**Бд как информационная модель**

БД- динамически обновляемая информационная модель предметной области.

Бд есть склад данных, предназначенных для совместного, различного использования.

Требования к организации БД:

* Неизбыточность (Бд неизбыточна если удаление любого элемента не ведёт к потере информации о предметной области)
* Непротиворечивость(целостность) (Бд непротиворечива если все хранящиеся в ней данные удовлетворяют определённым условиям)
* Независимость от приложений

Требования к управлению БД

* Защита данных
* Эффективность доступа (и управление им) для каждого пользователя

**Реляционная БД**

СУБД содержит

* Ядро
* Процессор языка СУБД
* Подсистему поддержки времени исполнения
* Сервисные подпрограммы

Метаданные – информация о БД: субканальная информация, структурированные данные, данные из более общей Информационной Системы, информация, содержащаяся на веб странице свойств информации.

Виды моделей данных

* Внешняя
* Концептуальная
* Логическая (внутренняя)
* Физическая

Реляционная бд основана на реляционной модели данных

Включает в себя следующие компоненты:

* Структурный аспект
* Аспект целостности
* Аспект обработки

**Нормальная форма**

Нормальная форма – свойство отношения в реляционной модели данных, характеризующее его с точки зрения избыточности, потенциально приводящей к логически ошибочным результатам выборки или изменения данных. Нормальная форма определяется как совокупность требований, которым должно удовлетворять отношение.

Переменная отношения находится в первой нормальной форме тогда и только тогда, когда каждый атрибут отношения содержит строго одно атомарное значение.

Вторая нормальная форма: Переменная отношения находится во второй нормальной форме тогда и только тогда, когда её непрерывный атрибут функционально полно зависит от первичного ключа.

Процесс преобразования отношения бд, к виду, отвечающему условиям той или иной нормальной формы, называется нормализация.

Отношения – фундаментальные понятия реляционной модели данных.

**Целостность**

Целостность бд есть соответствие содержащейся в ней информации её (БД) внутренней логике, структуре и заданным правилам

Целостность данных:

* Целостность сущности(таблицы) требует, чтобы все записи имели уникальный идентификатор (первичный ключ),
* Целостность столбцов, которая достигается ограничениями на данные (пример: not null), Ссылочная целостность, что обеспечивает связь между первичным (в родительской таблице) и внешним( в дочерней таблице), запись в родительской таблице не может удалена, если она присутствует в дочерней, для дочерней таблицы нельзя ввести первичный ключ, отсутствующий среди первичных ключей родительской таблицы

**Уровни моделей баз данных**

1. Инфологический (концептуальный, семантический)
2. Даталогический
3. Физический

Требования к проектированию и разработке бд.

Два подхода к проектированию: от предметной области и от запроса

**Модель Сущность-Связь и её элементы**

Модель сущность связь (или ER модель) используется при высокоуровневом проектировании БД. С её помощью можно выделить свойства сущности и обозначить связи, которые могут устанавливаться между этими сущностями.

Во время проектирования Бд происходит преобразование ER модели в конкретную схему Бд на основе выбранной модели данных.

ER модуль представляет собой формальную конструкцию которая сама по себе не предписывает никаких графических ср-в её реализации.

В качестве стандартной графической модели, с помощью которых можно визуализировать ER модель была предложена диаграмма сущность-связь.

Отношение — это математическая модель, таблицы базы данных.

Отношение — это множество сущностей обладающих одинаковым набором атрибутов

В реальном проектирование структуры базы данных применяется семантическое моделирование

Сущность — это класс однотипных объектов, информация о которых должна быть учтена в модели.

Экземпляр сущности есть конкретный представитель сущности. «Сотрудник Иванов» - представитель сущности «Сотрудник».

Атрибут сущности — это именованная характеристика являющимся некоторым свойством сущности.

