**Лекция 1**

*База данных* – совокупность данных, организованная по определенным правилам

*СУБД (система управления базами данных)* – программа для создания базы данных и работы с ее данными

*Функции СУБД* – управление данными, транзакциями, журнализация.

*Свойства бд* – быстродействие, независимость данных, безопасность, простота обновления данных

*Уровни восприятия данных в бд* – внешний(пользователи), логический(программисты, администраторы), внутренний(файловая система, системные программисты)

*Уровни представляения инфы в бд* – внешние(word документ), концептуальные(таблицы), физические(файлы)

*Виды баз данных* – по технология обработки данных -централизованные и распределенные, по способа доступа к данным – с локальным и удаленным доступом

*Преимущество централизованных бд* – однократное хранение каждого сообщения в бд, что сокращает избыточность

*Этапы проектирования бд* – анализ предметной области, описание модели, логическое проектирование модели, физическое проектирование бд.

*Уровни представления данных* – концептуальный (сущности, атрибуты, связи), логический(записи, элементы данных, связи между записями), физический(группирование данных, индексы, методы доступа)

*Концепция бд позволяет* – повысить надежность данных, сократить затраты, обеспечить легкость использования данных и скорость доступа к ним.

**Лекция 2**

*Реляционная бд* – связанная между собой совокупность таблиц, где (таблица – совокупность строк и столбцов, строки – экземпляры объекта, столбцы – параметры объекта).

*Понятия реляционной модели данных* – типы данных(целое, строки), домены(номер, имя, дата – название групп столбцов), атрибуты(номер телефона, номер карточки, имя человека – названия конкретных столбцов), кортежи(строки таблицы), отношение – таблица(атрибуты и кортежи), ключ – атрибут, по которому производится поиск.

*Свойства реляционных таблиц* - каждый элемент таблицы - один элемент данных, все элементы в столбце имеют одинаковый тип данных, каждый столбец имеет уникальное имя, в таблице нет одинаковых строк, порядок следования строк и столбцов может быть любым.

*Первичный ключ* – поле или набор полей по которому ищется запись в таблице. Значение этого поля должно быть уникальным для каждой строки в таблице.

*Связи между таблицами в бд* – “один ко многим” – одной записи в родительской таблице может соответсвовать много записей в дочерней, “один к одному”, одной записи в родительской соотв. одна запись в дочерней, “многие ко многим”, много записей в родительской могут соответсвовать многим записям в дочерней таблице

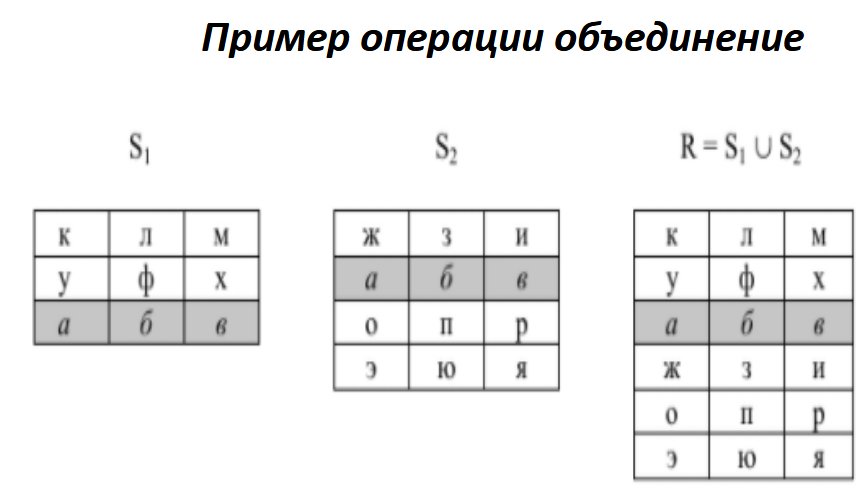
*Внешний ключ* – поле для связи дочерней таблицы с родительской(создается в родительской).

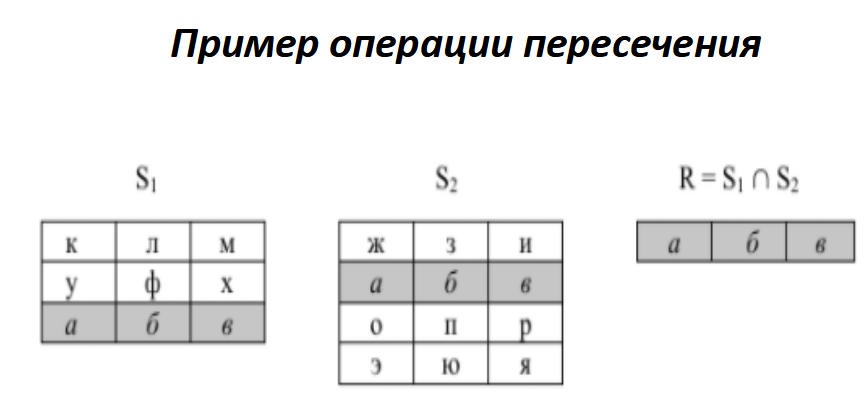
*Методы доступа к данным в бд* – последовательный(просмотр всех записей таблицы), инексно-последовательный(поиск ведется только по первой строке индекса), оба идут последовательно

