


Практическая работа №14

Тема: Создание базы данных и таблиц в оболочке MySQL Workbench.

Цель: научиться создавать базы данных и таблиц в оболочке MySQL Workbench с использованием ER – диаграмм и генерации скрипта SQL.

Задание1.

Создание базы данных и таблиц с помощью оболочки MySQL Workbench.

1. Запустите оболочку MySQL Workbench, используя меню Пуск; Нажмите на кнопку Local instance MySQL
2. В меню File выберите New Model. 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 **office**. В именах таблиц и столбцов нельзя использовать русские буквы.
3. В этом окне также нужно настроить так называемую «кодировку страницы» для корректного отображения русских букв внутри таблиц. Для этого выберите из списка пункт «cp 1251-general_ci».
4. Выберите пункт Add Diagram Создаем новую ER- диаграмму.
5. 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 **table 1**. 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 **table 1** в **k_dept** и начнем создавать столбцы.




Каждый столбец имеет:

- имя (не используйте русские буквы в имени!),
- тип данных.

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.

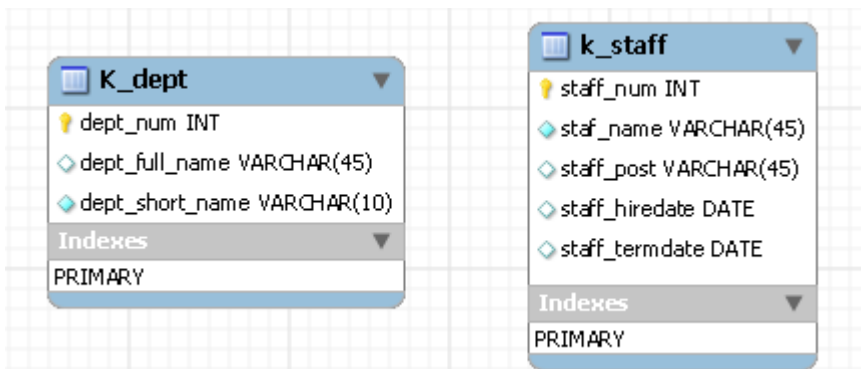
Таблица **Отделы** имеет следующий вид:

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"/>		
		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

6. Затем создадим таблицу **Сотрудники** со следующими столбцами: номер_сотрудника, имя_сотрудника, должность, дата_начала_контракта, дата_окончания_контракта


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"/>		

На диаграмме созданные таблицы выглядят следующим образом:

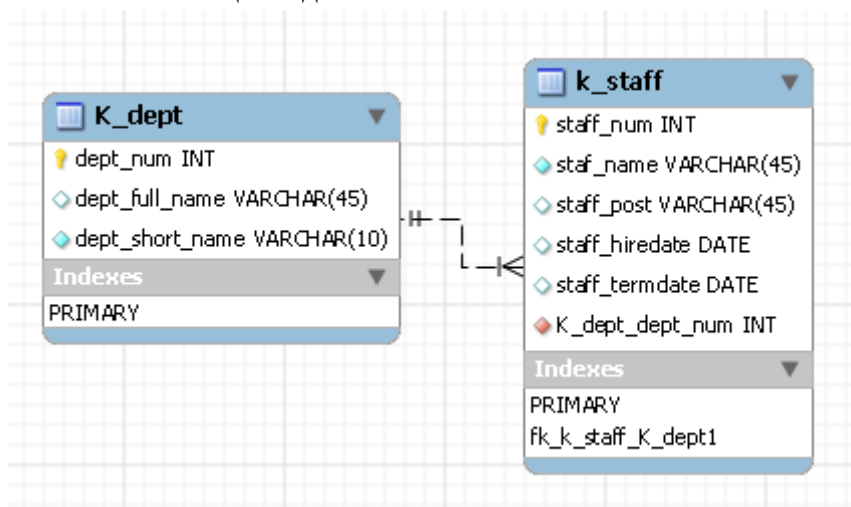


Обратите внимание, что при создании первичного ключа автоматически создается **индекс** по этому первичному ключу. **Индекс** представляет собой вспомогательную структуру, которая служит, прежде всего, для **ускорения поиска** и **быстрого доступа** к данным.

Теперь свяжем эти таблицы. Создадим связь «Работает» между **Сотрудником** (дочерняя таблица) и **Отделом** (родительская таблица), степень связи M:1. Для создания связей M: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**:



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.

