ANDROID LEVEL_2

ПРОГРАММА КУРСА

- Урок 1. База данных SQLite
- Урок 2. Интент-фильтры и контент-провайдеры
- Урок 3. Позиционирование и карты
- Урок 4. Многопоточность в Android. Сервисы
- Урок 5. Broadcast Receivers. Виджеты
- Урок 6. Телефония, сенсоры, bluetooth
- Урок 7. Компоненты UI. Графика. Анимация
- Урок 8. Отладка приложений

УРОК 1. БАЗА ДАННЫХ

-Введение в базы данных
-Основные понятия и определения
-Реляционная модель данных
-Примеры таблиц базы данных
-Основы языка SQL
-Назначение, синтаксис, основные конструкции

УРОК 1. SQLITE В ANDROID

- Основные возможности и сферы применения
- Краткий обзор пакетов и классов
- Класс SQLiteOpenHelper
- Жизненный цикл базы данных
- Примеры приложения с базой данных

ЧТО ТАКОЕ БД?

База данных — представленная в объективной форме совокупность самостоятельных материалов, систематизированных таким образом, чтобы эти материалы могли быть найдены и обработаны с помощью электронной вычислительной машины (ЭВМ).

База данных — совокупность данных, организованных в соответствии с концептуальной структурой, описывающей характеристики этих данных и взаимоотношения между ними.

ЧТО ТАКОЕ СУБД?

Система управления базами данных (СУБД) — совокупность программных и лингвистических средств общего или специального назначения, обеспечивающих управление созданием и использованием баз данных.

КЛАССИФИКАЦИИ СУБД

Примеры:

- **Иерархические**
- **Сетевые**
- **Реляционные**
- **Объектно-ориентированные**
- **Объектно-реляционные**

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Термин «реляционный» – означает, что теория основана на математическом понятии отношение (***relation***).

Реляционная модель данных (РМД) - логическая модель данных, прикладная теория построения баз данных, которая является приложением к задачам обработки данных таких разделов математики, как теория множеств и логика первого порядка.

На реляционной модели данных строятся реляционные базы данных.

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Relation(отношение) – набор кортежей, каждый элемент в котором является членом определенного домена данных.

Домен – допустимое множество значений. Условия по набору данных. На данный момент удобно стало использовать в качестве доменов типы данных, к которому относится атрибут.

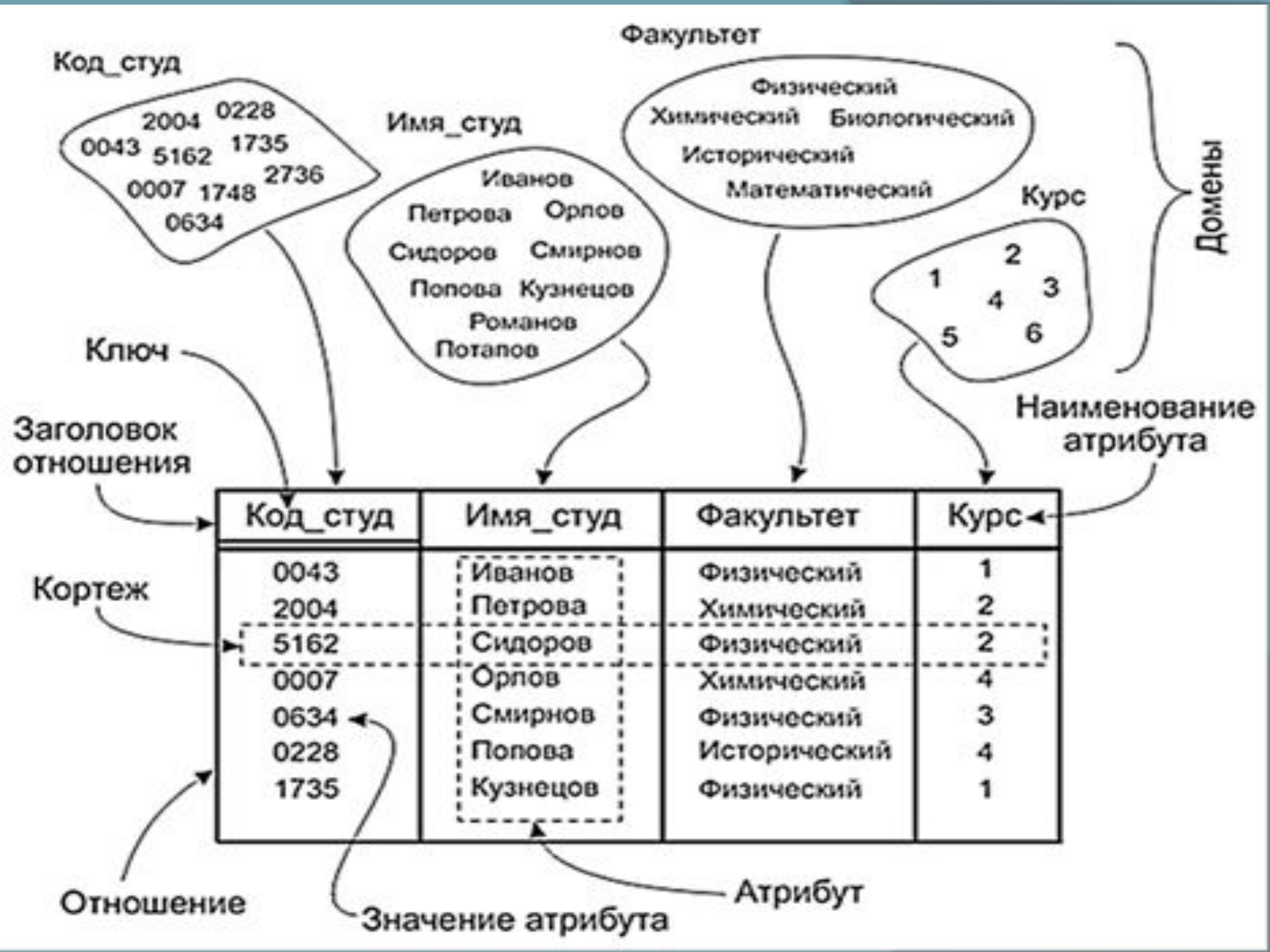
Атрибут – характеристика, описывающая «логический» тип данных.

