

Лабораторная работа №11

Представления

1. Использование представлений

Представление является еще одним объектом, составляющим логическую структуру любой базы данных. Представление для конечных пользователей выглядит как таблица, однако при этом не содержит данных, а лишь представляет их. Физически же представляемые данные расположены в таблицах базы данных. Представление создается в текущей БД и хранится как отдельный объект.

Представление реализуется в виде сохраненного запроса, на основе которого и производится выборка из различных таблиц базы данных. Представления обладают следующими преимуществами:

- позволяют сделать так, что разные пользователи БД будут видеть ее по-разному;
- обеспечивают конфиденциальность информации, т. к. позволяют отобразить только необходимую информацию, скрывая определенные поля или строки;
- управляют правами доступа к данным, например, вместо того чтобы предоставлять права на выполнение запросов к определенным полям таблиц, проще разрешить выполнение запросов через представление;
- упрощают визуальное восприятие данных, поскольку пользователь работает с представлением как с единой таблицей, которая создана на основе выборки данных из нескольких таблиц.

Наряду с преимуществами, представления обладают и двумя существенными недостатками.

- Представление создает лишь видимость существования соответствующей таблицы, и СУБД приходится преобразовывать запрос к представлению в запрос к исходным таблицам. Если представление отображает многотабличный запрос, то простой запрос к представлению становится сложным объединением, и на его выполнение может потребоваться много времени, что сказывается на производительности вычислений.
- Когда пользователь пытается обновить строки представления, СУБД должна установить их соответствие строкам исходных таблиц, а также обновить последние. Это возможно только для простых представлений (созданных на основе одной таблицы); сложные представления обновлять нельзя, они доступны только для выборки.

Указанные недостатки означают, что не стоит без разбора применять представления вместо исходных таблиц. В каждом конкретном случае необходимо учитывать перечисленные преимущества и недостатки представлений.

2. Создание представлений

Для создания представления используется команда CREATE VIEW:

```
CREATE VIEW ИмяПредставления
AS
ЗапросНаВыборку
```

При указании имени представления необходимо придерживаться ранее определенных правил именования объектов, также это имя не должно совпадать с именем уже существующей таблицы в базе данных. После имени представления в скобках могут быть перечислены его поля.

ЗапросНаВыборку представляет собой оператор SELECT, параметры которого и определяют содержимое представления. Имена полей представления задаются либо с помощью псевдонимов в операторе выборки, либо перечисляются после имени представления в скобках.

Например, для некоторой БД организации создадим представление, содержащее лишь следующую информацию о сотрудниках: номер, имя и фамилия сотрудника в одном поле, дата рождения в формате 00.00.0000.

```
CREATE VIEW InfoEmployees
AS
SELECT EmployeeKey, FirstName || ' ' || LastName, BirthDate
FROM DimEmployee
```

Содержимое представления можно просмотреть так же, как содержимое таблицы: через Обозреватель объектов или сформировав запрос: **SELECT * FROM InfoEmployees**.

С помощью данного представления ограничен доступ к некоторым полям исходной таблицы DimEmployees, в этом случае говорят, что на таблицу наложен **вертикальный фильтр**, т. е. ограничен доступ к части полей таблицы без защиты на уровне столбцов.

Если в коде запроса выборки определено условие отбора записей, то говорят, что на таблицу наложен **горизонтальный фильтр**.

Например, следующее представление обеспечивает доступ к информации только о тех заказчиках, образование которых High School:

```
CREATE VIEW InfoCustomersHighSchool
AS
SELECT FirstName, LastName, EnglishEducation
FROM DimCustomer
WHERE EnglishEducation='High School'
```

В запросе выборки может быть указана команда **SELECT** любой сложности, однако при этом **не рекомендуется использовать раздел ORDER BY**, который в дальнейшем можно применить при выборке данных из созданного представления. Также рекомендуется при создании представлений на основе нескольких таблиц использовать **только внутреннее соединение**.

Например, создадим представление, отображающее для каждого заказчика суммарную стоимость его заказов:

```
CREATE VIEW InfoOrders
AS
SELECT FirstName || ' ' || LastName, sum(UnitPrice*OrderQuantity)
FROM (DimCustomer INNER JOIN FactInternetSales
      ON DimCustomer.CustomerKey=FactInternetSales.CustomerKey)
GROUP BY FirstName || ' ' || LastName
```

3. Управление представлениями

Созданное представление может быть изменено выполнением команды **ALTER VIEW**, имеющей синтаксис, аналогичный команде **CREATE VIEW**.

Для удаления представления необходимо выполнить команду:

```
DROP VIEW ИмяПредставления
```

Задание 1. Создайте представление, содержащее следующую информацию о студентах: номер зачетной книжки, фамилия и инициалы в одном поле, дата рождения в формате 00.00.0000.

Задание 2.

А) Создайте представление «Занятость кабинетов», отражающее для каждого кабинета количество проводимых в нем занятий.

Б) Модифицируйте представление «Занятость кабинетов», выведя 3 наименее загруженных компьютерных класса.

Задание 3. Выведите статистику успеваемости студентов **по месяцам и годам**. В статистике отобразите средний балл и количество студентов, сдававших контрольную точку в указанный период.

Указание: Создайте сначала представление, содержащее номер зачетки, оценку, год и месяц контрольной точки. Затем на его основе создайте запрос.

Рекомендация: Чтобы получить год и месяц сдачи контрольной точки вам необходимо «отбросить» первые три символа из строки, содержащей дату.

Задание 4. Определите количество дней рождения студентов по временам года.

Указание: Создайте сначала представление, выводящее номера зачетов по сезонам (временам года).

Рекомендация: Для выделения сезонов можно использовать следующую конструкцию (после SELECT):

```
Case
    When extract (month ...) in (1, 2, 12) Then 'Зима'
    When extract (month ...) in (3, 4, 5) Then 'Весна'
    ...
    ...
End
as Сезон
```

Задание 5. Создайте представление, отражающее количество студентов, сдававших зимнюю и летнюю сессии.