

Базы данных



 Priem_UGNTU

 pkrusoil

 УФИМСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
ПРИЕМНАЯ КОМИССИЯ

Поиск по сайту

[Главная](#) [Об УГНТУ](#) [Бакалавриат и специалитет](#) [Магистратура](#) [Аспирантура](#) [СПО](#) [Контакты](#)

УГНТУ СООБЩАЕТ:
ПРИЕМ НА ЗАОЧНУЮ ФОРМУ ОБУЧЕНИЯ ПРОДОЛЖАЕТСЯ!

Более подробная информация на сайте pk.rusoil.net или по телефону 242-08-59.

Прием на заочное обучение

СТУДЕНЧЕСКИЙ БИЛЕТ

УФИМСКИЙ ГОСУДАРСТВЕННЫЙ НЕФТЯНОЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

ПРИЕМ ДОКУМЕНТОВ ONLINE

УГНТУ

до 70 000 Р

ПОБЕДИТЕЛЯМ И ПРИЗЕРАМ ОЛИМПИАД ДОПОЛНИТЕЛЬНЫЕ СТИПЕНДИИ

S7 Airlines Покупка и управление S7 Priority Информация Бизнесу Путешественникам

RU RUB Войти

которых все ждали
Более 100 направлений для летнего отдыха

Покупка | **Регистрация на рейс** | Мои бронирования | Статус рейса

Авиабилеты | Отели ↗ | Апартаменты ↗ | Аренда авто | Трансфер ↗

Уфа UFA ↔

Оплата милами У меня есть сертификат

АПТЕКА 02 ПЛЮС г. Уфа, ул. Первомайская, д. 27
8 (347) 266-37-49
Время работы: с 08:00 до 22:00
СМОТРЕТЬ СХЕМУ ЛОКАЦИИ

ЛИЧНЫЙ КАБИНЕТ
КОРЗИНА ПУСТА

скачать на iOS | Android | Скачать с ПК

КАТАЛОГ Введите название препарата, не менее 3х символов
Например: Найз табл

3% СКИДКА*

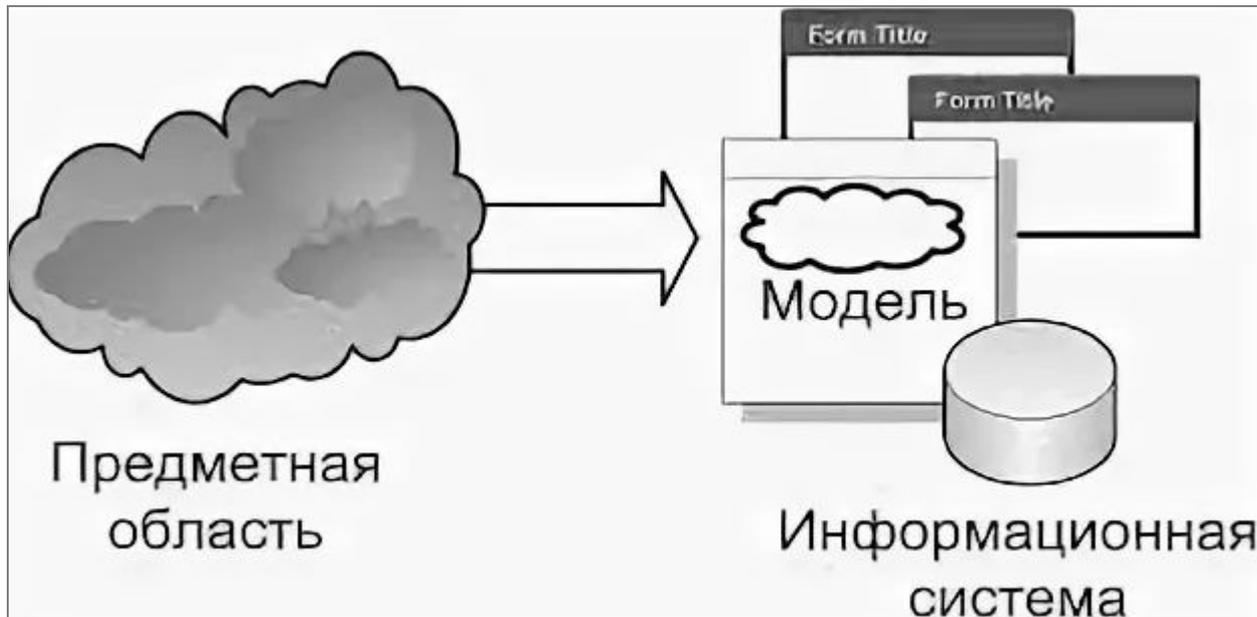
Скоро на нашем сайте... ДОСТАВКА ИЗ АПТЕКИ! 2

Информационные системы:

Основные понятия

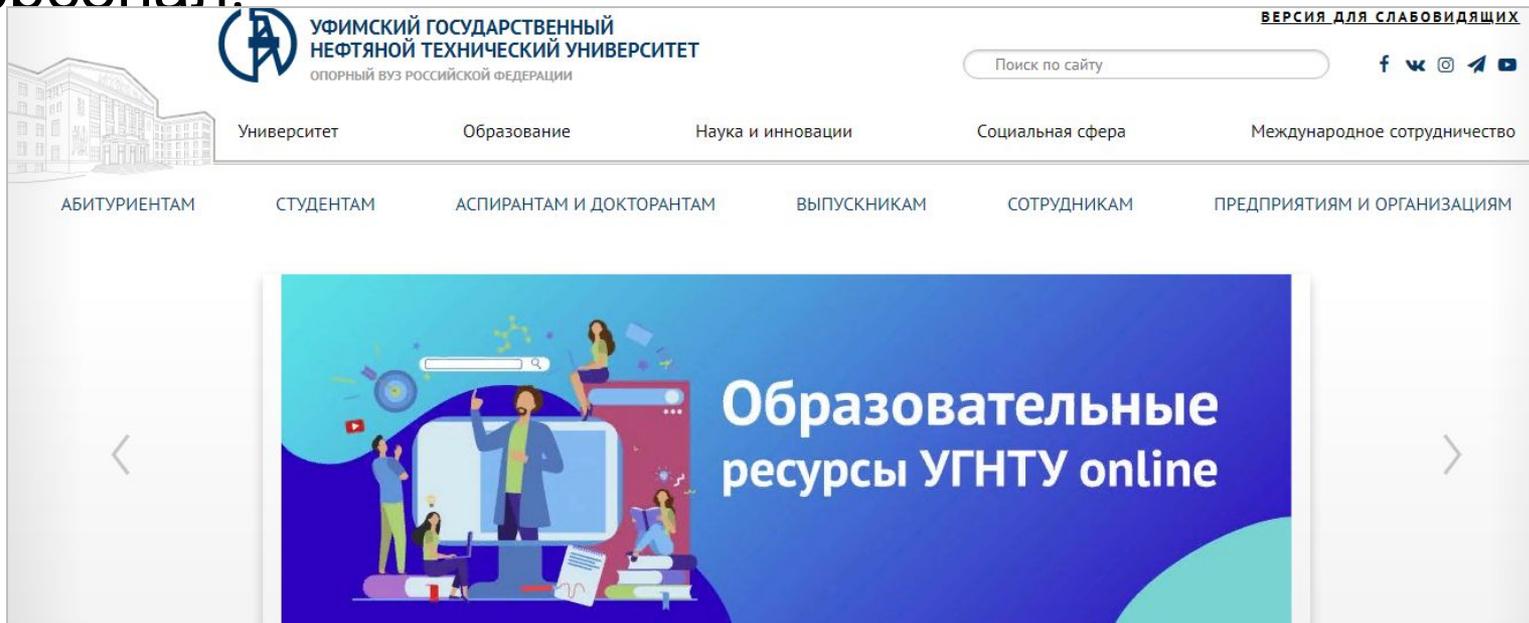
Данными обычно называют описания объектов, событий, явлений реального мира, сохраненные для последующего использования.

