

Лекция 15

«Моделирование данных»

Часть 1

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

Терминология: базы данных

ГОСТ 34.321-96 Информационные технологии (ИТ). Система стандартов по базам данных. Эталонная модель управления данными

2.1 база данных (database):

Совокупность взаимосвязанных данных, организованных в соответствии со схемой базы данных таким образом, чтобы с ними мог работать пользователь

2.7 данные (data):

Информация, представленная в формализованном виде, пригодном для передачи, интерпретации или обработки с участием человека или автоматическими средствами

2.53 схема базы данных (database schema):

Формальное описание данных в соответствии с конкретной схемой данных.

2.54 схема данных (data schema):

Логическое представление организации данных.

2.55 тип данных (data type):

Поименованная совокупность данных с общими свойствами

2.47 система управления базами данных (database management system):

Совокупность программных и языковых средств, обеспечивающих управление базами данных

Терминология: структуры данных

ГОСТ 34.003-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

2.7 программное обеспечение автоматизированной системы

Совокупность программ на носителях данных и программных документов, предназначенная для отладки, функционирования и проверки работоспособности АС

Структура данных ([англ. data structure](#)) — программная единица, позволяющая хранить и обрабатывать множество однотипных и/или логически связанных [данных](#) в [вычислительной технике](#)

Термин «структура данных» может иметь несколько близких, но тем не менее различных значений:

- [Абстрактный тип данных](#)
- Реализация какого-либо абстрактного типа данных
- Экземпляр типа данных, например, конкретный [список](#)
- В контексте функционального программирования — уникальная единица ([англ. unique identity](#)), сохраняющаяся при изменениях

Структуры данных формируются с помощью [типов данных](#), [ссылок](#) и операций над ними в выбранном [языке программирования](#).

Терминология: информационная база

ГОСТ 34.003-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

2.8 информационное обеспечение автоматизированной системы

Совокупность форм документов, классификаторов, нормативной базы и реализованных решений по объемам, размещению и формам существования информации, применяемой в АС при ее функционировании

2.19 информационная база автоматизированной системы; информационная база АС:

Совокупность упорядоченной информации, используемой при функционировании АС

2.20 немашинная информационная база автоматизированной системы;

немашинная информационная база АС: Часть информационной базы АС, представляющая собой совокупность документов, предназначенных для непосредственного восприятия человеком без применения средств вычислительной техники

2.21 машинная информационная база автоматизированной системы; машинная

информационная база АС: Часть информационной базы АС, представляющая собой совокупность используемой в АС информации на носителях данных

Терминология: информация в АС

ГОСТ 34.003-90 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Термины и определения

6.4 входная информация автоматизированной системы; входная информация АС: Информация, поступающая в АС в виде документов, сообщений, данных, сигналов, необходимая для выполнения функций АС

6.5 выходная информация автоматизированной системы; выходная информация АС: Информация, получаемая в результате выполнения функций АС и выдаваемая на объект ее деятельности, пользователю или в другие системы

6.6 оперативная информация автоматизированной системы; оперативная информация АС: Информация, отражающая на данный момент времени состояние объекта, на который направлена деятельность АС

6.7 нормативно-справочная информация автоматизированной системы; нормативно-справочная информация АС: Информация, заимствованная из нормативных документов и справочников и используемая при функционировании АС

Терминология: независимость программ от данных

Одним из главных вопросов разработки программного обеспечения ИС (и программирования как самостоятельной дисциплины) является вопрос о соотношении программ и данных, ибо решение этого вопроса, в конечном счете, определяет выбор алгоритмов обработки информации, аппаратных средств и технологической платформы.

Фундаментальным принципом в решении вопроса о соотношении программ и данных является концепция независимости прикладных программ от данных. Суть этой концепции состоит не столько в отделении программ от данных, сколько в рассмотрении их как самостоятельных взаимодействующих объектов

Торжество концепции независимости программ от данных привело к формированию в 1962 году концепции базы данных (БД) и созданию на ее основе метода баз данных для решения задач обработки информации.

До середины 60-х годов прошлого века основной концепцией построения программного обеспечения являлась концепция файловой системы и так называемый позадачный метод. Такой подход по-прежнему остается доминирующим в разработке и функционировании несущих операционных платформ.

В конце 80-х годов прошлого века была предложена концепция объектно-ориентированных баз данных и объектно-ориентированный подход разработки программ на основе обработки событий.

Терминология: DDL

Data Definition Language (DDL) (язык описания данных) — это семейство компьютерных языков, используемых в компьютерных программах для описания структуры баз данных. На текущий момент наиболее популярным языком DDL является SQL, используемый для получения и манипулирования данными в РСУБД, и сочетающий в себе элементы DDL, DML и DCL.

