Оглавление

[Введение 2](#_Toc72420649)

[2.5.1. Создание таблиц 4](#_Toc72420650)

[2.5.2. Создание представлений (готовых запросов) 7](#_Toc72420651)

[Заключение 9](#_Toc72420652)

[Список использованных источников 10](#_Toc72420653)

# Введение

Тема курсовой работы: Информационная система «Фирма 12»

В качестве предметной области проектирования БД выбрана торговая фирма, которая осуществляет выполнение заказов, поступающих от клиентов.

Цель курсового проектирования – применить на практике знания, полученные в процессе изучения дисциплины "Базы данных", получить теоретические знания и практические навыки реализации баз данных. Осуществить анализ предметной области. Освоить концептуальное проектирование и научиться определять сущности и атрибуты БД. Научиться разрабатывать инфологическую модель БД в виде ER-диаграмм. Получить теоретические знания и практические навыки при физическом проектировании баз данных (БД). Научиться создавать даталогическую модель БД.

В качестве исходных данных предоставлены объекты и атрибуты описывающие торговую фирму, клиентов, поставщиков и т.д.

# 1 Основные положения работы

1.1

У нас есть простая таблица City из базы данных World, в которой есть строки и столбцы. Но термины: таблица, строка, столбец – это термины стандарта SQL. Кстати: ни одна из существующих в мире СУБД не имеет полной поддержки того или иного стандарта SQL, но и ни один стандарт SQL полностью не реализует математику реляционных баз данных.

В терминологии реляционных баз данных: таблица – это отношение (принимается такое допущение), строка – это кортеж, а столбец – атрибут. Иногда вы можете услышать, как некоторые разработчики называют строки записями. Чтобы не было путаницы в дальнейшем предлагаю использовать термины SQL.

Если рассматривать таблицу, как объект (например книга), то столбец – это характеристики объекта, а строки содержат информацию об объекте.

Ниже приводятся отношения. Для каждого отношения указаны атрибуты с их внутренним названием, типом и длиной.

Таблица описывающая реквизиты фирмы(FIRMA):

Идентификатор `idfirma` INT

Название `nazvan` VARCHAR

Юридический адрес `uradr` VARCHAR

Телефон `phone` VARCHAR

Идентификатор счета `idschet` INT

Таблица описывающая счет клиента либо поставщика(SCHET):

Индентификатор `idschet` INT

Название банка `bank` VARCHAR

Номер счета `numchet` VARCHAR

Статус счета `status` INT

Дата счета `data` DATE

Кредитование `creditstat` INT

Сумма кредита `creditsum` FLOAT

Таблица клиентов фирмы(CLIENT):

Идентификатор `idclient` INT

ФИО `fio` VARCHAR

Юридический адрес `uradr` VARCHAR

Счет в банке `idschet` INT

Таблица описывающая поставщиков(POSTAV):

Идентификатор `idpostav` INT

ФИО `fio` VARCHAR

Юридический адрес `uradr` VARCHAR

Счет `idschet` INT

Таблица описывающая склады(SKLAD):

Идентификатор `Nsklad` INT

Адрес склада `adress` VARCHAR

Наименование `naimen` VARCHAR

Таблица описывающая товары(TOVAR):

Идентификатор `idtovar` INT

Наименование `naimen` VARCHAR

Ед. измерения `edizm` VARCHAR

Сорт товара `sort` INT

Цена `price` FLOAT

Поставщик `idpostav` INT

Склад `sklad` INT

Количество `Kolvo` INT

Таблица описывающая заказы(ZAKAZ):

Идентификатор `id\_zakaz` INT

Дата `data` DATE

Клиент `id\_client` INT

Товар `id\_tovar` INT

Количество `num` FLOAT

Таблица описывающая накладные отпуска товаров(NAKLAD):

Идентификатор `idnaklad` INT

Товар `idtovar` INT

Количество `kolvo` INT

Сумма `summa` FLOAT

Клиент `idclient` INT

Таблица описывающая кадры(KADR):

Идентификатор `idkadr` INT

ФИО `fio` VARCHAR

Должность `dolg` VARCHAR

Образование `edu` VARCHAR

Зарплата `salary` FLOAT

Дата приема `data` DATE

Потенциальными ключами отношения Поставщики являются атрибуты ФИО и Идентификатор поставщика. Идентификатор занимает меньше места, а ФИО к тому же может измениться, поэтому мы выбираем Идентификатор в качестве первичного ключа. Точно такая же ситуация с кадрами и клиентами, ФИО может изменяться а Идентификатор и короче и останется постоянным.

В заказах и накладных

### 2.5.1. Создание таблиц

CREATE TABLE `client` (

`idclient` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250DEFAULT 'Не указано',

`uradr` varchar(500DEFAULT 'Не указано',

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idclient`),

UNIQUE KEY `idclient` (`idclient`),

KEY `FKschet` (`idschet`),

CONSTRAINT `FKschet` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

);

CREATE TABLE `firma` (

`idfirma` int NOT NULL AUTO\_INCREMENT,

`nazvan` varchar(500) NOT NULL DEFAULT '',

`uradr` varchar(500) NOT NULL DEFAULT '',

`phone` varchar(50) NOT NULL DEFAULT '',

`idschet` int NOT NULL,

PRIMARY KEY (`idfirma`),

UNIQUE KEY `idfirma` (`idfirma`),

KEY `FKschetf` (`idschet`),

CONSTRAINT `FKschetf` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`);

