Пенза 2018

**ОТЧЕТ**

по лабораторной работе №1

«Создание и связывание таблиц базы данных в среде MySQL»

Министерство образования Российской Федерации

Пензенский государственный университет

Кафедра «Вычислительная техника»

### Цель работы

Выполнили:

студенты группы 15ВВ1

Буряков А.В.

Рахматуллина Р.Х.

Николайчев В.А.

Принял:

Карамышева Н.С.

Торкоз Д.А.

* создать базу данных;
* создать таблицы (не менее трех), определить поля таблиц, индексы;
* определить связи между таблицами и ограничения целостности;
* заполнить согласованными данными таблицы БД;
* при необходимости исправить введенную информацию;
* составить отчет по лабораторной работе.

### Листинг

### create database AutoParts;

### use AutoParts;

### create table providers (

### title nvarchar(50) primary key not null,

### provider\_address nvarchar(50) not null,

### phone nvarchar(12) null check(phone like('[+0123456789][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9][0-9]%'))

### );

### create table parts (

### title nvarchar(50) not null,

### article integer primary key auto\_increment not null,

### manufacturer nvarchar(50) null,

### price integer not null check (price > 0)

### );

### -- parts M:M providers

### -- Некоторые из поставщиков могут поставлять одинаковые детали (один артикул)

### -- У одного поставщика может быть множество деталей

### create table bridge\_providers\_parts (

### title nvarchar(50) not null references providers (title) on delete cascade,

### article integer not null references parts (article) on delete cascade

### );

### create table deals (

### deal\_date date not null,

### article integer not null references parts (article),

### deal\_parts\_count integer not null check (deal\_parts\_count > 0)

### );

### create table priceHistory (

### article integer not null references parts (article),

### price\_date date not null,

### price integer not null

### );

### delimiter $$

### create definer = CURRENT\_USER trigger writePriceHistoryEvent before update on parts

### for each row begin

### if new.price <> old.price then

### insert into priceHistory (article, price\_date, price)

### values (old.article, curdate(), new.price);

### end if;

### end$$

### insert into parts (title, manufacturer, price)values

### ('Рулевая рейка', 'Nissan Almera N16', 12000),

### ('Пепельница', 'ВАЗ 2114', 700),

### ('Салонный фильтр', 'Toyota Camry 40', 1300),

### ('Свечи зажигания', 'NGK', 100),

### ('Ремень привода водяной помпы', 'Opel Astra', 950),

### ('Блок ABS', 'Honda HR-V', 8000),

### ('Аккумулятор', 'ExtraStar', 3000),

### ('Рулевая тяга', 'Volkswagen Jetta', 1420),

### ('Тормозные колодки', 'Kia Rio', 2467),

### ('Фильтр топливный', 'ВАЗ Kalina', 1140),

### ('Радиатор двигателя', 'ВАЗ Granta', 7200),

### ('Стартер', 'Mitsubishi Pajero', 3600),

### ('Диск сцепления', 'Mitsubishi Lancer X', 1600),

### ('ШРУС', 'Skoda Oktavia', 4120),

### ('Катушка зажигания', 'Nissan Primera G11', 1800),

### ('Тросс стояночного тормоза', 'Volkswagen Passat', 1400),

### ('Подшипник ступицы', 'ВАЗ Kalina', 1873),

### ('Интеркулер', 'Subaru Forester', 9980),

### ('Рычаг подвески', 'Honda HR-V', 3710),

### ('Сайлентблок', 'LADA Vesta', 1300),

### ('Фильтр салона', 'BMW e53', 1890),

### ('Ролик ремня ГРМ', 'Volkswagen Passat', 2800);

### insert into providers values

### ('Автоток','ул. Литвинова 20а', '+78412296880'),

### ('Интер-авто' ,'ул. Кулакова 14б' ,'+78412540304'),

### ('Vag.Direction', 'ул. Красная 29' ,'+78412625323'),

### ('Автостоп', 'ул. Измайлова 32' , '+78412625323'),

### ('Магазин-разборка 58', 'ул. Кураева 49' ,'+78412684086'),

### ('Запчастье' ,' ул. Ставского 4', '+78412224610');

### insert into bridge\_providers\_parts values

### ('Запчастье', 6),

### ('Запчастье', 5),

### ('Запчастье', 12),

