name VARCHAR(255) NOT NULL
);

INSERT INTO people (name) VALUES ('Emil');

INSERT INTO people (name) VALUES ('Niklas');

INSERT INTO people (name) VALUES ('Grimmjow');

INSERT INTO people (name) VALUES ('Asmongold');

INSERT INTO people (name) VALUES ('Mågens');
```

Thereafter I made sure to make 3 users with different access levels. One that is completely powerless, one with read capabilities and one that has access to reading and writing. I came up with the following users and a shared password across them to make it easier for my partner to access them without having to rewrite everything:

USERNAME: slave PASSWORD: Lars123!

USERNAME: bruger PASSWORD: Lars123!

USERNAME: admin PASSWORD: Lars123!

I made sure they had the permissions with the following queries:

REVOKE CONTROL on dbo.people TO slave; Completely revoking his privileges

GRANT SELECT ON dbo.people TO bruger; Giving him access to read from the database

GRANT INSERT ON dbo.people TO admin; Giving him access to insert into the database

GRANT UPDATE ON dbo.people TO admin; Giving him access to update in the database

GRANT DELETE ON dbo.people TO admin; Giving him access to delete from the database

GRANT SELECT ON dbo.people TO admin; Giving him access to read from the database