Функции языков DDL определяются первым словом в предложении (часто называемом **запросом**), которое почти всегда является глаголом.

В случае с SQL это глаголы — «create» («создать»), «alter» («изменить»), «drop» («удалить»). Эти запросы или команды часто смешиваются с другими командами SQL, в связи с чем DDL не является отдельным компьютерным языком.

Запрос «create» используется для создания базы данных, таблицы, индекса, представления или хранимой процедуры. Запрос «alter» используется для изменения существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. Запрос «drop» используется для удаления существующего объекта базы данных (таблицы, индекса, представления или хранимой процедуры) или самой базы данных. И наконец, в DDL существуют понятия первичного и внешнего ключа, которые осуществляют соблюдение целостности данных. Команды "первичный ключ" primary key, "внешний ключ" foreign key включаются в запросы «create table", «alter table

Терминология: DML

Data Manipulation Language (DML) (язык управления (манипулирования) данными) — это семейство компьютерных языков, используемых в компьютерных программах или пользователями баз данных для получения, вставки, удаления или изменения данных в базах данных.

На текущий момент наиболее популярным языком DML является SQL, используемый для получения и манипулирования данными в РСУБД.

Другие формы DML использованы в IMS/DL1, базах данных CODASYL (таких как IDMS), и других.

Языки DML изначально использовались только компьютерными программами, но с появлением SQL стали также использоваться и людьми.

Функции языков DML определяются первым словом в предложении (часто называемом **запросом**), которое почти всегда является глаголом. В случае с SQL эти глаголы — «select» («выбрать»), «insert» («вставить»), «update» («обновить»), и «delete» («удалить»). Это превращает природу языка в ряд обязательных утверждений (команд) к базе данных.

Языки DML могут существенно различаться у различных производителей СУБД. Существует стандарт SQL, установленный ANSI, но производители СУБД часто предлагают свои собственные «расширения» языка.

Терминология: иерархическая модель данных

Иерархическая модель данных — это [модель данных](#), где используется представление [базы данных](#) в виде [древовидной](#) (иерархической) структуры, состоящей из объектов ([данных](#)) различных уровней.

Между объектами существуют связи, каждый объект может включать в себя несколько объектов более низкого уровня.

Такие объекты находятся в отношении предка (объект более близкий к корню) к потомку (объект более низкого уровня), при этом возможна ситуация, когда объект-предок не имеет потомков или имеет их несколько, тогда как у объекта-потомка обязательно только один предок.

Как и сетевая, иерархическая модель данных базируется на графовой форме построения данных, и на концептуальном уровне она является просто частным случаем сетевой модели данных. В иерархической модели данных вершине графа соответствует тип сегмента или просто сегмент, а дугам — типы связей предок — потомок. В иерархических структурах сегмент — потомок должен иметь в точности одного предка.

Иерархическая модель представляет собой связный [неориентированный граф](#) древовидной структуры, объединяющий сегменты. Иерархическая БД состоит из упорядоченного набора деревьев.

Терминология: сетевая модель данных

Сетевая модель данных — логическая [модель данных](#), являющаяся расширением [иерархического подхода](#), строгая математическая теория, описывающая структурный аспект, аспект целостности и аспект обработки данных в сетевых базах данных.

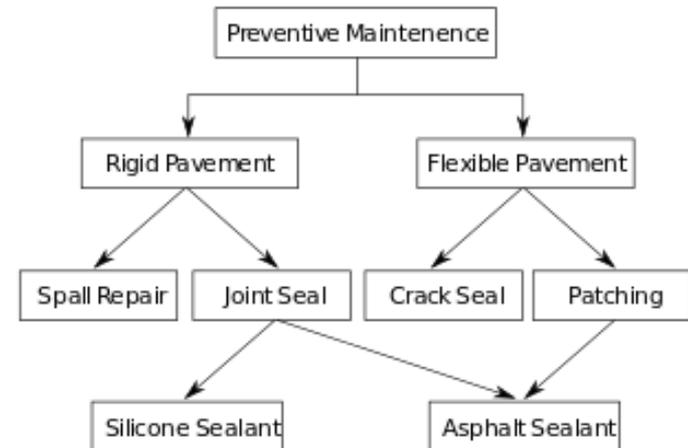
Разница между [иерархической моделью данных](#) и сетевой состоит в том, что в иерархических структурах запись-потомок должна иметь в точности одного предка, а в сетевой структуре данных у потомка может иметься любое число предков.