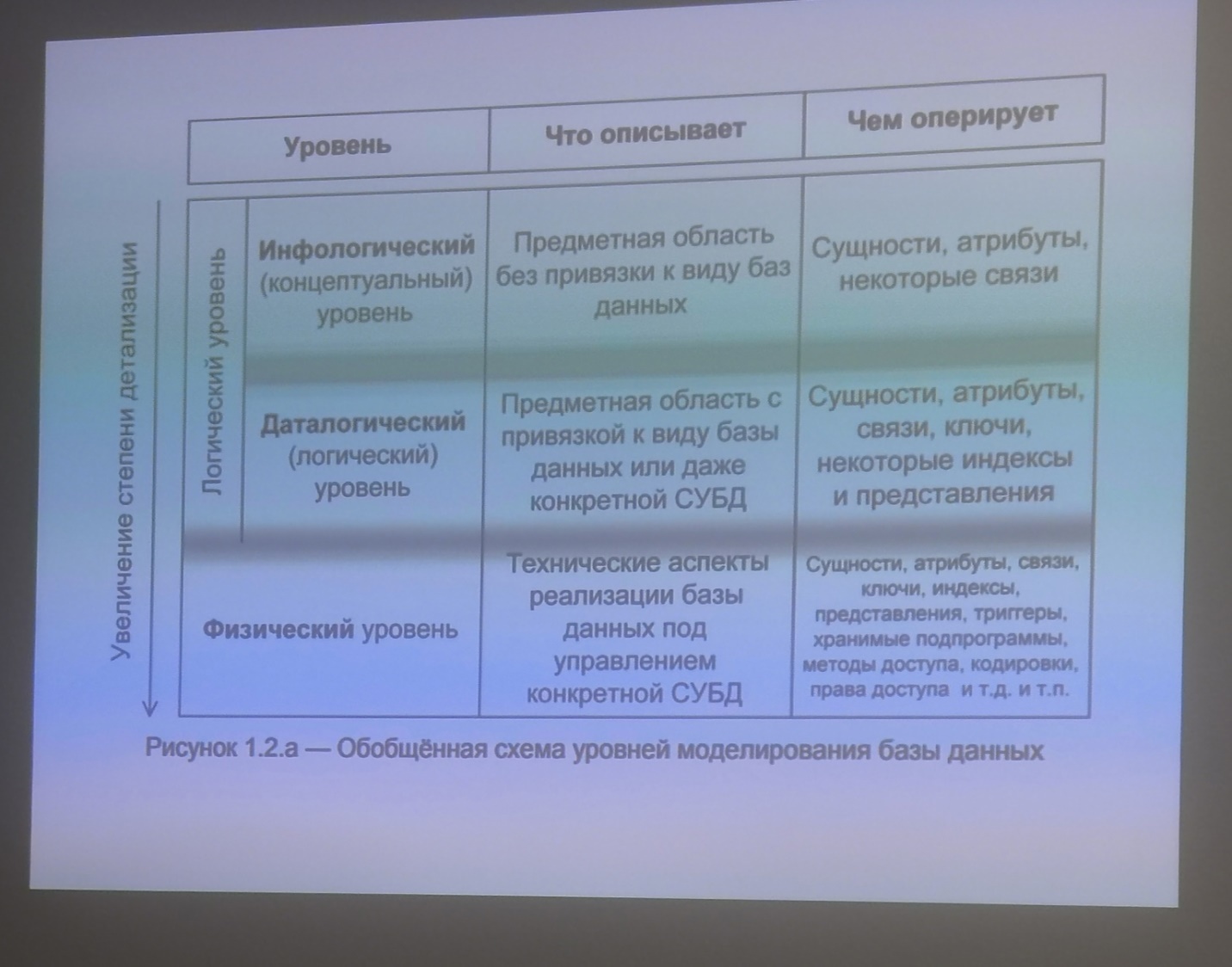
Ключ сущности есть неизбыточный набор атрибутов значения которых в совокупности являются уникальными для каждого экземпляра сущности.

Связь есть ассоциация между двумя сущностями.

<Каждый экземпляр Сущности 1> <Модальность связи (может/должен)> <Наименование связи> <Тип связи> <Экземпляр сущности 2>

Выводы: Диаграммы сущность-связь позволяют использовать наглядные графические обозначения. Они делятся на концептуальные и физические.