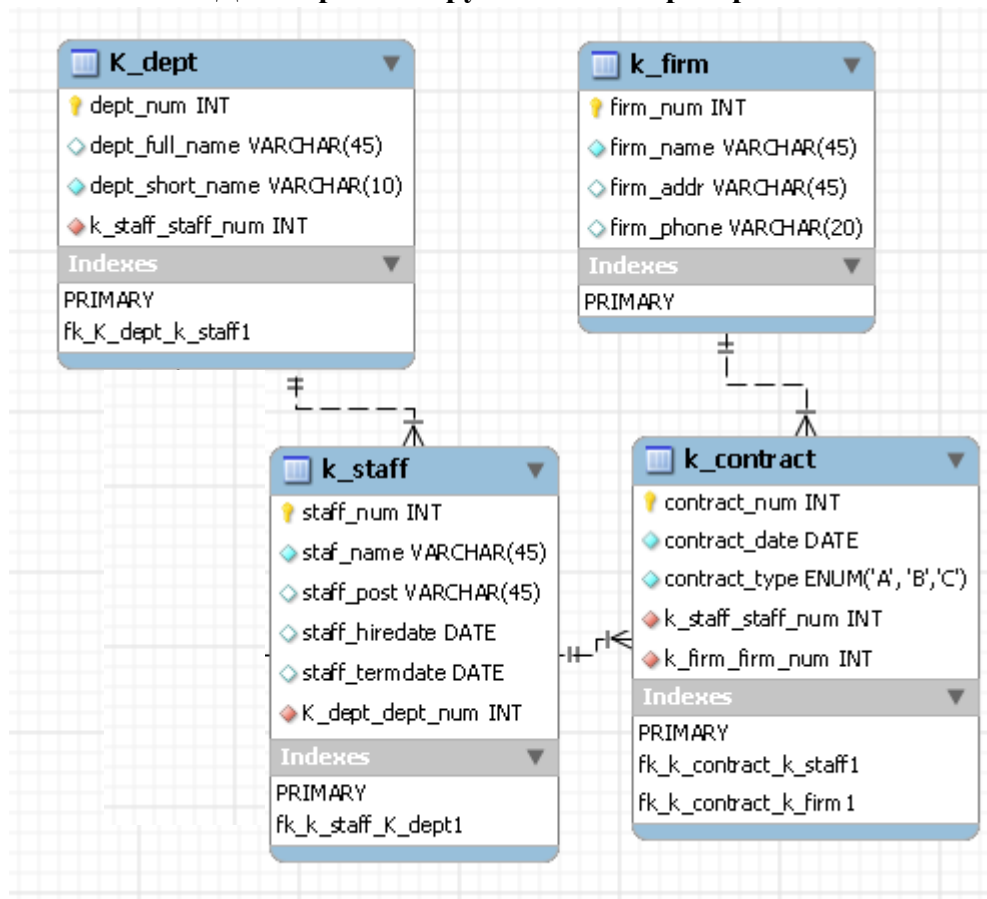
Создадим таблицу **Предприятия**:

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"/>	

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

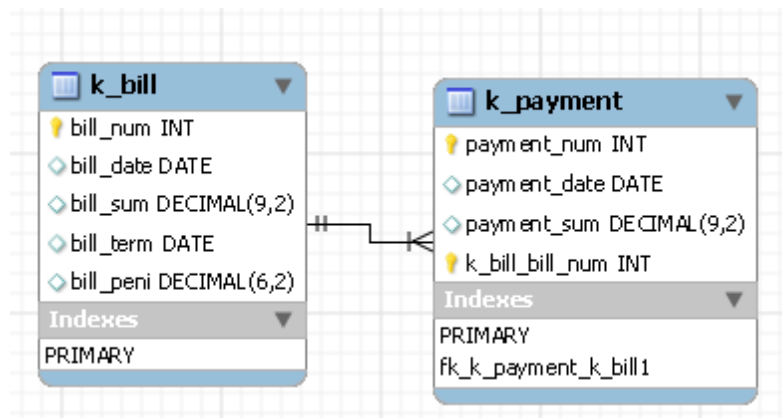
k_contract									
Column Name	Datatype	PK	NN	UQ	BIN	UN	ZF	AI	Default
contract_num	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	
contract_date	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"/>	

Свяжем **Договоры** с **Сотрудниками** и **Предприятиями** связями.




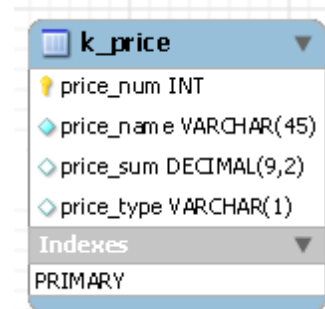
Задание 2.

