

# Информационное обеспечение САПР

# Информационное обеспечение и информационный фонд САПР

- Информационное обеспечение САПР состоит из информационного фонда и средств управления этим фондом.
- Информационный фонд (ИФ) - совокупность данных, используемых всеми компонентами САПР.
- Назначение информационного обеспечения (ИО) САПР- реализация информационных потребностей всех составных компонентов САПР. Основная функция ИО САПР - ведение информационного фонда, т.е. обеспечение создания, поддержки и организации доступа к данным. Таким ИО САПР есть совокупность информационного фонда и средств его ведения.

# Информационный фонд

Информационный фонд включает в себя информацию для выполнения автоматизированного проектирования, и представляется в виде печатных документов, чертежей, файлов на машинных носителях и т.п.

В информационный фонд входят данные о комплектующих деталях, узлах, материалах, технологической оснастке и оборудовании, типовых проектных решениях, текущем состоянии выполняемых проектов, сведения из ГОСТов, описание типовых проектных процедур и др.

# Банки данных

Основу информационного обеспечения САПР составляют банки данных, состоящие из баз данных и систем управления базами данных. Особенности банков данных в САПР обусловлены большим объемом данных, разнообразием их типов, изменчивостью данных по мере развития проектов и т.п. Структурирование описаний по иерархическим уровням и аспектам лежит в основе организации баз данных.

# Определения БД

- **База данных** — структурированная совокупность взаимосвязанных данных, относящихся к конкретной предметной области.
- **База данных** может быть определена как совокупность специально организованных данных, рассчитанных на применение в большом количестве прикладных программ или как совокупность данных, отражающая состояние объектов и их отношений в рассматриваемой предметной области.
- **База данных** определяется как совокупность взаимосвязанных, хранящихся вместе данных при наличии такой избыточности, которая допускает их использование оптимальным образом для одного или нескольких приложений; данные запоминаются так, чтобы они были независимы от программ, использующих эти данные.

