



Министерство образования Республики Беларусь

Учреждение образования
«Гомельский государственный технический
университет имени П. О. Сухого»

Институт повышения квалификации
и переподготовки

Кафедра «Информатика»

Т. В. Тихоненко

СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ ДАНЫХ

ПРАКТИКУМ

по одноименному курсу

для слушателей специальности

1-40 01 74 «Web-дизайн и компьютерная графика»

заочной формы обучения

Гомель 2016

УДК 004.65(075.8)
ББК 32.973-018.2я73
Т46

*Рекомендовано научно-методическим советом
факультета автоматизированных и информационных систем
ГГТУ им. П. О. Сухого
(протокол № 5 от 28.12.2015 г.)*

Рецензент: доц. каф. «Информационные технологии» ГГТУ им. П. О. Сухого
канд. техн. наук, доц. *В. В. Комраков*

Тихоненко, Т. В.

Т46 Системы управления базами данных : практикум по одноим. курсу для слушателей специальности 1-40 01 74 «Web-дизайн и компьютерная графика» заоч. формы обучения / Т. В. Тихоненко. – Гомель : ГГТУ им. П. О. Сухого, 2016. – 37 с. – Систем. требования: PC не ниже Intel Celeron 300 МГц; 32 Mb RAM; свободное место на HDD 16 Mb; Windows 98 и выше; Adobe Acrobat Reader. – Режим доступа: <https://elib.gstu.by>. – Загл. с титул. экрана.

Данное издание представляет собой пособие по практическому освоению понятий баз данных и систем управления базами данных. Правильное и последовательное выполнение всех заданий данного пособия позволит по окончании курса овладеть основными принципами и правилами создания баз данных.

Издание адресовано слушателям, начинающим изучать web-технологии.

**УДК 004.65(075.8)
ББК 32.973-018.2я73**

© Учреждение образования «Гомельский
государственный технический университет
имени П. О. Сухого», 2016

Содержание

Лабораторная работа № 1	4
Лабораторная работа № 2	15
Лабораторная работа № 3	25
Лабораторная работа № 4	30
Список литературы	37

Лабораторная работа № 1

Тема работы: проектирование базы данных. Работа в среде **mySQL Workbench**.

Цель работы: научиться проектировать базы данных. Получить навыки работы в среде **mySQL Workbench**.

Порядок выполнения работы

1. Убедиться, что на персональном компьютере установлена среда **MySQL Workbench** и СУБД **MySQL**. При необходимости выполнить инсталляцию программного обеспечения.
2. Изучить краткие теоретические сведения и методические указания.
3. Выполнить практическое задание.

Краткие теоретические сведения и методические указания

Среда **mySQL Workbench** предназначена для визуального проектирования баз данных и управления сервером **mySQL**.

Для построения моделей предназначена секция **Data Modeling** (рисунок 1).

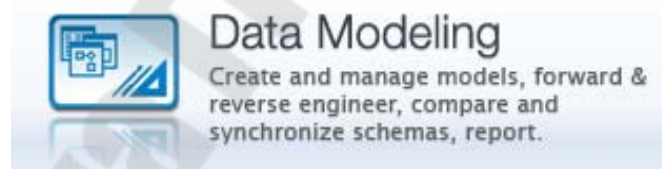


Рисунок 1 – Создание модели

Выберем пункт **Create new EER Model**.

EER model расшифровывается как **Extended Entity-Relationship Model** и переводится как **Расширенная модель сущностей-связей**.

По умолчанию имя созданной модели **myDB**. Щелкните правой кнопкой мыши по имени модели и выберите в появившемся меню пункт **Edit schema**. В появившемся окне **можно** изменить имя модели. Назовем ее, например, **kontora**. В именах таблиц и столбцов нельзя использовать русские буквы.

В этом окне также **нужно** настроить так называемую «кодировку» для корректного отображения русских букв **внутри** таб-

лиц. Для этого выберите из списка пункт «**cp1251-cp1251_general_ci**». Окно свойств можно закрыть.

Диаграмму будем строить с помощью визуальных средств. Щелкнем по пункту **Add diagram**, загрузится пустое окно диаграммы (рисунок 2).

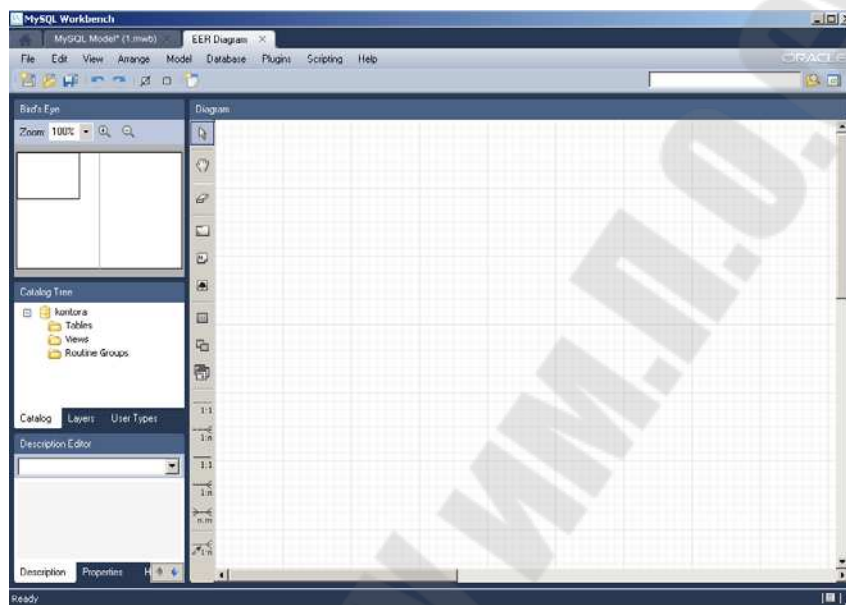



Рисунок 2 – Начало работы

Создать новую таблицу можно с помощью пиктограммы . Нужно щелкнуть по этой пиктограмме, а потом щелкнуть в рабочей области диаграммы. На этом месте появится таблица с названием по умолчанию **table1**. Двойной щелчок по этой таблице открывает окно редактирования, в котором можно изменить имя таблицы и настроить её структуру.

Будем создавать таблицу **Отделы** со следующими столбцами: номер_отдела, полное_название_отдела, короткое_название_отдела. Переименуем **table1** в **k_dept** и начнем создавать столбцы.

Каждый столбец имеет:

- имя (не используйте русские буквы в имени!),
- тип данных. Самые распространенные типы данных:
 - INT – целое число;
 - VARCHAR(размер) – символьные данные переменной длины, в скобках указывается максимальный размер;
 - DECIMAL(размер, десятичные_знаки) – десятичное число;
 - DATE – дата;

- DATETIME – дата и время.

Далее располагаются столбцы, в которых можно настроить дополнительные свойства поля, включив соответствующий флажок:

- PK (primary key) – первичный ключ;
- NN (not null) – ячейка не допускает пустые значения;
- UQ (unique) – значение должно быть уникальным в пределах столбца;
- AI (auto incremental) – это свойство полезно для простого первичного ключа, оно означает, что первичный ключ будет автоматически заполняться натуральными числами: 1, 2, 3, и т.п.;
- DEFAULT – значение по умолчанию, т.е., значение, которое при добавлении новой строки в таблицу автоматически вставляется в ячейку сервером, если пользователь оставил ячейку пустой.