Предметная область — это часть реального мира, рассматриваемая в рамках определённой деятельности и сведения о которой предполагается хранить и обрабатывать в информационной системе.



Основные понятия

Информационная система – это система, реализующая автоматизированный сбор, хранение, поиск, извлечение и модификацию данных и включающая технические средства обработки данных, программное обеспечение и соответствующий персонал.



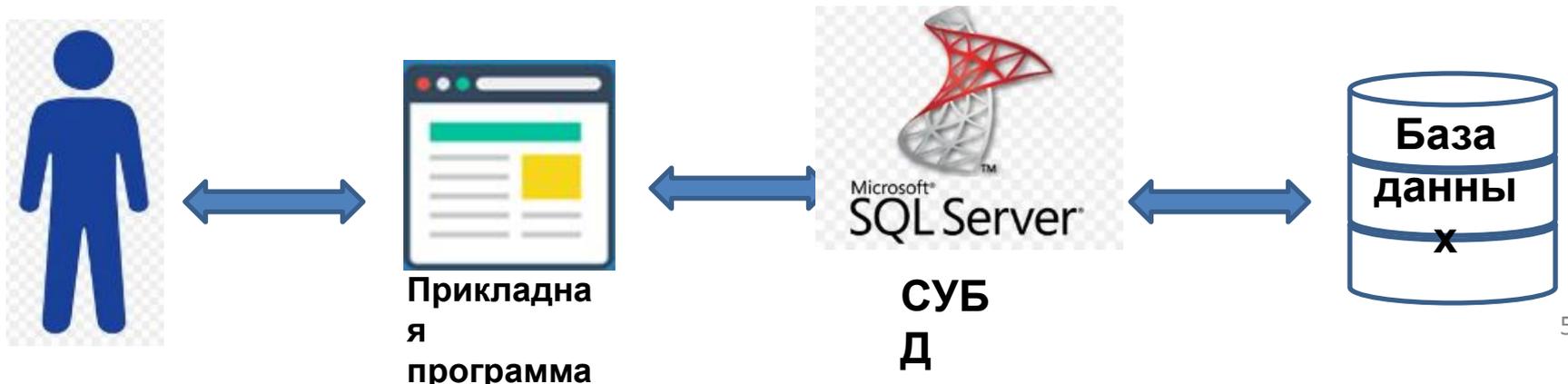
Информационная система = БД + СУБД

Основные понятия

База данных (БД) (Database, BD) – это организованная совокупность данных о некоторой предметной области, предназначенная для длительного хранения и постоянного применения.



Система управления базой данных (СУБД) (Database Management System, DBMS) – это программное обеспечение для работы с БД, т.е. совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.



Задачи СУБД:

- 1) добавление, обновление (изменение), удаление данных;
- 2) поиск данных по заданным условиям;
- 3) выполнение несложных расчетов;
- 4) обеспечение целостности данных;
- 5) возможность восстановления данных при необходимости.

Примеры СУБД:

- MS Access, MS SQL Server (Microsoft Corporation);
- Oracle, MySQL (Oracle Corporation);
- PostgreSQL (PostgreSQL Global Development Group) и т.д.



Информационные системы

Локальные

(БД, СУБД и клиентские программы установлены на рабочей станции (PC))



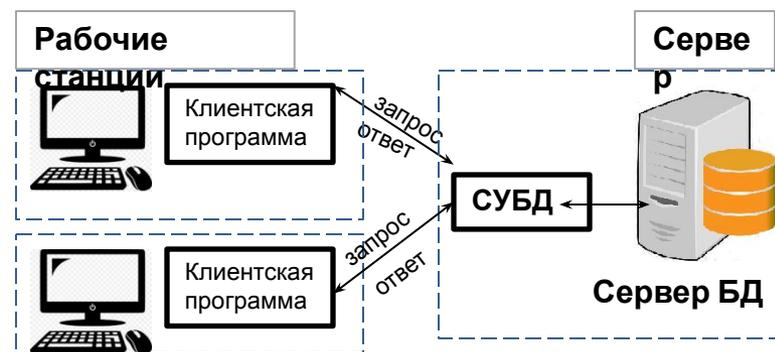
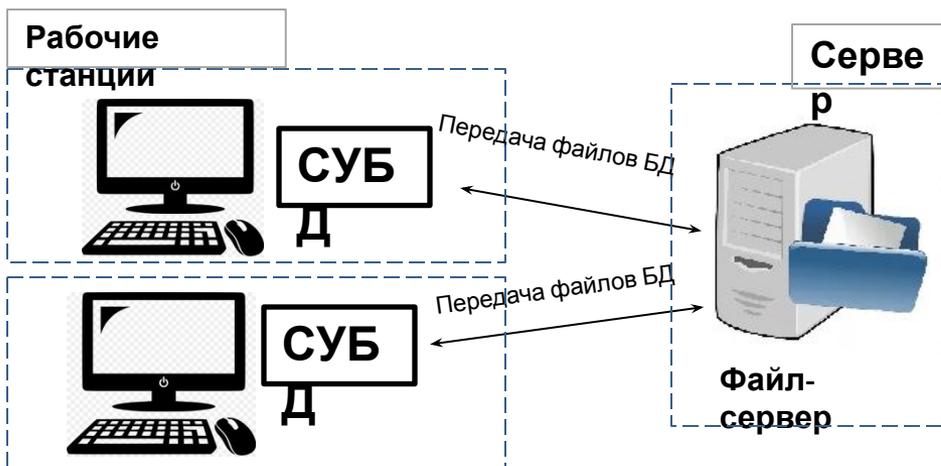
Файл – серверные
(БД находится на сервере сети (файловом сервере), а СУБД и клиентские программы на рабочей станции)

Удаленные (сетевые)

Клиент-серверные

(БД и СУБД находятся на сервере (сервер БД), а клиентские программы на рабочих станциях .

С рабочей станции (клиента) отправляются запросы на сервер (используется специальный язык запросов SQL), полученные результаты выводятся на экране рабочей станции (клиенте)



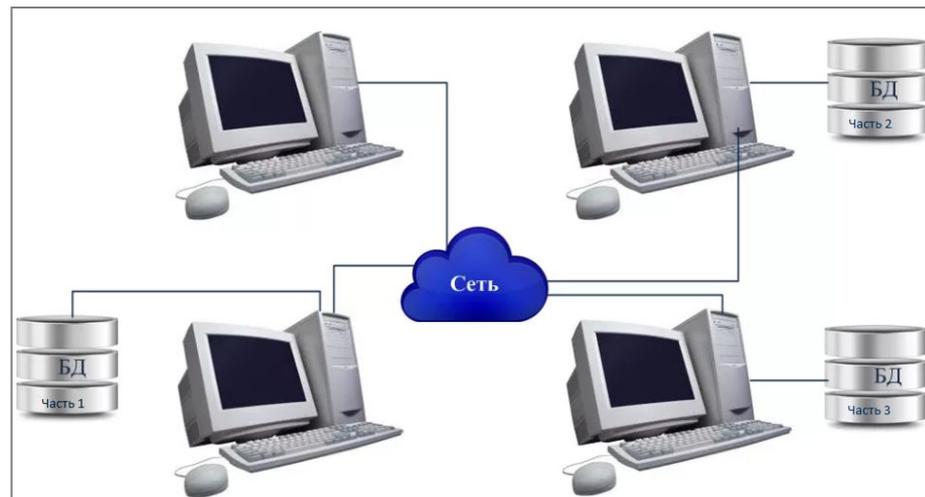
Классификация БД по способу хранения данных

