РАЗРАБОТКА БАЗ ДАННЫХ

ФИО преподавателя: Богомольная Г.В.

e-mail: bogomolnaya@mirea.ru

ТЕМА
МОДЕЛИРОВАНИЕ ДАННЫХ

План лекции

- Логическое моделирование данных:
  - ✔ метод Баркера;
  - ✔ метод IDEF1X;
  - ✔ Подход, используемый в CASE-средстве Silverrun.
- Алгоритм перехода от ER–модели к реляционной схеме данных.
- Физическое проектирование баз данных.

Моделирование данных

Цель моделирования данных - обеспечение разработчика ИС концептуальной схемой базы данных в форме одной или нескольких локальных моделей, которые могут быть отображены в любую систему баз данных.

Средство моделирования данных - диаграммы "сущность-связь".

Базовые понятия диаграммы «сущность-связь» (ERD):

Сущность (Entiry) - реальный либо воображаемый объект, имеющий существенное значение для рассматриваемой предметной области.

Связь (Relationship) - поименованная ассоциация между двумя сущностями, значимая для предметной области, при которой каждый экземпляр одной сущности ассоциирован с произвольным (в том числе нулевым) количеством экземпляров второй сущности, и наоборот.

Атрибут (Attribute) - любая характеристика сущности, значимая для предметной области и предназначенная для квалификации, идентификации, классификации, количественной характеристики или выражения состояния сущности.

Моделирование данных

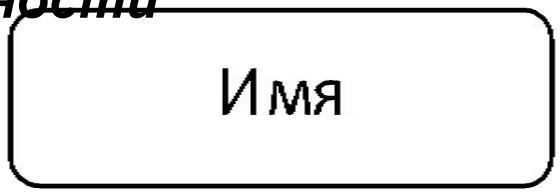
Свойства сущности:

- иметь уникальное имя:
  - к одному и тому же имени должна всегда применяться одна и та же интерпретация;
  - одна и та же интерпретация не может применяться к различным именам, если только они не являются псевдонимами;
- обладать одним или несколькими атрибутами, которые либо принадлежат сущности, либо наследуются через связь;
- обладать одним или несколькими атрибутами, которые однозначно идентифицируют каждый экземпляр сущности.

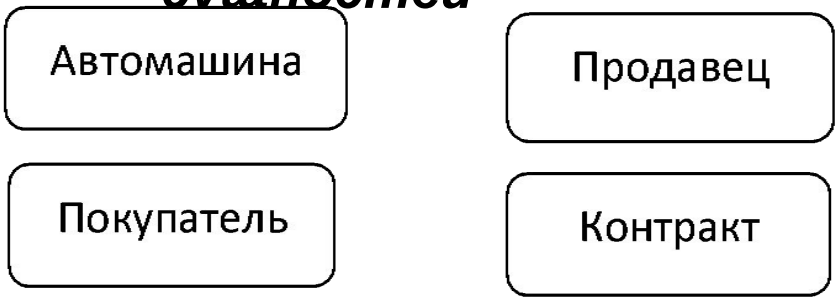
Метод Баркера

Первый шаг моделирования - извлечение информации из интервью и выделение сущностей.

Графическое изображение сущности



Выделение сущностей

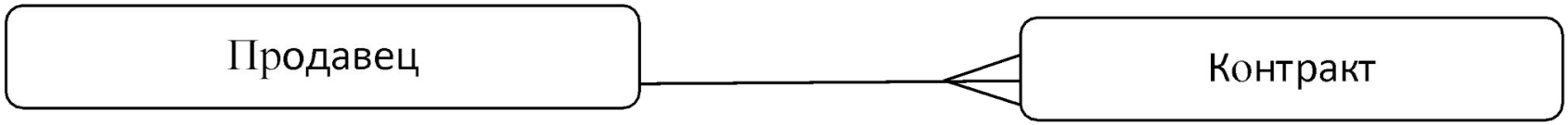


Метод Баркера

Второй шаг моделирования - идентификация связей.

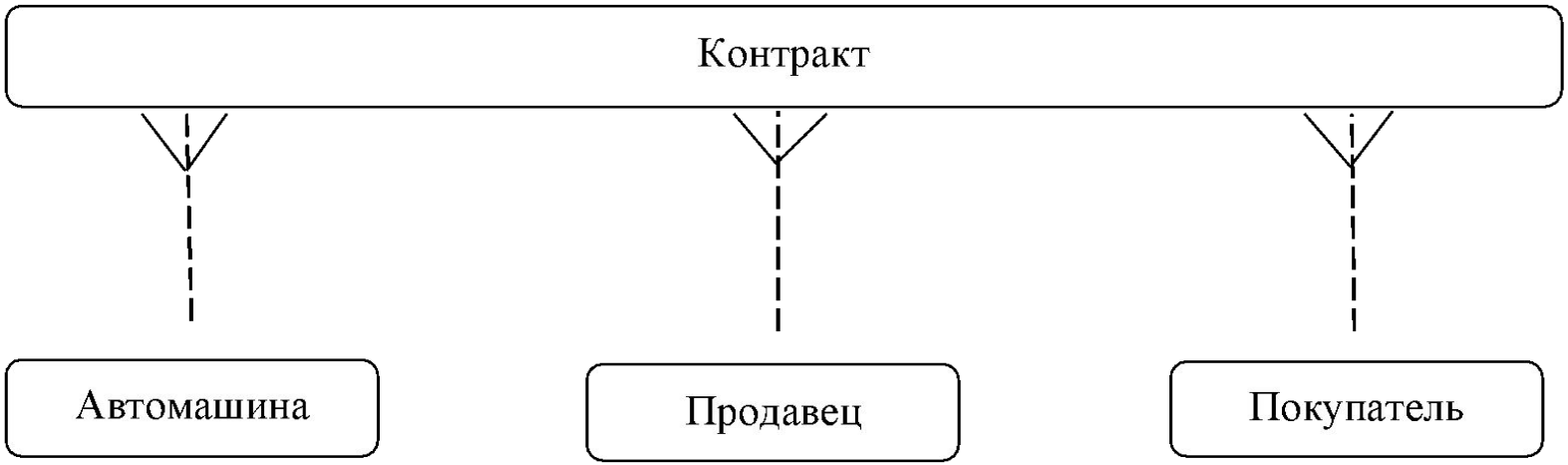
Графическое изображение степени и обязательности связи