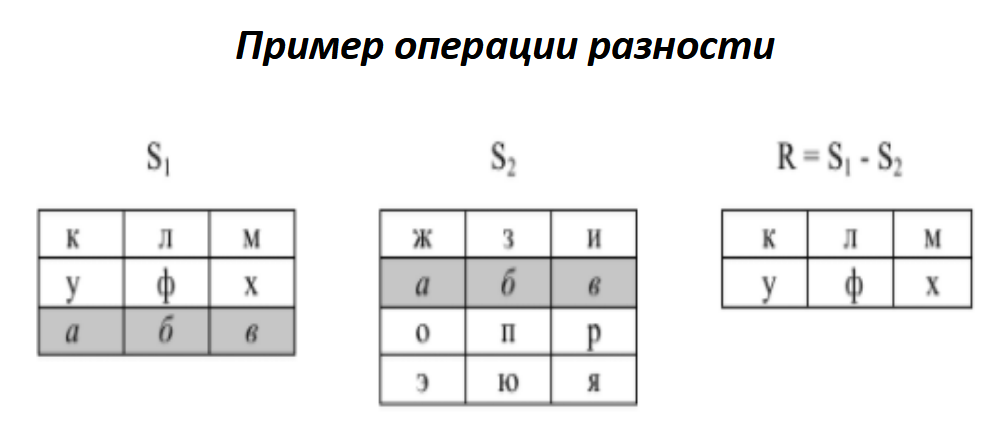
*Индексы* – механизм быстрого доступа к данным в таблицах (для каждой записи создается индекс, из них составляются B, R и прочие деревья или хеш таблицы, плюсы – быстрее поиск, минусы – дольше добавлять данные, нужно соблюдать порядок индексов)

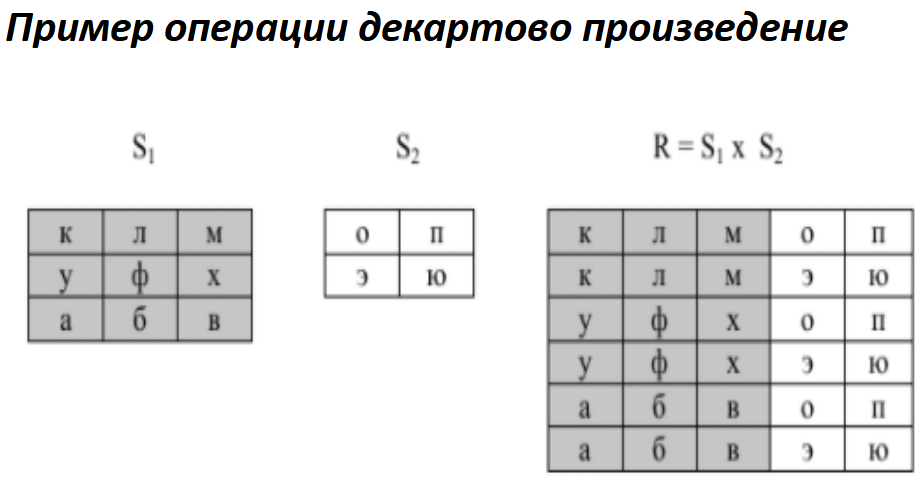
*Реляционная алгебра* – алгебра таблиц

*Операции реляционной алгебры* – **теоретико-множественные**(**объединение**(ИЛИ - данные одной таблицы вставляются в конец другой), **пересечение**(И - остаются только одинаковые строки для обеих таблиц), **разность**(И - остаются только неодинаковые строки из обеих таблиц), **декартово произведение**(И – справа от каждой строки из таблицы1 пишутся все строки из таблицы2 поочереди) – обе таблицы, над которыми производятся эти операции должны иметь одинаковое название столбцов),