Базы данных

Централизованные
(БД хранится на одном компьютере)



Распределенные
(составные части единой БД хранятся на нескольких компьютерах, объединенных в сеть)

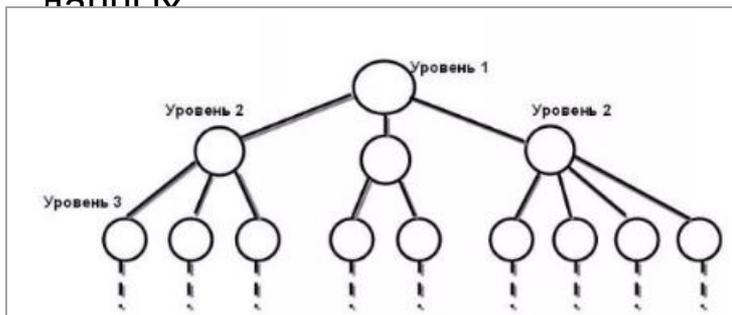


Модели данных БД

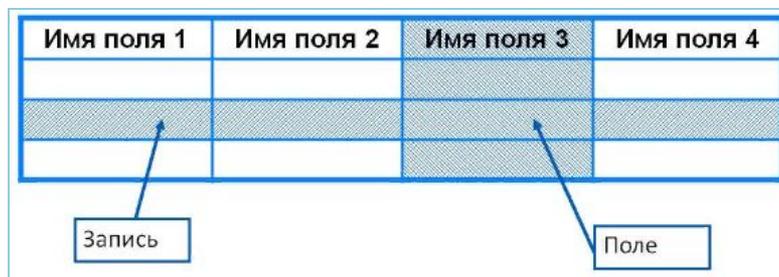
Модель данных – это метод (принцип) логической организации данных, используемый СУБД.

По способу установления связей между данными исторически сложились 3-и классические модели: **иерархическая, сетевая, реляционная**.

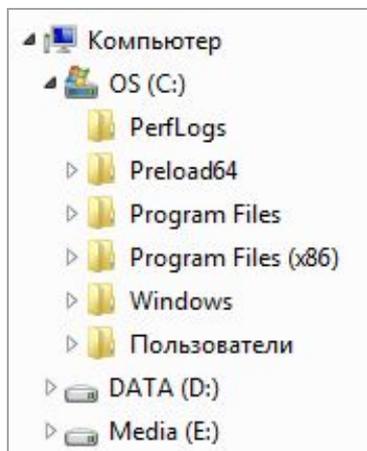
Далее появились постреляционная, многомерная и объектно-ориентированная модели данных.



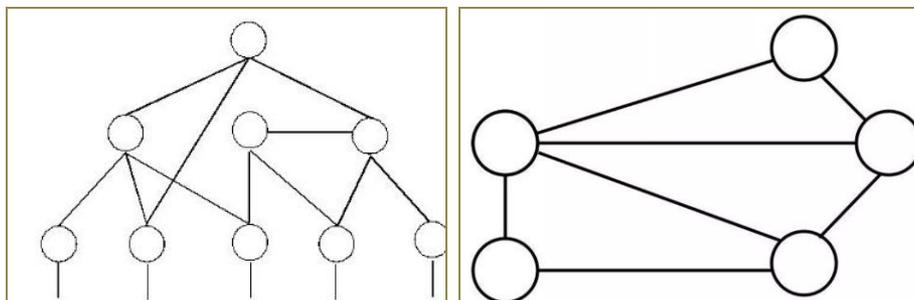
Графическое представление иерархической модели



Структура таблицы реляционной БД



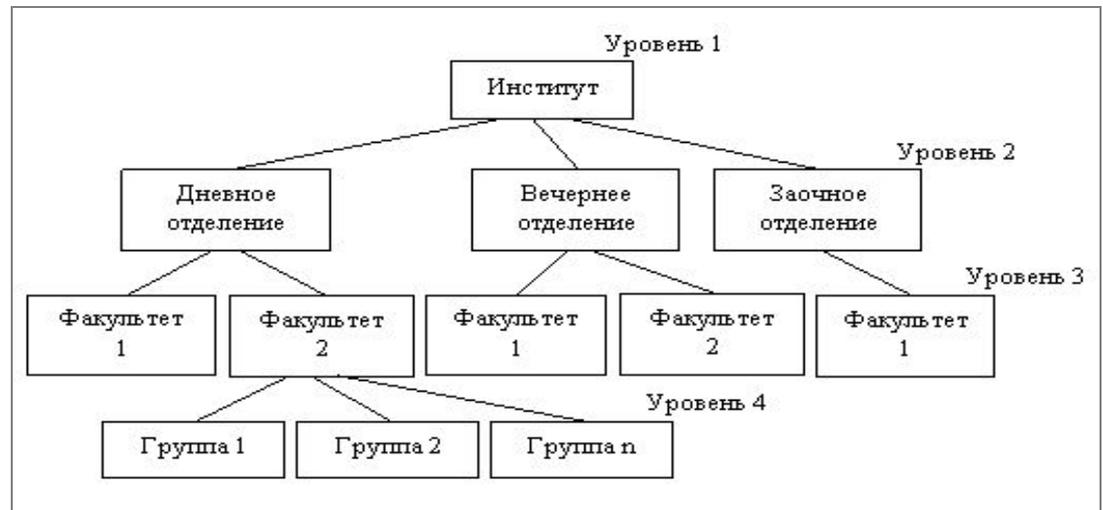
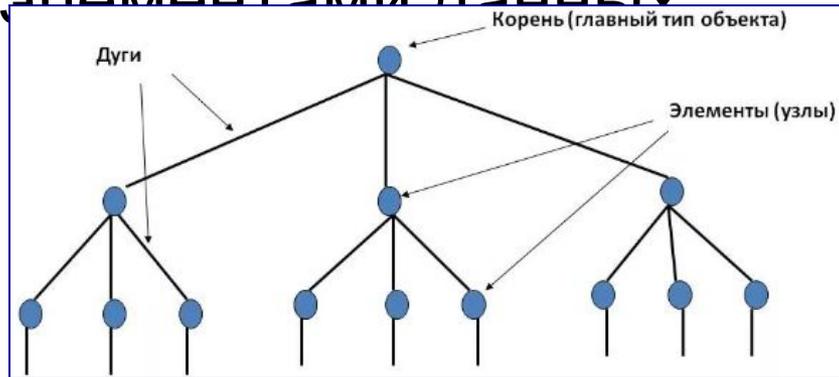
Иерархическая модель схожа по принципу построения с файловой системой



Графические представления сетевой модели данных

Иерархическая модель данных

Иерархическая модель данных имеет форму дерева с дугами-связями и узлами-элементами данных.



Сетевая модель данных

Сетевую модель данных можно рассматривать как расширенную версию иерархической модели.

Основное различие между этими двумя моделями состоит в том, что в сетевой модели запись может иметь связи со многими другими записями, а не только с одной родительской.



Реляционная модель данных

Реляционная база данных – это набор простых таблиц (отношений, сущностей), между которыми установлены связи с помощью ключей.

- Edgar Frank Codd.
- Основные концепции модели опубликована в 1970 г.
- Модель основывается на понятии «отношения» (Relation).

Запись - это строка таблицы.

Поле - это столбец таблицы.

Имя поля содержит название столбца вынесенное в заголовок.

Поле (имя + тип (свойства: размер, формат и др.))