Таблица **Отделы** имеет следующий вид (рисунок 3):

K_dept										
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
dept_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
dept_full_name	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
dept_short_name	VARCHAR(10)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Рисунок 3 – Таблица «Отделы»

Затем создадим таблицу **Сотрудники** со следующими столбцами: номер_сотрудника, имя_сотрудника, должность, дата_начала_контракта, дата_окончания_контракта (рисунок 4).

k_staff										
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default	
staff_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		
staf_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
staff_post	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
staff_hiredate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		
staff_termdate	DATE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

Рисунок 4 – Таблица «Сотрудники»

Созданные таблицы выглядят следующим образом (рисунок 5):

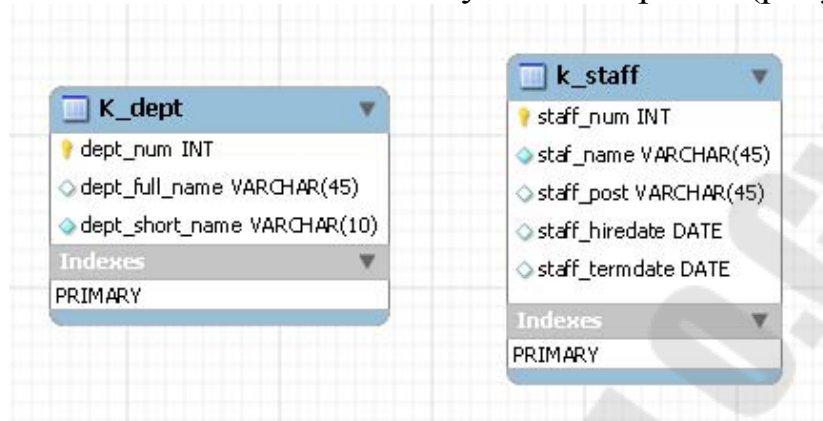



Рисунок 5 – Вид созданных таблиц

Обратите внимание, что при создании первичного ключа автоматически создается **индекс** по этому первичному ключу. **Индекс** представляет собой вспомогательную структуру, которая служит, прежде всего, для **ускорения поиска** и **быстрого доступа** к данным.

Теперь свяжем эти таблицы. Сначала создадим связь «Работает» между **Сотрудником** (дочерняя таблица) и **Отделом** (родительская таблица), степень связи М:1. Для создания связей М:1 служит пиктограмма на панели инструментов  (с пунктирной линией). С ее помощью создается так называемая «неидентифицирующая связь», т.е. обыкновенный внешний ключ, при этом первичный ключ родительской таблицы добавляется в список столбцов дочерней таблицы.

Итак, щелкнем на пиктограмме, затем щелкнем на дочерней таблице **Сотрудники**, затем на родительской таблице **Отделы** (рисунок 6).

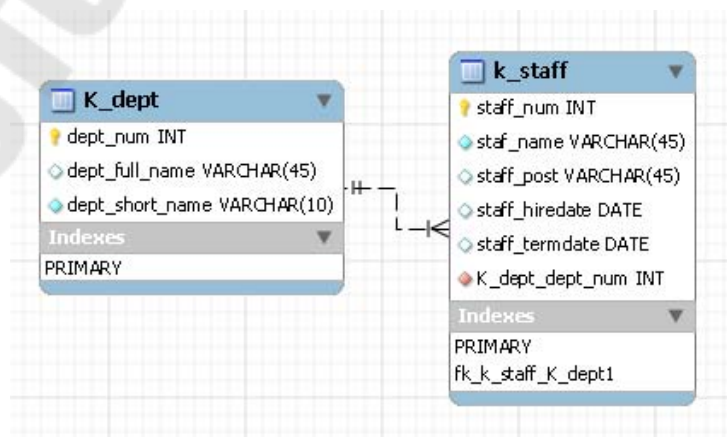



Рисунок 6 – Создание связи 1:n

Обратите внимание, что при этом произошло. Между таблицами образовалась пунктирная линия; в сторону «к одному» она отмечена двумя черточками, в сторону «ко многим» - «куриной лапкой». Кроме того, в таблице **Сотрудники** образовался дополнительный столбец, которому автоматически присвоено имя *k_dept_dept_num* (т.е., имя родительской таблицы плюс имя первичного ключа родительской таблицы). А в группе **Индексы** создан индекс по внешнему ключу.

Теперь добавим связь между этими же таблицами «Руководит» 1:1 (рисунок 7). Выберем пиктограмму , затем щелкнем по **Отделам**, затем по **Сотрудникам**.

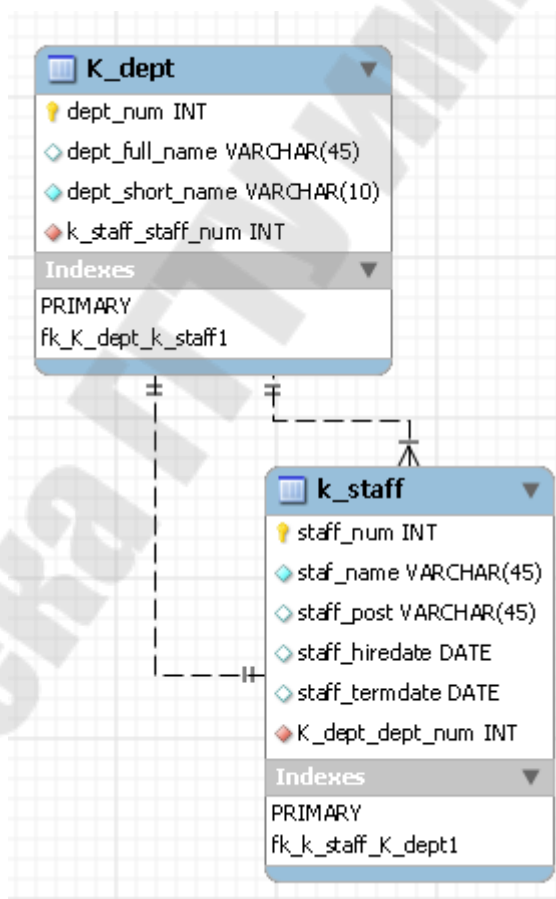


Рисунок 7 – Создание связи 1:1

Чтобы 2 связи на картинке не «завязывались узлом», мы их разместили друг под другом.

Обратите внимание, что в таблицу **Отделы** был автоматически добавлен столбец *k_staff_staff_num*, а также индекс по внешнему ключу.

Создадим таблицу **Предприятия** (рисунок 8):

k_firm									
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
firm_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
firm_name	VARCHAR(45)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
firm_addr	VARCHAR(45)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
firm_phone	VARCHAR(20)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 8 – Таблица «Предприятия»

Создадим таблицу **Договоры**. У столбца **Тип_договора** зададим следующий формат: это буква из списка 'A', 'B', 'C' (рисунок 9).

k_contract									
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
contract_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
contract_date	DATE	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
contract_type	ENUM('A', 'B', 'C')	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 9 – Таблица «Договоры»

Свяжем **Договоры** с **Сотрудниками** и **Предприятиями** связями М:1 (рисунок 10).