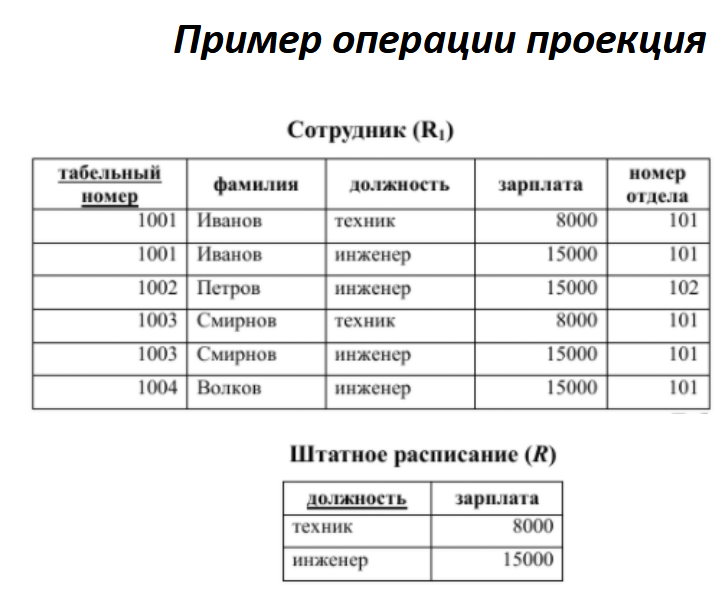


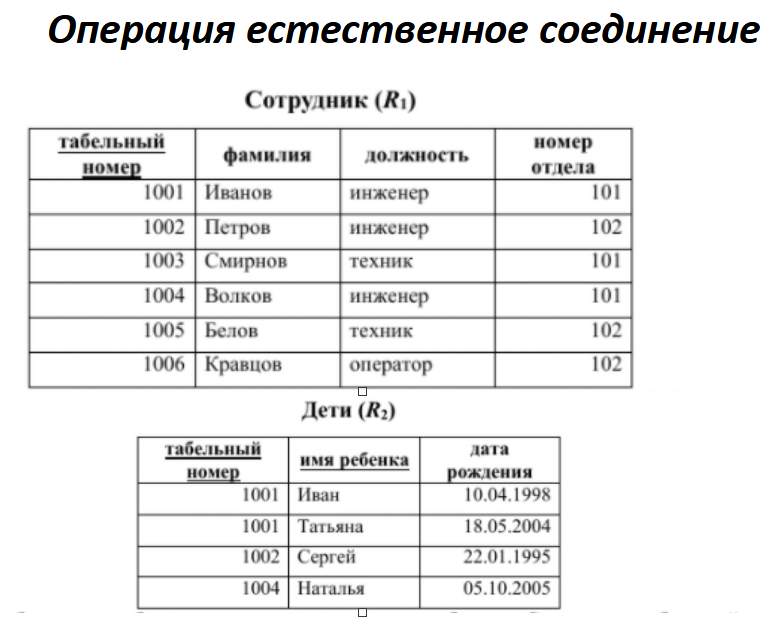






испециальные(выборка(WHERE user\_age>19... таблица, состоящая из строк, для которых истинно какое-то условие), проекция(Из таблцы убирают ненужные нам столбцы, например из таблцы [имя, возраст, пол] получим только[имя, пол]), естественное соединение(JOIN, соеденение таблиц по внешнему ключу, например, id\_родителя = 10, id\_ребенка этого родителя(двое детей) равно 5, 6, будет таблица с записями (пишу столбикик в строчку) {10, 5}, {10, 6}, ...), соединение по условию(JOIN WHERE...) деление(Таблица1 без стобцов таблицы2, записи которой полностью соостветствуют всем или некоторым записям из таблицы2 )





****

В результат деления - (R) попали те чуваки из таблицы R1, у которых есть все проекты из таблицы R2

**Лекция 3**

*Задачи инфологического проектирования* - определение и анализ предметной области системы, описание структуры и динамики предметной области, моделирование предметной области.

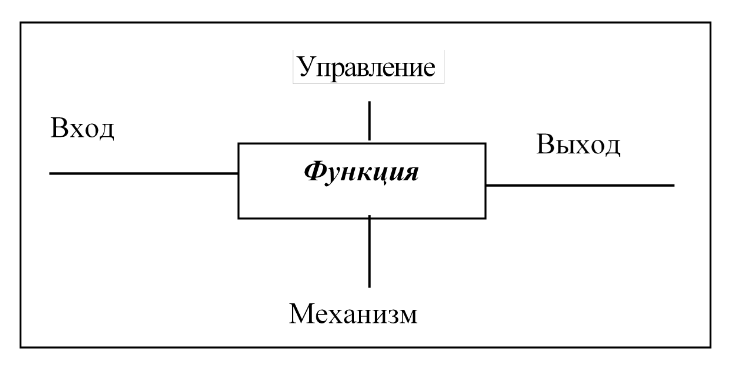
*Подходы к выбору структуры предметной области* – функциональный(известны функции пользователей для которых создается бд), предметный(предполагаются объекты и взаимосвязи между ними, используется при решении заранее не определенных задач)

*Отличия структурного и объектно-ориентированного подхода* - структурный подход - за основу принимается функциональная (алгоритмическая) декомпозиция, объектно-ориентированный подход – объектная декомпозиция.

