

Министерство образования Республики Беларусь

Учреждение образования  
Белорусский государственный университет информатики и  
радиоэлектроники

Факультет компьютерного проектирования  
Кафедра инженерной психологии и эргономики  
Дисциплина: Базы данных

Лабораторная работа № 4

«Модификация данных. Создание представлений»

Выполнил:  
ст.гр. 113802  
Разумов Д.А.

Проверила:  
Василькова А.Н.

Минск 2022

Предположим, что нашей фирме нужно знать гражданство водителя. Для этого в таблицу drivers надо добавить столбец с гражданством водителя. Для добавления столбцов в таблицу используется оператор ALTER TABLE - ADD COLUMN. Его синтаксис следующий:

ALTER TABLE имя\_таблицы ADD COLUMN имя\_столбца;

На рисунке 1 приведен пример запроса с оператором ADD COLUMN.

```
ALTER TABLE users ADD COLUMN citizenship varchar(20);
SELECT * FROM drivers
```

rs	fullName	category	address	phone	callCode	dateOfBirth	experience	citizenship
	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	447104585	+375	1975-12-01	24	NULL
	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	445629451	+375	1991-05-22	8	NULL
	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	331144185	+375	1971-03-12	27	NULL
	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Слуцкое шоссе 61-67	294104582	+375	1968-01-22	26	NULL
	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	295012127	+375	1988-10-13	9	NULL
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 1 – Пример добавления столбца в конец таблицы

Для того, чтобы указать местоположение столбца используются ключевые слова: FIRST - новый столбец будет первым, и AFTER - указывает после какого столбца поместить новый.

Пример, добавим еще два столбца: один – familyStatus – семейное положение, а другой – rating - рейтинг водителя. Оба столбца вставим после поля address.

На рисунке 2 приведен пример запроса добавления столбца в таблицу в определенное место с помощью AFTER.

```
4 ALTER TABLE drivers ADD COLUMN familyStatus VARCHAR(20) AFTER address,
5 ADD COLUMN rating VARCHAR(20) AFTER address;
```

	idDrivers	fullName	category	address	rating	familyStatus	phone	callCode	dateOfBirth	experience	citizenship
1	1	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	NULL	NULL	447104585	+375	1975-12-01	24	NULL
2	2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	NULL	NULL	445629451	+375	1991-05-22	8	NULL
3	3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	NULL	NULL	331144185	+375	1971-03-12	27	NULL
4	4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Слуцкое шоссе 61-67	NULL	NULL	294104582	+375	1968-01-22	26	NULL
5	5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	NULL	NULL	295012127	+375	1988-10-13	9	NULL
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 2 – Добавление столбца с помощью AFTER

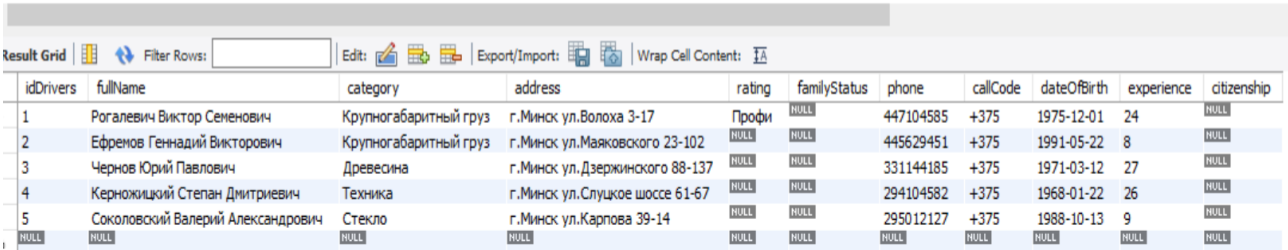
Теперь надо указать рейтинг какому-нибудь водителю, пусть это будет Рогалевич с id=1. Для обновления уже существующих данных служит оператор UPDATE.

Его синтаксис, следующий:

UPDATE имя\_таблицы SET имя\_столбца=значение\_столбца  
WHERE условие;

На рисунке 3 приведен данный пример добавления Рогалевичу рейтинга.

```
8 • UPDATE drivers SET rating = 'Профи'
9   WHERE idDrivers = 1;
10 • SELECT * FROM drivers
11
```



