

Erzeugen und Verwenden von Views

Eine View ist wie ein Fenster, durch das Daten in Tabellen betrachtet und auch geändert werden können. Z.B. kann eine View aus der Tabelle EMP definiert werden, der nur die Columns EMPNO, ENAME, JOB und nur die Mitarbeiter aus Abteilung 10 enthält:

EMP10

EMPNO	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO	PROJNO
7839	KING	PRESIDENT		17-NOV-81	5000		10	102
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10	101
7934	MILLER	CLERK	7782	23-JAN-82	1300		10	102
7369	SMITH	CLERK	7902	17-DEC-80	800		20	101
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30	103
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30	103
7566	JONES	MANAGER	7839	02-APR-81	2975		20	101
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30	103
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30	101
7788	SCOTT	ANALYST	7566	09-DEC-82	3000		20	101
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30	102
7876	ADAMS	CLERK	7788	12-JAN-83	1100		20	101
7900	JAMES	CLERK	7698	03-DEC-81	950		30	102
7902	FORD	ANALYST	7566	03-DEC-81	3000		20	101
7954	CARTER	CLERK	7698	07-APR-84	1000		30	102
7955	WILSON	SALESMAN		22-APR-84	1650		30	102
7657	MASON	ANALYST	7566	24-APR-84	3910		20	101
7658	CHAN	ANALYST	7566	03-MAY-84	3450		20	101

Eine View ist wie eine Tabelle organisiert mit den eigenen Zeilen und Spalten und kann, abgesehen von einigen Einschränkungen, auch so behandelt werden.

Eine View besitzt keine eigenen Daten und benötigt daher auch keinen Speicherplatz.

Es gibt zwei wichtige Gründe für die Verwendung von Views:

- **Sicherheit:** Soll z.B. den Mitarbeitern der Zugriff auf die Tabelle EMP, jedoch nicht auf die Columns SAL und COMM, gewährt werden, so kann eine View definiert werden, der die Columns SAL und COMM nicht enthält. Der Zugriff auf die Tabelle EMP wird nicht gewährt, der Zugriff auf den View schon (Vergabe von Zugriffsberechtigungen siehe Kapitel 3.4).
- **Erleichterung:** Um bestimmte Daten zu erhalten, kann anstatt einer komplexen Query eine View erzeugt werden, um diese Information durch eine einfache Query zu erhalten.

Namenskonventionen:

Um Views von Tabellen zu unterscheiden, sollte den View-Namen **_view** hinzugefügt werden.

Zur Erzeugung einer Views wird das CREATE VIEW Kommando verwendet:

CREATE VIEW *viewname*
AS [*query*];

Um die View für Abteilung 10 mit den Columns E_EMPNO, E_ENAME und E_JOB zu erhalten, wird folgendes Kommando eingegeben:

```
create view emp10_view
as select e_empno, e_ename, e_job
from e_emp
where e_d_deptno=10;
```

Bei der Erzeugung einer View kann jedes erlaubte SELECT Kommando verwendet werden mit einer Ausnahme: die ORDER BY Klausel darf nicht verwendet werden.

Sollten die Zeilen in bestimmter Reihenfolge dargestellt werden, so muss die ORDER BY Klausel beim SELECT Kommando mit der View verwendet werden.

Es ist auch möglich, aus einer View eine weitere View zu erzeugen.

Eine Query auf eine View kann genauso durchgeführt werden wie auf eine Tabelle. Um die Informationen aus der View EMP10_VIEW zu erhalten, ist folgendes Kommando einzugeben:

```
select * from emp10_view;
```

Natürlich kann auch eine WHERE Bedingung verwendet werden:

```
select *
from emp10_view
where e_empno>7800;
```

Werden die Daten in der Tabelle geändert, so werden diese Änderungen auch in den zu dieser Tabelle gehörenden Views sichtbar.

Wird z.B. der Job von MILLER geändert, so wird diese Änderung auch im View EMP10 angezeigt:

```
update e_emp
set e_job='analyst'
where e_ename='MILLER';
```

```
select * from emp10_view;
```

Werden Zeilen in den Tabellen hinzugefügt oder gelöscht, so werden auch diese Änderungen im View sichtbar. Im folgenden Beispiel wird ein neuer Mitarbeiter hinzugefügt und danach die Daten des Views in alphabetischer Reihenfolge ausgegeben:

```
insert into e_emp  
values (7999, 'brown', 'clerk', 7902, '1983-01-  
25', 3800, NULL, 10, 102);
```

```
select *  
from emp10_view  
order by e_ename;
```

View ändern

Update:

In Views können die Daten auch direkt upgedated werden. Das SELECT, das beim Erzeugen des Views verwendet wurde, muss allerdings folgende Bedingungen erfüllen:

- Daten werden nur aus einer Tabelle selektiert
- Keine Verwendung der GROUP BY Klausel, der DISTINCT Klausel oder einer Gruppenfunktion
- Eine Column, die verändert wird, darf nicht durch einen Ausdruck erzeugt worden sein

Soll z.B. der Job von MILLER wieder umgeändert werden, so kann diese Änderung auch in der View EMP10_VIEW durchgeführt werden:

```
update emp10_view  
set e_job = 'clerk'  
where e_ename = 'miller';
```

Delete:

Um Zeilen aus einer View zu löschen, muss das SELECT, das beim Erzeugen der View verwendet wurde, folgende Bedingungen erfüllen:

- Daten werden nur aus einer Tabelle selektiert
- Keine Verwendung der GROUP BY Klausel, der DISTINCT Klausel oder einer Gruppenfunktion

Insert:

Um Zeilen in eine Tabelle über einen View einzufügen, muss das SELECT, das beim Erzeugen der View verwendet wurde, folgende Bedingungen erfüllen:

- Daten werden nur aus einer Tabelle selektiert
- Keine Verwendung der GROUP BY Klausel, der DISTINCT Klausel oder einer Gruppenfunktion
- Eine Column darf nicht durch einen Ausdruck erzeugt worden sein
- Jede NOT NULL Column der Tabelle muss in der View ebenfalls vorhanden sein

Views aus mehr als einer Tabelle

Um eine View aus mehreren Tabellen zu erzeugen, wird beim SELECT ein Join verwendet.

Erzeugen einer Views PROJSTAFF_VIEW aus E_EMP und P_PROJ:

```
create view projstaff_view (employee, project, project_number)
as select e_ename, p_pname, e_emp.e_p_projnr
from e_emp join p_proj
on e_emp.e_p_projnr = p_proj.p_projnr;
```

Hier werden in Klammern die Namen der Columns im View definiert, die den Colum-Namen im Query zugeordnet werden (E_ENAME wird in der View zu EMPLOYEE, P_PNAME zu PROJECT und P_PROJNR zu PROJECT_NUMBER).

Um jetzt die Mitarbeiter am Projekt 101 aufzulisten, kann diese View verwendet werden:

```
select employee, project
from projstaff_view
where project_number=101;
```

In dieser View ist kein UPDATE, INSERT oder DELETE möglich, da bei der Erzeugung mehr als eine Tabelle verwendet worden ist.

Es ist auch möglich, mehr als zwei Tabellen bei der View-Erzeugung zu verwenden. Im folgenden Beispiel, wird die View PROJECTS_VIEW erzeugt, welche die Projektbezeichnungen, die Mitarbeiternamen, die Mitarbeiternummern und die entsprechenden Städtenamen enthält:

```
create view projects_view
(project, employee, emp_number, location)
AS SELECT p_pname, e_ename, e_empno, d_loc
from p_proj join e_emp on e_emp.e_p_projnr = p_proj.p_projnr
            join d_dept on e_emp.e_d_deptno = d_dept.d_deptno;
```

Um die Projekte und Mitarbeiter aus New York aufzulisten, kann die View PROJECTS_VIEW verwendet werden:

```
select project, employee, location
from projects_view
where location = 'NEW YORK';
```

Es können auch Views mit Tabellen oder anderen Views gejoint werden.

Verwendung von Ausdrücken und Funktionen in Views

Werden bei der Erzeugung von Views Ausdrücke oder Funktionen verwendet, um Columns im View zu definieren, so müssen diesen Columns Namen gegeben werden. Das kann entweder in der CREATE VIEW Klausel oder im Query über einen Alias-Namen durchgeführt werden.

Soll z.B. eine View definiert werden, die zu jedem Mitarbeiter das Jahresgehalt enthält, so ist folgendes Kommando einzugeben:

```
create view pay_view (name,monthly_sal,annual_sal,deptno)
as select e_ename, e_sal,e_sal*12,e_d_deptno
from e_emp;
```

```
select * from pay_view where deptno=30;
```

Weiters können in der Query beim Erzeugen von Views auch Gruppenfunktionen und die GROUP BY Klausel verwendet werden.

In folgender View werden Gehaltsstatistiken über die Abteilungen zusammengefasst:

```
create view dept_comp_view(deptno,lo_sal,med_sal,hi_sal,tot_sal)
as select e_d_deptno,min(e_sal),avg(e_sal),max(e_sal),sum(e_sal)
from e_emp
group by e_d_deptno;
```

Über die View können nun die Statistikdaten ausgegeben werden:

```
select * from dept_comp_view;
```

Wird bei einer Tabelle, aus der auch Views erzeugt worden sind, die Tabellenstruktur verändert, so sollen die davon betroffenen Views gelöscht und neu erzeugt werden.