# Электронные таблицы и реляционные СУБД (РСУБД):

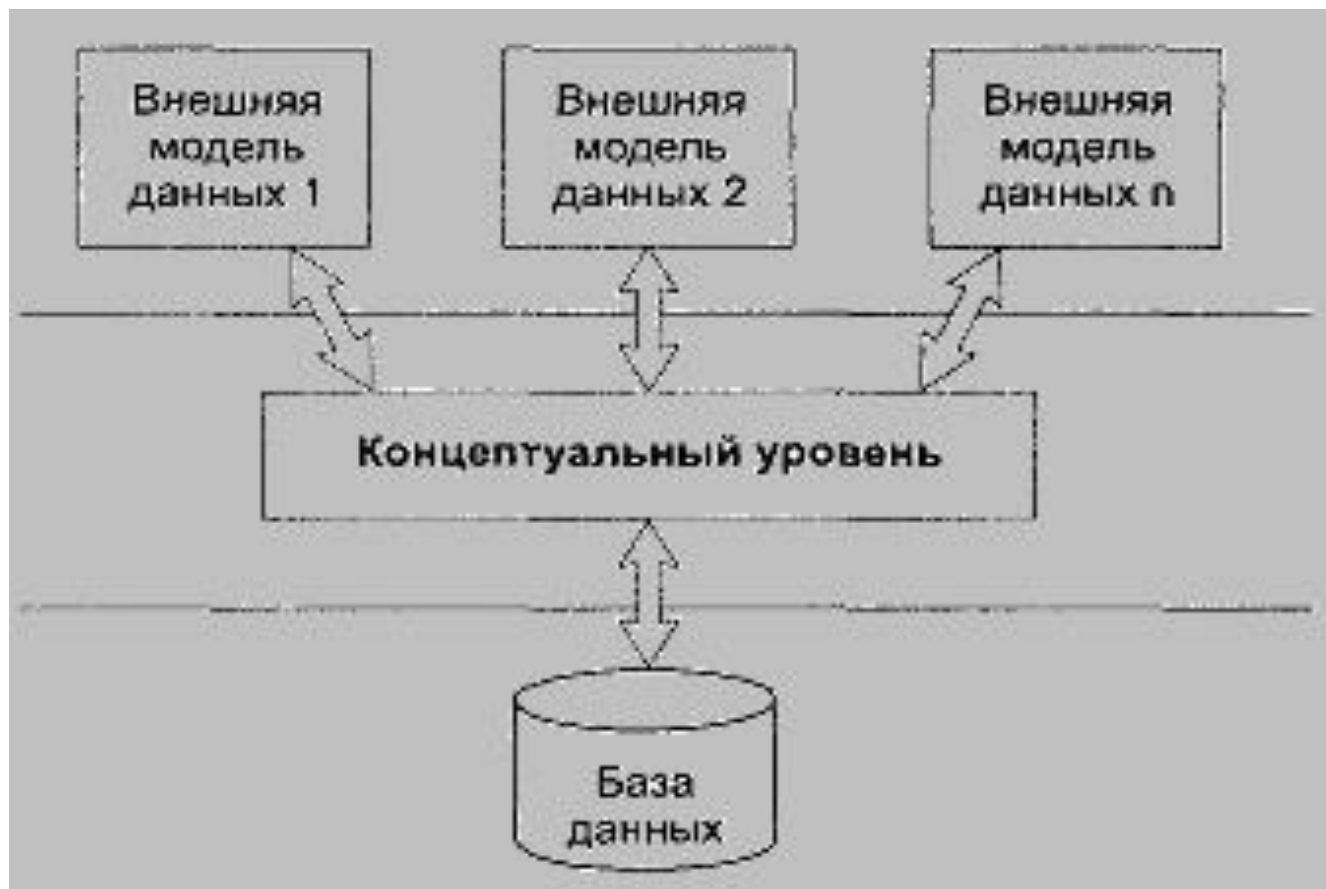
- Все РСУБД предназначены для обработки очень больших объемов данных — гораздо больших, чем те, с которыми могут оперировать электронные таблицы.
- В РСУБД таблицы можно связывать, представляя их данные в виде единой таблицы. Пользуясь электронными таблицами сделать это нелегко, если вообще возможно.
- В РСУБД дублирование информации сведено к минимуму. Практически повторяются только коды, связывающие данные нескольких таблиц. Поддерживается целостность данных.

- Любая база данных должна обладать следующими свойствами:
- интегрированностью, направленной на решение широкого круга задач;
- структурированностью;
- взаимосвязанностью;
- независимостью описания данных от прикладных программ.

- Независимость описания данных достигается за счет построения двух уровней представления данных: логического и физического. На логическом уровне данные представляются в виде удобном для использования прикладными программами и проектировщиками. Физический уровень представления данных отражает способ хранения и структуру данных с учетом их расположения на носителях информации.



# Уровни представления данных



# Независимость данных

- Физический уровень( первый) – внешние модели и концептуальная модель не подвержены изменениям физической памяти и метода доступа
- Логический уровень(второй) - Если концептуальная модель спроектирована т. о., чтобы отражать будущие расширения, то вносимые в нее изменения не должны оказывать влияния на существующие внешние модели

- В современной технологии БД предполагается, что создание БД, ее поддержка и обеспечение доступа пользователей к ней осуществляется централизованно с помощью специального программного инструментария — системы управления базами данных (СУБД).

**Система управления базами данных (СУБД)** — совокупность языковых и программных средств, предназначенных для создания, ведения и совместного использования БД многими пользователями.

- СУБД реализует два внутренних интерфейса:
- между логическими структурами данных в программных модулях и различными программными модулями системы;
- между логической и физической структурами в БД.

# Преимущества централизованного подхода в управлении данными.

- Возможность сокращения избыточности хранимых данных
- возможность устранения (до некоторой степени) противоречивости данных, что является следствием предыдущего пункта
- возможность общего доступа к данным
- возможность соблюдения стандартов
- возможность введения ограничений для обеспечений безопасности
- возможность обеспечения целостности данных, т.е. правильности и точности
- возможность сбалансировать противоречивые требования, например учесть требования всего предприятия, а не отдельного пользователя
- обеспечение независимости данных ( строго говоря это цель, а не преимущество)

# Основные функции СУБД

- Создание БД, включая формирование описаний объектов, их свойств и отношений.
- Первоначальная загрузка реализаций.
- Поиск объектов, реализаций и их свойств.
- Удаление отдельных свойств объектов, описаний объектов и уничтожение его связей в БД.
- Добавление новых свойств объекта, новых объектов и установление связей нового объекта в БД.
- Модификация свойств и характеристик объекта.
- Обеспечение защиты и целостности данных.

