

## Представления

На предыдущем этапе мы выполняли запросы к базе данных. Существенным недостатком запросов является то, что их формулировки не сохраняются в базе данных.

Для того чтобы преодолеть этот недостаток, придумали **представления** (view). *Представления* – это объекты базы данных, которые можно рассматривать как виртуальные таблицы. На самом деле хранится только формулировка команды SELECT, с помощью которой производится выборка данных из реальных таблиц.

Необходимость в использовании представлений возникает, например, в том случае, когда нужно запретить доступ пользователя к отдельным столбцам или строкам таблицы – тогда можно просто написать представление, в котором эти столбцы или строки не будут присутствовать, и предоставить доступ пользователю именно к этому представлению, а не к реальной таблице.

Другой полезной возможностью является вычисление значений, которые не хранятся непосредственно в таблице, но всегда могут быть рассчитаны.

Представление, как и запрос, может содержать информацию из разных таблиц.

Представления могут быть **обновляемыми** (т.е., предоставлять возможность не только чтения, но и изменения данных в исходных таблицах) и **необновляемыми**. Представление будет обновляемым только в том случае, если его структура такова, что SQL server может точно определить, в какие строки каких таблиц нужно поместить измененные данные. Необновляемыми будут, например, представления, содержащие итоговые данные и группировки.

Для создания представлений используется команда CREATE VIEW.

Краткий формат этой команды:

```
CREATE VIEW имя_представления AS  
Команда_SELECT
```

Например, создадим представление, содержащее список договоров и их кураторов для отдела с номером 1. *Будет ли это представление обновляемым?*

```
CREATE VIEW k_contract1
AS
SELECT k_contract.contract_num, k_contract.contract_date,
       k_contract.contract_type, k_contract.k_firm_firm_num,
       k_staff.staff_name
FROM k_contract INNER JOIN
     k_staff ON k_contract.k_staff_staff_num = k_staff.staff_num
WHERE k_dept_dept_num = 1
```

Для просмотра представления следует выполнить команду

```
SELECT * FROM k_contract1
```

Overview

Output

History

Result (1)

Export

	contract_num	contract_date	contract_type	k_firm_firm_num	staff_name
▶	1	2011-11-01	A	1	Иванов
	3	2011-09-01	C	1	Иванов
	6	2011-07-15	C	3	Иванов
	7	2011-11-12	A	4	Иванов

Проверим, является ли это представление обновляемым. Попробуем изменить, например, дату счета 1:

```
UPDATE k_contract1 SET contract_date='2011-11-02'
WHERE contract_num=1
```

Команда выполнена успешно, и повторный запрос на выборку дает следующий результат:

```
SELECT * FROM k_contract1
```

Overview

Output

History

Result (1)

Export

	contract_num	contract_date	contract_type	k_firm_firm_num	staff_name
▶	1	2011-11-02	A	1	Иванов
	3	2011-09-01	C	1	Иванов
	6	2011-07-15	C	3	Иванов
	7	2011-11-12	A	4	Иванов

Создадим вспомогательное представление для запросов о полностью оплаченных и частично оплаченных счетах (см. предыдущее занятие). Это представление для каждого счета содержит его номер и сумму оплаты.

```
CREATE VIEW k_pay_sum
AS
SELECT k_bill_bill_num, SUM(payment_sum) AS pay_sum
FROM k_payment
GROUP BY k_bill_bill_num
```

Для просмотра представления следует выполнить команду

```
SELECT * FROM k_pay_sum.
```

Overview	Output	History	Result (1)
Export			
	k_bill_bill_num	pay_sum	
►	2	1000.00	
	3	2000.00	
	4	1000.00	
	7	1000.00	
	8	1000.00	

Это представление не будет обновляемым. Проверим:

```
UPDATE k_pay_sum SET pay_sum=1500 WHERE k_bill_bill_num=1
```

Получим ошибку:

**ERROR 1288: The target table k\_pay\_sum of the UPDATE is not updatable.**

Действительно, **невозможно** изменить значение поля, в котором находится **сумма** чисел из разных строк.

Но нам и не требовалось обновлять данные в этом представлении. Теперь с помощью этого представления можно переформулировать запрос на выборку полностью оплаченных счетов, этот запрос станет проще:

*Полностью оплаченные счета*

```
SELECT    b.bill_num AS "Номер счета",
          b.bill_date AS "Дата счета",
          b.bill_sum AS "Сумма счета",
          p.pay_sum AS "Сумма оплаты"

FROM k_bill b, k_pay_sum p
WHERE b.bill_num=p.k_bill_bill_num AND
      b.bill_sum<=p.pay_sum
```

**Задание.** Создайте несколько (не менее 3) представлений для вашей базы данных. Будут ли они обновляемыми или нет? Проверьте.