### ('Запчастье', 13),

### ('Запчастье', 14),

### ('Запчастье', 17),

### ('Запчастье', 18),

### ('Запчастье', 20),

### ('Запчастье', 21),

### ('Запчастье', 22),

### ('Автоток', 4),

### ('Автоток', 15),

### ('Автоток', 7),

### ('Vag.Direction', 8),

### ('Vag.Direction', 16),

### ('Vag.Direction', 22),

### ('Автостоп', 1),

### ('Автостоп', 3),

### ('Автостоп', 9),

### ('Автостоп', 10),

### ('Автостоп', 15),

### ('Автостоп', 20),

### ('Магазин-разборка 58', 2),

### ('Магазин-разборка 58', 11),

### ('Магазин-разборка 58', 12),

### ('Магазин-разборка 58', 14),

### ('Магазин-разборка 58', 16),

### ('Магазин-разборка 58', 18),

### ('Магазин-разборка 58', 22);

### insert into deals values

### ('2018.08.15', 2, 1),

### ('2018.08.19', 2, 4),

### ('2018.08.19', 1, 2),

### ('2018.08.21', 1, 3),

### ('2018.09.06', 1, 1),

### ('2018.08.26', 1, 1),

### ('2018.08.26', 3, 2),

### ('2018.08.30', 3, 5),

### ('2018.09.02', 4, 2),

### ('2018.09.03', 5, 7),

### ('2018.09.07', 6, 4),

### ('2018.09.08', 6, 1),

### ('2018.09.08', 7, 3),

### ('2018.09.09', 7, 2),

### ('2018.09.11', 8, 6),

### ('2018.08.23', 8, 1),

### ('2018.08.27', 8, 2),

### ('2018.08.15', 9, 1),

### ('2018.08.19', 9, 4),

### ('2018.08.19', 10, 2),

### ('2018.08.21', 10, 3),

### ('2018.09.06', 10, 1),

### ('2018.09.26', 10, 1),

### ('2018.09.26', 11, 2),

### ('2018.08.30', 12, 5),

### ('2018.09.02', 13, 2),

### ('2018.08.09', 13, 2),

### ('2018.08.03', 14, 7),

### ('2018.08.28', 15, 4),

### ('2018.09.18', 15, 1),

### ('2018.08.17', 16, 3),

### ('2018.09.25', 16, 2),

### ('2018.08.13', 17, 6),

### ('2018.09.24', 17, 1),

### ('2018.09.26', 17, 2),

### ('2018.09.17', 18, 6),

### ('2018.08.14', 18, 1),

### ('2018.08.20', 18, 2),

### ('2018.09.12', 19, 1),

### ('2018.09.11', 19, 4),

### ('2018.09.13', 20, 2),

### ('2018.08.29', 20, 3),

### ('2018.09.25', 20, 1),

### ('2018.08.26', 20, 1),

### ('2018.09.26', 21, 2),

### ('2018.08.17', 22, 5),

### ('2018.09.17', 22, 2);

### select \* from autoparts.deals;

### select \* from autoparts.parts;

### select \* from autoparts.providers;

### select \* from autoparts.bridge\_providers\_parts;

### select \* from autoparts.pricehistory;

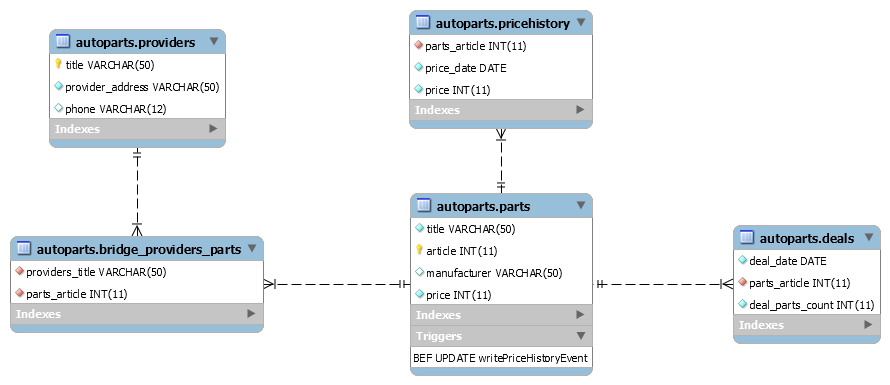
### update `autoparts`.`parts` set `price` = '1350' where (`article` = '3');

### update `autoparts`.`parts` set `price` = '1600' where (`article` = '16');

### Задание

*Фирма по продаже запчастей*. Фирма продает запасные части для автомобилей. Фирма имеет определенный набор поставщиков, по которым известны название, адрес и телефон. У поставщиков приобретаются детали. Каждая деталь характеризуется названием, артикулом и ценой. Некоторые из поставщиков могут поставлять одинаковые детали (один артикул). Каждый факт покупки запчастей у поставщика фиксируется в базе данных, причем обязательными для запоминания являются дата покупки и количество приобретенных деталей. Цена детали может меняться от поставки к поставке. Поставщики заранее ставят фирму в известность о дате изменения цены и ее новом значении. Нужно хранить не только текущее значение цены, но и всю историю изменения цен.

### Результаты работы программы



В результате работы скрипта создана база данных ‘AutoParts’, состоящая из 5 таблиц: запчасти, поставщики, сделки, история покупки и связующая таблица bridge\_providers\_parts для создания связи многие-ко-многим.

Таблица истории цен связана благодаря триггеру на таблице деталей, который при обновлении цены любой детали записывает в историю цен ее номер, дату и старую цену. Одна деталь может фигурировать во множестве сделок, различные поставщики могут поставлять одинаковые детали.

**Вывод**

Мы построили базу данных MySQL и создали связи согласно заданию, заполнили таблицы значениями. База данных готова к работе с запросами.