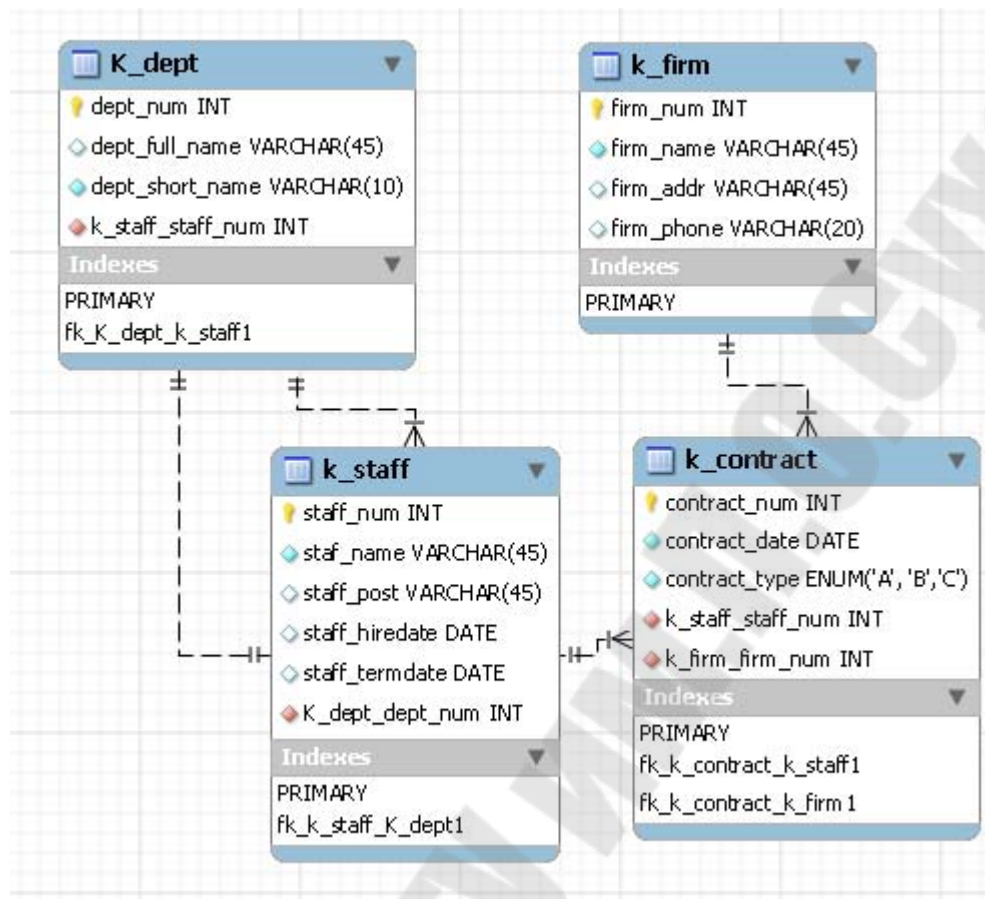


Рисунок 10 – Создание связи М:1

Затем создадим таблицы **Счета** и **Платежи** (рисунок 11):

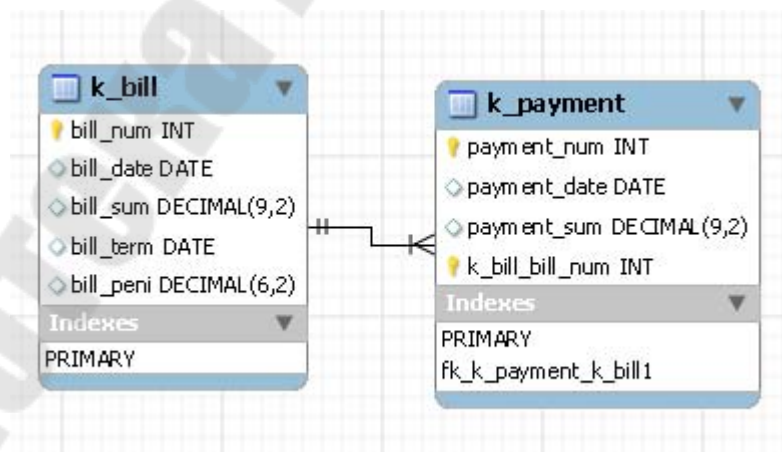



Рисунок 11 – Таблицы «Счета» и «Платежи»

Поскольку сущность **Платеж** была «слабой», у нее нет полноценного первичного ключа, и каждый платеж однозначно идентифи-

цируется группой атрибутов (номер_счета, номер платежа). Отметим в качестве ключевого поля *payment_num*, а затем создадим **идентифицирующую** связь между **Счетом** и **Платежом**. Идентифицирующая связь создается с помощью пиктограммы  (со сплошной линией). При этом новый столбец *k_bill_bill_num* становится не только внешним ключом в таблице **Платеж**, но и частью первичного ключа.

Далее создадим таблицу **Прайс-лист** со столбцами (номер_товара, название_товара, цена_товара и тип_товара), рисунок 12.

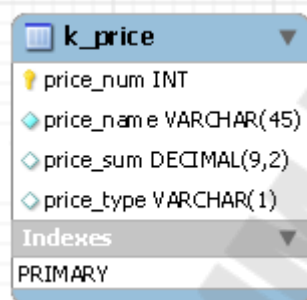



Рисунок 12- Таблица «Прайс-лист»

Между объектами **Счет** и **Прайс-лист** имеется связь «многие - ко многим». Для создания этой связи нужно использовать пиктограмму . Следует щелкнуть мышью по этой пиктограмме, а затем последовательно щелкнуть по связываемым таблицам. Между ними появится новая таблица, обратите внимание на ее столбцы, первичный ключ и внешние ключи (рисунок 13):

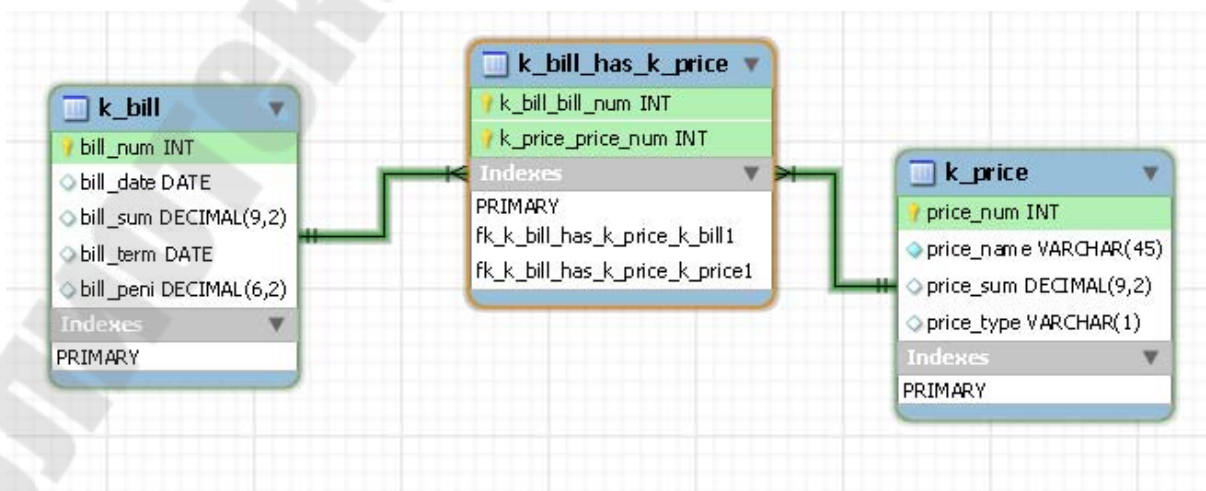


Рисунок 13 – Создание связи N:M

Для удобства переименуем эту таблицу в *k_protokol* (ПротоколСчета), добавим столбцы *kolvo* и *price_sum* (рисунок 14).

Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
k_bill_bill_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
k_price_price_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
kolvo	INT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
price_sum	DECIMAL(9,2)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

Рисунок 14 – Таблица «ПротоколСчета»

Теперь EER-диаграмма имеет такой вид (рисунок 15):

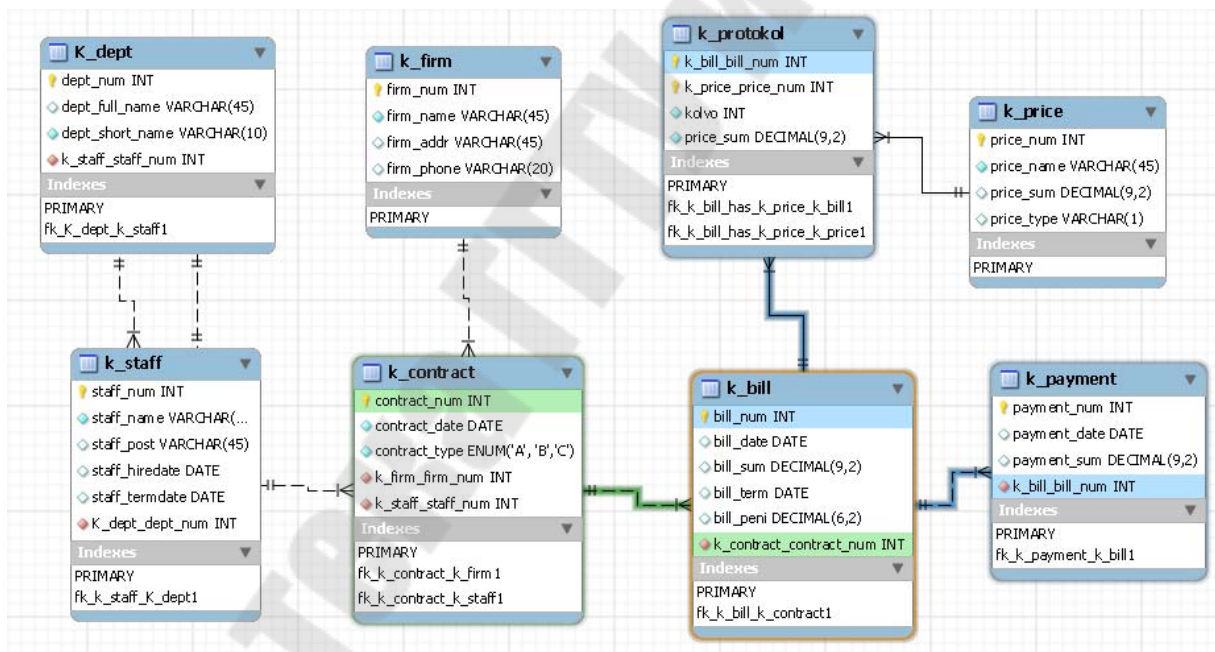


Рисунок 15 – EER-диаграмма

Практическое задание

Создайте в *mySQL Workbench* EER-диаграмму для своей задачи.