Кортеж – упорядоченный набор значений, по одному для каждого атрибута.

Отношения – набор кортежей. Хорошо представляются в графическом виде, как таблица, где

Кортежи – строки таблицы.

Атрибуты – столбцы.


Код_студ
2004 0228
0043 5162 1735
0007 1748 2736
0634
Имя_студ
Иванов
Петрова Орлов
Сидоров Смирнов
Попова Кузнецов
Романов
Потапов
Факультет
Физический
Химический Биологический
Исторический
Математический
Курс
1 2 3
4
5 6
Домены
Ключ
Заголовок отношения
Наименование атрибута
Код_студ
Имя_студ
Факультет
Курс
0043 Иванов Физический 1
2004 Петрова Химический 2
5162 Сидоров Физический 2
0007 Орлов Химический 4
0634 Смирнов Физический 3
0228 Попова Исторический 4
1735 Кузнецов Физический 1
Кортеж
Отношение
Значение атрибута
Атрибут

РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

- Таблица – данные о всех экземплярах.
- Строка – логический связанный набор атрибутов одного экземпляра объекта.
- Все экземпляры описываются одинаковым набором атрибутов.
- Атрибуты имеют уникальные имена.

СВОЙСТВА ТАБЛИЦ

- Порядок столбцов неважен
- Столбцы не зависят друг от друга
- Данные столбца имеют одинаковый тип

Поле

Запись

Номер	ФИО	Должность	Телефон
001	Иванов П.С.	Директор	123-12-12
002	Сидоров В.Н.	Водитель	234-23-23
003	Петров А.А.	Продавец	345-34-34
004	Ромашкова Е.А.	Продавец	456-45-45
005	Умялова Л.С.	Бухгалтер	567-56-56
006	Черенин С.В.	Зав. складом	678-67-67

ЧТО ТАКОЕ SQL?

SQL Structured Query Language - Структурированный язык запросов - язык управления базами данных для реляционных баз данных.

SQL состоит из четырех отдельных частей:

- язык определения данных (DDL)
- язык манипуляции данными (DML)
- язык определения доступа к данным (DCL)
- язык управления транзакциями (TCL)

ЧТО ТАКОЕ SQL?

Основные запросы (для работы с данными)

- **SELECT – извлечение данных из БД**
- **UPDATE – обновление данных в БД**
- **DELETE – удаление данных из БД**
- **INSERT – добавление данных в БД**

ЧТО ТАКОЕ КУРСОР (CURSOR)?