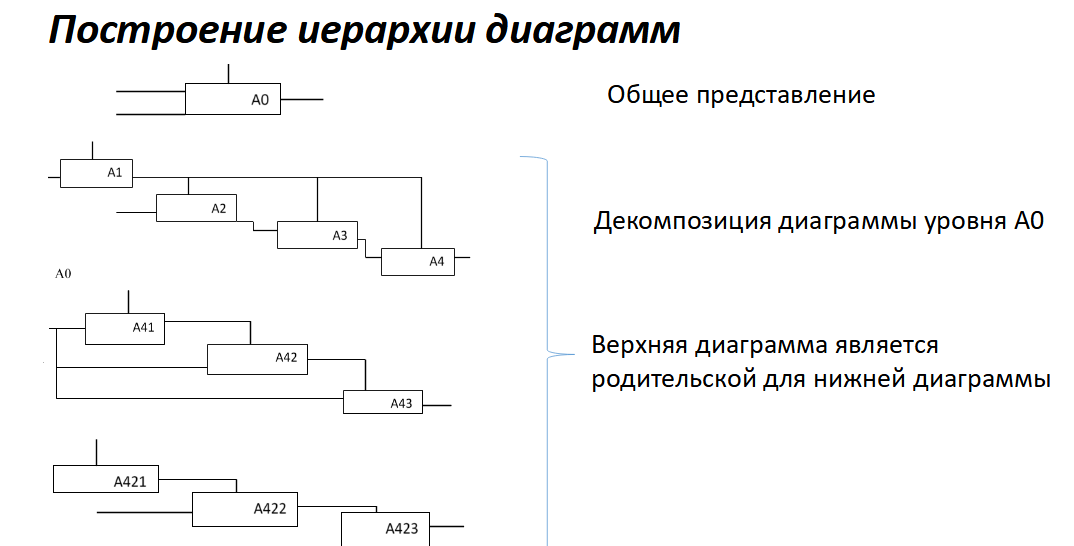
*Функциональный(структурный) подход:*



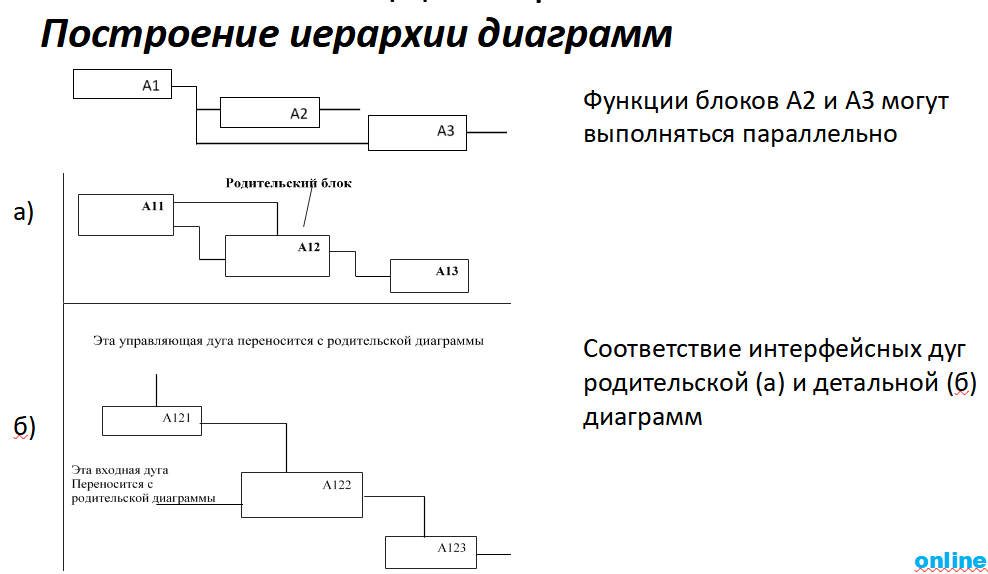
*SADT (structured analysis and design technique)* — методология структурного анализа и проектирования. На выходе у нее Функциональная модель, вот такая:



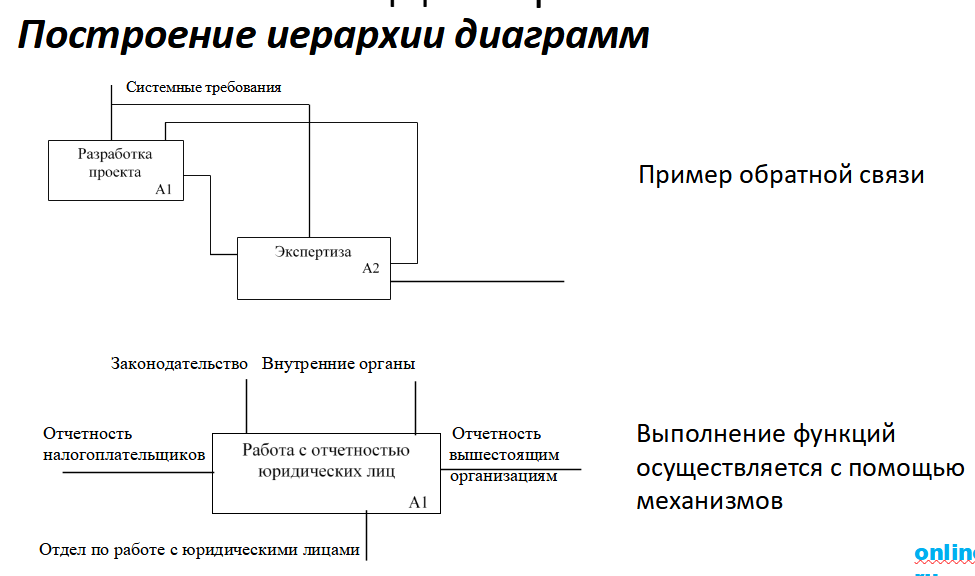
*Построение иерархии диаграмм функциональной модели:*



