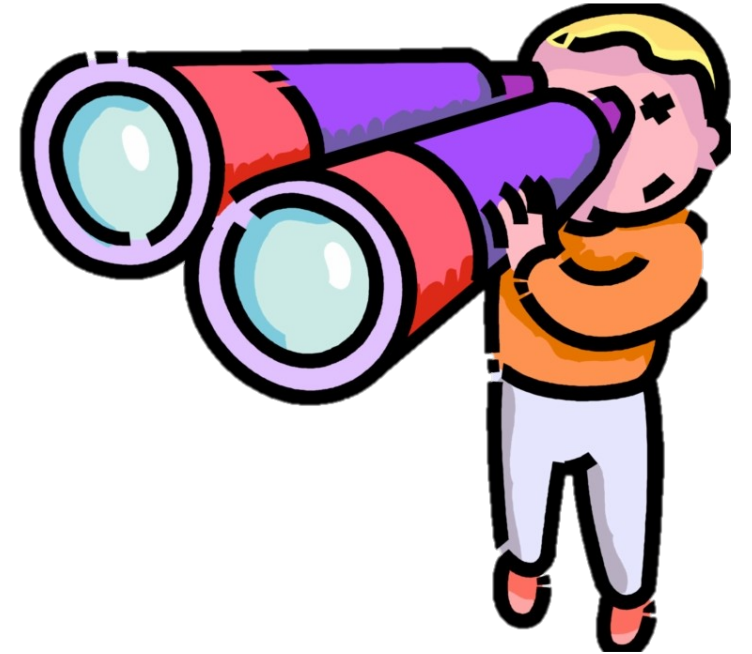


Views

Beveiliging



wim.bertels@ucll.be

Naamsvermelding-NietCommercieel-GelijkDelen 4.0
Unported Licentie

Views

- **CREATE VIEW**

- View: tabel die zichtbaar is voor gebruiker maar geen opslagruimte inneemt.
- Je kan conceptueel een view beschouwen als zijnde een virtuele tabel, dewelke wordt opgebouwd als die opgevraagd wordt.
- Een view wordt gemaakt op basis van een query.

```
CREATE VIEW leeftijden (spelersnr, leeftijd) AS  
    SELECT spelersnr, age(geb_datum)  
    FROM spelers;
```

```
CREATE VIEW leeftijden AS  
    SELECT spelersnr, age(geb_datum) AS leeftijd  
    FROM spelers;
```

Nut?

- Vereenvoudigen van routinematige instructies
- Reorganiseren van tabellen
- Stapsgewijs opzetten van SELECT-instructies
- **Beveiligen** van gegevens
- ..

Beveiliging

- Beveiliging
- SQL gebruiker:
 - moet gekend zijn
 - wachtwoord
 - expliciete toekenning van bevoegdheden:
 - Kolombevoegdheden
 - Tabelbevoegdheden
 - Databasebevoegdheden
 - Gebruikersbevoegdheden



Beveiliging

1. Invoeren/verwijderen van gebruikers
2. Tabel- en kolombevoegdheden
3. Databasebevoegdheden
4. Gebruikersbevoegdheden
5. WITH GRANT OPTION
6. Werken met rollen
7. Intrekken van bevoegdheden
8. Beveiliging van en met views

1. Invoeren/verwijderen van gebruikers

- CREATE USER: creëert een user

Vb. CREATE USER Frank IDENTIFIED BY Frank_pw

- ALTER USER: verandert het paswoord

Vb. ALTER USER Frank IDENTIFIED BY Frank_pasw

- DROP USER: verwijdt een user

Vb. DROP USER Frank

2. Tabel- en kolombevoegdheden

- GRANT: kent bevoegdheden toe
- Soorten tabelbevoegdheden:
 - SELECT: bevoegdheid tot SELECT en VIEW
 - INSERT: rijen toevoegen m.b.v. INSERT
 - DELETE: rijen verwijderen m.b.v. DELETE
 - TRUNCATE: tabel leegmaken m.b.v. TRUNCATE
 - UPDATE: rijen wijzigen m.b.v. UPDATE
 - REFERENCES: refererende sleutels naar deze tabel creëren
 - *TRIGGER: trigger op deze tabel aanmaken*
 - (alter en drop zijn gereserveerd voor de eigenaar, via ALTER TABLE gaan)
 - *CREATE: bv INDEX*
 - ALL/ALL PRIVILEGES: alle bevoegdheden

3. Databasebevoegdheden

- Soorten databasebevoegdheden:
 - CREATE: je mag objecten aanmaken
 - CONNECT: je mag de db gebruiken
 - TEMP: je mag tijdelijke objecten aanmaken

<https://www.postgresql.org/docs/current/sql-grant.html>

<https://www.postgresql.org/docs/current/sql-revoke.html>

4. Gebruikersbevoegdheden

- Databasebevoegdheden toekennen aan gebruiker(s)
- CREATE USER: gebruiker aanmaken

Vb. GRANT SELECT
ON ALL TABLES IN SCHEMA x
TO franky;

5. WITH GRANT OPTION

- WITH GRANT OPTION:
de gebruikers die toegang krijgen, kunnen deze machtiging doorgeven aan andere gebruikers

Vb. GRANT ALL ON spelers TO Frank
WITH GRANT OPTION;

6. Werken met rollen

- CREATE ROLE: Creëert een rol met bevoegdheden voor een aantal users, als de rol verandert, veranderen de bevoegdheden voor al de betrokken users.

Vb. CREATE ROLE ADMIN

GRANT SELECT, INSERT ON spelers

TO admin;

GRANT admin TO Frank, Marc, Ann, Greet;

- DROP ROLE: verwijdert de rol
Vb. DROP ROLE admin;

7. Intrekken van bevoegdheden

- REVOKE: verwijdert bevoegdheid (en de afhankelijke bevoegdheden)

Vb:- REVOKE ALL
ON spelers
FROM Frank;

- REVOKE admin
FROM Marc;

- REVOKE SELECT
ON spelers
FROM admin;

8. Beveiliging van en met views

- Analooq voor bevoegdheden op views
- Kan gebruikt worden voor de beveiliging van gegevens (gebruiker krijgt enkel machtiging op een view, waarin een deel van de tabel gedefinieerd wordt).

8. Beveiliging van en met views

Vb.

```
CREATE USER Frank;
```

```
CREATE VIEW zichtbaar_deel AS  
  SELECT spelersnr, naam, voorletters  
  FROM spelers;
```

```
GRANT SELECT  
ON zichtbaar_deel  
TO Frank;
```

Vragen

- CREATE MATERIALIZED VIEW
 - Wat is verschil met CREATE VIEW?
 - Waarom zou dit nuttig kunnen zijn?
- REFRESH MATERIALIZED VIEW [CONCURRENTLY]
 - Waarvoor dient refresh en wat is concurrently?
- Is dit ISO sql?
- <https://www.postgresql.org/docs/current/sql-creatematerializedview.html>

Wim Bertels (CC)BY-SA-NC

References:

- * Slides Databanken, H. Martens
- * SQL Leerboek, R. Van der lans