Варианты заданий

Вариант 1. Спроектировать структуру базы данных о студентах, для их распределения по местам практики: фамилия, год рождения, пол, группа, факультет, средний балл, место работы, город.

Вариант 2. Спроектировать структуру базы данных об автомобилях: номер, год выпуска, марка, цвет, состояние, фамилия владельца, адрес.

Вариант 3. Спроектировать структуру базы данных о квартирах, предназначенных для продажи: район, этаж, площадь, количество комнат, сведения о владельце, цена.

Вариант 4. Спроектировать структуру базы данных о книгах, купленных библиотекой: название, автор, год издания, адрес автора, адрес издательства, цена, книготорговая фирма.

Вариант 5. Спроектировать структуру базы данных о сотрудниках, имеющих компьютер: фамилия, номер комнаты, название отдела, данные о компьютерах.

Вариант 6. Спроектировать структуру базы данных о заказах, полученных сотрудниками фирмы: фамилия, сумма заказа, наименование товара, название фирмы - клиента, фамилия заказчика.

Вариант 7. Спроектировать структуру базы данных об оценках, полученных студентами на экзаменах: фамилия, группа, предмет, номер билета, оценка, преподаватель.

Вариант 8. Спроектировать структуру базы данных о преподавателях кафедры: фамилия, должность, степень, номер комнаты, читаемые курсы.

Вариант 9. Спроектировать структуру базы данных об авторах web-сайта и их статьях: имя, адрес, учетная запись, пароль, тема, заголовков, текст статьи, иллюстрации.

Вариант 10. Спроектировать структуру базы данных о списке рассылки и подписчиках: тема и содержание письма, дата отправки, имена и адреса подписчиков, их учетные записи и пароли.

Вариант 11. Спроектировать структуру базы данных о сотрудниках научно-исследовательского института: Табельный номер, ФИО, номер отдела, должность, звание, степень, стаж, процент надбавки.

Вариант 12. Спроектировать структуру базы данных о врачах больниц города: код больницы, код отделения, фамилия врача, домашний адрес, год рождения, специальность.

Вариант 13. Спроектировать структуру базы данных о собаках в клубе собаководов: код собаки, порода, кличка, дата рождения, пол, ФИО владельца, домашний адрес.

Вариант 14. Спроектировать структуру базы данных об автобусах автобусного парка: номер автобуса, ФИО водителя, номер маршрута, дата тех. осмотра, год выпуска, пробег.

Вариант 15. Спроектировать структуру базы данных о серверах дата-центра: код сервера, физическое имя, число процессоров, объем памяти, объем дискового пространства, дата ввода в эксплуатацию.

Вопросы для самоконтроля

1. Дайте определение базы данных, назовите основные отличительные признаки.
2. Дайте определения основным понятиям реляционной модели данных (поле, запись, таблица, ключ и др.)
3. Как происходит процесс проектирования базы данных.
4. Назовите типы связей между информационными объектами.
5. Что такое нормализация отношений в базе данных.

Лабораторная работа № 2

Тема работы: администрирование сервера баз данных MySQL. Разработка простейшего клиентского приложения для работы с базой данных.

Цель работы: ознакомиться с процессом инсталляции СУБД MySQL и оболочки XAMPP. Научится устанавливать соединение с сервером из программных систем, создавать базу данных и произвести элементарные действия на удаленном и локальном компьютере.

Порядок выполнения работы

1. Убедиться, что на персональном компьютере установлена СУБД MySQL и инструментальный набор WEB-разработчика XAMPP. При необходимости выполнить инсталляцию СУБД и программного обеспечения.
2. Изучить краткие теоретические сведения и методические указания.
3. Создать новую базу данных с помощью утилиты phpMyAdmin.
4. Создать таблицу в соответствии с вариантом задания (см. варианты в лабораторной работе № 1) и заполнить ее данными, используя интерфейсные средства оболочки (в таблице должно быть не менее 7 записей).
5. Разработать приложение на языке скриптов PHP для установки соединения и просмотра содержимого таблицы.

Краткие теоретические сведения и методические указания

Создание новой базы данных

Запустить `xampp_control`. Из окна стартовать серверы Apache и MySQL. После этого запустить веб-браузер (Opera) и набрать в адресной строке `http://localhost`. После открытия меню оболочки XAMPP выбрать в меню слева внизу PHPMyAdmin. По умолчанию логин `root` пароль пустой.

При выполнении работы всегда следите за тем, чтобы все ваши объекты (базы данных, таблицы, текстовые поля) получали кодировку (так называемое «сравнение») равную `utf8_general_ci`, а не заданную по умолчанию (например, `latin` или `swedish`), иначе у вас возникнут

проблемы с русскими значениями строк полей при работе с базой из PHP-скрипта (рисунок 2.1).



Рисунок 2.1 – Создание новой базы данных

Обратите внимание на SQL-скрипт, расположенный сверху страницы после создания базы данных (рисунок 2.2).

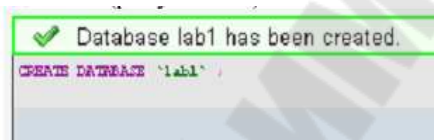


Рисунок 2.2 – SQL-скрипт для создания базы данных

Создание отношения (таблицы).

Так как база данных является моделью реального мира перед созданием таблицы необходимо описать предметную область, которую она представляет. Рассмотрим простейшую модель: телефонный справочник. Данные этой модели определяются следующими атрибутами: уникальный номер абонента (он должен создаваться автоматически), фамилия, имя, отчество, дата рождения (рисунок 2.3).

PHONELIB	
ID	INTEGER
FIO	VARCHAR(50)
BDATE	DATE

Рисунок 2.3 – Атрибуты отношения «Телефонный справочник»

Для создания таблицы в среде phpMyAdmin воспользуемся интерфейсом CreateNewTable (рисунок 2.4).

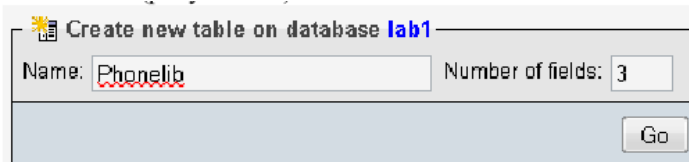


Рисунок 2.4 – Интерфейс создания таблицы

Описываем в интерфейсе имена атрибутов и их типы (рисунок 2.5).

