

Views

In SQL ist eine View eine virtuelle Tabelle, die aus den Ergebnissen einer SELECT-Abfrage erstellt wird. Views ermöglichen es, komplexe Abfragen in benannten, leicht zugänglichen Ansichten zu speichern, die von anderen Abfragen und Anwendungen wie normale Tabellen verwendet werden können. Hier ist eine Erklärung und einige Beispiele für die Verwendung von Views:

Erstellen einer View:

Um eine View zu erstellen, verwenden Sie die CREATE VIEW-Anweisung. Hier ist die grundlegende Syntax:

```
CREATE VIEW view_name AS
SELECT Spalte1, Spalte2, ...
FROM Tabelle
WHERE Bedingung;
```

Nun einige Beispiele:

Beispiel 1: Einfache View erstellen:

Angenommen, wir haben eine Tabelle "Kunden" und möchten eine View erstellen, die nur die Kunden aus einem bestimmten Land anzeigt:

```
CREATE VIEW KundenAusDeutschland AS
SELECT Vorname, Nachname, Land
FROM Kunden
WHERE Land = 'Deutschland';
```

Nachdem diese View erstellt wurde, können Sie sie wie eine normale Tabelle verwenden, um die Kunden aus Deutschland abzurufen:

```
SELECT * FROM KundenAusDeutschland;
```

Beispiel 2: Join in einer View verwenden:

Sie können auch Joins in einer View verwenden, um Daten aus mehreren Tabellen zu kombinieren. Angenommen, wir haben Tabellen "Kunden" und "Bestellungen", und wir möchten eine View erstellen, die Kundeninformationen und Bestelldaten kombiniert:

```
CREATE VIEW KundenBestellungen AS
SELECT Kunden.Name, Bestellungen.Bestelldatum
FROM Kunden
INNER JOIN Bestellungen ON Kunden.KundenID = Bestellungen.KundenID;
```

Jetzt können Sie die View verwenden, um Kunden und ihre Bestellungen abzurufen:

```
SELECT * FROM KundenBestellungen;
```

Beispiel 3: Aggregatfunktionen in einer View verwenden:

Views können auch Aggregatfunktionen verwenden, um zusammengefasste Informationen bereitzustellen. Angenommen, wir haben eine Tabelle "Bestellungen" und möchten eine View erstellen, die die Gesamtzahl der Bestellungen pro Kunde anzeigt:

```
CREATE VIEW BestellungenProKunde AS  
SELECT KundenID, COUNT(Bestellnummer) AS AnzahlBestellungen  
FROM Bestellungen  
GROUP BY KundenID;
```

Jetzt können Sie die View verwenden, um die Anzahl der Bestellungen pro Kunde abzurufen:

```
SELECT * FROM BestellungenProKunde;
```

Beispiel 4: View bearbeiten oder aktualisieren:

Sie können auch eine vorhandene View ändern oder aktualisieren. Angenommen, Sie möchten die View "KundenAusDeutschland" um eine zusätzliche Spalte erweitern:

```
CREATE OR REPLACE VIEW KundenAusDeutschland AS  
SELECT Vorname, Nachname, Land, Stadt  
FROM Kunden  
WHERE Land = 'Deutschland';
```

Die **CREATE OR REPLACE**-Syntax ermöglicht es Ihnen, eine vorhandene View zu aktualisieren, wenn sie bereits existiert.

Views sind äußerst nützlich, um komplexe Abfragen zu vereinfachen, die Sicherheit zu verbessern (da Sie Zugriffsrechte auf Views verwalten können) und die Wiederverwendbarkeit von Abfragen zu fördern. Sie ermöglichen es, Daten in einer für Benutzer und Anwendungen leicht verständlichen Form bereitzustellen, ohne die zugrunde liegende Datenstruktur zu ändern.