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.

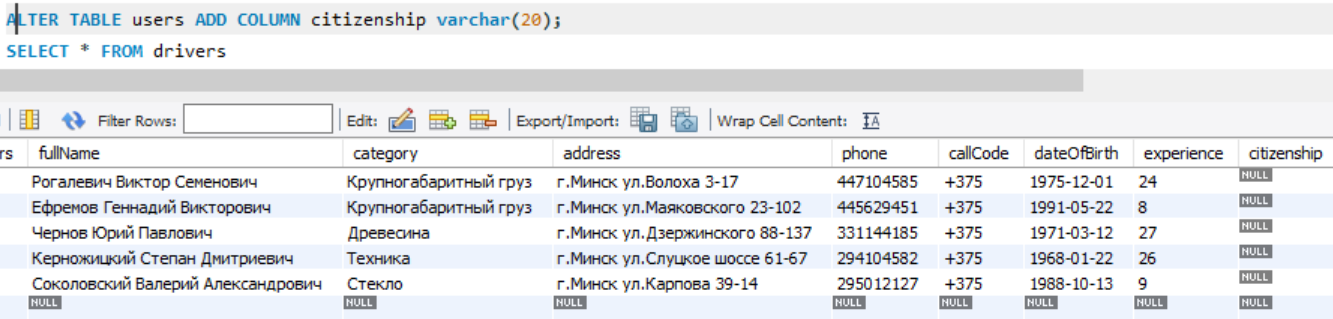


Рисунок 1 – Пример добавления столбца в конец таблицы

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 addres.

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

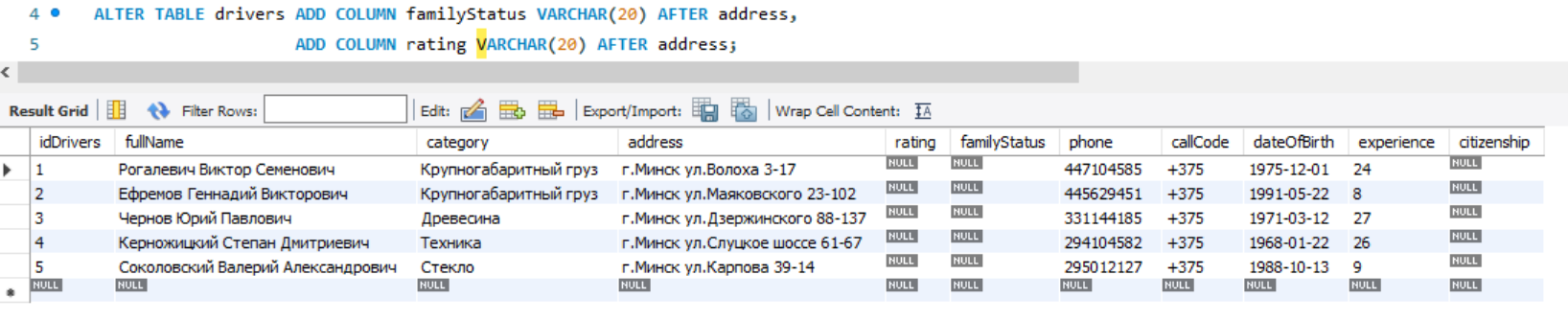


Рисунок 2 – Добавление столбца с помощью AFTER

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.