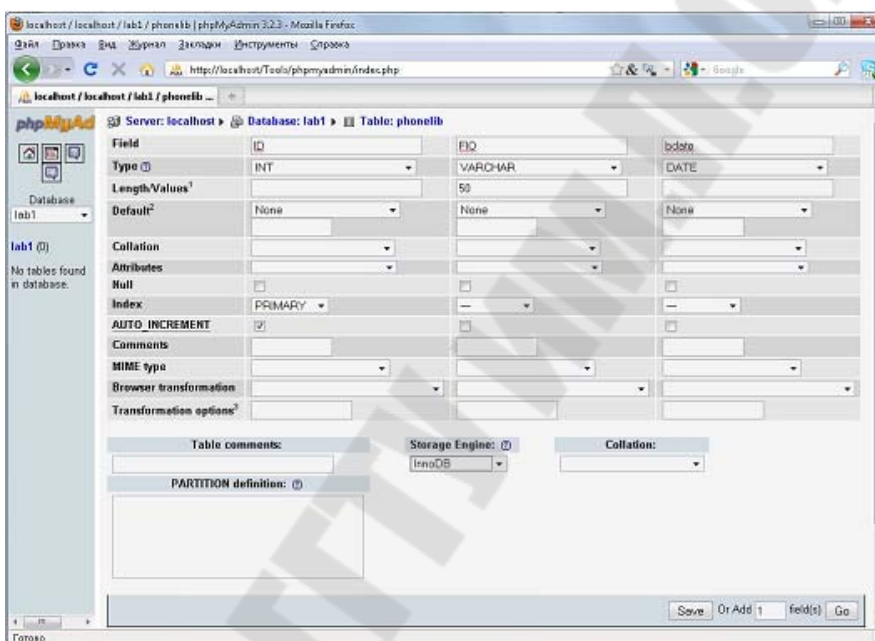


Рисунок 1.5 – Описание атрибутов

Обратите внимание, что поле для поля ID установлен признак AUTO_INCREMENT, для того чтобы при добавлении новой записи (картежа) значение поля генерировалось автоматически. На будущее, установим для этого атрибута признак PRIMARY в поле index. Атрибут FIO (фамилия, имя, отчество) является текстовым, поэтому для него указываем длину 50 (в символах).

До для создания таблицы после описания атрибутов нажимаем кнопку «Save».

Среда phpMyAdmin автоматически генерирует SQL-скрипт для создания новой таблицы:

```
CREATE TABLE `lab1`, `phonelib` (
  `ID` INT NOT NULL AUTO_INCREMENT PRIMARY KEY,
```

```

`FIO` VARCHAR(50) NOT NULL,
`bdata` DATE NOT NULL
) ENGINE = InnoDB;

```

Синтаксис операции CREATE TABLE в MySQL, в целом, соответствует стандарту SQL-92, однако имеет ряд особенностей. В частности, фрагмент фразы ENGINE = InnoDB определяет тип «движка» для работы с данными, каждый из которых имеет свои характерные особенности.

MyISAM. Движок по умолчанию. Не поддерживает транзакций, средняя надежность хранения данных. Превосходная производительность чтения данных (через SELECT). Блокирует всю таблицу при записи в нее данных, отчего маленькая производительность при частых записях.

InnoDB. Транзакционный тип движка, применяемый при интенсивных операциях записи, благодаря возможности блокировки уровня строк таблицы. Великолепная восстанавливаемость и высокая надежность хранения данных. Поддерживает внешние ключи. Именно этот движок рекомендуется использовать при выполнении лабораторных работ.

Ввод данных в таблицу.

Для ввода данных в таблицу выбираем операцию INSERT (рисунок 2.6).

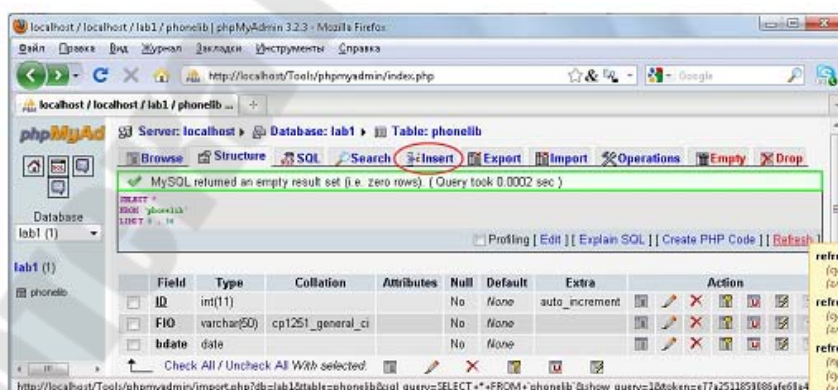


Рисунок 2.6 – Переход к вводу данных

Заполняем колонку «VALUE» в интерфейсе ввода данных и нажимаем на кнопку «Go» (рисунок 2.7). Обратите внимание, что поле ID оставляем пустым.

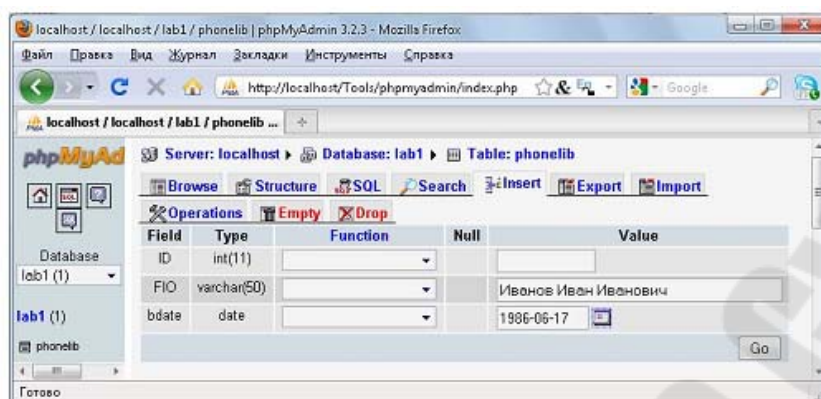


Рисунок 2.7 – Ввод данных в таблицу

Создание нового пользователя.

При покупке хостинга у большинства хостинг-провайдеров для вас автоматически создается база данных и аккаунт пользователя с логином и паролем, имеющий все необходимые привилегии. На нормальном хостинге есть точно такой же phpMyAdmin посредством которого вы сможете управлять таблицами и их содержимым, а также импортировать данные из локальной базы данных в базу данных хостинга. Логин и пароль для доступа к базе данных, а также ссылка на phpMyAdmin высылается хостинг-провайдером на ваш email. На локальном же компьютере нам придется самим создавать пользователя.

Для этого выбираем базу данных lab1 и вкладку Privileges (рисунок 2.9).



Рисунок 2.9 – Редактирование привилегий

Переходим по ссылке Addnewusers в интерфейсе администратора создаем нового пользователя и пароль (рисунок 1.10). В нашем случае имя пользователя будет совпадать с паролем (student).

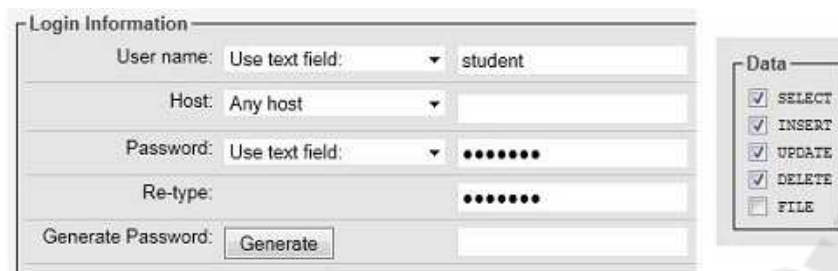


Рисунок 2.10 – Создание нового пользователя, пароля и привилегий

В расположенном выше интерфейсе определяем права пользователя при работе с базой данных. Как минимум, для обеспечения функциональности необходимо разрешить добавлять записи в таблицу и выбрать данные из нее (операции Insert и Select). В нашем случае разрешим также обновлять и удалять записи (Update и Delete).

Разработка клиентского приложения.

Клиентское приложение – это аппаратный или программный комплекс вычислительной системы, посылающий запросы серверу.

Программа, являющаяся клиентом, взаимодействует с сервером, используя определенный протокол. Она может запрашивать с сервера какие-либо данные, манипулировать данными непосредственно на сервере, запускать на сервере новые процессы и т.д. Полученные от сервера данные клиентская программа может предоставлять пользователю или использовать как-либо иначе, в зависимости от назначения программы. Программа-клиент и программа-сервер могут работать как на одном и том же компьютере, так и на разных. Во втором случае для обмена информацией между ними используется сетевое соединение.