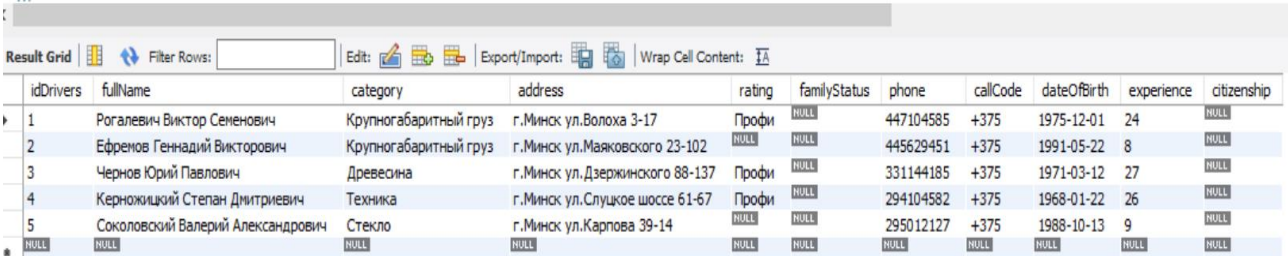
idDrivers	fullName	category	address	rating	familyStatus	phone	callCode	dateOfBirth	experience	citizenship
1	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	Профи	NULL	447104585	+375	1975-12-01	24	NULL
2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	NULL	NULL	445629451	+375	1991-05-22	8	NULL
3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	NULL	NULL	331144185	+375	1971-03-12	27	NULL
4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Случокское шоссе 61-67	NULL	NULL	294104582	+375	1968-01-22	26	NULL
5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	NULL	NULL	295012127	+375	1988-10-13	9	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 3 – Добавление рейтинга с помощью UPDATE

Изменять данные можно сразу в нескольких строках и во всей таблице.

Пример, зададим рейтинг «Профи» водителям, у которых стаж больше 20 лет.

```
12 • SET SQL_SAFE_UPDATES = 0;
13 • UPDATE drivers SET rating='Профи'
14   WHERE experience>20;
15 • SELECT * FROM drivers;
16
```



idDrivers	fullName	category	address	rating	familyStatus	phone	callCode	dateOfBirth	experience	citizenship
1	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	Профи	NULL	447104585	+375	1975-12-01	24	NULL
2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	NULL	NULL	445629451	+375	1991-05-22	8	NULL
3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	Профи	NULL	331144185	+375	1971-03-12	27	NULL
4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Случокское шоссе 61-67	Профи	NULL	294104582	+375	1968-01-22	26	NULL
5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	NULL	NULL	295012127	+375	1988-10-13	9	NULL
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 4 – Добавление рейтинга с помощью UPDATE и WHERE

Данные изменились в трех строках, согласно заданному условию. Понятно, что если в запросе опустить условие, то данные будут обновлены во всех строках таблицы.

Предположим, что нам не нравится название столбца Рейтинг, и мы хотим переименовать столбец с новым названием Репутация - reputation. Для изменения

имени существующего столбца используется оператор CHANGE. Его синтаксис, следующий:

ALTER TABLE имя\_таблицы CHANGE старое\_имя\_столбца новое\_имя\_столбца тип;

Пример на рисунке 5, поменяем rating на reputation:

ALTER TABLE users CHANGE rating reputation varchar (20);

```
17 • ALTER TABLE drivers CHANGE rating reputation VARCHAR(20);
18 • SELECT * FROM drivers;
```

19

	idDrivers	fullName	category	address	reputation	familyStatus	phone	callCode	dateOfBirth	experience	citizenship
▶	1	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	Профи	NULL	447104585	+375	1975-12-01	24	NULL
	2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	NULL	NULL	445629451	+375	1991-05-22	8	NULL
	3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	Профи	NULL	331144185	+375	1971-03-12	27	NULL
	4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Случокое шоссе 61-67	Профи	NULL	294104582	+375	1968-01-22	26	NULL
	5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	NULL	NULL	295012127	+375	1988-10-13	9	NULL

Рисунок 5 – Пример работы ALTER TABLE CHANGE

Если нам понадобится изменить только тип столбца, то мы будем использовать оператор MODIFY. Его синтаксис, следующий:

ALTER TABLE имя\_таблицы MODIFY имя\_столбца новый\_тип;

```
17 • ALTER TABLE drivers MODIFY familyStatus INT;
18 • SELECT * FROM drivers;
```

19

	idDrivers	fullName	category	address	reputation	familyStatus	phone	callCode	dateOfBirth	experience	citizenship
	1	Рогалевич Виктор Семенович	Крупногабаритный груз	г.Минск ул.Волоха 3-17	Профи	NULL	447104585	+375	1975-12-01	24	NULL
	2	Ефремов Геннадий Викторович	Крупногабаритный груз	г.Минск ул.Маяковского 23-102	NULL	NULL	445629451	+375	1991-05-22	8	NULL
	3	Чернов Юрий Павлович	Древесина	г.Минск ул.Дзержинского 88-137	Профи	NULL	331144185	+375	1971-03-12	27	NULL
	4	Керножицкий Степан Дмитриевич	Техника	г.Минск ул.Случокое шоссе 61-67	Профи	NULL	294104582	+375	1968-01-22	26	NULL
	5	Соколовский Валерий Александрович	Стекло	г.Минск ул.Карпова 39-14	NULL	NULL	295012127	+375	1988-10-13	9	NULL
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