- Программы, с помощью которых пользователи работают с базой данных, называются **приложениями**.
- В общем случае с одной базой данных могут работать множество различных приложений. При рассмотрении приложений, работающих с одной базой данных, предполагается, что они могут работать параллельно и независимо друг от друга, и именно СУБД призвана обеспечить работу множества приложений с единой базой данных таким образом, чтобы каждое из них выполнялось корректно, но учитывало все изменения в базе данных, вносимые другими приложениями.

# Лекция 2

- **Жизненный цикл базы данных**
- **Этапы проектирования БД**
- **Методология IDEF1X**





# Этапы проектирования БД

- Инфологическое проектирование
- Датологическое проектирование
  - Логическое проектирование
  - Физическое проектирование

# Инфологическое проектирование

- Задача инфологического проектирования - получение семантических или смысловых моделей, отражающих информационное содержание конкретной предметной области.
- Инфологическая модель - это обобщенное, не привязанное к каким либо СУБД описание предметной области.

- Предметная область - часть реального мира, которая описывается или моделируется с помощью баз данных и использующих их приложений.
- Информационный объект - идентифицируемый объект реального мира, некоторое понятие или процесс, относящиеся к предметной области, о которой хранятся данные.
- Концептуальная модель представляет собой общее интегрированное описание, а внешняя модель - это представление пользователя или группы пользователей.

# Логическое проектирование

- Задача логического проектирования - организация данных, выделенных на предыдущем этапе проектирования в форму принятую в выбранной конкретной СУБД. Т.е. здесь разрабатывается схема концептуальной модели(логическая схема) и схемы внешних моделей данных о предметной области, пользуясь только теми типами моделей данных и их особенностями, которые поддерживаются этой СУБД.

# Физическое проектирование

- Задача физического проектирования - выбор рациональной структуры хранения данных и методов доступа к ним, исходя из методов и средств, которые предоставляются разработчику системой управления базой данной.

# ***Основные этапы проектирования реляционной СУБД***

- Выделение ПО, анализ требований пользователей. Разработка представлений пользователей , то есть внешних инфологических моделей.
- Интеграция представлений пользователей в единую концептуальную модель, описываемую с помощью концептуальной схемы.
- Представление концептуальной схемы в виде отношений и нормализация отношений (логическая схема).
- Оптимизация и проектирование внутренней схемы (возможна денормализация).

# Методология IDEF1X

- Методология IDEF1X определяет стандарты терминологии, используемой при информационном моделировании, и графического изображения типовых элементов на диаграммах.
- Компонентами IDEF1X-модели являются сущности, связи между сущностями и атрибуты сущностей.



# Процесс построения информационной модели

- определение сущностей;
- определение связей между сущностями;
- задание первичных и альтернативных ключей;
- определение атрибутов сущностей;
- приведение модели к требуемому уровню нормальной формы;

# Определение сущностей

- Каждая сущность должна иметь уникальное имя, обладать уникальным идентификатором. Каждый экземпляр сущности должен однозначно идентифицироваться и отличаться от всех других экземпляров данного типа сущности. Каждая сущность может обладать любым количеством связей с другими сущностями модели.