Веб-сервер – клиент-серверное приложение, у котором клиент выступает браузер, а сервер – веб-сервер и сервер баз данных. Логика веб-приложение распределена между сервером и клиентом, хранение данных осуществляется, преимущественно, на сервере, обмен информацией происходит по сети. Одним из преимуществ такого подхода является тот факт, что клиенты не зависят от конкретной операционной системы пользователя, поэтому веб-приложения являются межплатформенными сервисами. Для создания веб-приложений на стороне сервера используются разнообразные технологии и любые языки

программирования, способные осуществлять вывод в стандартную консоль.

Приложение-клиент будет разрабатываться средствами языков HTML, Java-скрипт, PHP.

HTML-скрипт титульной страницы проекта:

```
<html>
  <head>
    <meta charset=windows-1251>
    <meta name=``robots`` content=``all``>
    <title>Лабораторная работа №2. Соединение с ба-
зой данных</title>
  </head>
  <body>
    <a href=``phlib.php``>Перечень абонентов</a>
  </body>
</html>
```

Если в адресной строке браузера ввести адрес «<http://localhost/lab1/>», увидим ссылку «перечень абонентов».

Задачей приложения-клиента в данной лабораторной работе является вывод состояния отношения в виде таблицы. Для этого необходимо выполнить следующие действия:

- 1) установить соединение с базой данных;
- 2) выполнить выборку данных;
- 3) отобразить данные в виде таблицы.

Для общения с MySQL из PHP понадобятся следующие функции:

1) создать соединение с MySQL: `intmysql_connect(string hostname, string username, string password)`. Параметры: `Hostname` – имя хоста, на котором находится база данных, `Username` – имя пользователя, `Password` – пароль пользователя. Функция возвращает параметр типа `int`, который больше 0, если соединение прошло успешно, и равен 0 в противном случае.

2) выбрать базу данных для работы: `intmysql_select_db(string database_name, intlink_identifier)`. Параметры: `Database_name` – имя базы данных; `link_identifier` – ID соединения, которое получено в функции `mysql_connect` (параметр обязательный, если он не указывается, то используется ID от последнего вызова `mysql_connect`). Функция возвращает значение `true` или `false`.

3) запрос к базе данных: `intmysql_query` (string query, int-link_identifier). Параметры: Query - строка, соединяющая запрос, link_identifier – см. предыдущую функцию. Функция возвращает ID результата или 0, если произошла ошибка.

4) В цикле выбираем записи, полученные в результате запроса: `arraymysql_fetch_array`(resource result [, int result_type]). Параметры: result – указатель на ресурс результатов выборки, второй (необязательный) – флаг, отвечающий за содержимое возвращаемого функцией ассоциативного массива. Функция возвращает массив, соответствующий извлеченному ряду, или FALSE, если рядов больше нет. Функция `mysql_fetch_array()` – это расширенная версия `mysql_fetch_row()`. Помимо сохранения данных в числовых индексах результирующего массива, она также хранит данные в ассоциативных индексах, используя в качестве ключей имена полей.

5) закрыть соединение `intmysql_close`(int link_identifier). Параметры: link_identifier – см. выше. Функция возвращает значение true или false.

Скрипт будет иметь вид:

```
<html>
<head>
  <meta charset=windows-1251>
  <meta name=``robots`` content=``all``>
  <title>Лабораторная работа №2. Соединение с ба-
зой данных</title>
</head>
<body>
  <?
  /*Переменные для соединения с базой данных*/
  $hostname = ``localhost``;
  $username = ``student``;
  $password = ``student``;
  $dbName = ``lab1``;
  /*Таблица MySQL, в которой хранятся данные*/
  $usertable = ``phonelib``;
  /*Создать соединение*/
  mysql_connect ($hostname, $username, $password)
OR DIE (``Не могу создать соединение``);
  /*Выбрать базу данных. Если произойдет ошибка -
вывести ее*/
```

```

mysql_select_db($dbName) OR DIE(mysql_error(
));
/*Создать запрос для выборки информации*/
$query = ``select * from $usertable``;
/*Выполнить запрос*/
$result = mysql_query($query) OR
DIE(mysql_error( ));
echo ``<table border=1>``;
echo ``<tr><th> Идентификатор </th><th> Ф.И.О.
</th><th> Дата рождения </th></tr>``;
/*Выбрать очередную запись из таблицы*/
while($row=mysql_fetch_array($result))//берем
//результаты из каждой строки
{
echo;
/*Вывести ее в виде HTML*/
echo
`<tr><td>``.$row[`ID`].``</td><td>``.$row[`FIO`].``</
td><td>``.$row[`bdate`].``</td><td>``;
};
echo ``</table>``;
/*Закреть соединение*/
mysql_close();
?>
</body>
</html>

```

Результат выполнения скрипта приведен на рисунке 2.11.

Идентификатор	Ф.И.О.	Дата рождения
1	Иванов Иван Иванович	1986-06-17
2	Петров Петр Петрович	2007-04-12

Рисунок 2.11 – Результат выполнения скрипта выборки из базы данных

Важное замечание: для того, чтобы данные из таблицы, полученные с помощью `mysql_fetch_array()`, отображались на странице верно (они в кодировке Юникод, а браузер по умолчанию выводит в кодировке windows-1251 в виде вопросительных знаков), нужно произвести следующие действия:

1) указать браузеру, что страница набрана в кодировке UTF-8, для этого в `<head>` страницы нужно поместить тэг `<meta charset=utf-8">`

2) сменить кодировку страницы (в Notepad++ это делается командой Кодировки – Преобразовать в UTF-8)

3) указать драйверу базы данных, что данные в таблице в UTF-8 (после установления соединения с базой `mysql_select_db()` следует выполнить запрос следующего вида: “SET NAMES UTF-8”)

Вопросы для самоконтроля

- 1) Дайте определение СУБД.
- 2) Дайте определение сервера и клиента ресурса компьютерной сети.
- 3) Назовите основные архитектуры сетевых информационных систем, достоинства и недостатки.
- 4) Назовите существующие модели данных.
- 5) Назовите основные характеристики реляционной модели данных.
- 6) Какая информация указывается при создании базы данных?
- 7) Какие команды языка SQL используются для создания базы данных, таблиц, ввода данных?

Лабораторная работа № 3

Тема работы: создание таблиц в СУБД MySQL.

Цель работы: научиться создавать таблицы и устанавливать связи между ними.

Порядок выполнения работы

1) Запустить XAMPP, из браузера зайти на <http://localhost/>, оттуда запустить ссылку phpMyAdmin. Все действия по созданию таблиц и записей производить из веб-интерфейса.

2) Создать базу данных своей предметной области в соответствии со своим вариантом. База данных должна включать в себя таблицы (выделены полужирным шрифтом), которые состоят из полей, описанных в скобках. Все названия (БД, таблиц, полей) должны быть набраны латиницей с использованием при необходимости знака подчеркивания (например, **bolnica**, **postuplenie_pacientov**, **kod_pacienta**). Разные таблицы содержат одинаковые поля – именно по ним осуществляются связи. Ключевые поля таблиц (обычно они идут в таблицах первыми) должны иметь атрибут AUTO_INCREMENT.

3) После создания всех таблиц нужно наполнить все таблицы данными (не менее 3 записей в каждой таблице, причем заполнить таблицы данными таким образом, чтобы ключевые поля имели одинаковые значения в разных таблицах (например, если есть пациент с каким-то номером, то в таблицах поступления и выписки пациента должны быть записи с тем же номером пациента)).

4) Экспортировать базу данных в SQL-файл через phpMyAdmin для проверки ее преподавателем.

Варианты заданий

Вариант 1.

Предметная область: **Страховое агентство**

Информационные объекты:

Штаты (Табельный номер сотрудника, ФИО, Должность, Образование, Стаж работы, Оклад)

Виды страхования (Код страховки, Название страховки, Примечание)

Список застрахованных (Код застрахованного, ФИО застрахованного, Адрес, Код страховки, Страховая сумма, Табельный номер сотрудника, Дата страховки)

Страховые взносы (Код застрахованного, Период страховки, Страховой взнос, Дата взноса)

Страховые возмещения (Код застрахованного, Код страховки, Сумма выплаты, Дата выплаты)

Вариант 2.