У ватников есть аргумент, "что тебе сделал Путин", на первый взгляд,



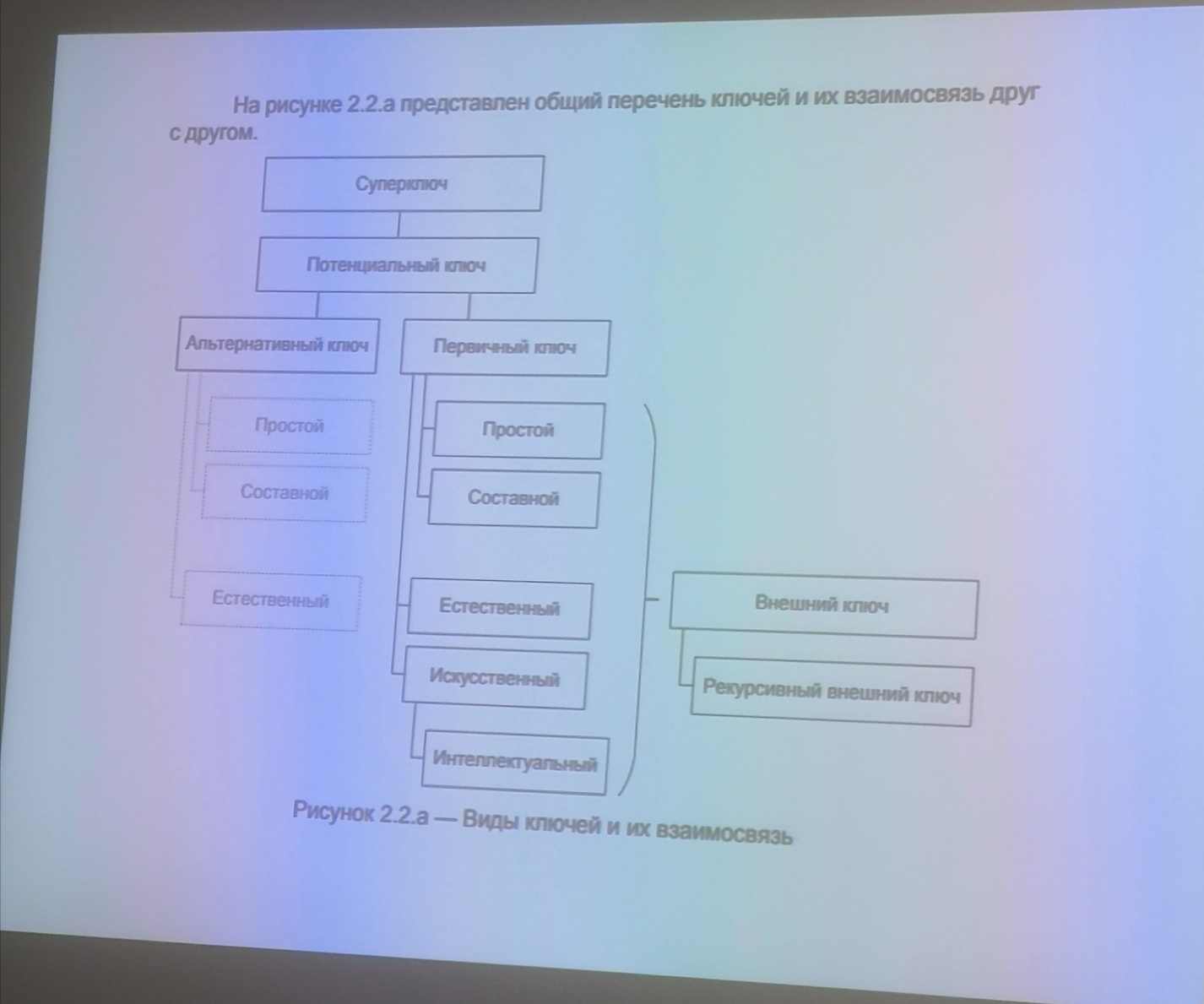
Нисходящие проектирование есть проектирование от предметной области от инфологического к физическому уровням. «Сначала тема, потом база»

Восходящее проектирование – от физического к инфологическому. «База есть, но зачем?»

Требования к БД:

1. Адекватность предметной области (Сама база, информация и связи в ней должны быть адекватны.)
2. Удобство использования
3. Производительность
4. Защищённость данных

**Ключи**

Ключ есть идентификатор, являющийся частью набора элементов данных.

Суперключ есть набор полей в котором могут оказаться лишние поля.

Потенциальный ключ- отношения не выбранный в качестве первичного.

Альтернативный ключ

Первичный ключ есть потенциальный ключ, выбранный в качестве средства гарантированный идентификации кортежей отношения.

Естественный ключ-ключ, построенный на множестве атрибутов отношения, несущих смысловую нагрузку

Искусственный ключ (суррогатный) -Ключ построенный на

Внешний ключ есть атрибут или поле, что содержит копии значений первичного ключа от другого отношения

**Связи**

Связь есть ассоциация, объединяющая несколько сущностей.

Мощность связи – свойство связи, определяющее допустимые мощности взаимосвязанных подмножеств кортежей отношений, объединённых данной связью. (максимум связей у таблицы)

1 ко многим – ассоциация, определяющая 2 отношения таким образом, что одному кортежу родительского отношения может соответствовать произвольное количество кортежей дочернего отношения.

