

Oracle-SQL – Weitere Schema-Objekte

Stephan Karrer

Weitere Datenbankobjekte unter Oracle

- **View**

Logische Sicht auf Daten in potentiell verschiedenen Tabellen

- **Sequenz**

Generierung von eindeutigen Zahlenwerten

- **Index**

Verbesserung der Zugriffsperformanz

- **Synonym**

Alias-Mechanismus für Objektnamen

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;
```

```
CREATE OR REPLACE VIEW emp_view (LASTNAME, ANN_SALARY) AS
  SELECT last_name, salary*12
  FROM employees
  WHERE department_id = 20;

SELECT * FROM emp_view;
```

Erstellen eines komplexen Views

```
CREATE OR REPLACE 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 Operationen auf Views

- Auf einfachen Views üblicherweise möglich
- Entfernen von Zeilen ist nicht möglich bei Verwendung von:
 - Gruppenfunktionen
 - GROUP BY -Klausel
 - Pseudospalte ROWNUM
- Modifizieren bzw Einfügen von Zeilen ist nicht möglich bei Verwendung von:
 - Gruppenfunktionen
 - GROUP BY -Klausel
 - Pseudospalte ROWNUM
 - DISTINCT
 - Spalten, die durch Ausdrücke definiert sind
 - NOT NULL Spalten, die nicht durch den View selektiert sind (nur bei INSERT)

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.

READONLY VIEW

```
CREATE OR REPLACE 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 READ ONLY
;
```

- WITH READ ONLY definiert von vornherein einen nicht schreibbaren VIEW
- CREATE OR REPLACE ersetzt den View, falls schon vorhanden

Entfernen eines View

```
DROP VIEW clerk;
```

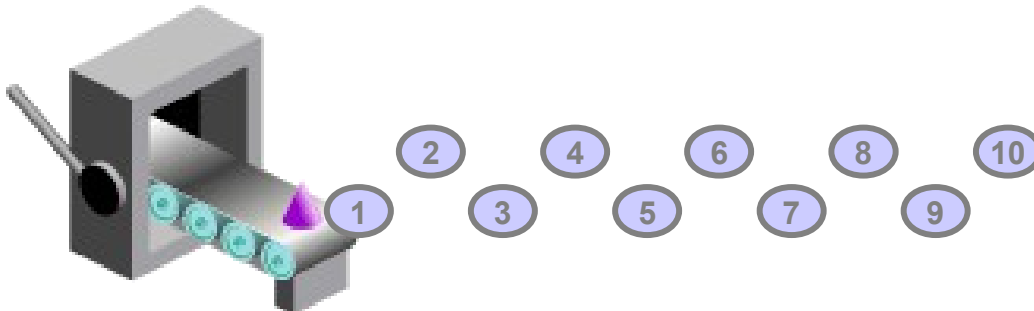
```
View dropped.
```

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

Sequenzen

■ Eigenschaften:

- Kann automatisch eindeutige Nummern erzeugen (Ticketautomat)
- Kann gemeinsam genutzt werden
- Kann für die Erzeugung eindeutiger Schlüsselwerte benutzt werden
- Die Werte können im Speicher vorgehalten (caching) werden, was den Zugriff beschleunigt



CREATE SEQUENCE: Syntax

```
CREATE SEQUENCE sequence
    [INCREMENT BY n]
    [START WITH n]
    [{MAXVALUE n | NOMAXVALUE}]
    [{MINVALUE n | NOMINVALUE}]
    [{CYCLE | NOCYCLE}]
    [{CACHE n | NOCACHE}];
```

```
-- Beispiel
CREATE SEQUENCE dept_deptid_seq
    INCREMENT BY 10
    START WITH 120
    MAXVALUE 9999
    NOCACHE
    NOCYCLE;
```

Pseudospalten für den Zugriff

```
INSERT INTO departments (department_id,  
                        department_name, location_id)  
VALUES      (dept_deptid_seq.NEXTVAL,  
            'Support', 2500);
```

```
SELECT      dept_deptid_seq.CURRVAL  
FROM        dual;
```

- **NEXTVAL**
liefert den nächsten verfügbaren Sequenzwert;
jeder Aufruf verbraucht damit einen Wert.
- **CURRVAL**
Zugriff auf den aktuell gehaltenen Wert;
ist mehrfach möglich.

Ändern und Löschen einer Sequenz

```
ALTER SEQUENCE dept_deptid_seq  
          INCREMENT BY 20  
          MAXVALUE 999999  
          NOCACHE  
          NOCYCLE;
```

- Nur die zukünftige Nummernvergabe ist betroffen
- Für einen anderen Startpunkt muss die Sequenz gelöscht und neu erzeugt werden

```
DROP SEQUENCE dept_deptid_seq;
```

Indizes

- Ist ein Schemaobjekt
- Kann Performanzgewinn bringen
- Existiert unabhängig von der Tabelle
- Wird automatisch durch den Server benutzt und verwaltet

Erzeugung und Löschen eines Index

- Oracle erzeugt automatisch einen Index für Primärschlüssel- und Unique-Spalten bei der Tabellenerzeugung.
- Der Benutzer kann zusätzliche Indizes auf (nicht-unique) Spalten erzeugen

```
CREATE INDEX emp_last_name_idx  
ON      employees(last_name);
```

```
DROP INDEX emp_last_name_idx;
```


Richtlinien für die Erzeugung eines Index

Create an index when:



A column contains a wide range of values



A column contains a large number of null values



One or more columns are frequently used together in a `WHERE` clause or a join condition



The table is large and most queries are expected to retrieve less than 2% to 4% of the rows in the table

Do not create an index when:



The columns are not often used as a condition in the query



The table is small or most queries are expected to retrieve more than 2% to 4% of the rows in the table



The table is updated frequently



The indexed columns are referenced as part of an expression

Synonyme

- Sind Aliase und können den Zugriff bequemer gestalten
- Nur Administratoren können PUBLIC-Synonyme erzeugen.

```
CREATE [PUBLIC] SYNONYM synonym  
FOR      object;
```

```
-- Beispiel  
CREATE SYNONYM d_sum  
FOR dept_sum_vu;  
  
-- ...  
  
DROP SYNONYM d_sum;
```