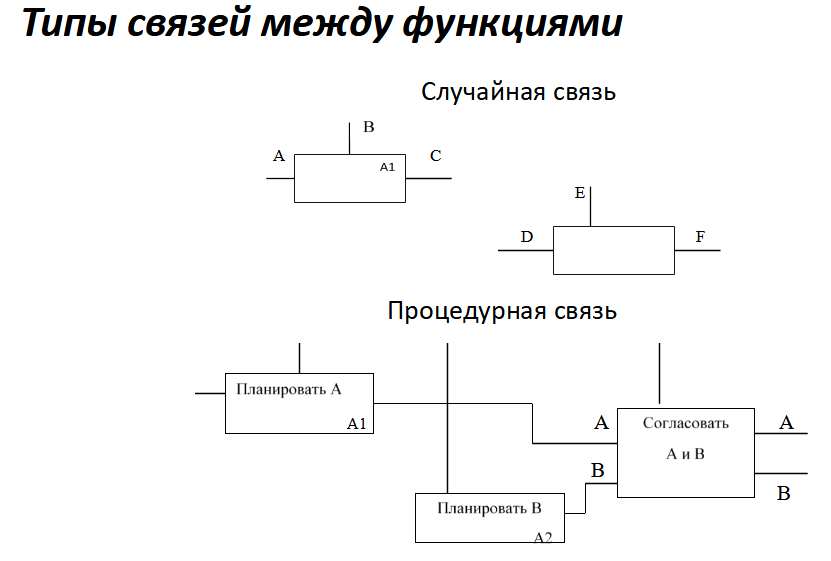
И вот так там соединяются палки:

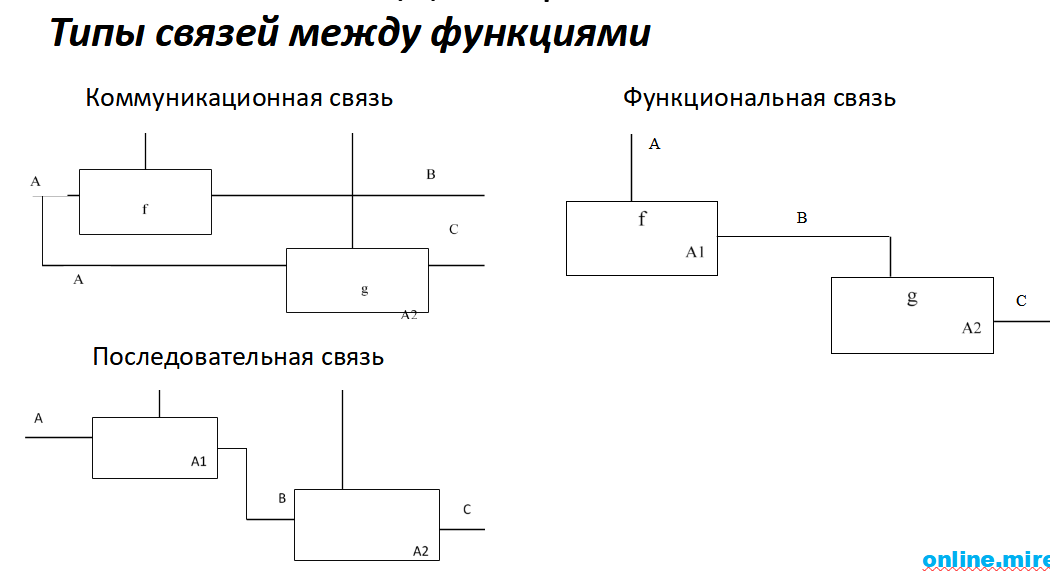
****

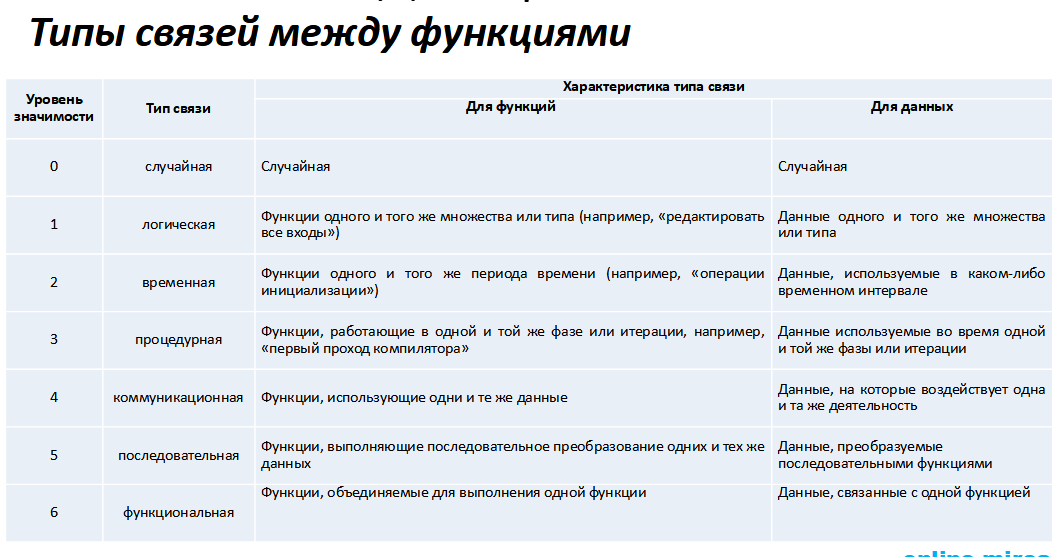
И вот так работают эти штуки “сверху” и “снизу” от которых мы тянем стрелки:



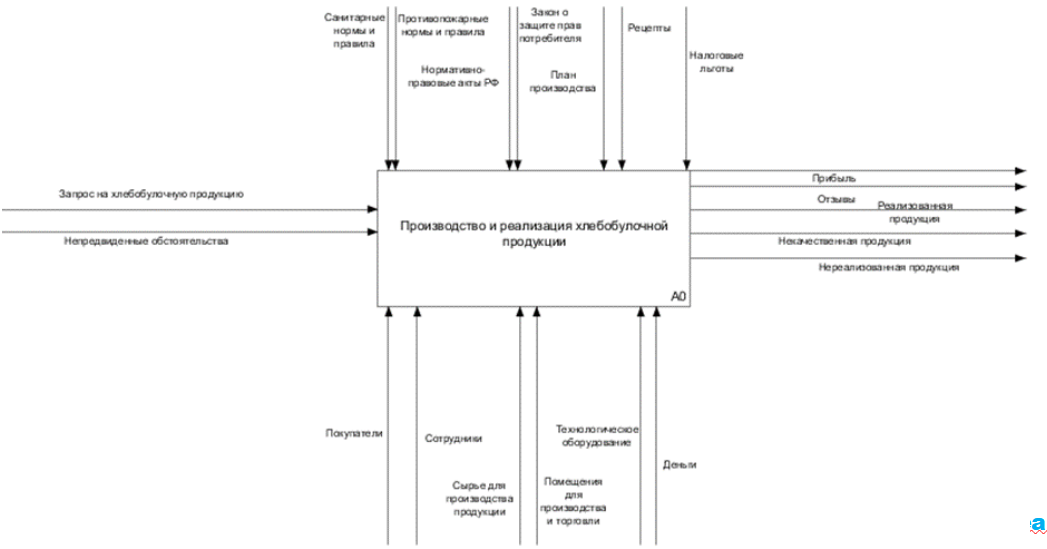
*Типы связей между функциями:*

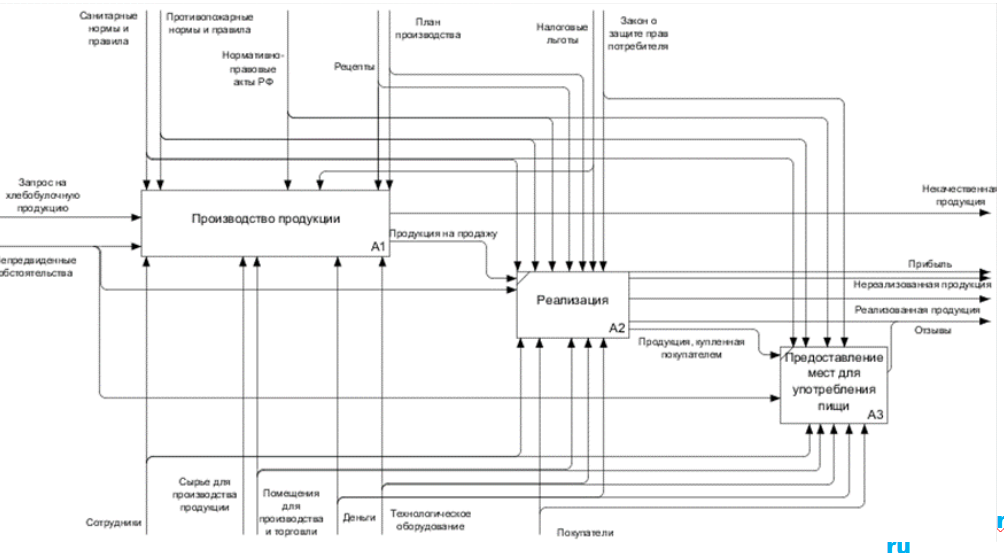




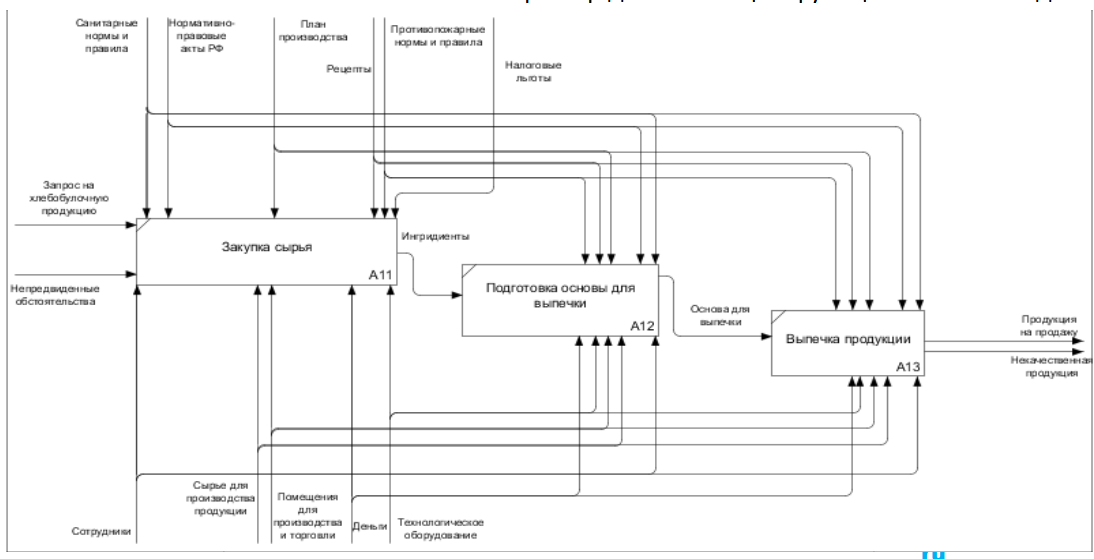


*Пример функциона модели:*

*льной*

*Пример декомпозиции функциональной модели:*

*Пример декомпозиции функциональной модели (еще глубже):*

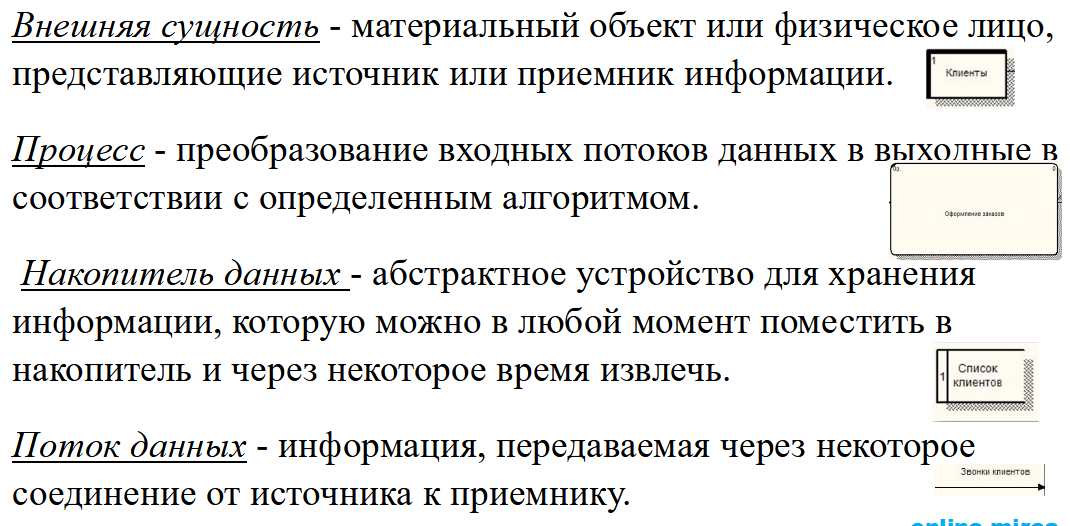