Сетевая БД состоит из набора экземпляров определенного типа записи и набора экземпляров определенного типа связей между этими записями. Тип связи определяется для двух типов записи: предка и потомка. Экземпляр типа связи состоит из одного экземпляра типа записи предка и упорядоченного набора экземпляров типа записи потомка. Для данного типа связи L с типом записи предка P и типом записи потомка C должны выполняться следующие два условия:

- каждый экземпляр типа записи P является предком только в одном экземпляре типа связи L ;
- каждый экземпляр типа записи C является потомком не более чем в одном экземпляре типа связи L .

[Сетевая модель данных](#)

Network Model



Терминология: объектная модель данных

Объектно-ориентированная база данных (ООБД) — [база данных](#), в которой данные моделируются в виде объектов, их атрибутов, методов и классов

ORM ([англ.](#) Object-Relational Mapping, [рус.](#) *объектно-реляционное отображение*) — технология программирования, которая связывает [базы данных](#) с концепциями [объектно-ориентированных языков программирования](#), создавая «виртуальную [объектную базу данных](#)»

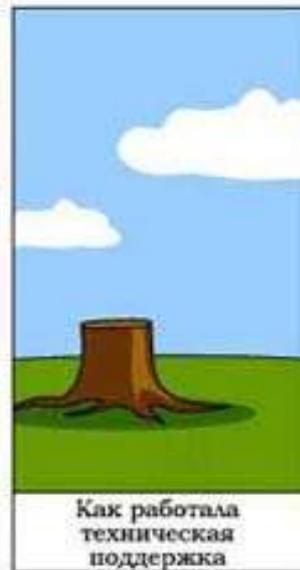
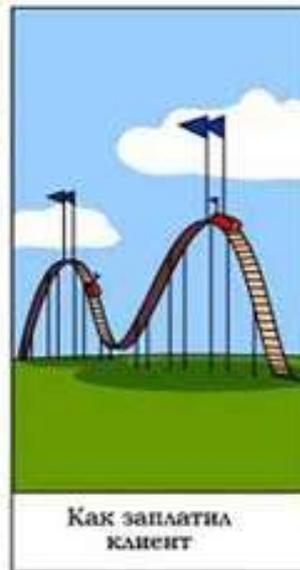
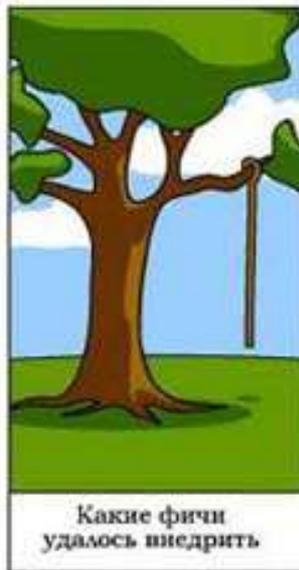
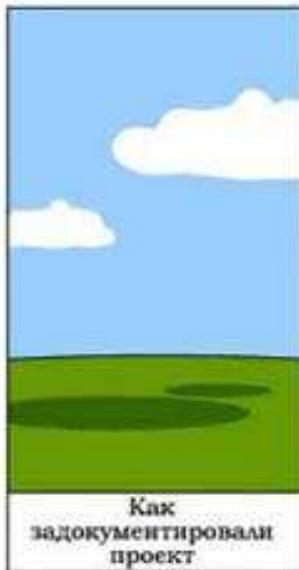
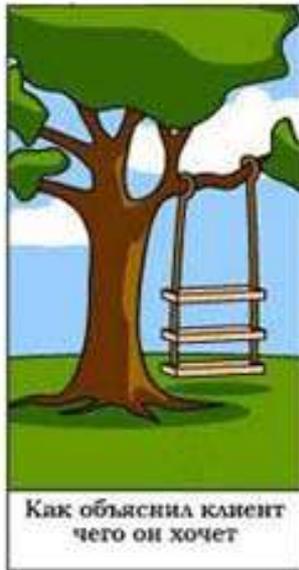
NoSQL ([англ.](#) *not only SQL, не только SQL*), в [информатике](#) — термин, обозначающий ряд подходов, направленных на реализацию хранилищ [баз данных](#), имеющих существенные отличия от моделей, используемых в традиционных [реляционных СУБД](#) с доступом к данным средствами языка [SQL](#).