	Имя поля		

таблица «Предприятие»

Код	Наименование	Отрасль	Вид деятельности	Дата основания
1	Уфимское моторостроительное производственное объединение	авиадвигателестроение	производство авиационных двигателей	1925
2	Туймазинский картонно-бумажный комбинат	целлюлозно-бумажная	производство и продажа бумаги	1962
3	Учалинский горно-обогатительный комбинат	горное дело	добыча и производства	1961

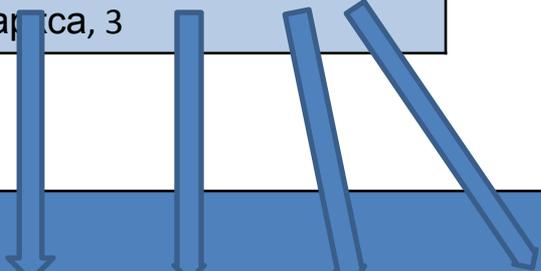
Атомарные значения полей

БД ВУЗа, таблица «Студент»

Код	Фамилия	Имя	Отчество	Год рождения	Пол	Адрес
1	Иванов	Иван	Иванович	2001	мужской	450075 Уфа, б-р Славы, 2
2	Сидоров	Петр	Петрович	2001	мужской	450064 Уфа, Мира, 7
3	Синицына	Инна	Петровна	2002	женский	450008 Уфа, пл. Карла Маркса, 3

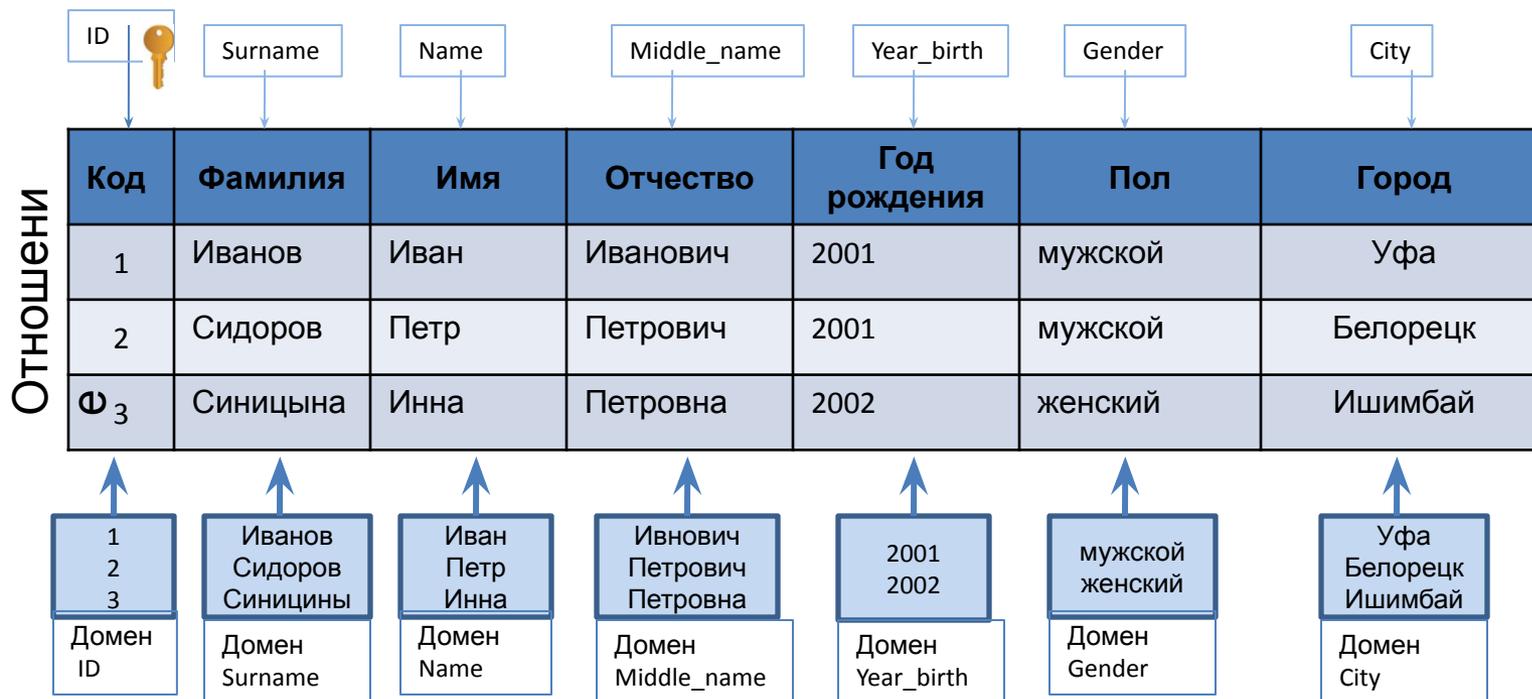
БД Издательства, таблица «Подписчик»

Код	Фамилия	Имя	Отчество	Год рождения	Пол	Индекс	Город	Улица	Дом
1	Иванов	Иван	Иванович	2001	мужской	450075	Уфа	б-р Славы	2
2	Сидоров	Петр	Петрович	2001	мужской	450064	Уфа	Мира	7
3	Синицына	Инна	Петровна	2002	женский	450008	Уфа	пл. Карла Маркса,	3



Основные понятия

Термины реляционной модели	Термины «табличные» и языка SQL	Термины обработки данных
Отношение	Таблица	Файл
Кортеж	Строка	Запись
Атрибут	Столбец	Поле
Домен	Множество допустимых значений	Базовый или пользовательский тип данных (с условиями)
Мощность (кардинальность) отношения	Количество строк	Количество записей
Степень отношения	Количество столбцов	Количество полей



Ключи

Первичный ключ (сокращенно РК - Primary Key) – это поле (или совокупность полей), значения которого не могут повторяться.



таблица «Студент»						
Фамилия	Имя	Отчество	Дата рождения	Пол	Номер зачетной книжки	Адрес
Иванов	Иван	Иванович	01.01.2001	мужской	12345678	450075, Уфа, б-р Славы, 2
Сидоров	Петр	Петрович	02.02.2001	мужской	89123456	450900 д. Жилино, Семейная, 7
Синицына	Инна	Петровна	03.03.2002	женский	56789012	450008 Уфа, пл. Карла Маркса, 3



**Первичный
ключ**

Ключи по типу

Простой первичный ключ состоит из одного поля.

Составной первичный ключ состоит из более чем одного поля.

таблица «Студент»						
Фамилия	Имя	Отчество	Дата рождения	Пол	Номер зачетной книжки	Адрес
Иванов	Иван	Иванович	01.01.2001	мужской	12345678	450075, Уфа, б-р Славы, 2
Сидоров	Петр	Петрович	02.02.2001	мужской	89123456	450900 д. Жилино, Семейная, 7
Синицын а	Инна	Петровна	03.03.2002	женский	56789012	450008 Уфа, пл. Карла Маркса, 3

Составной первичный
ключ

Простой первичный
ключ

таблица «Студент»					
Фамилия	Имя	Отчество	Дата рождения	Пол	Адрес
Иванов	Иван	Иванович	01.01.2001	мужской	450075, Уфа, б-р Славы, 2
Сидоров	Петр	Петрович	02.02.2001	мужской	450900 д. Жилино, Семейная, 7
Синицын а	Инна	Петровна	03.03.2002	женский	450008 Уфа, пл. Карла Маркса, 3

Ключи по способу задания

Логический (естественный) первичный ключ – поле, данные в котором логически связаны с информационным содержимым записи и уникально ее идентифицируют **Логический первичный**

ключ

таблица «Пользователь»		
Имя	E-mail	Пароль
Иван	ivan@mail.ru	A@s123
Семен	semen@bk.ru	@A345s
Инна	Inna@yandex.ru	f5@As

Суррогатный (искусственный) первичный ключ - поле добавленное искусственным путем для однозначной идентификации записей.