**

**Лекция 4**

*Диаграммы потоков данных (DFD)* - средство моделирования функциональных требований к проектируемой базе данных.

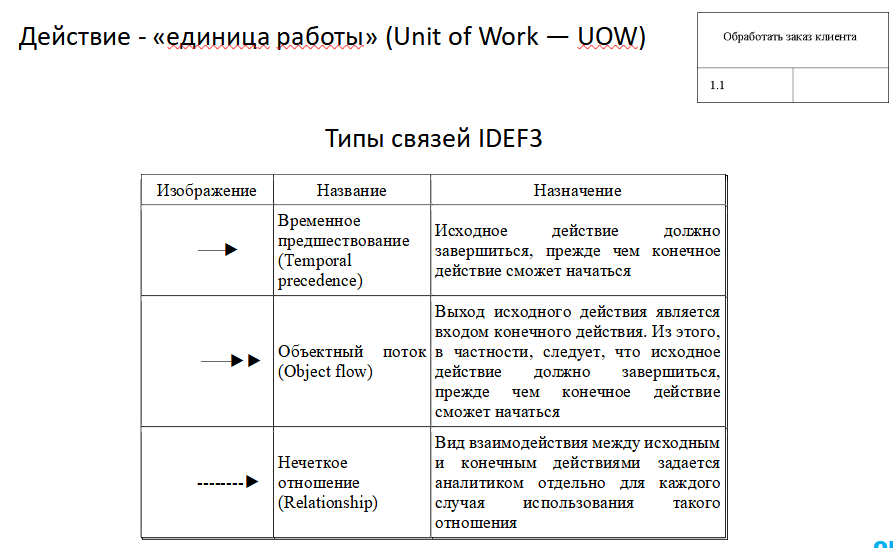
*Цель DFD* - продемонстрировать, как каждый процесс превращает свои входные данные в выходные и показать отношения между процессами.