Применяется к базам данных, в которых делается попытка решить проблемы [масштабируемости](#) ([англ.](#) *scalability*) и [доступности](#) ([англ.](#) *availability*) за счёт [атомарности](#) ([англ.](#) *atomicity*) и [согласованности данных](#) ([англ.](#) *consistency*)

Лекция 16
«Моделирование данных»
Часть 2

Овчинников П.Е.
МГТУ «СТАНКИН»,
ст.преподаватель кафедры ИС

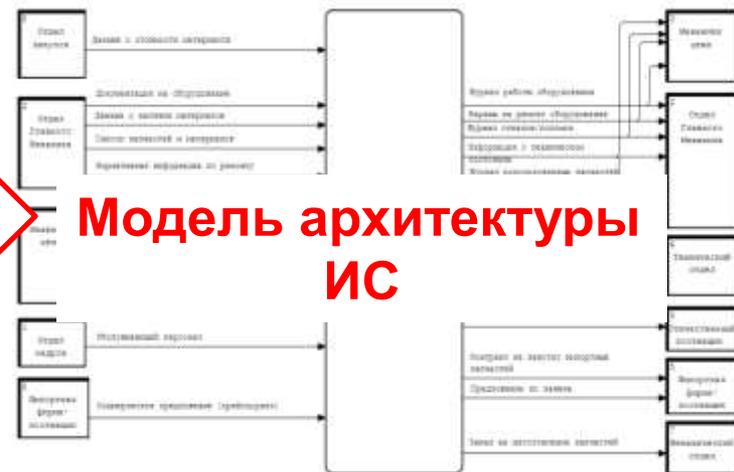
Проблематика