Затем создадим **Счета** и **Платежи**:




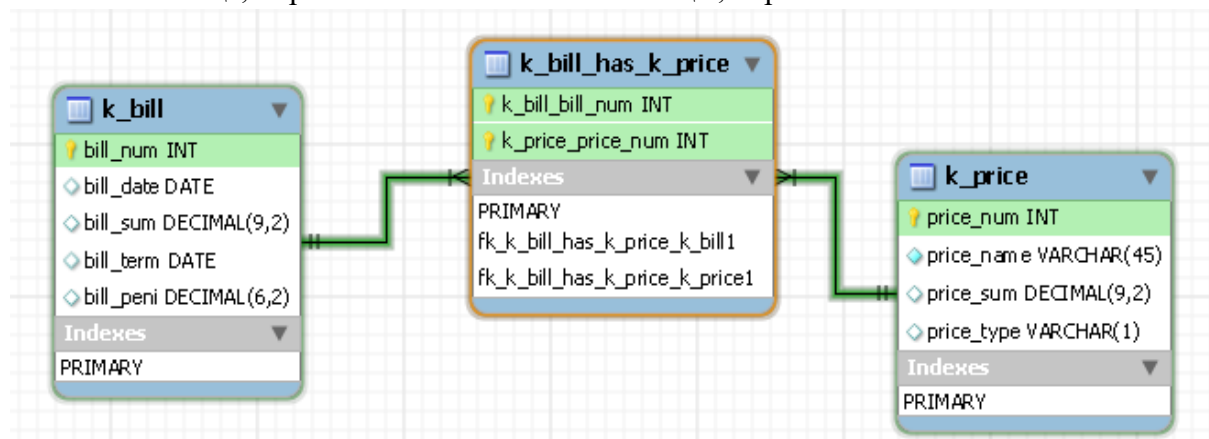
Поскольку сущность **Платеж** была «слабой», у нее нет полноценного первичного ключа, и каждый платеж однозначно идентифицируется группой атрибутов (номер_счета, номер платежа). Отметим в качестве ключевого поля *payment_num*, а затем создадим **идентифицирующую** связь между **Счетом** и **Платежом**.

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).

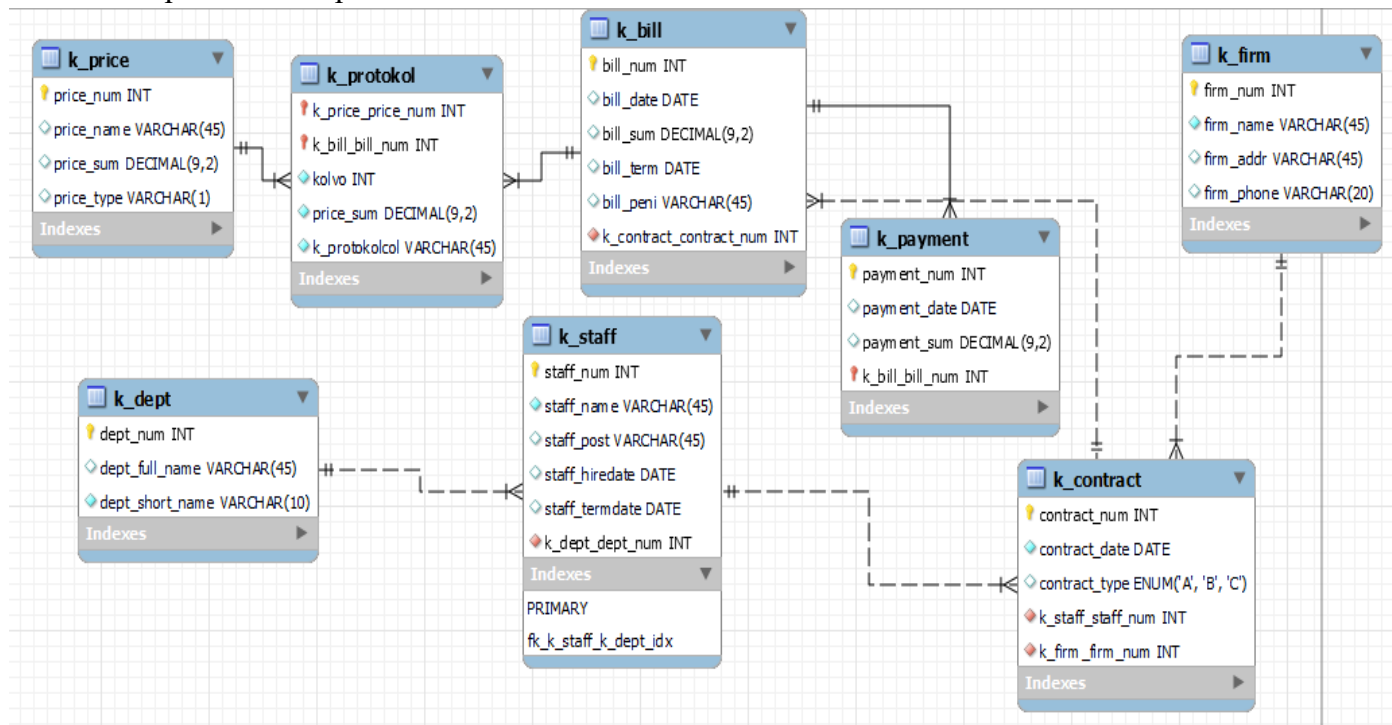
Между объектами **Счет** и **Прайс-лист** имеется связь «многие - ко многим». Для создания этой связи нужно использовать пиктограмму . Следует щелкнуть мышью по этой пиктограмме, а затем последовательно щелкнуть по связываемым таблицам. Между ними появится новая таблица, обратите внимание на ее столбцы,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