- Запрос к базе данных (SELECT) возвращает некий набор данных (записей) – «виртуальную таблицу» (результирующий набор данных)
- В большинстве случаев приложение обрабатывает не весь этот набор данных целиком, а единичные записи
- При этом существует необходимость навигации по результирующему набору данных для выборки очередной записи
- Cursor – это получаемый при выполнении запроса результирующий набор данных и связанный с ним указатель текущей записи

ВОПРОСЫ ?

ЧТО ТАКОЕ SQLITE?

• SQLite – это встраиваемая кроссплатформенная БД с открытым исходным кодом, которая поддерживает достаточно полный набор команд SQL

• SQLite не использует парадигму клиент-сервер, в качестве протокола взаимодействия БД и приложения используются вызовы функций (API) библиотеки SQLite

• SQLite хранит всю базу данных (включая определения, таблицы, индексы и данные) в одном файле

ЧТО ТАКОЕ SQLITE?

Начнем работу из под консоли.
https://www.sqlite.org/

Удобный браузер для работы с SQLite
http://sqlitebrowser.org

SQLITE В ANDROID

- Android предоставляет полную поддержку базы данных SQLite
- Базы данных в Android в основном используются для хранения повторяющихся и структурированных данных, таких как контактная информация, данные пользователя (заметки, списки дел, закладки) и тд
- Все базы данных SQLite созданные в приложении хранятся в защищенной (внутренней) области памяти приложения (на диске) и доступны только этому приложению

SQLITE В ANDROID

- Для работы с SQLite в Android существует определенный набор классов (пакет android.database.sqlite)
- SQLiteDatabase – используется для управления базой данных SQLite. Содержит методы для создания и удаления БД, управления транзакциями, методы для выполнения SQL запросов и работы с данными (создание, удаление, изменение)
- SQLiteCursor – реализация курсора для обработки результатов запросов к БД SQLite (SQLiteDatabase)
- SQLiteOpenHelper – класс помощник (helper) для управления созданием базы данных и версионностью

SQLITEOPENHELPER

- SQLiteOpenHelper – абстрактный класс, реализующий API для взаимодействия с базой данных SQLite
- Конструктор передает Фреймворку необходимую информацию (имя БД и версию)
- **onCreate** – вызывается, когда база данных создается впервые. Этот метод должен создавать необходимые таблицы и заполнять их начальными данными (если это необходимо)
- **onUpgrade** – вызывается, когда необходимо обновить базу данных. Обычно содержит методы для изменения структуры БД

INSERT

- insert(String table, String nullColumnHack, ContentValues values)
- table – имя таблицы для вставки записи
- nullColumnHack – SQL не позволяет добавлять пустую запись в таблицу не указав хотя бы одно имя столбца, если нужно добавить пустую запись, в этом параметре необходимо указать имя столбца, в которое будет помещено значение NULL
- values – имена столбцов (атрибутов) и их значения.

UPDATE

- update(String table, ContentValues values, String whereClause, String[] whereArgs)
- table – имя таблицы для обновления записи
- values - имена столбцов (атрибутов) и их значения
- whereClause – выражение SQL WHERE (условие), если равно null, будут обновлены все записи
- whereArgs – аргументы для выражения SQL WHERE

DELETE

- delete(String table, String whereClause, String[] whereArgs)
- table – имя таблицы для обновления записи
- whereClause – выражение SQL WHERE (условие), если равно null, будут удалены все записи
- whereArgs – аргументы для выражения SQL WHERE

QUERY

- **query(String table, String[] columns, String selection, String[] selectionArgs, String groupBy, String having, String orderBy)**
  - **columns** – список полей, которые мы хотим получить
    selection – строка условия WHERE
    selectionArgs – массив аргументов для selection. В selection можно использовать знаки *?* , а которые будут заменены этими значениями.
    groupBy - группировка
    having – использование условий для агрегатных функций
    orderBy - сортировка

ДОМАШНЕЕ ЗАДАНИЕ

- **Разобрать все примеры урока**
- **Написать приложения (минимум одно)**
  - **Список дел**
  - **Список покупок**

Упростить модель БД и вывести значения в виде списка.

Дополнительное задание*: Использовать модель DML, для динамического изменения элементов списка (вставка, удаление, обновление).

** **При создании БД, подгружать данные из .xml**