Суррогатный первичный

ключ

таблица «Пользователь»			
Код пользователя	Имя	E-mail	Пароль
5	Иван	ivan@mail.ru	A@s123
6	Семен	semen@bk.ru	@A345s
7	Инна	Inna@yandex.ru	f5@As

Ключи

Внешний ключ (сокращенно FK - Foreign Key) – неключевое поле (совокупность полей) таблицы, связанное с первичным ключом другой таблицы.



Ключи

Ключ или Потенциальный ключ (Candidate key) - простой или составной ключ, который уникально идентифицирует каждую запись таблицы.

Если отношение имеет более одного потенциального ключа, то один из них рассматривается как первичный, остальные являются **альтернативными (Alternative key)**.

таблица «Пользователь»				
Код пользовател я	Имя	Логин	E-mail	Пароль
5	Иван	Ivan	ivan@mail.ru	A@s123
6	Семен	Semen	semen@bk.ru	@A345s
7	Инна	Inna	Inna@yandex.ru	f5@As

Потенциальные ключи:

- 1) Код пользователя (например, первичный ключ)
- 2) Логин
- 3) E-mail
- 4) Имя + Пароль

Ограничения целостности данных

Целостностью данных можно назвать механизм поддержания соответствия базы данных предметной области.

Ограничения целостности предназначены для предупреждения возможности добавления в базу недопустимым данным .

Базовые требования обеспечения целостности:

- 1) **Ссылочную целостность** обеспечивается системой первичных и внешних ключей. Правило: значение внешнего ключа подчиненной таблицы должно быть соответствовать одному из значений первичного ключа главной таблицы.
- 2) **Целостность сущностей**. Правило: любая таблица (отношение) должна иметь первичный ключ или проверку уникальности. Поле первичного ключа не должно содержать пустые значения (*NULL*).
- 3) **Целостность домена** обеспечивается заданием условий на значения, ключами, хранимыми процедурами, триггерами, запретом пустых значений. Например, при создании структуры таблицы можно сразу указать количество цифр в поле Номер телефона или Фамилия.

Домен (в реляционной модели данных) - множество допустимых значений поля.

Задачи

1) Какие поля в таб.1 и таб.2 могут быть первичными ключами?

Определите названия ключей по типу и по способу задания.

Дата	Время	Температура воды	Температура воздуха	Скорость ветра	Волны
01.06.2019	9:00	19	29	8	2
26.07.2019	9:00	22	30	7	2
23.08.2019	10:00	24	36	9	3

Фамилия	Имя	Адрес	Телефон
Иванов	Иван	г.Уфа, Айская,11	9271112233
Евдокимов	Петр	г.Сибай, Ленина,2	9167779900
Петров	Иван	г.Уфа, Мира,10	9064455612

Задачи

2) Какие из приведенных ниже полей могут стать простым естественным первичным ключом?

- фамилия;
- имя;
- номер и серия паспорта;
- номер дома;
- город проживания;
- адрес электронной почты;
- дата выполнения работы;
- марка стиральной машины.

3) Сколько заказчиков из Перми?

Заказчики

Код	Название	Код города
1	ООО «Альфа»	3
2	ЗАО «Бета»	2
3	ООО «Гамма»	3
4	ОАО «Дельта»	2
5	ООО «Каппа»	1

Города

Код	Название
1	Москва
2	Санкт-Петербург
3	Пермь
4	Воронеж
5	Липецк

Задачи

- 4) Определите какие товары отправлены в каждый из городов?
- 5) Сколько бумаги отправлено в каждый из городов?
- 6) Посчитайте общую стоимость товаров, отправленных в каждый из городов.

Заказчики

Код	Название	Код города
1	ООО «Альфа»	3
2	ЗАО «Бета»	2
3	ООО «Гамма»	3
4	ОАО «Дельта»	2
5	ООО «Каппа»	1

Города

Код	Название
1	Москва
2	Санкт-Петербург
3	Пермь
4	Воронеж
5	Липецк

Товары

Артикул	Название	Цена за упаковку
7576	Бумага	150 руб.
2325	Карандаши	200 руб.
3889	Фломастеры	350 руб.
2987	Дневники	400 руб.
7825	Пеналы	250 руб.

Заказы

Накладная	Код заказчика	Артикул	Кол-во упаковок
1011	3	7576	10
1012	5	7576	20
1013	4	3889	25
1014	1	7825	30
1015	3	7576	10

Виды связей между реляционными таблицами

Реляционная база данных — это совокупность взаимосвязанных таблиц.

Виды связей между таблицами:

- 1) Один к одному (1:1, 1 – 1).
- 2) Один ко многим (1:M, 1 – ∞).
- 3) Многие ко многим (M:M, ∞ – ∞).

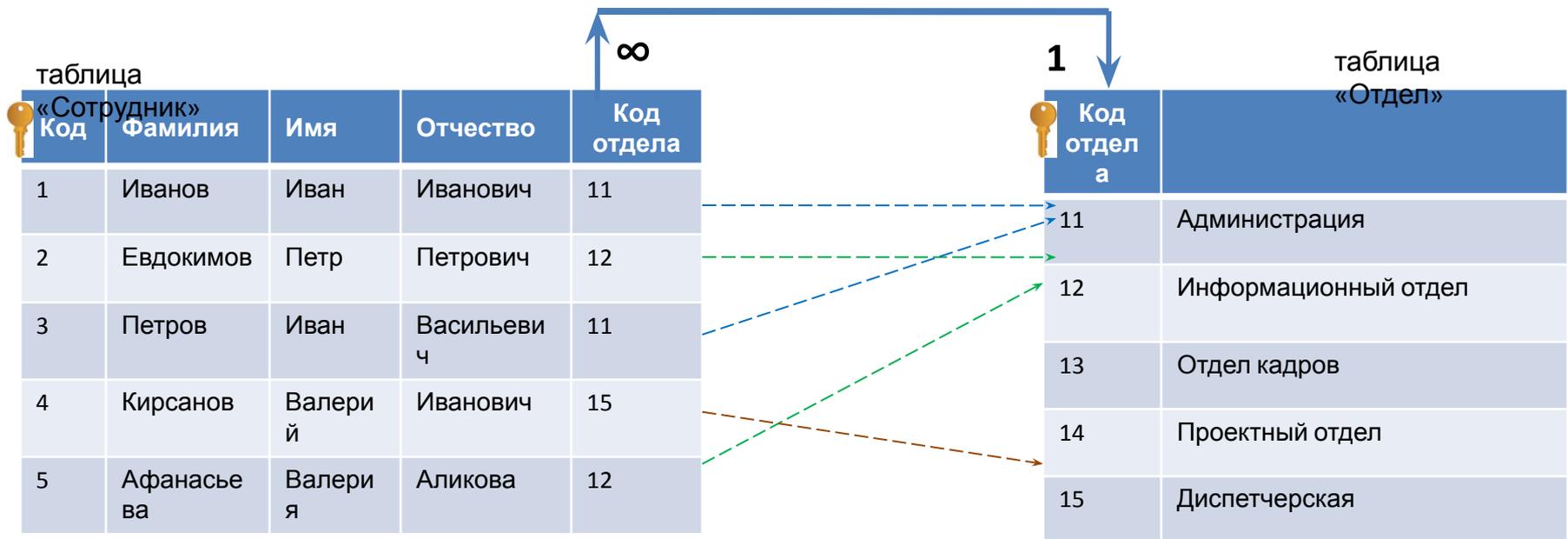
Виды связи между реляционными таблицами

Связь **один к одному** означает, что одной записи в первой таблице соответствует только одна запись во второй.