# Определение сущностей

Группа/1

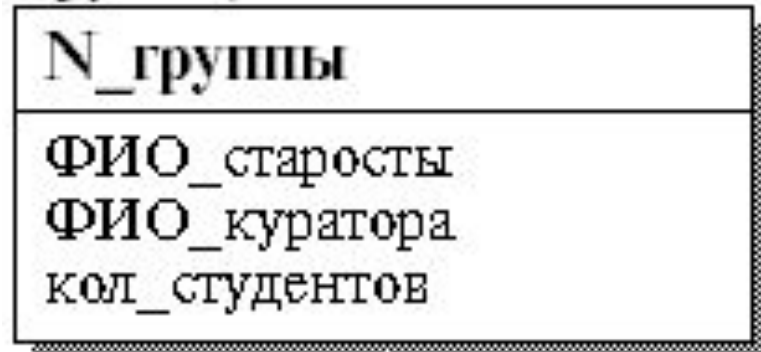
Номер\_группы

ФИО\_старосты

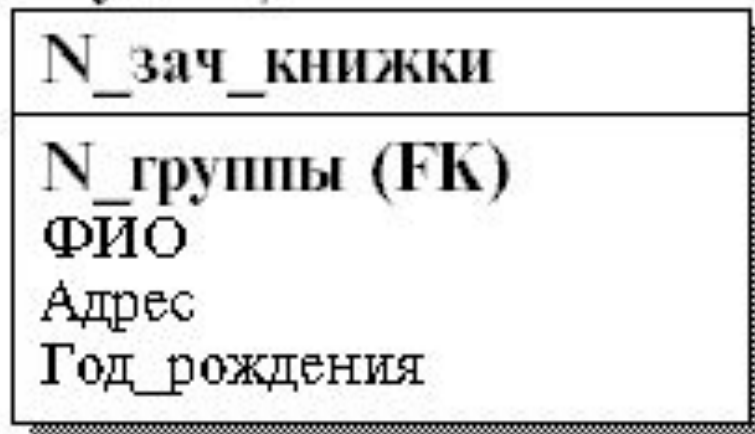
ФИО\_куратора

Кол-во\_студентов

## Группа/1

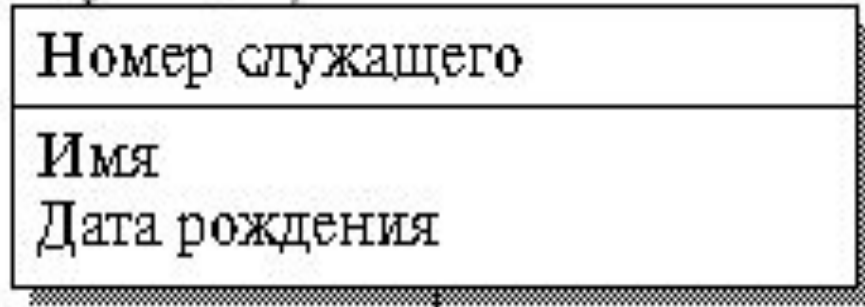


## Студент/2



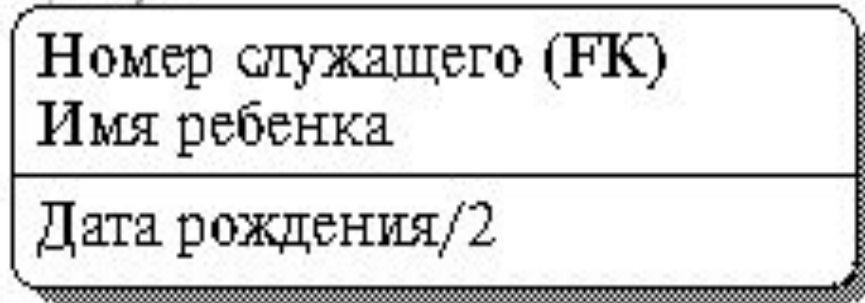
- Сущность называется независимой, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями

## Служащий/1



Является родителем/1

## Дети/2



- Сущность называется зависимой, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности

# Определение связей

На диаграмме связи представляются пятью основными элементами информации:

- тип связи (идентифицирующая, неидентифицирующая, полная/неполная категория, неспецифическая связь);
- родительская сущность;
- дочерняя (зависимая) сущность;
- мощность связи (cardinality);
- допустимость пустых (null) значений.

# Определение связей

- Наиболее часто в модели используется связь "родитель-потомок" или функциональная зависимость между двумя сущностями, при которой каждый экземпляр одной сущности, называемой родительской сущностью (сущностью - родителем), ассоциирован с произвольным количеством экземпляров второй сущности, называемой дочерней или сущностью-потомком, а каждый экземпляр сущности-потомка ассоциирован в точности с одним экземпляром сущности-родителя.

