

# PostgreSQL – Views

Stephan Karrer

## Weitere Datenbankobjekte unter Oracle

- **View**

Logische Sicht auf Daten in potentiell verschiedenen Tabellen

- **Sequenz**

Generierung von eindeutigen Zahlenwerten

- **Index**

Verbesserung der Zugriffsperformanz

## Sichten (Views)

**Base  
Table**

employees						
employee_id	last_name	job_id	manager_id	hire_date	salary	department_id
203	marvis	hr_rep	101	07-Jun-94	6500	40
204	baer	pr_rep	101	07-Jun-94	10000	70
205	higgins	ac_rep	101	07-Jun-94	12000	110
206	gietz	ac_account	205	07-Jun-94	8300	110

**View**

staff				
employee_id	last_name	job_id	manager_id	department_id
203	marvis	hr_rep	101	40
204	baer	pr_rep	101	70
205	higgins	ac_rep	101	110
206	gietz	ac_account	205	110

## Eigenschaften von Views

```
CREATE [OR REPLACE] [FORCE|NOFORCE] VIEW view  
    [(alias[, alias]...)]  
    AS subquery  
    [WITH CHECK OPTION [CONSTRAINT constraint]]  
    [WITH READ ONLY [CONSTRAINT constraint]];
```

- Views sind gespeicherte SQL-Anweisungen
- Können komplexe Abfragen verkörpern
- Unterliegen als DB-Objekte eigener Zugriffskontrolle
- Views können auf Basis anderer Views erstellt werden
- DML-Anweisungen sind mit Einschränkungen möglich

## Erzeugen, Ersetzen und Nutzen von Views

```
CREATE VIEW emp_view AS
  SELECT last_name, salary*12 annual_salary
  FROM employees
  WHERE department_id = 20;

SELECT * FROM emp_view;
```

## Erstellen eines komplexen Views

```
CREATE VIEW dept_sum_vu
  (name, minsal, maxsal, avgsal)
AS SELECT    d.department_name, MIN(e.salary),
             MAX(e.salary), AVG(e.salary)
  FROM      employees e JOIN departments d
  ON        (e.department_id = d.department_id)
  GROUP BY d.department_name;
```

- Komplexe Views
  - Verwendung von Funktionen und Gruppierungen
  - Verwendung mehrerer Tabellen
  - Verwendung von Unterabfragen
- DML ist auf einfachen Views üblicherweise möglich
- DML ist in der Regel nicht möglich bei komplexen Views

## View-Anomalie

```
CREATE VIEW clerk AS
    SELECT employee_id, last_name, department_id, job_id
    FROM employees
    WHERE job_id = 'PU_CLERK'
        or job_id = 'SH_CLERK'
        or job_id = 'ST_CLERK';

UPDATE clerk SET job_id = 'PU_MAN' WHERE employee_id = 118;

CREATE VIEW clerk AS
    SELECT employee_id, last_name, department_id, job_id
    FROM employees
    WHERE job_id = 'PU_CLERK' or job_id = 'SH_CLERK'
        or job_id = 'ST_CLERK'
    WITH CHECK OPTION ;
```

- Ein erfolgreich eingefügter oder veränderter Datensatz ist möglicherweise über den View nicht sichtbar!
- WITH CHECK OPTION verhindert diese Anomalie.

## Entfernen eines View

```
DROP VIEW clerk;
```

```
View dropped.
```

- Das Löschen eines Views führt zu keinem Datenverlust, da ja die Basistabellen nach wie vor vorhanden sind.