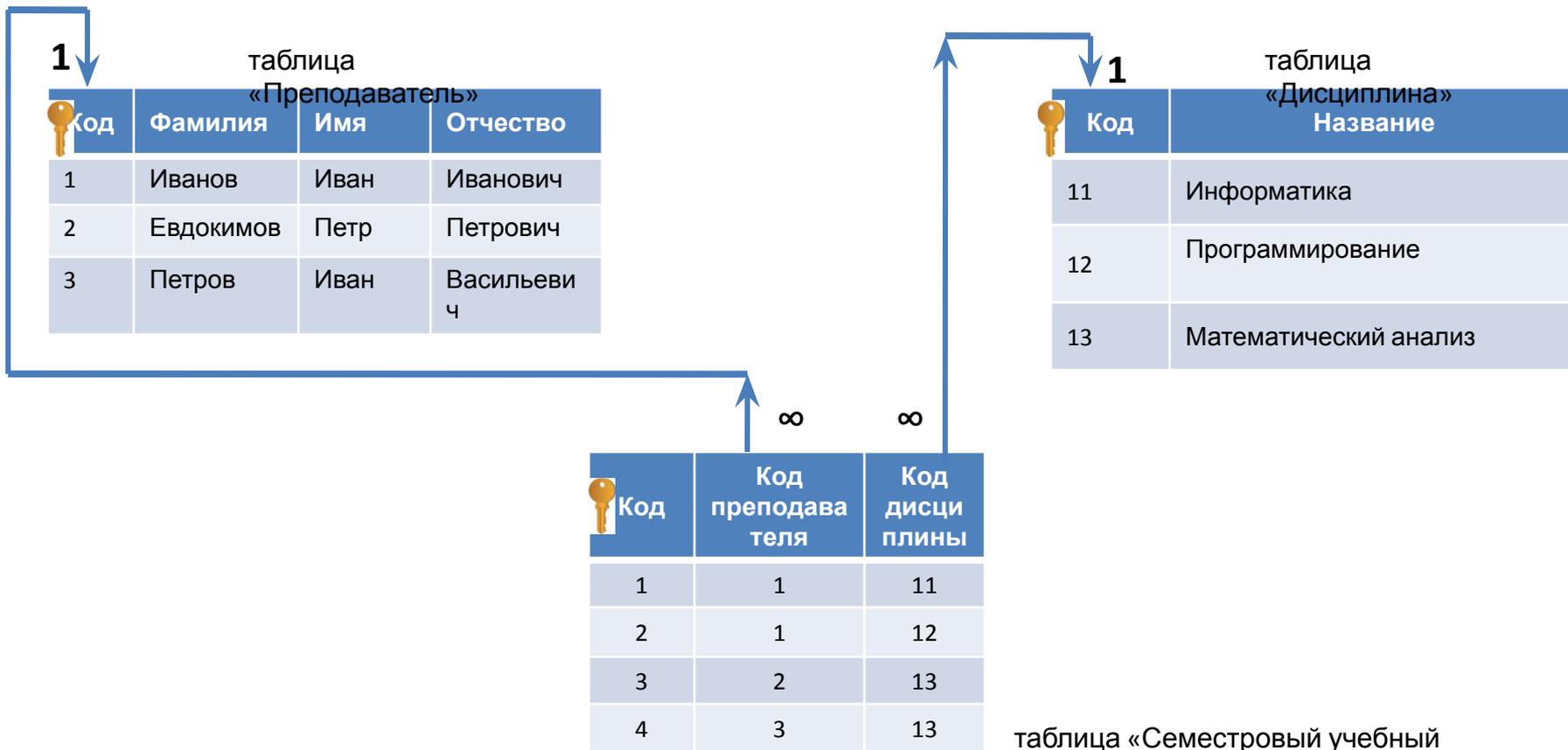
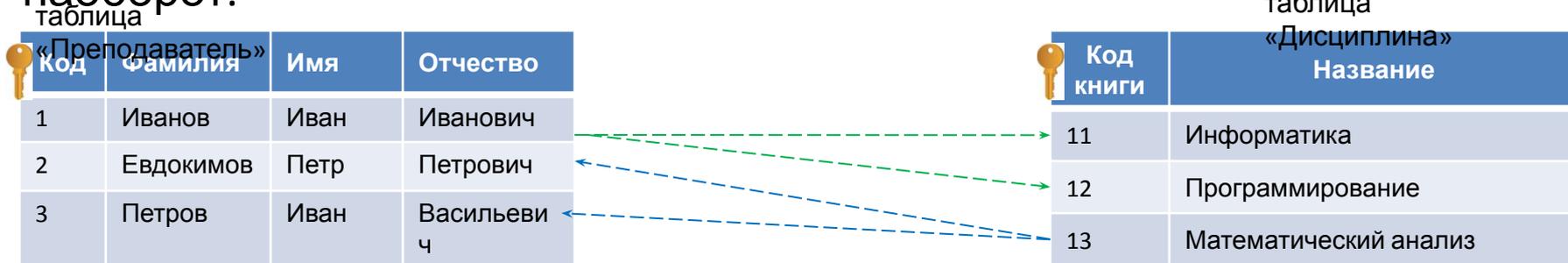
Виды связей между реляционными таблицами

Связь **один ко многим** означает, что одной записи в первой таблице может соответствовать множество записей во второй.



Виды связей между реляционными таблицами

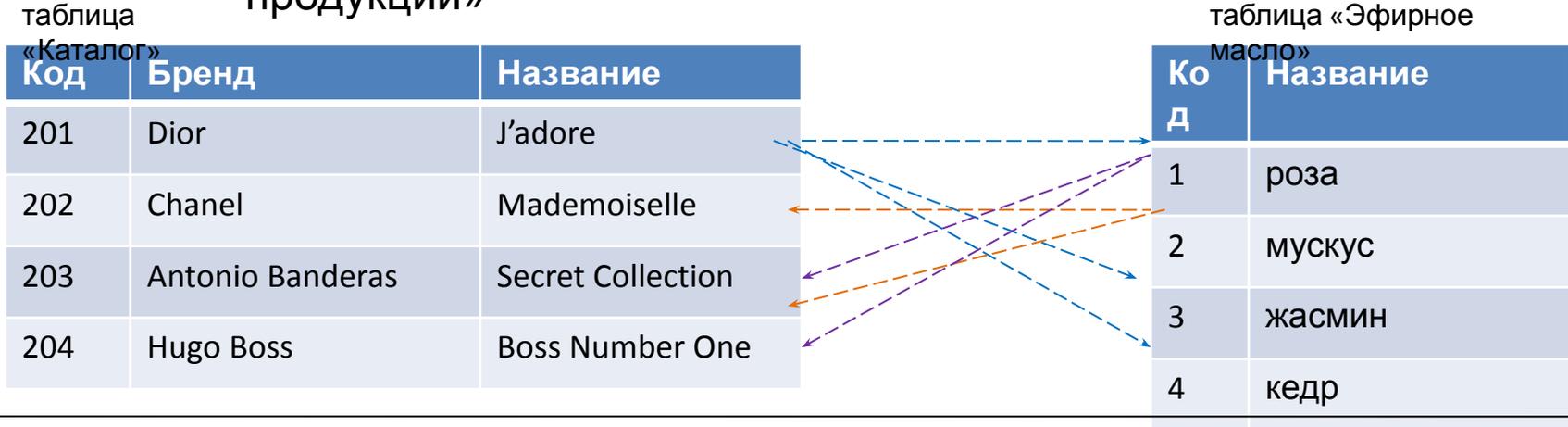
Связь **многие ко многим** означает, что одной записи в первой таблице может соответствовать множество записей во второй, и наоборот.