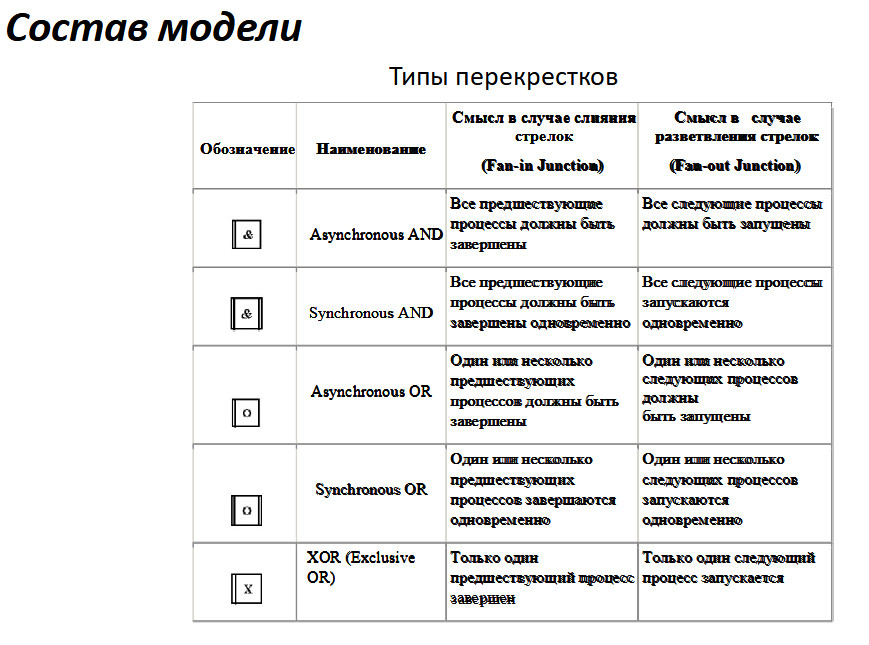
*Состав диаграмм потоков данных*: внешние сущности, системы и подсистемы, процессы, накопители данных, потоки данных.

**

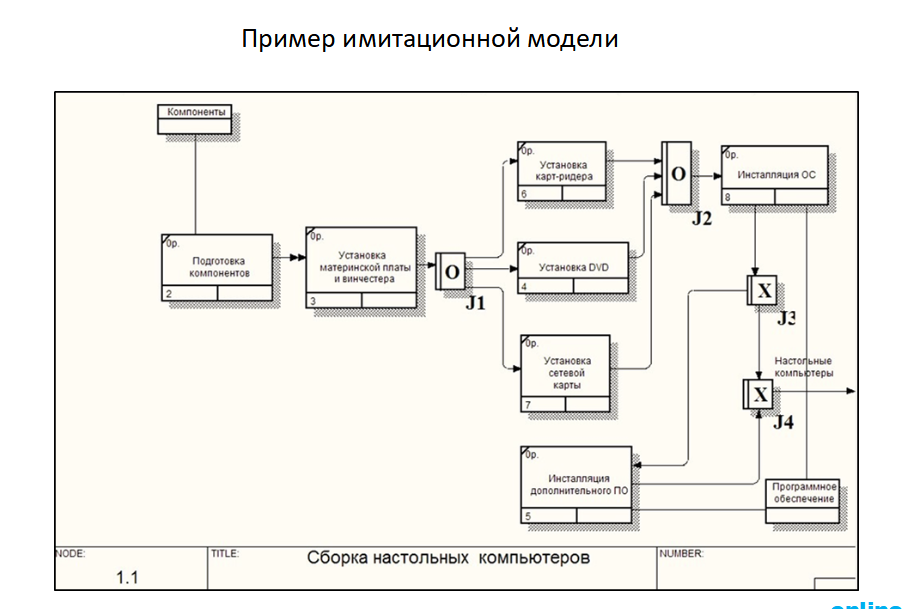
**

*Диаграмма последовательностей экранных форм (IDEF3)* - абстрактная модель пользовательского интерфейса, отражающая последовательность появления экранных форм в приложении.

**

**

*Пример IDEF3:*

**

**Лекция 5**

*Логическое проектирование БД* - процесс разработки корректной схемы реляционной БД.

*Корректная схема БД* - в которой отсутствуют нежелательные зависимости между атрибутами отношений (столбцами таблиц).

*Этап логического или даталогического проектирования*: описание концептуальной схемы БД в терминах выбранной СУБД, описание внешних моделей в терминах выбранной СУБД, описание декларативных правил поддержки целостности базы данных, разработка процедур поддержки семантической целостности БД.

*Проектирование схемы БД может быть выполнено*: **путём декомпозиции** (разбиения), когда исходное множество отношений, входящих в схему БД заменяется другим множеством отношений, являющихся проекциями исходных отношений; **путем синтеза**, т.е. путем компоновки из заданных исходных элементарных зависимостей между объектами предметной области схемы БД.

*Процесс проектирования с использованием декомпозиции* - процесс последовательной нормализации схем отношений, при этом каждая последующая итерация соответствует нормальной форме более высокого уровня и обладает лучшими свойствами по сравнению с предыдущей.

*Декомпозиция должна сохранять эквивалентность схем БД при замене одной схемы на другую -* Схемы БД называются эквивалентными, если содержание исходной БД может быть получено путем естественного соединения отношений, входящих в результирующую схему, и при этом не появляется новых кортежей в исходной БД.

*Основные свойства нормальных форм*:

каждая следующая нормальная форма улучшает свойства предыдущей;

при переходе к следующей нормальной форме свойства предыдущих нормальных форм сохраняются.

Следующий слайд, который я еще не разбирал – 8. И 5-ую лекцию не пытался писать своими словами, ее стоит упросить.