

## Extra oefeningen hoofdstuk 11

Maak onderstaande oefeningen in de volgorde van opgave. Noteer bij elke oefening als iets niet gaat of anders verloopt dan je zou verwachten.

1. Maak een view loc\_us\_vu, die het location\_id, het adres, de stad en het country\_id geeft van de vestigingen in de VS. Je mag de namen van de kolommen behouden.

```
CREATE OR REPLACE VIEW loc_us_vu
AS SELECT location_id, street_address, city, country_id
FROM locations
WHERE country_id = 'US';
```

2. Maak bovenstaande oefening view met naam loc\_us\_cl\_vu en geef andere kolomnamen in de view.

```
CREATE OR REPLACE VIEW loc_us_cl_vu
(location, address, city, country)
AS SELECT location_id, street_address, city, country_id
FROM locations
WHERE country_id = 'US';
```

3. Voer nu volgende wijziging door in de view uit oefening 1: verander voor de locatie 1700 de naam van de stad in 'Seattle'.

```
UPDATE loc_us_vu
SET initcap(city)='Seattle'
WHERE location_id=1700;
```

4. Voeg een constraint toe aan de view uit oefening 1: enkel rijen met locaties uit de VS mogen gemanipuleerd worden.

```
CREATE OR REPLACE VIEW loc_us_vu
AS SELECT location_id, street_address, city, country_id
FROM locations
WHERE country_id = 'US',
```

```
WITH CHECK OPTION CONSTRAINT country_us_cs ;
```

5. Verander daarna het land\_id voor de locatie 1700 naar 'CA'.

```
UPDATE loc_us_vu
```

```
SET country_id='CA'
```

```
WHERE location_id=1700;
```

➔ dit lukt niet wegens constraint violation.

6. Schrijf een view die de namen van zowel de werknemers als hun baas toont, maar enkel indien de werknemer meer verdient dan zijn baas. De gebruiker die de view raadpleegt, mag niets kunnen wijzigen aan de gegevens.

```
CREATE OR REPLACE VIEW emp_mgr_vu
```

```
AS SELECT wn.last_name werknemer, wg.last_name werkgever
```

```
FROM employees wn, employees wg
```

```
WHERE wn.manager_id=wg.employee_id
```

```
AND wn.salary > wg.salary
```

```
WITH READ ONLY CONSTRAINT emp_mgr_vu_ch;
```

controle view:

```
SELECT *
```

```
FROM emp_mgr_vu;
```

met join uitgeschreven:

```
CREATE OR REPLACE VIEW emp_mgr_vu
```

```
AS SELECT wn.last_name werknemer, wg.last_name werkgever
```

```
FROM employees wn JOIN employees wg
```

```
ON wn.manager_id=wg.employee_id
```

```
WHERE wn.salary > wg.salary
```

```
WITH READ ONLY CONSTRAINT emp_mgr_vu_ch;
```

7. Schrijf een interactieve code die toelaat het personeelsnummer, de naam, de datum van indiensttreding en het departementsnummer van een aantal pas aangeworven personeelsleden in te geven. De datum wordt steeds

ingegeven op volgende manier: 25/02/1999.

Nagaan of er een sequentie bestaat voor het personeelsnummer (hopen op goede naamgevingen):

```
SELECT sequence_name  
FROM user_sequences;
```

```
INSERT INTO employees (employee_id, last_name, hire_date,  
department_id)  
VALUES(employees_seq.nextval, UPPER('&naam'), TO_DATE('&datum  
indiensttreding', 'DD/MM/YYYY'), &departementsnummer);  
SAVE newEmployee.sql;  
run dit script zo vaak als nodig.
```