Предметная область: **Фирма комиссионной продажи автомобилей**

Информационные объекты:

Марка авто (Код марки, Название, Производитель)

Цвет авто (Код цвета, Название)

Продажа авто (Номер авто, Код марки, Код цвета, Номер владельца, Дата выпуска, Дата приема на продажу, Дата продажи, Цена, Код покупателя, Процент фирме)

Владельцы (Код владельца, ФИО, Адрес)

Покупатели (Код покупателя, ФИО, Адрес)

Вариант 3.

Предметная область: **Банк**

Информационные объекты:

Штаты (Код должности, Должность, Оклад, Количество должностных единиц)

Персонал (Табельный номер сотрудника, ФИО, Код должности, Адрес, Телефон)

Вклады (Вид вклада, Номер счета, Дата создания, Вид валюты)

Клиенты (Номер счета, ФИО, Адрес)

Операции (Код операции, Наименование)

Движение денег (Номер движения, Номер счета, Код операции, Дата, Табельный номер сотрудника, Сумма операции)

Вариант 4.

Предметная область: **Отдел сбыта предприятия**

Информационные объекты:

Изделие (Код изделия, Наименование, Цена, Срок реализации)

Склад (Код поступления, Код изделия, Дата изготовления, Количество)

Сотрудники отдела (Табельный номер сотрудника, ФИО, Код должности, Код категории)

Должности (Код должности, Название, Оклад)

Категория (Код категории, Название)

Покупатели (Код покупателя, Название, Адрес)

Продажа (Код продажи, Дата продажи, Код покупателя, Код изделия, Количество, Отпускная цена, Код сотрудника)

Вариант 5.

Предметная область: **Научно-исследовательский институт (НИИ)**

Информационные объекты:

Отдел (Номер отдела, Название, Надбавка, Тема, Дата окончания работы)

Сотрудники (Табельный номер, ФИО, Номер отдела, Код должности, Код звания, Код степени, Стаж, Процент надбавки)

Должности (Код должности, Название, Оклад, Доплата)

Звание (Код звания, Звание)

Степень (Код степени, степень)

Вариант 6.

Предметная область: **Школа**

Информационные объекты:

Учитель (Код учителя, ФИО, Адрес, Семейное положение, Наличие детей, Стаж, Код категории, Код звания)

Категория (Код категории, Название)

Звание (Код звания, звание)

Оплата (Код оплаты, Код категории, ставка, премия)

Доплата (Код доплаты, Наименование, Размер)

Начисление (Код начисления, Код учителя, Код оплаты, Количество часов в неделю, Код доплаты)

Вариант 7.

Предметная область: **Киоск**

Информационные объекты:

Виды товаров (Код вида, Название вида)

Товары (Код товара, Код вида, Название, Цена)

Поступление товаров (Код поступления, Код товара, Дата поступления, Количество)

Продажа товаров (Код продажи, Код товара, Дата продажи, Количество, отпускная цена)

Вариант 8.

Предметная область: **Общежитие**

Информационные объекты:

Штаты (Код должности, Название, Количество штатных единиц, зарплата)

Персонал (Табельный номер сотрудника, ФИО, Код должности, Кабинет)

Студент (Код студента, ФИО, Номер комнаты, Домашний адрес, Код факультета, Курс, Группа)

Факультет (Код факультета, Название, Декан, Телефон деканата)

Вариант 9.

Предметная область: **Книжный магазин**

Информационные объекты:

Область знаний (Код области знаний, Название)

Книги (Код области знаний, Индекс в каталоге, Авторы, Название, Год выпуска, Цена)

Поступление книг (Код поступления, Код области знаний, Индекс в каталоге, Количество, Дата поступления)

Продажа книг (Код продажи, Код области знаний, Индекс в каталоге, Количество, Отпускная цена, Дата продажи)

Вариант 10.

Предметная область: **Склад товаров**

Информационные объекты:

Товар (Код товара, Наименование, Единица измерения, Стоимость)

Поступление товаров (Код поступления, Код товара, Дата, Количество, Номер накладной)

Выдача товаров (Код выдачи, Код товара, Количество, Дата, Код МОЛ, сумма прибыли)

МОЛ (Код МОЛ, ФИО)

/МОЛ – материально-ответственное лицо/

Вариант 11.

Предметная область: **сотрудники университета**

Информационные объекты:

Штаты (Код должности, Название, Количество штатных единиц, зарплата)

Кафедра(Код кафедры, Название, Код факультета, Телефон, Кабинет)

Сотрудник(Код сотрудника, ФИО, Код кафедры, Домашний адрес, Код должности, Адрес)

Факультет (Код факультета, Название, Декан, Телефон деканата, Кабинет)

Лабораторная работа № 4

Тема работы: Построение запросов выборки данных из связанных таблиц.

Цель работы: научиться .

Порядок выполнения работы

1) В соответствии со своим вариантом предметной области составить запросы на выборку данных из связанных таблиц. Связи между таблицами по полям следует указывать прямо внутри каждого запроса после ключевого слова WHERE (например, *SELECT fio, adres, data_postupleniya FROM pacienti, postuplenie WHERE pacienti.kod_pacienta=postuplenie.kod_pacienta*). В запросах, где нужно указать конкретные данные (например, конкретное ФИО или код), нужно указать одно значение на свое усмотрение (для примера).

2) Протестировать запросы через закладку SQL из интерфейса phpMyAdmin. Все запросы, а также их описания, следует сохранить в текстовый файл для последующей сдачи преподавателю.

3) Разработать и разместить на сервере XAMPP скриптовый файл. Скрипт должен выдавать на экран результат действия любого SELECT-запроса, который вводится в многострочное поле редактирования. На форме задается также база данных, в которой запрос должен быть выполнен, имя пользователя и пароль для подключения к серверу MySQL.

4) Используя данный скрипт, нужно выполнить все запросы, содержащиеся в вашем варианте, и сохранить результаты в виде HTML-файлов через браузер. При сдаче работы нужно будет также объяснить, как работает данный скрипт.

Варианты заданий

Вариант 1.

Страховое агентство

- Построить запрос выдачи списка застрахованных клиентов с указанием ФИО, суммы страховки, даты страховки.
- Выдать список застрахованных с указанием общей суммы его страховых взносов.

- Выдать список лиц, которым по страховке сделали возмещение, указать ФИО, сумму возмещения, дату выплаты.
- Выдать список сотрудников, указав сколько страховок каждого вида они заключили и на какую сумму.
- Выдать список клиентов, указав общую сумму взносов с длительностью страховки более чем на месяц.
- Выдать информацию о застрахованном клиенте по его ФИО (когда и на какую сумму застрахован и какой вид страховки).
- Выдать по введенному виду страховки список застрахованных.
- Выдать список работников с высшим образованием, ФИО которых начинается с заданной буквы, отсортировать список по убыванию ФИО.
- Выдать список сотрудников, указав ФИО застрахованного сотрудником и общую страховую сумму (запрос с агрегатной функцией)

Вариант 2.

Фирма комиссионной продаже автомобилей

- Выдать список авто, выставленных на продажу.
- Выдать список авто по введенной марке.
- Выдать список авто по введенным марке и цвету.
- Выдать список марок авто, выставленных на продажу, указав их количество (по каждой марке) и среднюю стоимость.
- Выдать список покупателей авто. Указать ФИО, дату, суммарную стоимость проданных авто (запрос с агрегатной функцией)
- Выдать списка автомобилей, поступивших на продажу в прошлом месяце. Указать их владельцев, год выпуска, цену;
- Выдать владельцев, автомобили которых проданы. Указать цену и дату продажи;
- Выдать покупателей, которые приобрели автомобиль марки, начинающейся с определенной буквы;
- Выдать сведения об объеме продаж фирмы за заданный день и суммы доходов фирмы;

Вариант 3.