### Служащий/1

|                      |
|----------------------|
| Номер служащего      |
| Имя<br>Дата рождения |

Является родителем/1

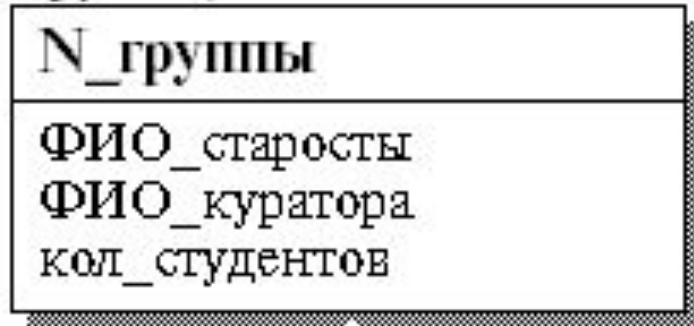
### Дети/2

|                                     |
|-------------------------------------|
| Номер служащего (FK)<br>Имя ребенка |
| Дата рождения/2                     |

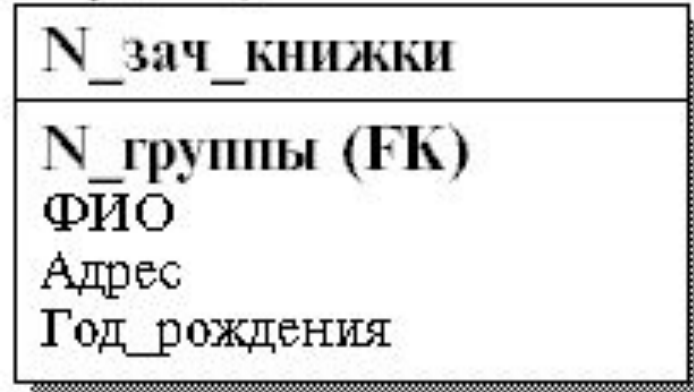
- Связь называется идентифицирующей, если экземпляр дочерней сущности идентифицируется через ее связь с родительской сущностью.
- Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в первичный ключ дочерней сущности.
- Дочерняя сущность при идентифицирующей связи всегда является зависимой.



## Группа/1



## Студент/2



- Связь называется неидентифицирующей, если экземпляр дочерней сущности идентифицируется иначе, чем через связь с родительской сущностью.
- Атрибуты, составляющие первичный ключ родительской сущности, при этом входят в состав неключевых атрибутов дочерней сущности.

# Миграция атрибутов

- При определении связи происходит миграция атрибутов первичного ключа родительской сущности в соответствующую область атрибутов дочерней сущности. Поэтому такие атрибуты не вводятся вручную.
- Атрибуты первичного ключа родительской сущности по умолчанию мигрируют со своими именами. IDEF1X позволяет ввести для них роли, т.е. новые имена, под которыми мигрирующие атрибуты будут представлены в дочерней сущности. В случае неоднократной миграции атрибута такое переименование необходимо.

# Мощность связи

- Мощность связи представляет собой отношение количества экземпляров родительской сущности к соответствующему количеству экземпляров дочерней сущности. Для любой связи, кроме неспецифической, эта связь записывается как 1:n.

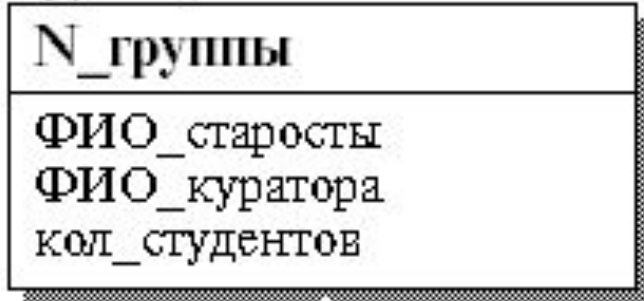
# МОЩНОСТЬ СВЯЗИ

В соответствии с методологией IDEF1X, предоставляется 4 варианта для  $n$ , которые изображаются дополнительным символом у дочерней сущности:

- ноль, один или больше (по умолчанию);
- один или больше (изображается буквой "P");
- ноль или один (изображается буквой "Z");
- ровно  $N$ , где  $N$  - конкретное число (изображается числом  $N$ ).