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

k_protokol									
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"/>	

Теперь EER-диаграмма имеет такой вид:



Задание 3.

На основе этой диаграммы создадим физическую базу данных. Выберем пункт меню Database - Forward Engineer. Запустится мастер построения базы данных.

На первом и втором шаге можно задать некоторые дополнительные действия. Ничего на этих страниц не выбираем, нажимаем Next .

На следующей странице включен флажок, указывающий, что мы создаем таблицы (всего 8 шт.). Других объектов пока у нас нет.


На следующей странице показывается текст Скрипта для создания базы данных. Сохраните в файл (кнопка **Save to file...**).



Далее запрашивается логин и пароль для подключения к серверу: password
Если нет никаких ошибок, то получим окно с сообщением об успешном результате:
В противном случае можно нажать на кнопку **Show logs** и посмотреть протокол ошибок.

Задание 4. Внимание!!! При выполнении заданий 4 и 5 сохраняйте запрос и результат запроса в текстовом документе.

1. Самый универсальный и гибкий способ создания данных состоит в использовании SQL-команды INSERT.

SQL-команды нужно набирать в окне Local instance MySQL. Для выполнения команд нужно выбрать меню **Query – Execute** или кнопку  на панели инструментов или нажать Ctrl+Enter.

Можно набрать несколько команд и выполнить их **все вместе**, или выделить **отдельную** команду (как для копирования) и выполнить только ее.

2. Заполним таблицу **Предприятия**:

выберем базу данных

USE office;

добавим строки

INSERT INTO k_firm (firm_name, firm_addr)

VALUES('Альфа', 'Москва');

INSERT INTO k_firm (firm_name, firm_addr)

VALUES('Бета', 'Казань');

INSERT INTO k_firm (firm_name, firm_addr)

VALUES('Гамма', 'Париж');

INSERT INTO k_firm (firm_name, firm_addr)

VALUES('Дельта', 'Лондон');

INSERT INTO k_firm (firm_name, firm_addr)

VALUES('Омега', 'Токио');

посмотрим результат

SELECT * FROM k_firm;

Обратите внимание, что мы не задавали значения для столбца *firm _ num*, поскольку этот столбец имеет свойство **Auto increment**, и сервер его заполняет сам, натуральными числами.

Overview	Output	History	Result (1)
Export			
firm_num	firm_name	firm_addr	firm_phone
1	Альфа	Москва	NULL
2	Бета	Казань	NULL
3	Гамма	Париж	NULL
4	Дельта	Лондон	NULL
5	Омега	Токио	NULL

Fetches: 5

Заполним **Отдел**

INSERT INTO k_dept (dept_short_name, dept_full_name)

VALUES('Sales', 'Отдел продаж');

INSERT INTO k_dept (dept_short_name, dept_full_name)

VALUES('Mart', 'Отдел маркетинга');

INSERT INTO k_dept (dept_short_name, dept_full_name)

VALUES('Cust', 'Отдел гарантийного обслуживания');

SELECT * FROM k_dept;

	dept_num	dept_full_name	dept_short_name
▶	1	Отдел продаж	Sales
	2	Отдел маркетинга	Mart
	3	Отдел гарантийного обслуживания	Cust
★	NULL	NULL	NULL

Заполним таблицу **Сотрудник**. Обратите внимание, что в этой таблице можно указывать **только** такой номер отдела, который **существует** в таблице **Отдел**! Пример запроса для одной строки:

```
INSERT INTO k_staff (staff_name, K_dept_dept_num, staff_hiredate, staff_post)
VALUES('Иванов', 1, '1999-01-01', 'Менеджер');
```