Проблематика: разрыв связей



Модель процессов



Модель архитектуры ИС

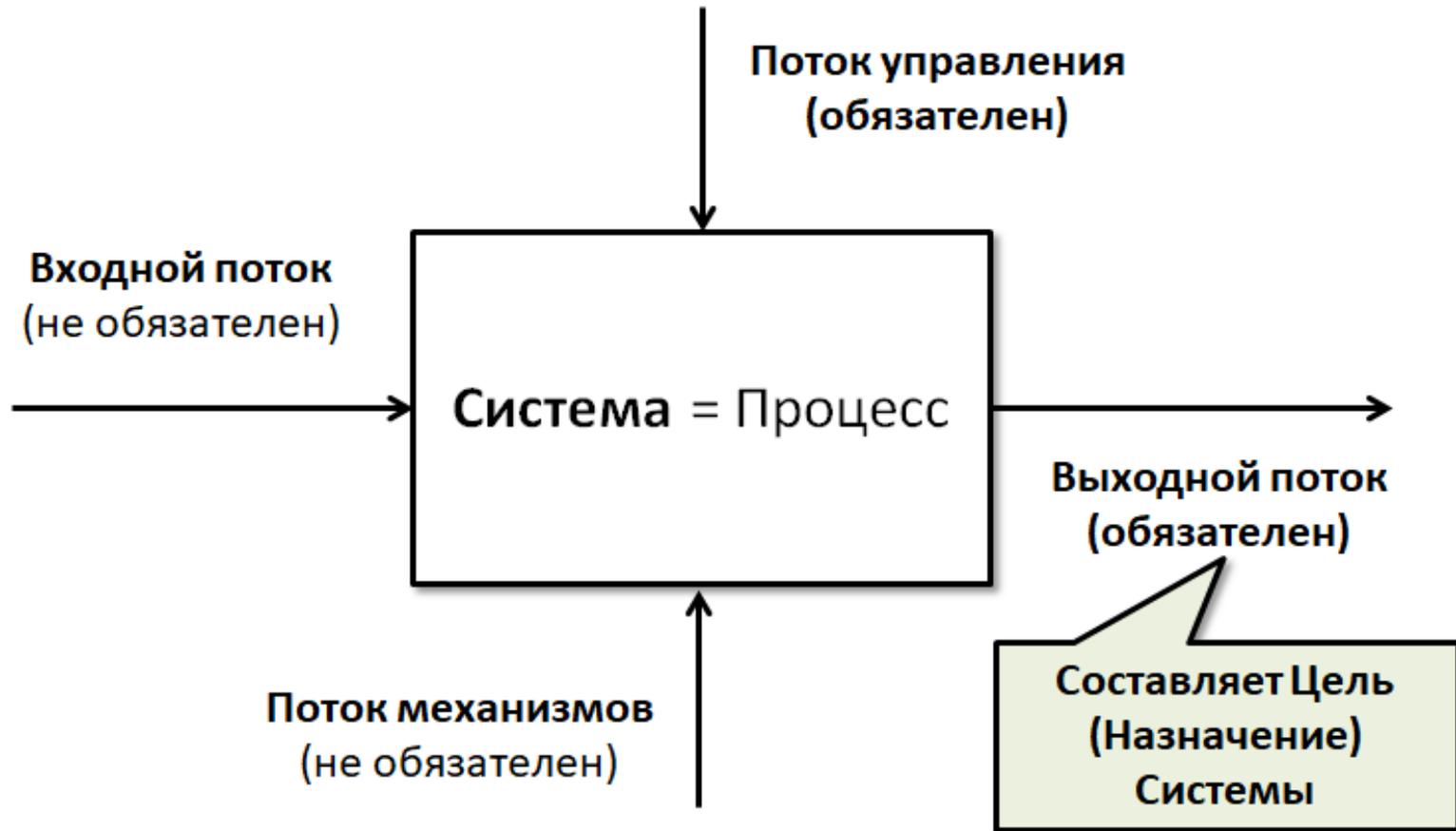


Модель данных ИС

Упрощенная модель отношения

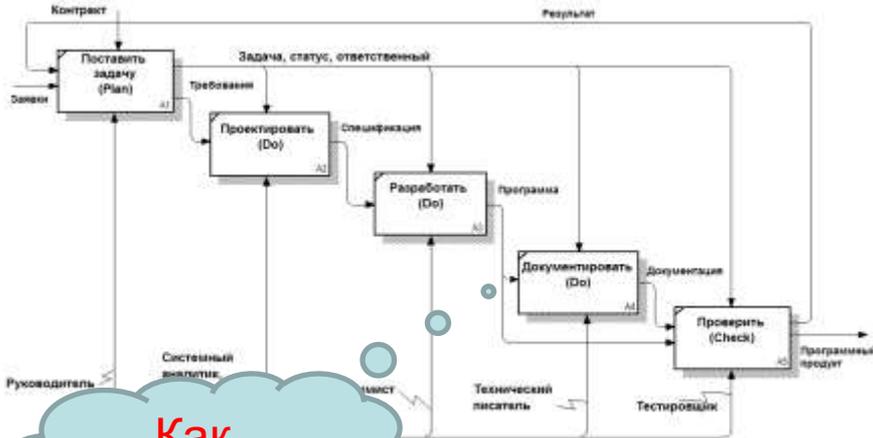
Детализированная модель отношения

Структурно-функциональный подход (система и ее модель)



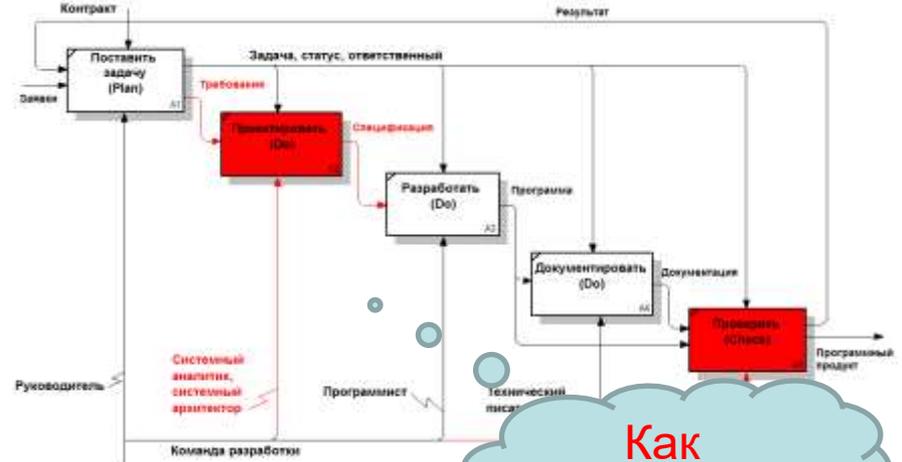
Проблематика

AsIs (как есть)



Как узнать?

ToBe (как будет)



Как выбрать?

ToDo (что сделать?)