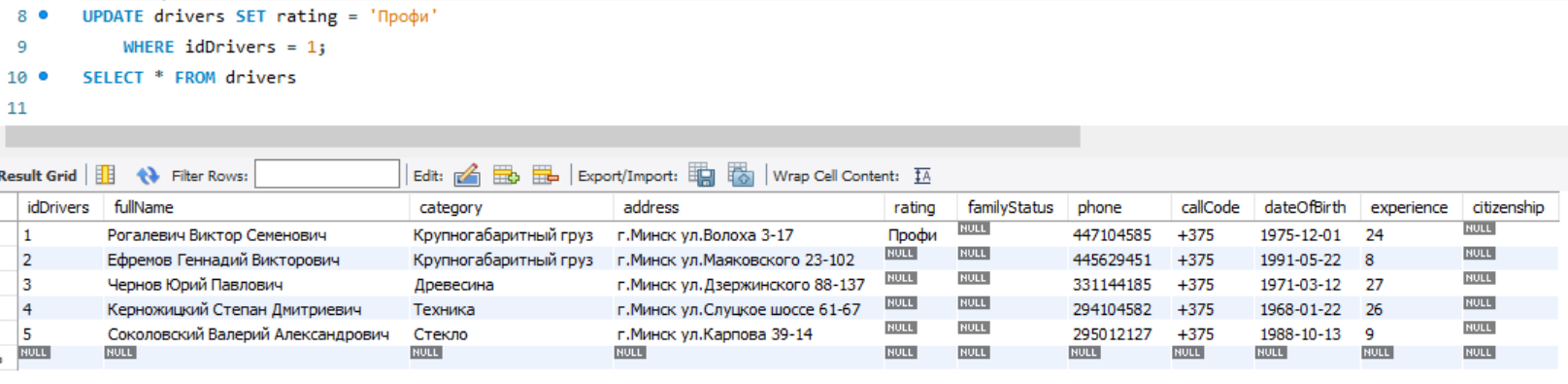


Рисунок 3 – Добавление рейтинга с помощью UPDATE

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

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

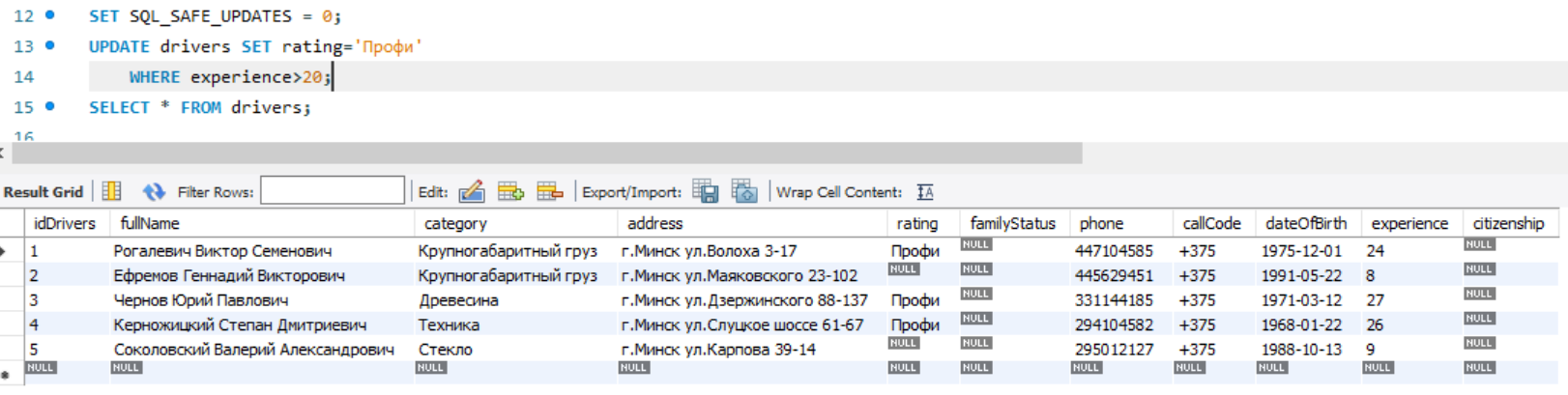


Рисунок 4 – Добавление рейтинга с помощью UPDATE и WHERE

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);

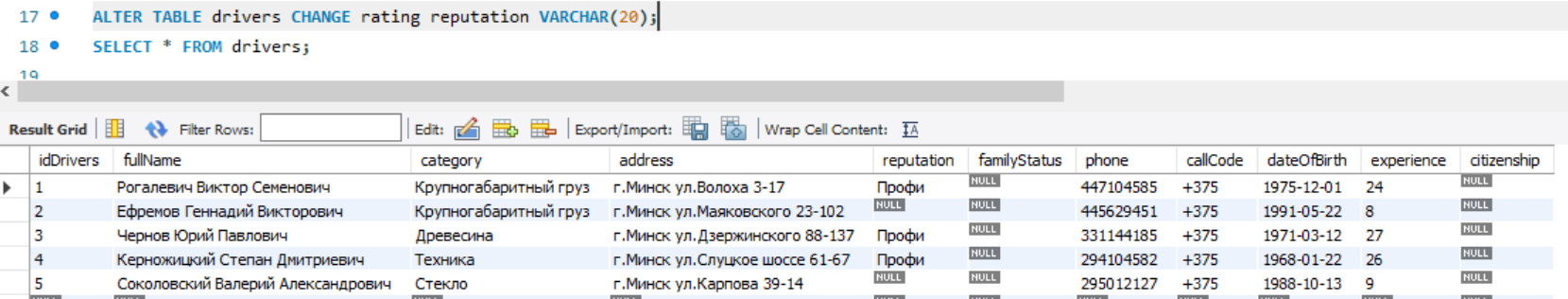


Рисунок 5 – Пример работы ALTER TABLE CHANGE

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;