Многие к одному, то же самое, но наоборот, но с возможным нулём в дочернем кортеже.

Связь многие к одному используется если

* Есть сущности разных типов, объединённых именно такой связью
* Есть ограничения СУБД на количество полей
* В целях оптимизации
* В описываемой предметной области есть множество различных сущностей

Идентифицирующая связь – ассоциация, объединяющая 2 отношения таким образом, что любому кортежу дочернего отношения всегда соответствовал кортеж родительского отношения.

Неидентифицирующая связь – ассоциация объединяющая 2 отношения таким образом, что кортежу дочернего отношения может не соответствовать ни одного кортежа родительского отношения

**Основные задачи SQL**

Определение данных DDL Data Definition Language (Создание таблиц и баз данных) (create, alter, drop)

DML манипулирование данными и построение запросов Data Manipulation Language (insert, update, delete)

Ограничение доступа к данным (grant, revoke), управление курсором (open cursor, close cursor)

**Типы данных**

Предопределённые типы: Int, small int, big int, decimal, numeric

Приближённые: Float, real, double precision

Битовые последовательности

Bit, binary large object

Непредопределённые типы

Массив (Array)

Множество и мультимножество

Пример использования последовательности

Пользовательский тип

**Встроенные функции**

Bit\_length

Cast – преобраование данных

Char\_length

Current\_date

Current\_time

Current\_timesstamp

Lower

Position

Substring – часть строки с n символа с указанной длиной

Translate – пребразование строки с учётом указанной функции

Trim – удаление первых и/или последних символов

Upper

**Манипулирование данными SQL**

Шаблон Like ищет строку по заданной текстовой формуле

Пример: для отбора всех поставщиков, у которых во второй позиции названия есть символ «К» и окончание «О» выражение будет иметь вид «\_К%О»

Алгоритмизация функций

* Count
* Avg
* Sum
* Max
* Min
* Group by
* Having

Слияние Union

**Индексы**

Индексы есть специальная структура баз данных, используемая для ускорения поиска записей и физического доступа к записям.

По кол-ву полей:

Простой – 1 поле

Составной - 2+ полей

Уникальный индекс – настроенный на уникальных полях

Не уникальный индекс- на не уникальных полях

По соотношению с расположением записи:

Кластерный – на поле с проведённым физическом

Не кластерный – на поле без физического упорядочивания

Первичный – на поле с уникальными значениями **и** с физическим упорядочиванием

По структуре хранения

Разделённые – хранимый и обрабатываемый по частям, что быстрее

Не разделённые – едино, что надёжнее

По детализации

Плотный – с указателем на отдельное значение

Не плотный – с указатель на расположение блока записей

По базовой структуре

На основе b-дерева

На основе t-дерева

На основе хеш-таблицы

**Команды для работы в mysql**

DESCRIBE table\_name – вывод на экран структуры таблицы

Truncate

Функции обработки строк

Concat- конкатинация объединение строк

Текст для поиска %-частичное вхождение в строку

^ строка начинающаяся с текста для поиска

$- строка заканчивающаяся текстом для поиска

\_ один и только один неизвестный символ

RLike- начинается с

Select \* from sport where fio RLike ‘^A’ and extract (year from YMD) =1992;

Having= where но только для столбца group by

Управляющие функции mysql

If, case

**Хранимые подпрограммы**

Хранимая процедура есть подпрограмма для выполнения операций с данными, вызываемая как напрямую, так и другими программами.

Хранимая функция есть подпрограмма, расширяющая возможности языка SQL и работающая аналогично встроенным функциям. ОБЯЗАНА возвращать значения.

Отличия

1. Функция обязана возвращать значение, процедура-нет
2. Вызов: call/set
3. Процедура может вызывать другие процедуры, тогда как функция только процедуры, но не иные функции
4. Процедура не может быть использована в проверке
5. Процедура может быть изменена после создания

**Транзакции в MYsql**

Транзакция есть последовательность операций, рассматриваемая как единая сущность.

Оператор открывающий транзакцию: start transaction

Для завершения транзакции и внесения изменений в БД используется оператор commit.

Autocommit – автосохранение

Set autocommit= 0 его отключение

Delimiter #

Begin

Set autocommit =0;

Statt transaction;

Update sport set ves=ves-N;

Commit;

Select \* from sport;

End;

**Администрирование**

Create user (name)

Drop user (name)

Set password = password(‘password’)

Set password for User\_name =password(‘password’)

Grant privilege on object to User\_name with grant opinon

Привилегии

Create

Delete

Execute

Index

Select

Update

Alter

Drop