Терминология: система

Внешняя среда

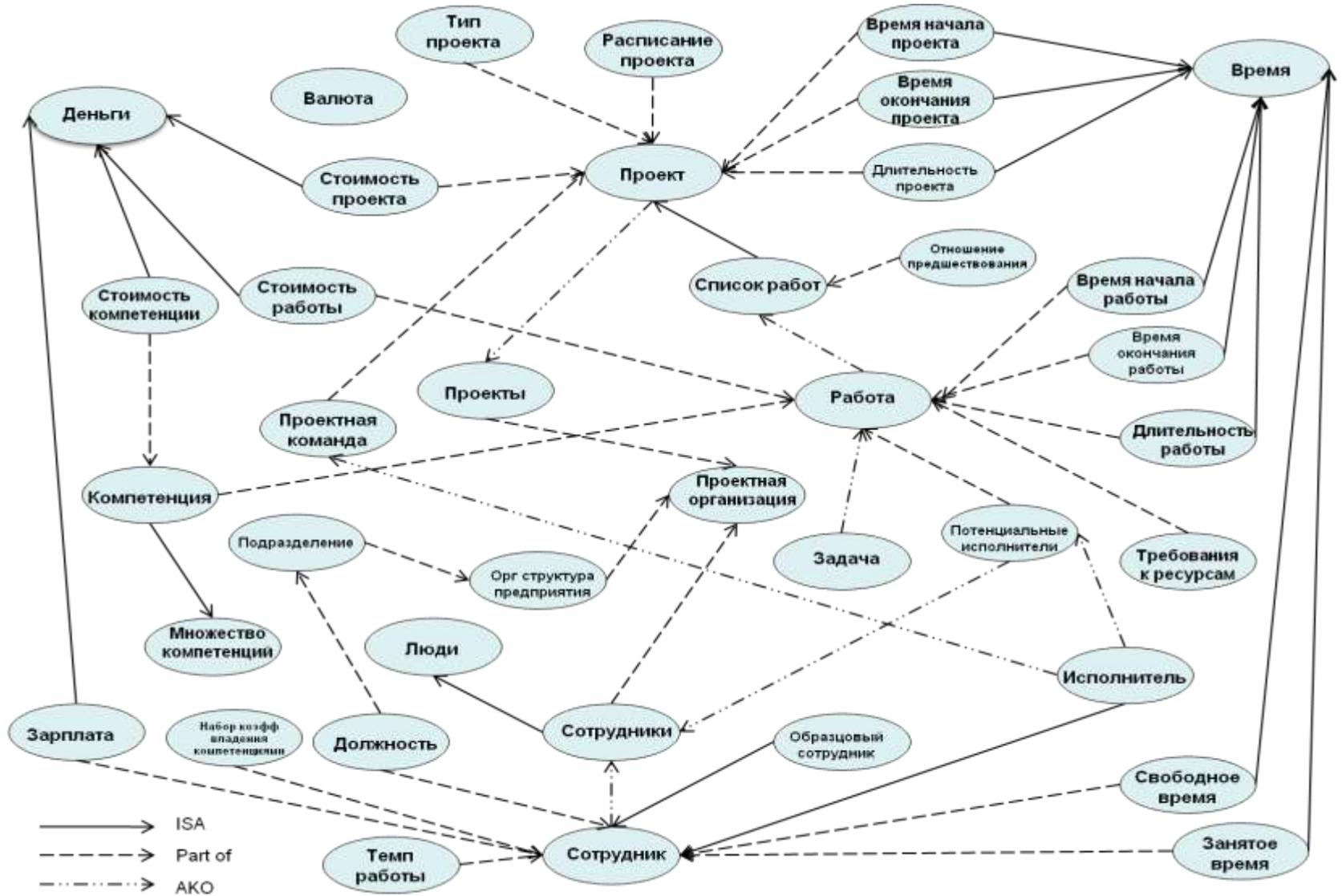
Объект = Система

Субъект = Наблюдатель

Граница системы



Проектирование БД

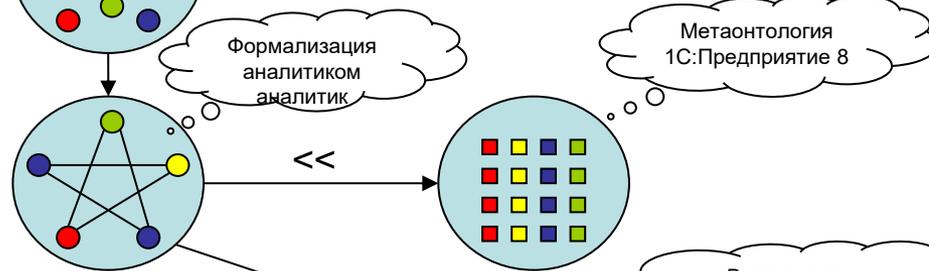


Проектирование БД

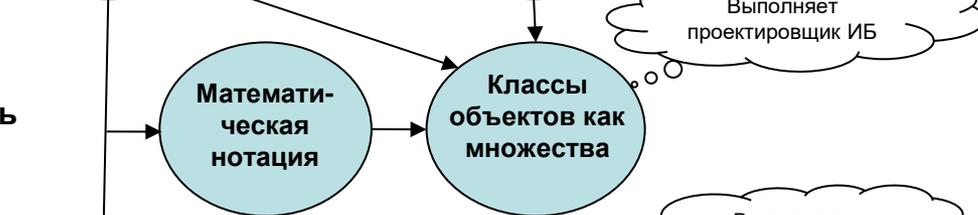
1. Семантическая сеть



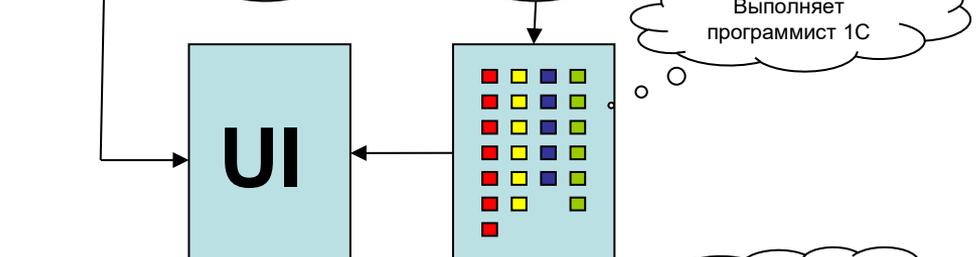
2. Онтология



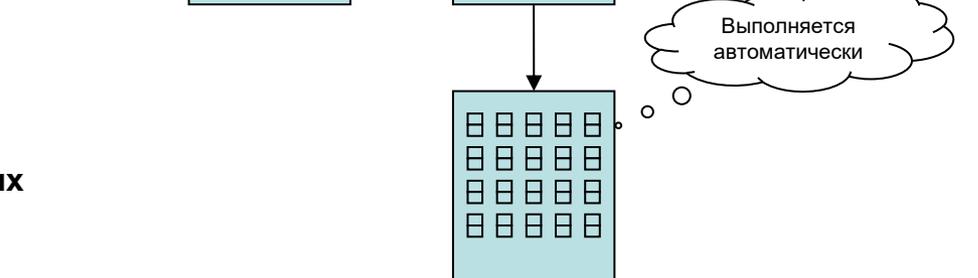
3. Концептуальная модель



4. Объектная модель информационной базы



5. Логическая модель реляционной базы данных



Проектирование БД

ПРОФЕССИОНАЛЬНЫЙ СТАНДАРТ

Архитектор программного обеспечения

67

Регистрационный номер

I. Общие сведения

Проектно-конструкторская деятельность