SELECT * FROM k_staff;

(Запросы для остальных строк напишите самостоятельно)

Overview

Output

History

Result (1)

Export

	staff_num	staff_name	staff_post	staff_hiredate	staff_termdate	K_dept_dept_num
▶	1	Иванов	Менеджер	1999-01-01	NULL	1
	2	Петров	Менеджер	2010-10-13	NULL	2
	3	Сидоров	Менеджер	2005-12-01	NULL	3
	4	Семенов	Директор	1990-01-01	NULL	1
	5	Григорьев	Программист	2008-12-19	NULL	3

Fetches: 5

3. Заполним таблицу **Договор** (Пример запроса для одной строки):

```
INSERT INTO k_contract
```

```
(contract_type, k_firm_firm_num, k_staff_staff_num, contract_date) VALUES('A', 1, 1, '2011-11-01');
```

SELECT * FROM k_contract;

(Запросы для остальных строк напишите самостоятельно)

Overview

Output

History

Result (1)

Export

	contract_num	contract_date	contract_type	k_staff_staff_num	k_firm_firm_num
▶	1	2011-11-01	A	1	1
	2	2011-10-01	B	2	1
	3	2011-09-01	C	1	1
	4	2011-11-15	A	2	2
	5	2011-08-01	B	2	2
	6	2011-07-15	C	1	3
	7	2011-11-12	A	1	4

Fetches: 7

Задание 5.

Заполним таблицу **Счет** и посмотрим содержимое:

Пример запроса для одной строки:

```
INSERT INTO k_bill (bill_date, bill_sum, bill_term, bill_peni, k_contract_contract_num)
VALUES('2011-11-12', 1000, '2011-12-12', null, 1);
```

SELECT * FROM k_bill;

(Запросы для остальных строк напишите самостоятельно)

Overview		Output	History	Result (1)		
Export						
	bill_num	bill_date	bill_sum	bill_term	bill_peni	k_contract_contract_num
▶	1	2011-11-12	1000.00	2011-12-12	NULL	1
	2	2011-12-12	2000.00	2012-01-12	NULL	1
	3	2012-01-12	2000.00	2012-02-12	NULL	1
	4	2011-12-12	6000.00	2012-01-12	NULL	2
	5	2012-01-12	2000.00	2012-02-12	NULL	2
	6	2012-01-12	2500.00	2012-02-12	NULL	3
	7	2011-12-12	1500.00	2012-01-12	NULL	4
	8	2011-12-12	1200.00	2012-01-12	NULL	5
	9	2012-01-12	10000.00	2012-02-12	NULL	5

Fetches: 9 |

И остальные таблицы заполните самостоятельно:

Overview	Output	History	Result (1)	
Export				
	payment_num	payment_date	payment_sum	k_bill_bill_num
▶	1	2011-12-15	1000.00	2
	1	2012-01-13	1500.00	3
	1	2012-01-12	1000.00	4
	1	2012-01-05	100.00	7
	1	2011-12-25	1000.00	8
	2	2012-01-15	500.00	3
	2	2012-01-12	900.00	7

SELECT * FROM k_payment;

SELECT * FROM k_price;

Overview	Output	History	Result (1)	
Export				
	price_num	price_name	price_sum	price_type
►	1	Материализация духов	1000.00	У
	2	Раздача слонов	100.00	У
	3	Слоновый бивень	3000.00	Т
	4	Моржовый клык	1500.00	Т
	5	Копыто Пегаса	5000.00	Т

'У' означает услугу, 'Т' – товар.

SELECT * FROM k_protokol;

Overview		Output	History	Result (1)
Export				
	kolvo	price_sum	k_price_price_num	k_bill_bill_num
► 1	1000.00	1	1	
2	1000.00	1	2	
1	1000.00	1	5	
2	1000.00	1	6	
1	1000.00	1	8	
20	100.00	2	3	
10	100.00	2	5	
5	100.00	2	6	
2	100.00	2	8	
2	3000.00	3	4	
1	1500.00	4	7	
2	5000.00	5	9	

Критерии оценки работы:

- выполнено без ошибок и недочетов 5 заданий - выставляется отметка **«отлично»**;
- выполнено без ошибок и недочетов 4 задания, выставляется отметка **«хорошо»**;
- выполнено без ошибок и недочетов 3 задания - выставляется отметка **«удовлетворительно»**;
- выполнено без ошибок и недочетов менее 3 заданий - выставляется отметка **«неудовлетворительно»**.

Студенты, получившие за работу неудовлетворительную оценку, должны выполнить ее повторно.