# Допустимость пустых (NULL) значений

Группа/1



Студент/2



- Допустимость пустых (NULL) значений в неидентифицирующих связях изображается пустым ромбиком на дуге связи со стороны родительской сущности.

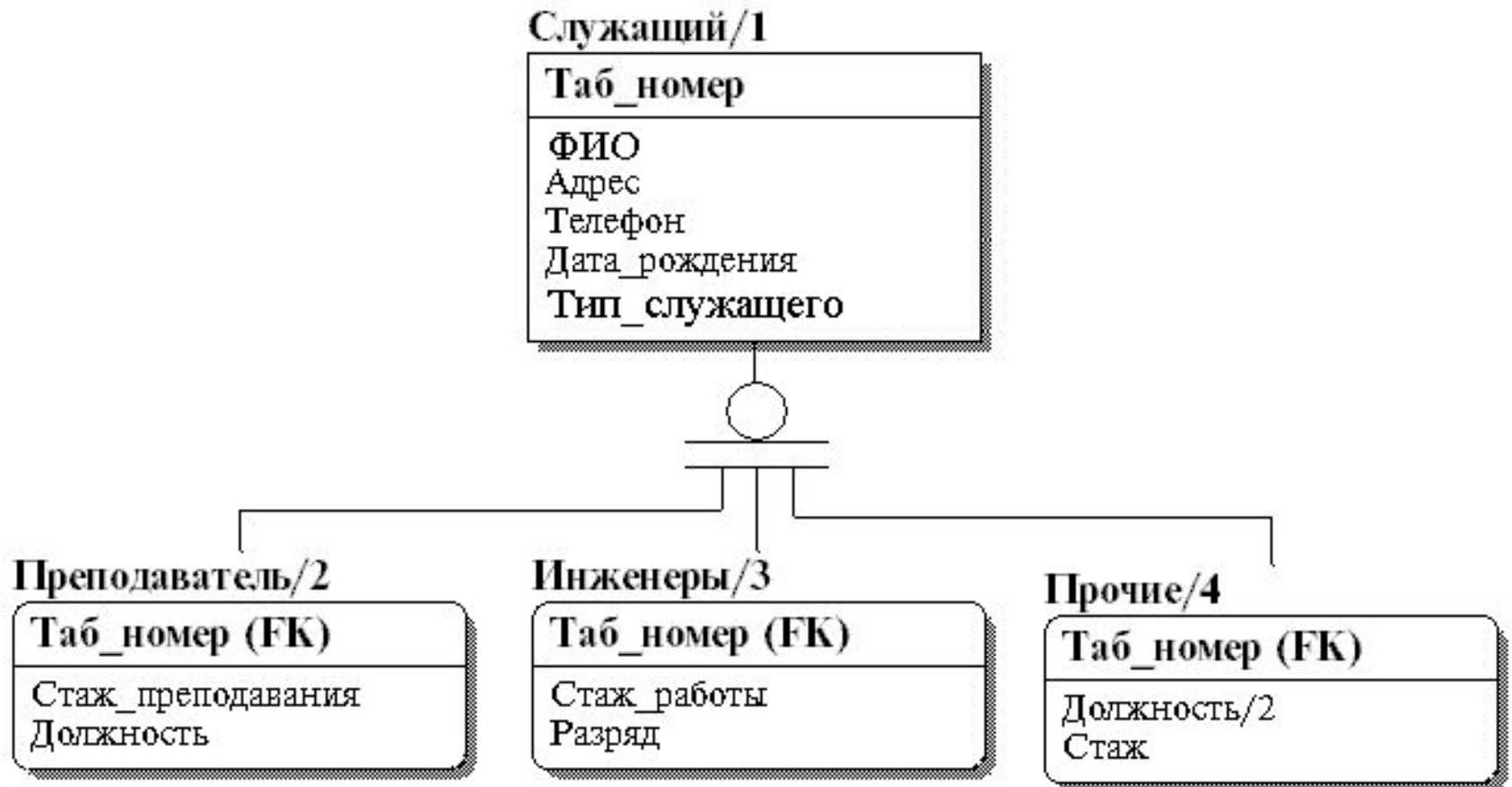
# Связи категоризации

- Некоторые сущности определяют целую категорию объектов одного типа. Для выражения такой связи вводится тип связи категоризации, т.е. связи между двумя или более сущностями, при которой каждой экземпляр одной сущности называемой общей или супертипом, связан в точности с одним экземпляром одной и только одной из других сущностей, называемых сущностями категориями или подтипами.

# Связь неполной категоризации



# Связь полной категоризации





- Первичный ключ - это атрибут или минимальный набор атрибутов, уникально идентифицирующий экземпляр сущности.
- Если несколько наборов атрибутов могут уникально идентифицировать сущность, то выбор одного из них осуществляется разработчиком на основании анализа предметной области.

Ни одна из частей ключа не может быть NULL, незаполненной или отсутствующей. Ключ не может изменяться со временем.

# Приведение модели к требуемому уровню нормальной формы

- В первой нормальной форме ER-схемы устраняются повторяющиеся атрибуты или группы атрибутов, т.е. производится выявление неявных сущностей, "замаскированных" под атрибуты.

# Приведение модели к требуемому уровню нормальной формы

- Во второй нормальной форме устраняются атрибуты, зависящие только от части первичного ключа. (Все неключевые атрибуты должны функционально зависеть от всего первичного ключа). Эта часть уникального идентификатора определяет отдельную сущность.

# Приведение модели к требуемому уровню нормальной формы

- В третьей нормальной форме устраняются атрибуты, зависящие от атрибутов, не входящих в первичный ключ. (Каждый неключевой атрибут должен функционально зависеть только от ключевых атрибутов). Эти атрибуты являются основой отдельной сущности.

# Case - средство ERwin

- ERwin - средство концептуального моделирования БД, использующее методологию IDEF1X. ERwin реализует
  - проектирование схемы БД,