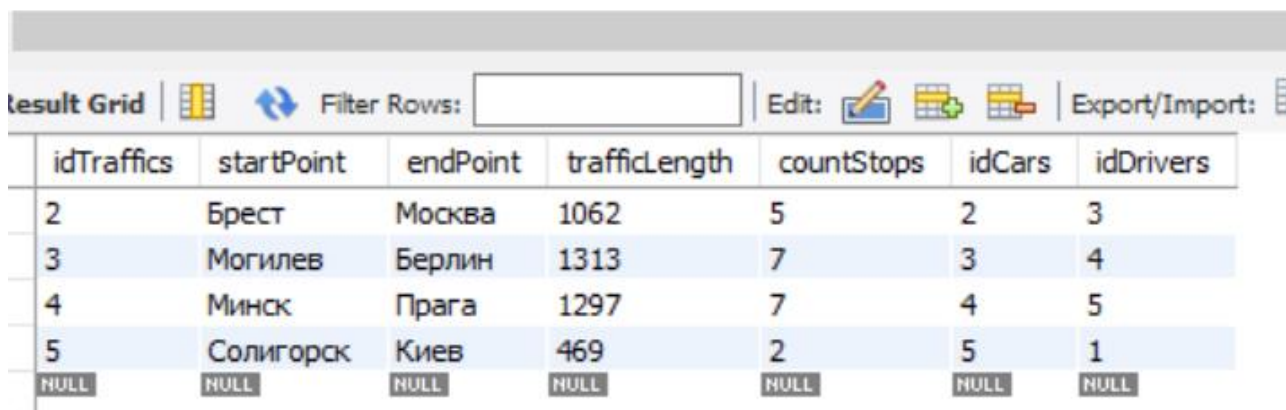
Рисунок 6 – Изменение типа столбца

Рассмотрим - оператор DELETE, который позволяет удалять строки из таблицы. Его синтаксис, следующий:

DELETE FROM имя\_таблицы  
WHERE условие;

Например, удалим из таблицы traffics те маршруты, по которым ехал водитель Ефремов (id=2):

```
9 • DELETE FROM traffics
10 WHERE idDrivers=2;
11 • SELECT * FROM traffics
```



idTraffics	startPoint	endPoint	trafficLength	countStops	idCars	idDrivers
2	Брест	Москва	1062	5	2	3
3	Могилев	Берлин	1313	7	3	4
4	Минск	Прага	1297	7	4	5
5	Солигорск	Киев	469	2	5	1
NULL	NULL	NULL	NULL	NULL	NULL	NULL

Рисунок 7 – Удаление из таблиц записи

Понятно, если опустить условие, то из таблицы будут удалены все данные. Следует помнить, что данные СУБД даст удалить только в том случае, если они не являются внешними ключами для данных из других таблиц (поддержка целостности БД).

## Представление

Представления или Views представляют виртуальные таблицы. Но в отличие от обычных стандартных таблиц в базе данных представления содержат запросы, которые динамически извлекают используемые данные.

Представления дают нам ряд преимуществ. Они упрощают комплексные SQLоперации. Они защищают данные, так как представления могут дать доступ к части таблицы, а не ко всей таблице. Представления также позволяют возвращать отформатированные значения из таблиц в нужной и удобной форме.

Для создания представления используется команда CREATE VIEW, которая имеет следующую форму:

```
CREATE VIEW название_представления [(столбец_1 столбец_2,...)]
AS выражение_SELECT
```

Рисунок 8 – Создание представления



Добавим в базу данных, в которой содержатся наши таблицы, представление. Код и само представление показаны на рисунке 9.