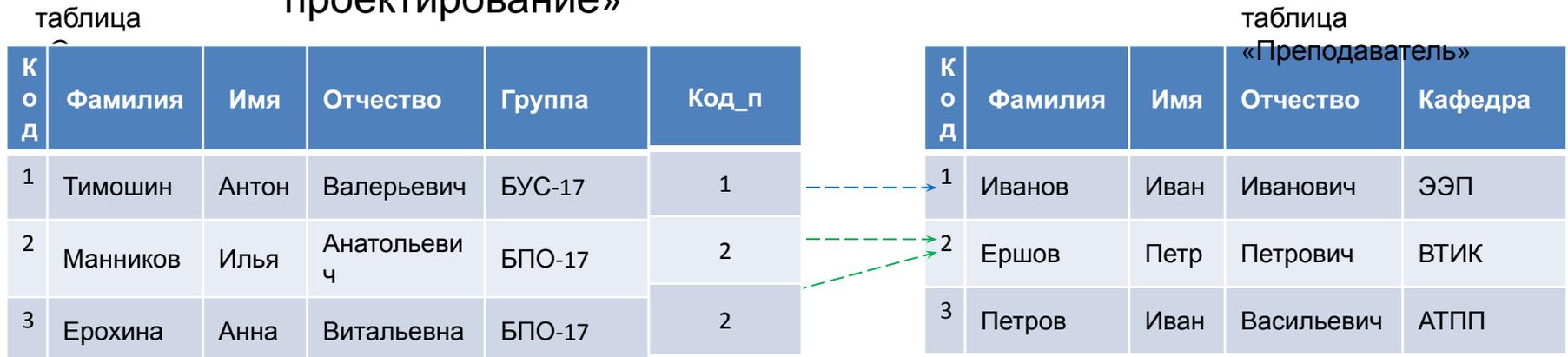
Задачи

1) Какой вид связи между таблицей 1 и таблицей 2?

Часть предметной области «Информация по парфюмерной продукции»



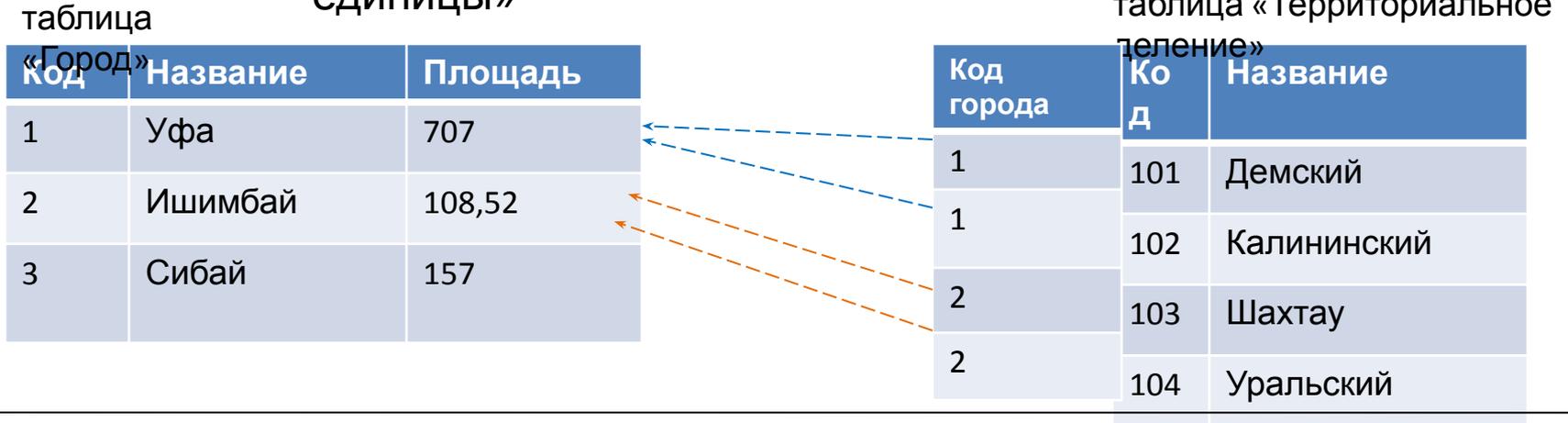
Часть предметной области «Дипломное проектирование»



Задачи

2) Какой вид связи между таблицей 1 и таблицей 2?

Часть предметной области «Административные единицы»



Часть предметной области «География»



Язык SQL



SQL

Structured Query Language

Язык SQL, предназначенный для взаимодействия с базами данных.

Первый стандарт (первоначально 1074 Американским

национальным институтом стандартов (ANSI) и одобрен действующим стандартом,

принятый в 2003 году (SQL:2003) с небольшими модификациями,

применяется свой диалект языка. Transact SQL применяется в Microsoft SQL



Команды SQL

CREATE, ALTER, DROP, TRUNCATE	Команды языка определения данных (DDL - Data Definition Language)
INSERT, UPDATE, DELETE, SELECT	Команды языка манипулирования данными (DML - Data Manipulation Language)
COMMIT, ROLLBACK, SAVEPOINT	Команды языка управления транзакциями (TCL - Transaction Control Language)
GRANT, REVOKE, DENY	Команды языка управления данными (DCL - Data Control Language)

Типы данных MS SQL Server

Числовые типы данных:

BIT: хранит значение 0 или 1. Фактически является аналогом булевого типа в языках программирования. Занимает 1 байт.

TINYINT: хранит числа от 0 до 255. Занимает 1 байт. Хорошо подходит для хранения небольших чисел.

SMALLINT: хранит числа от -32 768 до 32 767. Занимает 2 байта

INT: хранит числа от -2 147 483 648 до 2 147 483 647. Занимает 4 байта. Наиболее используемый тип для хранения чисел.

BIGINT: хранит очень большие числа от -9 223 372 036 854 775 808 до 9 223 372 036 854 775 807, которые занимают в памяти 8 байт.

DECIMAL(p , s)] и NUMERIC(p , s)]: числа с фиксированной точностью и масштабом. При использовании максимальной точности числа могут принимать значения в диапазоне от $-10^{38}+1$ до $10^{38}-1$.

Синонимами типа decimal по стандарту ISO являются типы dec и dec(p , s).

Тип numeric функционально эквивалентен типу decimal.

p (точность) - максимальное общее число хранимых десятичных разрядов. Это число включает символы слева и справа от десятичной запятой. Точность должна быть значением в диапазоне от 1 до максимум 38. Точность по умолчанию составляет 18.

s (масштаб) - максимальное число хранимых десятичных разрядов справа от десятичной запятой. Это число отнимается от p для определения максимального количества цифр слева от десятичной запятой. Масштаб должен иметь значение от 0 до p и может быть указан только при заданной точности. По умолчанию масштаб принимает значение 0, поэтому $0 \leq s \leq p$. Максимальный размер хранилища зависит от точности.

SMALLMONEY: хранит дробные значения от -214 748.3648 до 214 748.3647. Предназначено для хранения денежных величин. Занимает 4 байта. Эквивалентен типу DECIMAL(10,4).

MONEY: хранит дробные значения от -922 337 203 685 477.5808 до 922 337 203 685 477.5807.

Представляет денежные величины и занимает 8 байт. Эквивалентен типу DECIMAL(19,4).

FLOAT: хранит числа от $-1.79E+308$ до $1.79E+308$. Занимает от 4 до 8 байт в зависимости от дробной части.