  - генерацию ее описания на языке целевой СУБД,
  - разработку отчетов и
  - реинжиниринг(обратное проектирование) существующей БД.
- ERwin поддерживает прямой интерфейс с основными СУБД: DB2, Informix, Ingres, NetWare SQL, ORACLE, Progress, Rdb, SQL/400, SQLBase, SQL Server, Sybase System 10, Watcom SQL. ERwin поддерживает также настольные (desktop) СУБД: Microsoft Access, FoxPro, Paradox.

# Процесс проектирования с помощью ERwin

- описание модели на логическом уровне (построение инфологической модели определение: сущностей, связей, атрибутов, ключей)
- приведение модели к требуемому уровню нормальной формы;
- переход к физическому описанию модели: назначение соответствий имя сущности - имя таблицы, атрибут сущности - атрибут таблицы; задание триггеров, процедур и ограничений;
- генерация базы данных.

# Уровни представления ERwin

- **Логический** уровень представления соответствует этапу инфологическому проектирования или концептуальной схеме.
- Целевая СУБД, имена объектов и типы данных, индексы составляют второй (**физический**) уровень модели ERwin.

# преимущества применения ERwin

- существенное повышение скорости разработки за счет мощного редактора диаграмм, автоматической генерации базы данных, автоматической подготовки документации;
- нет необходимости ручной подготовки SQL-предложений для создания базы данных;
- возможность легко вносить изменения в модель при разработке и расширении системы;
- возможность автоматической подготовки отчетов по базе данных; важно, что эти отчеты всегда в точности соответствуют реальной структуре БД;
- разработчики прикладного программного обеспечения снабжены удобными в работе диаграммами;



# преимущества применения ERwin

- интеграция со средствами 4GL позволяет уже на стадии информационного моделирования задавать отображение данных в приложениях;
- обратное проектирование позволяет документировать и вносить изменения в существующие информационные системы;
- поддержка однопользовательских СУБД позволяет использовать для персональных систем современные технологии, что значительно упрощает переход от настольных систем к системам в технологии клиент-сервер (upsizing).