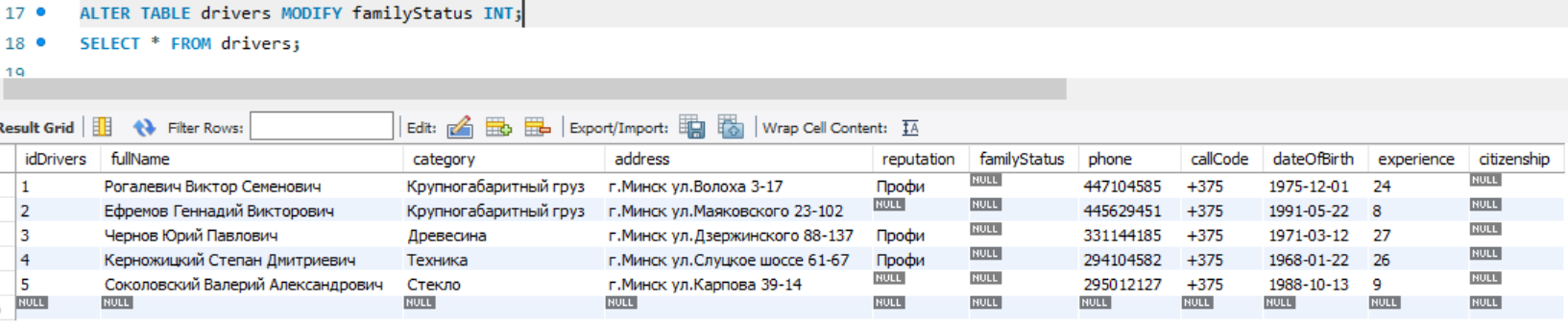


Рисунок 6 – Изменение типа столбца

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):

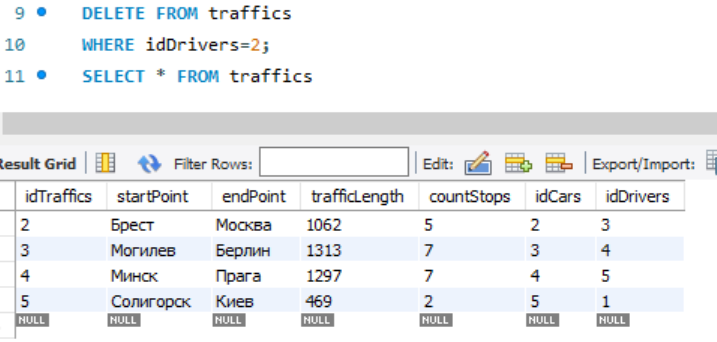


Рисунок 7 – Удаление из таблиц записи

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 отличии 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:



Рисунок 8 – Создание представления

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

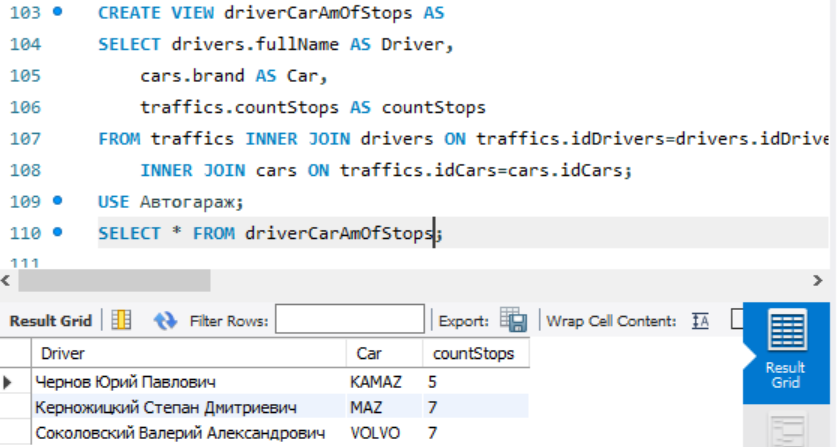


Рисунок 9 – Создание представления и его получение

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:

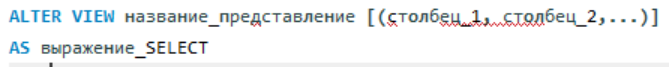


Рисунок 10 – Изменение представления

Например, изменим выше созданное представление driverCarAmOfStops.

Результат показан на рисунке 11.

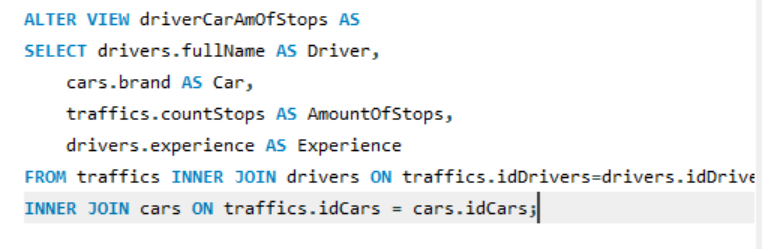


Рисунок 11 – Изменения представления

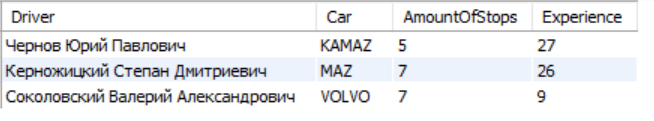


Рисунок 12 – Получение измененного представления

Для удаления представления вызывается команда DROP VIEW:

Результат удаления показан на рисунке 11.



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.