Может иметь форму определения в виде FLOAT(n), где n представляет число бит, которые используются для хранения десятичной части числа (мантиссы). По умолчанию n = 53.

REAL: хранит числа от $-3.40E+38$ до $3.40E+38$. Занимает 4 байта. Эквивалентен типу FLOAT(24).

Типы данных MS SQL Server

Типы данных, представляющие дату и время:

DATE: ГГГГ-ММ-ДД. Хранит даты от 1 января 0001 года до 31 декабря 9999 года. Занимает 3 байта.

DATETIME: хранит даты и время от 01/01/1753 до 31/12/9999. Занимает 8 байт.

DATETIME2: ГГГГ-ММ-ДД чч:мм:сс[.доли секунды], хранит даты и время в диапазоне от 01/01/0001 00:00:00.0000000 до 31/12/9999 23:59:59.9999999.

Занимает от 6 до 8 байт в зависимости от точности времени.

Может иметь форму `DATETIME2(n)`, где `n` представляет количество цифр от 0 до 7 в дробной части секунд.

SMALLDATETIME: хранит даты и время в диапазоне от 01/01/1900 до 06/06/2079, то есть ближайшие даты.

Занимает от 4 байта.

DATETIMEOFFSET: хранит даты и время в диапазоне от 0001-01-01 до 9999-12-31. Сохраняет детальную информацию о времени с точностью до 100 наносекунд. Занимает 10 байт.

TIME: хранит время в диапазоне от 00:00:00.0000000 до 23:59:59.9999999. Занимает от 3 до 5 байт.

Может иметь форму `TIME(n)`, где `n` представляет количество цифр от 0 до 7 в дробной части секунд

Типы данных MS SQL Server

Строковые типы данных:

CHAR: хранит строку длиной от 1 до 8 000 символов. На каждый символ выделяет по 1 байту. Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

Количество символов, которое может хранить столбец, передается в скобках. Например, для столбца с типом CHAR(10) будет выделено 10 байт. И если мы сохраним в столбце строку менее 10 символов, то она будет дополнена пробелами.

VARCHAR: хранит строку. На каждый символ выделяется 1 байт. Можно указать конкретную длину для столбца - от 1 до 8 000 символов, например, VARCHAR(10). Если строка должна иметь больше 8000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб: VARCHAR(MAX).

Не подходит для многих языков, так как хранит символы не в кодировке Unicode.

В отличие от типа CHAR если в столбец с типом VARCHAR(10) будет сохранена строка в 5 символов, то в столбце будет сохранено именно пять символов.

NCHAR: хранит строку в кодировке Unicode длиной от 1 до 4 000 символов. На каждый символ выделяется 2 байта. Например, NCHAR(15)

NVARCHAR: хранит строку в кодировке Unicode. На каждый символ выделяется 2 байта.

Можно задать конкретный размер от 1 до 4 000 символов: . Если строка должна иметь больше 4000 символов, то задается размер MAX, а на хранение строки может выделяться до 2 Гб.

Еще два типа **TEXT** и **NTEXT** являются устаревшими и поэтому их не рекомендуется использовать. Вместо них применяются VARCHAR и NVARCHAR соответственно.

Типы данных MS SQL Server

Бинарные типы данных:

BINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт.

VARBINARY: хранит бинарные данные в виде последовательности от 1 до 8 000 байт, либо до $2^{31}-1$ байт при использовании значения MAX (VARBINARY(MAX)).

Еще один бинарный тип - тип IMAGE является устаревшим, и вместо него рекомендуется применять тип VARBINARY.

Остальные типы данных:

UNIQUEIDENTIFIER: уникальный идентификатор GUID (по сути строка с уникальным значением), который занимает 16 байт.

TIMESTAMP: некоторое число, которое хранит номер версии строки в таблице. Занимает 8 байт.

CURSOR: представляет набор строк.

HIERARCHYID: представляет позицию в иерархии.

SQL_VARIANT: может хранить данные любого другого типа данных T-SQL.

XML: хранит документы XML или фрагменты документов XML. Занимает в памяти до 2 Гб.

TABLE: представляет определение таблицы.

GEOGRAPHY: хранит географические данные, такие как широта и долгота.

GEOMETRY: хранит координаты местонахождения на плоскости.

Службы Microsoft SQL Server

Наименование службы	Назначение
Database Engine	Управление реляционными БД
Analysis Services	Управление OLAP – кубами и интеллектуальный анализ данных
Integration Services	Поддержка решений по извлечению, преобразованию и загрузке данных
Reporting Services	Управление отчетами, построенными на основе SQL – запросов к реляционным БД
Full-Text Search	Управление полнотекстовым поиском
SQL Server Agent	Автоматизация административных задач
SQL Server Browser	Управление соединениями

Database Engine

Database Engine является ядром системы управления реляционной БД.

Может быть установлено несколько экземпляров службы Database Engine.

Один экземпляр Database Engine может быть службой по умолчанию (с именем MS SQL SERVER), другие экземпляры должны иметь уникальные имена.

Каждый экземпляр службы Database Engine требует отдельной инсталляции, конфигурации и настройки безопасности.

Один Database Engine может обеспечить доступ к нескольким БД.

Системные базы данных

Системная БД	Назначение
master	Хранит все системные данные Database Engine, а также информацию о других БД
msdb	Используется службами SQL Server Agent (выполнение заданий по расписанию), Database Mail (формирование уведомлений по электронной почте), а также хранит информацию о резервном копировании БД
tempdb	Пространство для временных объектов Database Engine и пользовательских временных таблиц. БД пересоздается при каждой перезагрузке
model	Шаблон, используемый при создании всех БД, управляемых экземпляром Database Engine
resource	БД, используемая только для чтения. Содержит системные объекты экземпляра Database Engine. Файлы БД являются скрытыми и не отображаются в MS SQL Server

УТИЛИТЫ Microsoft SQL Server:

- SQL Server Management Studio.
- SQL Server Books Online.
- SQLCMD Microsoft.
- SQL Configuration Manager.

Система управления базами данных

Основные функции СУБД:

- 1) управление данными во внешней памяти (на дисках);
- 2) управление данными в оперативной памяти с использованием дискового кэш;
- 3) журнализация изменений, резервное копирование и восстановление базы данных после сбоев;
- 4) поддержка языков БД (язык определения данных, язык манипулирования данными).

Система управления базами данных

Обычно современная СУБД содержит следующие компоненты:

- 1) ядро, которое отвечает за управление данными во внешней и оперативной памяти и журнализацию;
- 2) процессор языка базы данных, обеспечивающий оптимизацию запросов на извлечение и изменение данных и создание, как правило, машинно-независимого исполняемого внутреннего кода;
- 3) подсистему поддержки времени исполнения, которая интерпретирует программы манипуляции данными, создающие пользовательский интерфейс с СУБД;
- 4) сервисные программы (внешние утилиты), обеспечивающие ряд дополнительных возможностей по обслуживанию информационной системы.

Источники информации

- К.Дж. Дейт. Введение в системы баз данных. Восьмое издание. – М.: Вильямс, 2005. – 1328 С.
- Голицина О.Л. Базы данных. - Изд. «ФОРУМ», 2009. - 400 с.
- Дунаев В.В. Базы данных. Язык SQL для студента. - СПб.: БХВ-Петербург, 2007. - 320 с.
- Советов Б.Я. Базы данных. – Изд. «Высшая школа», 2007. - 463 с.
- Харрингтон Д. Проектирование объектно-ориентированных баз данных. Год 2007. "Лань" Электронная библиотечная система
- *К.Ю. Поляков и т.д.)*