06.003

(наименование вида профессиональной деятельности)

Код

Основная цель вида профессиональной деятельности:

Создание и сопровождение архитектуры программных средств, заключающейся

- в синтезе и документировании решений о структуре;
- компонентном устройстве;
- **основных показателях назначения;**
- порядке и способах реализации программных средств в рамках системной архитектуры;
- реализации требований к программным средствам;
- контроле реализации и ревизии решений

Показатели назначения ИС

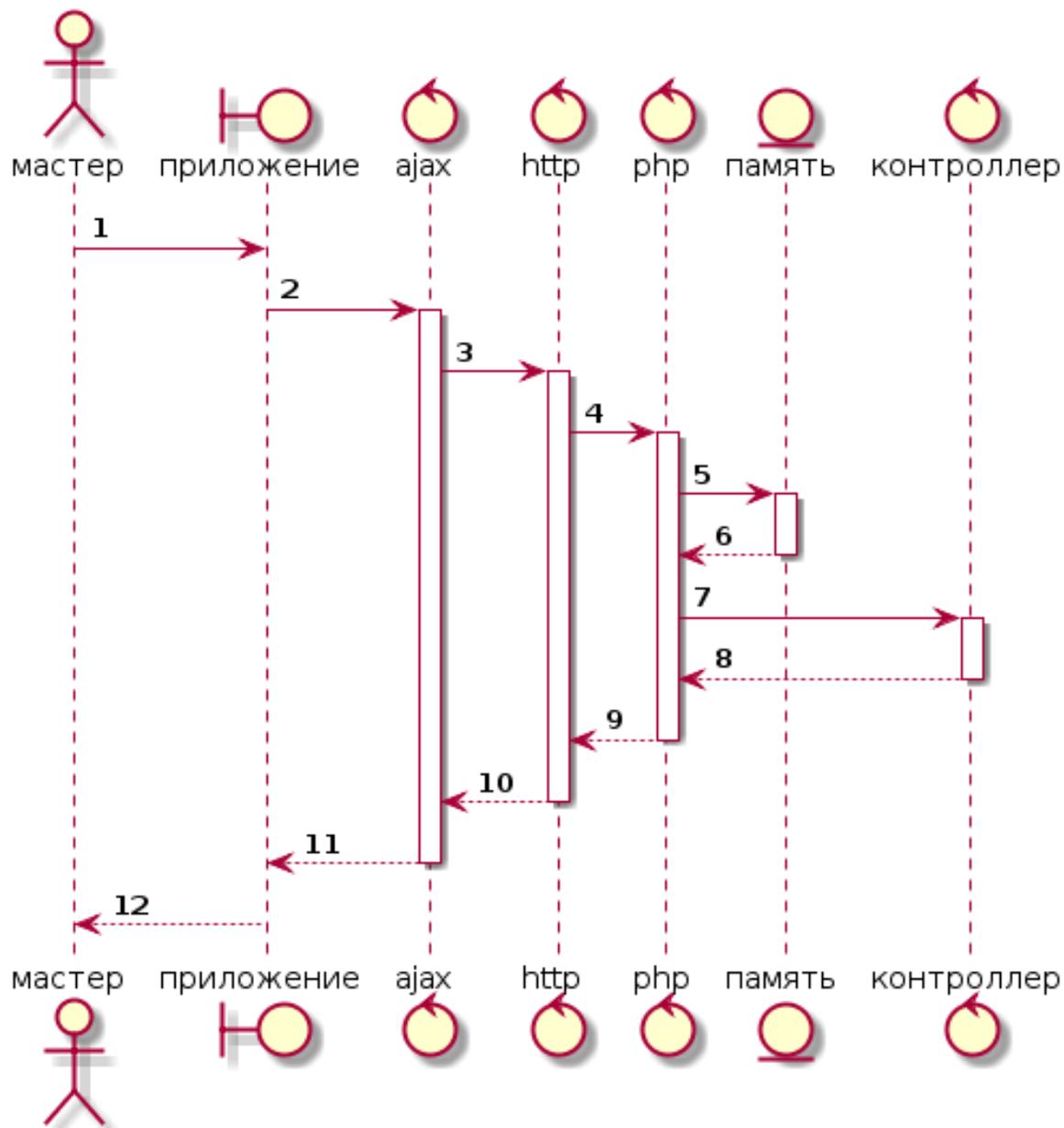
ГОСТ 34.602-89 Информационная технология (ИТ). Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы

2.6.1.3. В требованиях к показателям назначения АС приводят значения параметров, характеризующие степень соответствия системы ее назначению.

Для АСУ указывают:

- **степень приспособляемости** системы к изменению процессов и методов управления, к отклонениям параметров объекта управления;
- **допустимые пределы модернизации** и развития системы;
- **вероятностно-временные характеристики**, при которых **сохраняется целевое назначение** системы.

Потоки данных: web-приложение



Потоки данных: обработка запросов

Вложенные циклы — это вложенные итеративные процессы поиска данных в каждой из соединяемых таблиц.

Внешний цикл извлекает все необходимые строки из внешней таблицы. Если часть или все ограничения для внешней таблицы могут быть использованы для поиска по индексу, то на каждой итерации цикла в индексе ищутся расположения всех необходимых строк и выполняется прямой доступ к таблице. В противном случае таблица сканируется целиком. Оставшиеся ограничения используются для фильтрации выбранных строк. Для каждой оставшейся строки вызывается внутренний цикл.

Внутренний цикл по условиям соединения и данным внешнего цикла ищет строки во внутренней таблице. Если часть или все ограничения для внутренней таблицы, а также ограничения, полученные от внешнего цикла, могут быть использованы для поиска по индексу, то на каждой итерации цикла в индексе ищутся расположения всех необходимых строк и выполняется прямой доступ к таблице. В противном случае таблица сканируется целиком. Оставшиеся ограничения используются для фильтрации выбранных строк.

Циклы могут вкладываться произвольное число раз. В этом случае внутренний цикл становится внешним для следующего цикла и т. д.

На каждой итерации самого глубокого цикла выбранные из таблиц строки конкатенируются, для получения одной строки итогового результата.

Потоки данных: обработка запросов

Query 1: Query cost (relative to the batch): 0,00%
Query text: select * from Users u, Deps


SELEC
Cost:

SQL> EXPLAIN PLAN FOR
2 SELECT E.EMPNO,
3 E.ENAME,
4 D.DNAME
5 FROM EMP E,
6 DEPT D
7 WHERE E.DEPTNO = D.DEPTNO
8 AND E.DEPTNO = 10;
Explained.

SQL> SELECT * FROM TABLE(dbms_xplan.display(null,null,'basic'));

PLAN_TABLE_OUTPUT

Plan hash value: 568005898

/ Id	/ Operation	/ Name	/
/ 0	/ SELECT STATEMENT	/	/
/ 1	/ NESTED LOOPS	/	/
/ 2	/ TABLE ACCESS BY INDEX ROWID/ DEPT	/	/
/ 3	/ INDEX UNIQUE SCAN	/ PK_DEPT	/
/ 4	/ TABLE ACCESS FULL	/ EMP	/