CREATE TABLE `kadr` (

`idkadr` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(250) NOT NULL,

`dolg` varchar(500) NOT NULL,

`edu` varchar(500) NOT NULL,

`salary` float NOT NULL,

`data` date NOT NULL,

PRIMARY KEY (`idkadr`),

UNIQUE KEY `idkadr` (`idkadr`)

);

CREATE TABLE `naklad` (

`idnaklad` int NOT NULL AUTO\_INCREMENT,

`idtovar` int NOT NULL,

`kolvo` int NOT NULL,

`summa` float NOT NULL,

PRIMARY KEY (`idnaklad`),

UNIQUE KEY `idnaklad` (`idnaklad`),

KEY `FKtovarn` (`idtovar`),

CONSTRAINT `FKtovarn` FOREIGN KEY (`idtovar`) REFERENCES `tovar` (`idtovar`)

);

CREATE TABLE `postav` (

`idpostav` int NOT NULL AUTO\_INCREMENT,

`fio` varchar(500DEFAULT 'Не указано',

`uradr` varchar(500DEFAULT 'Не указано',

`idschet` int DEFAULT NULL,

PRIMARY KEY (`idpostav`),

UNIQUE KEY `idpostav` (`idpostav`),

KEY `FKschetp` (`idschet`),

CONSTRAINT `FKschetp` FOREIGN KEY (`idschet`) REFERENCES `schet` (`idschet`)

);

CREATE TABLE `schet` (

`idschet` int NOT NULL AUTO\_INCREMENT,

`bank` varchar(50DEFAULT 'Не указано',

`numchet` varchar(50) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`fio` varchar(250) CHARACTER SET utf8mb4 COLLATE utf8mb4\_0900\_ai\_ci NOT NULL,

`status` int NOT NULL DEFAULT '0',

`data` date NOT NULL,

`creditstat` int NOT NULL DEFAULT '0',

`creditsum` float NOT NULL,

PRIMARY KEY (`idschet`),

UNIQUE KEY `idschet` (`idschet`)

);

CREATE TABLE `sklad\_0` (

`Nsklad` int unsigned NOT NULL AUTO\_INCREMENT,

`adress` varchar(5000) DEFAULT NULL,

`zakazn` int DEFAULT NULL,

`tovarname` varchar(5000) DEFAULT NULL,

`edizm` varchar(50) DEFAULT NULL,

`tovarkolvo` int DEFAULT NULL,

`tovarprice` float DEFAULT NULL,

`oplata` char(100) DEFAULT NULL,

PRIMARY KEY (`Nsklad`)

);

CREATE TABLE `tovar` (

`idtovar` int NOT NULL AUTO\_INCREMENT,

`naimen` varchar(500DEFAULT 'Не указано',

`edizm` varchar(50DEFAULT 'Не указано',

`sort` int NOT NULL,

`price` float NOT NULL DEFAULT '0',

`idpostav` int NOT NULL,

PRIMARY KEY (`idtovar`),

UNIQUE KEY `idtovar` (`idtovar`),

KEY `FKpostavt` (`idpostav`),

CONSTRAINT `FKpostavt` FOREIGN KEY (`idpostav`) REFERENCES `postav` (`idpostav`)

);

CREATE TABLE `zakaz` (

`id\_zakaz` int NOT NULL AUTO\_INCREMENT,

`data` date DEFAULT NULL,

`id\_client` int NOT NULL,

`id\_tovar` int NOT NULL,

`num` float NOT NULL,

`price` float NOT NULL,

PRIMARY KEY (`id\_zakaz`),

UNIQUE KEY `id\_zakaz` (`id\_zakaz`),

KEY `FKtovar` (`id\_tovar`),

KEY `FKclientz` (`id\_client`),

CONSTRAINT `FKclientz` FOREIGN KEY (`id\_client`) REFERENCES `client` (`idclient`),

CONSTRAINT `FKtovar` FOREIGN KEY (`id\_tovar`) REFERENCES `tovar` (`idtovar`)

);

### 2.5.2. Создание представлений (готовых запросов)

#### Задание а.

В каком количестве и по какой цене имеется данный товар на данном складе.

SELECT t.naimen, s.adress, t.price, t.kolvo

from tovar t, sklad s

WHERE

t.sklad=s.Nsklad

AND t.naimen='Товар'

AND s.naimen='Склад'

#### Задание b

Кто из поставщиков не поставляет данный товар по данной цене.

Невнятный вопрос

SELECT \* FROM

postav p, tovar t

WHERE

t.idpostav=p.idpostav

AND t.naimen='Молоко'

#### Задание c.

Какой товар и в каком количестве был отпущен на данную сумму по данной накладной.

SELECT t.naimen, n.kolvo FROM

naklad n, tovar t

WHERE

n.idtovar=t.idtovar

AND n.idnaklad=2

AND n.summa=100

#### Задание d.

Кто из сотрудников данной фирмы является поставщиком товаров.

SELECT k.fio, k.dolg FROM

kadr k, postav p

WHERE

k.fio=p.fio

#### Задание e.

В каком банке открыл счет данный поставщик.

SELECT p.fio, s.bank FROM

schet s, postav p

WHERE

p.fio= 'ООО "Бабушкина крынка"'

AND p.idschet=s.idschet

# Заключение

Целью создания любой БД является упрощение использования больших массивов информации. БД позволяют собирать, хранить, обновлять и выводить информацию в понятной пользователю форме.

В ходе курсовой работы была разработана и создана база данных "Школа", которая значительно упрощает работу с данными и во многом экономит время сотрудников. Созданная ы, в нее могут быть введены дополнительные средства формирования данных.

# Список использованных источников