Банк

- Выдать список сотрудников, с указанием ФИО, должности, оклада.
- Выдать информацию о клиенте по введенной фамилии, указав номер счета, дату создания.
- Выдать список клиентов, указав номер их счета, по введенному виду вклада.
- Определить объем сумм по каждой операции (с указанием операции и суммы).
- Выдать информацию о работе сотрудника по введенной ФИО, указав ФИО, должность, количество операций, общую сумму.
- Выдать список клиентов, создавших вклады в прошлом месяце.
- Выдать отчет о проделанных операциях и их объеме за вчерашний день.

В

ыдать список сотрудников, совершивших операцию в введенную дату, с указанием ФИО сотрудника, адреса сотрудника, наименования операции, суммы операции.

- Выдать список сотрудников с указанием совершенных ими операций и общей суммы операции (запрос с агрегатной функцией)

Вариант 4.

Отдел сбыта предприятия

- Выдать список сотрудников с указанием ФИО, должности, категории, оклада.
- Выдать список покупателей с указанием количества приобретенных изделий и на какую сумму.
- Выдать информацию о количестве проданных изделий заданным сотрудником.
- Выдать список проданных изделий по заданной дате с указанием названия изделия, количества и общей суммы.
- Выдать список изделий, имеющихся на складе.
- Выдать список сотрудников, ФИО которых начинается с заданной буквы, указав количество проданных изделий в прошлом месяце.
- Выдать список изделий, проданных в заданный день с указанием ФИО покупателя, количества, отпускной цены.

- Выдать список проданных изделий, с указанием даты продажи, суммарного проданного количества, суммарной отпускной цены (запрос с агрегатной функцией)

Вариант 5.

Научно-исследовательский институт (НИИ)

- Выдать список сотрудников, указав ФИО, степень, звание, должность, отдел.
- Выдать сотрудников заданного отдела с указанием должности, оклада.
- Выдать перечень должностей для заданного отдела.
- Выдать перечень вакантных должностей.
- Выдать список отделов с указанием количества должностей и общей суммы окладов (запрос с агрегатной функцией)
- Выдать список сотрудников, работающих по заданной теме, указав процент надбавки за ее выполнение для каждого сотрудника.
- Выдать список тем, которые заканчиваются в этом году.
- Выдать список сотрудников, с указанием зарплаты с учетом надбавок по темам.

Вариант 6.

Школа

- Выдать список учителей введенной категории.
- Выдать список учителей, имеющих звание заслуженного.
- Выдать список учителей без категории.
- Выдать список учителей, не имеющих детей.
- Выдать список учителей с общей суммой зарплаты (ставка+премия).
- Выдать список учителей, с заданным стажем с указанием ФИО, адреса, семейного положения, наличия детей.
- Выдать зарплату за прошлый месяц заданного учителя.
- Выдать список учителей, ФИО которых начинается с заданной буквы и указать зарплату за прошлый месяц.
- Выдать список учителей, с видами и размерами доплат за прошлый месяц.
- Выдать список учителей с указанием категории, общей суммы оплаты (ставка+премия) (запрос с агрегатной функцией)

Вариант 7.

Киоск

- Выдать товары, которые поступили в киоск.
- Выдать список поступивших товаров, указав по каждому их количество, общую стоимость, для введенной даты.
 - Выдать список проданных товаров, указав их количество, общую стоимость по каждому.
 - Определить какие товары есть в наличии, сколько и на какую сумму.
 - Выдать список товаров, которые поступили в указанном месяце, указать сколько и на какую сумму.
 - Создать запрос о продаже товаров за указанный день.
 - Выдать список товаров, указав дату поступления товара, общее поступившее количество товаров и общую сумму стоимости (запрос с агрегатной функцией)
 - Выдать список товаров, которые были проданы в прошлом месяце, указав количество и сумму стоимости.
 - Создать запрос о поступлении товара за указанный день.

Вариант 8.

Общежитие

- Выдать список сотрудников, с указанием их ФИО, должности, зарплаты.
- Выдать список студентов, с указанием их ФИО, номера комнаты, названия факультета, группы.
 - Выдать список студентов указанной группы, с указанием их ФИО, названия факультета, номера комнаты.
 - Выдать список студентов указанного факультета, с указанием их ФИО, курса, группы.
 - Выдать список факультетов, с указанием их названия, декана и общего числа студентов.
 - Выдать список сотрудников, работающих в указанном кабинете.
 - Выдать список студентов, ФИО которых начинается с заданной буквы, с указанием их группы, курса, названия факультета.

- Выдать список должностей, с указанием их названия, количества штатных единиц, оклада, общего количества занятых должностей.
- Выдать список факультетов с указанием номера комнаты, где проживают студенты указанного факультета и количества студентов, указанного факультета в этой комнате (запрос с агрегатной функцией)

Вариант 9.

Книжный магазин

- Выдать список книг, поступивших в заданную дату.
- Выдать перечень книг по заданной области знаний.
- Выдать перечень книг, указав их количество, общую стоимость, по заданному году выпуска.
- Выдать перечень книг, проданных в заданную дату.
- Выдать перечень книг, имеющихся в наличии, указав их количество, цену.
- Выдать список книг, поступивших в прошлом месяце, указав название, автора и стоимость.
- Выдать отчет о продажах за прошлый месяц, указав название, год выпуска, количество и общую сумму.
- Выдать список книг, название которых начинается с заданной буквы и поступили в этом году в продажу.
- Выдать список авторов проданных книг с указанием названия книг, общего количества проданных книг (запрос с агрегатной функцией)

Вариант 10.

Склад товаров

- Выдать список товаров, поступивших по заданной накладной.
- Выдать список товаров, поступивших в заданную дату, указав название, количество и общую сумму.
- Выдать список товаров, выданных указанному МОЛ.
- Выдать список МОЛ с указанием количества выданных товаров и их общей стоимости.
- Выдать список товаров, имеющихся в наличие с указанием названия, цены и количества.

- Выдать список МОЛ, получивших товары в прошлом месяце, указав количество и общую стоимость.
- Выдать список товаров, поступивших в этом году, указав название, количество и сумму.
- Выдать список товаров, для которых не указана единица измерения.
- Выдать список МОЛ с указанием наименования выданного товара, его общего количества и его общей стоимости (запрос с агрегатной функцией)

Вариант 11.

Сотрудники университета

- Выдать список деканов, указав факультет, телефон, кабинет.
- Построить запрос выдачи ФИО сотрудника, название кафедры где он работает, домашний адрес.
- Выдать информацию о сотруднике по его ФИО.
- Выдать список факультетов с указанием числа кафедр.
- По введенной кафедре выдать сотрудников.
- Выдать список кафедр с указанием числа преподавателей, которые работают на каждой кафедре (запрос с агрегатной функцией)

Список литературы

1. Тихоненко, Т.В. Системы управления базами данных [Электронный ресурс] : курс лекций по одноименной дисциплине для слушателей специальности 1-40 01 74 "Web-дизайн и компьютерная графика" заочной формы обучения / Т. В. Тихоненко ; Министерство образования Республики Беларусь, Учреждение образования "Гомельский государственный технический университет имени П. О. Сухого", Институт повышения квалификации и переподготовки кадров, Кафедра "Информатика". - Гомель : ГГТУ, 2014 - 74 с.

2. Маклафлин, Б. PHP и MySQL : исчерпывающее руководство / Бретт Маклафлин ; [перевел с англ. О. Сивченко]. - 2-е изд.. - Санкт-Петербург [и др.] : Питер, 2014. - 543 с.

Тихоненко Татьяна Владимировна

**СИСТЕМЫ УПРАВЛЕНИЯ БАЗАМИ
ДАННЫХ**

ПРАКТИКУМ

по одноименному курсу

для слушателей специальности

1-40 01 74 «Web-дизайн и компьютерная графика»

заочной формы обучения

Подписано к размещению в электронную библиотеку
ГГТУ им. П. О. Сухого в качестве электронного
учебно-методического документа 28.11.16.

Рег. № 95Е.

<http://www.gstu.by>