## Представления

Идея представления (*View*) состоит в следующем: определить запрос, который предполагается использовать достаточно часто, сохранить его в базе данных в виде объекта ORACLE и разрешить пользователям обращаться к нему по имени, как к обычной таблице. Когда пользователь выбирает данные из представления, СУБД выполняет соответствующий запрос, организует результаты так, как определено в представлении, и выдает их пользователю. Для пользователя представление выглядит как таблица, из которой поступают данные. Однако на самом деле данные поступают через представление, из одного или нескольких других источников – исходных таблиц или других представлений.

Запрос, непосредственное выполнение которого можно заменить использованием представлений, может быть какой угодно сложности. В сущности, это и есть одно из назначений использования представлений – скрытие сложности запросов и структуры данных от пользователей.

Кроме того, представления обеспечивают дополнительный уровень безопасности базы данных, поскольку позволяют ограничивать диапазон строк и столбцов, возвращаемых пользователям. Если нужно скрыть от пользователей определенные столбцы или строки той или иной таблицы (таблиц), эти строки или столбцы достаточно исключить из результатов запроса при помощи операций горизонтальной и вертикальной фильтрации, тем самым, исключив и из представления. Если после этого закрыть доступ к исходным таблицам, из которых при помощи запроса выбираются данные, то для пользователей, имеющих доступ только к представлению, скрытые столбцы или строки просто не будут существовать.

Наконец, представления могут сделать работу с таблицами более удобной. При создании представлений можно как угодно называть имена столбцов, а также изменять порядок их отображения. Эта возможность часто используется тогда, когда необходимо подвести под некий общий стандарт, действующий внутри той или иной организации или рабочей группе, имена всех столбцов таким образом, чтобы они были понятным определенной категории пользователей.

### Создание и удаление представлений

Для создания представлений используется следующий синтаксис:

CREATE [OR REPLACE] VIEW *имя\_представления* AS

*оператор SELECT*

[WITH CHECK OPTION];

Использование необязательного параметра OR REPLACE позволяет заменить уже существующее представление с таким же именем новым. Если существующего представления с таким именем нет, то этот параметр игнорируется.

Примеры создания представлений:

-- создание представления, содержащего все поля таблицы

-- Students за исключением поля N\_tel (скрытие данных)

CREATE OR REPLACE VIEW Students\_V1 AS

SELECT N\_z, S\_Name, F\_Name, N\_gr,B\_Date FROM Students;

 Удаление представлений, как и всех других объектов базы данных, осуществляется при помощи оператора DROP по имени представления, например:

DROP VIEW Students\_V1;

### Операции выборки из представлений

Для пользователя выборка данных из представления ничем не отличается от выборки данных из таблицы. Иногда пользователи и не догадываются, пользуются они реальной таблицей или представлением. Выборка данных из представлений осуществляется так же, как и из таблиц – при помощи оператора SELECT, например:

-- выборка всей информации обо всех студентах

-- за исключением номера телефона(его мы изначально не включили в представление) и даты рождения, отсортированной по номеру группы.

SELECT N\_z, S\_Name, F\_Name, N\_gr FROM Students\_V1

ORDER BY N\_gr;

### Обновляемые представления

Если к представлению можно применить операторы обновления (INSERT, UPDATE или DELETE), то представление является *обновляемым* (updateble), иначе оно является *читаемым* (read-only).

Ниже приведены критерии того, является ли представление обновляемым в SQL:

* оно базируется на одной таблице;
* оно должно включать первичный ключ таблицы;
* оно не должно включать полей, полученных в результате применения функций агрегирования;
* оно не может содержать спецификации DISTINCT;
* оно не должно использовать GROUP BY или HAVING;
* оно не должно использовать подзапросы;
* оно может быть определено на другом представлении, но это представление должно быть обновляемым;
* оно не может содержать константы, строки или выражения в списке выбираемых выходных полей;
* для INSERT оно должно включать поля из таблицы, которые имеют ограничения NOT NULL.

Для обновляемых представлений применимы любые операторы обновления, которые при выполнении по отношению к представлению транслируются на исходную таблицу, на основе которой было создано представление. Однако это правило на первый взгляд не согласуется с функцией представлений по защите данных, описанной ранее. Действительно, если представить, что при помощи представления были скрыты столбцы или строки, но пользователь знает об их существовании, он может выполнить операции обновления данных, затрагивающие скрытую в представлении информацию.

Для предотвращения описанной ситуации применяется параметр WITH CHECK OPTION, который может указываться при создании обновляемых представлений после оператора SELECT. При наличии этого параметра СУБД предотвращает операции обновления данных, если они нарушают условия горизонтальной или вертикальной фильтрации, определенные в предложении в операторе SELECT:

-- создание обновляемого представления с защитой от

-- изменений скрытого столбца

CREATE OR REPLACE VIEW V2 AS

SELECT N\_z, S\_Name, F\_Name, N\_gr FROM Students\_V1

WITH CHECK OPTION;