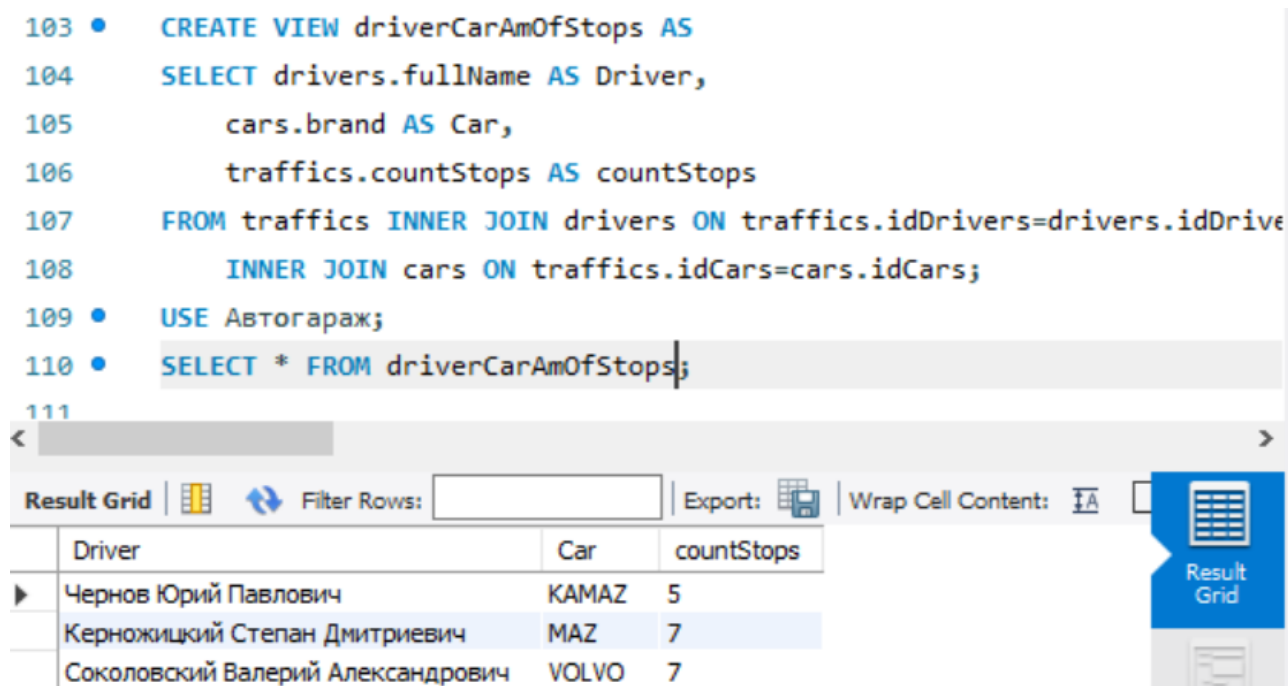


Рисунок 9 – Создание представления и его получение

При создании представлений следует учитывать, что представления, как и таблицы, должны иметь уникальные имена в рамках той же базы данных.

Представления могут иметь не более 1024 столбцов и могут обращаться не более чем к 256 таблицам.

Также можно создавать представления на основе других представлений. Такие представления еще называют вложенными (nested views). Однако уровень вложенности не может быть больше 32-х.

Команда SELECT, используемая в представлении, не может включать выражения INTO или ORDER BY (за исключением тех случаев, когда также применяется выражение TOP или OFFSET). Если же необходима сортировка данных в представлении, то выражение ORDER BY применяется в команде SELECT, которая извлекает данные из представления.

Для изменения представления используется команда ALTER VIEW. Эта команда имеет практически тот же самый синтаксис, что и CREATE VIEW:

```
ALTER VIEW название_представление [(столбец_1, столбец_2,...)]
AS выражение_SELECT
```

Рисунок 10 – Изменение представления

Например, изменим выше созданное представление driverCarAmOfStops. Результат показан на рисунке 11.

```
ALTER VIEW driverCarAmOfStops AS
SELECT drivers.fullName AS Driver,
       cars.brand AS Car,
       traffics.countStops AS AmountOfStops,
       drivers.experience AS Experience
FROM traffics INNER JOIN drivers ON traffics.idDrivers=drivers.idDriver
INNER JOIN cars ON traffics.idCars = cars.idCars;
```

Рисунок 11 – Изменения представления

Driver	Car	AmountOfStops	Experience
Чернов Юрий Павлович	KAMAZ	5	27
Керножицкий Степан Дмитриевич	MAZ	7	26
Соколовский Валерий Александрович	VOLVO	7	9

Рисунок 12 – Получение измененного представления

Для удаления представления вызывается команда DROP VIEW:  
Результат удаления показан на рисунке 11.

```
DROP VIEW driverCarAmOfStops
```

Рисунок 13 – Удаление представления

Также стоит отметить, что при удалении таблиц также следует удалить и представления, которые используют эти таблицы.

Исходя из полученных результатов мы получили навыки модификации данных и создания представлений на языке SQL в среде разработки MySQL Workbench.