Графическое изображение - связь продавца с контрактом



Метод Баркера

Диаграмма «сущность-связь» без атрибутов

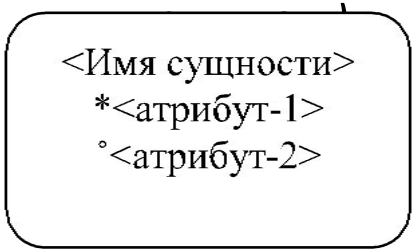


Метод Баркера

Третий шаг моделирования - идентификация атрибутов.

Графическое изображение атрибутов:

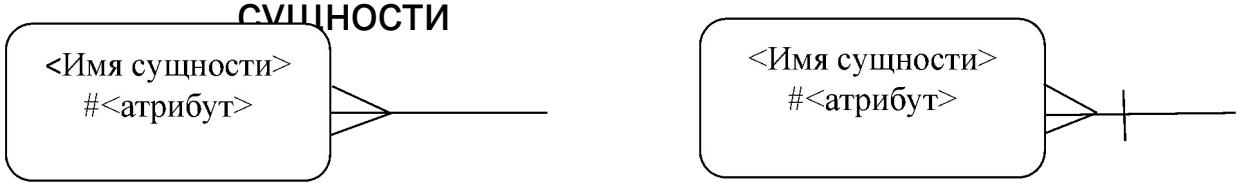
обязательный (помечен звездочкой), необязательный (помечен



Виды идентификации:

а - полная идентификация;

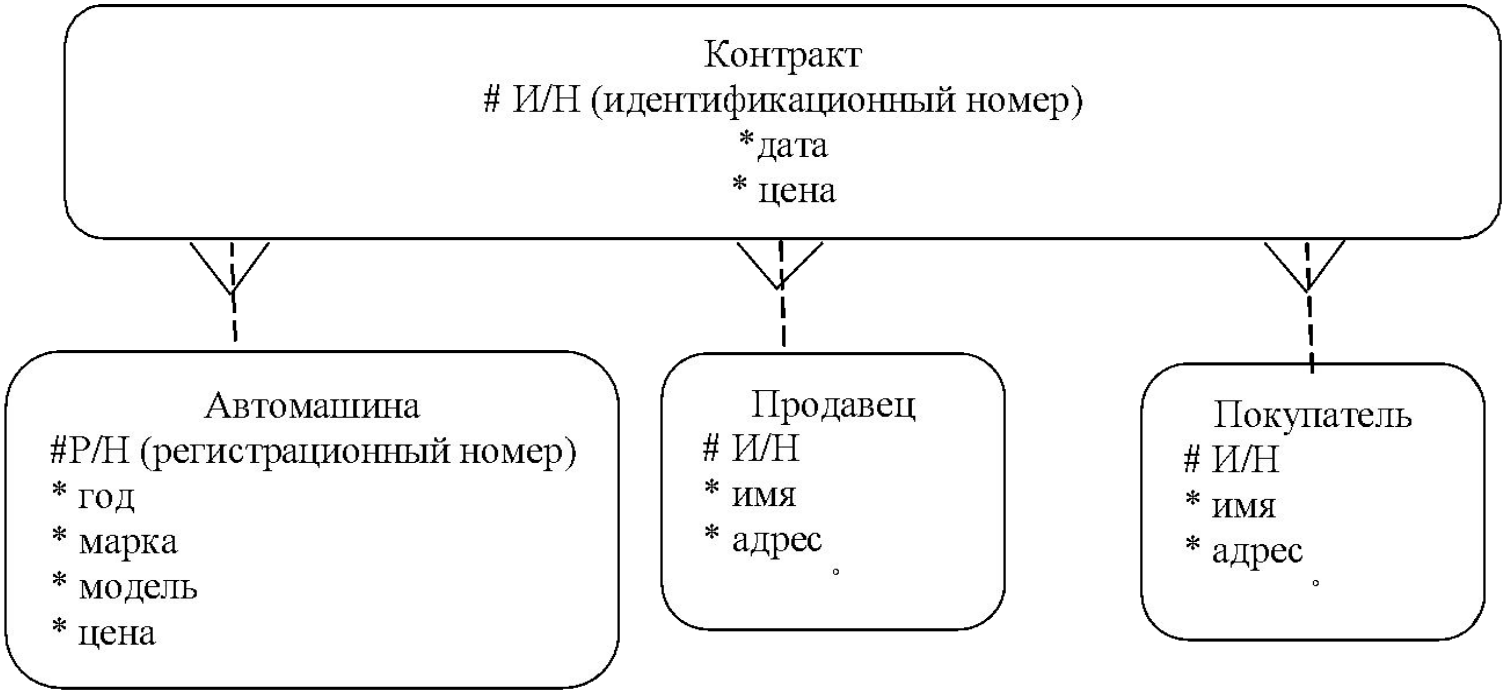
б- идентификация посредством другой сущности



а б

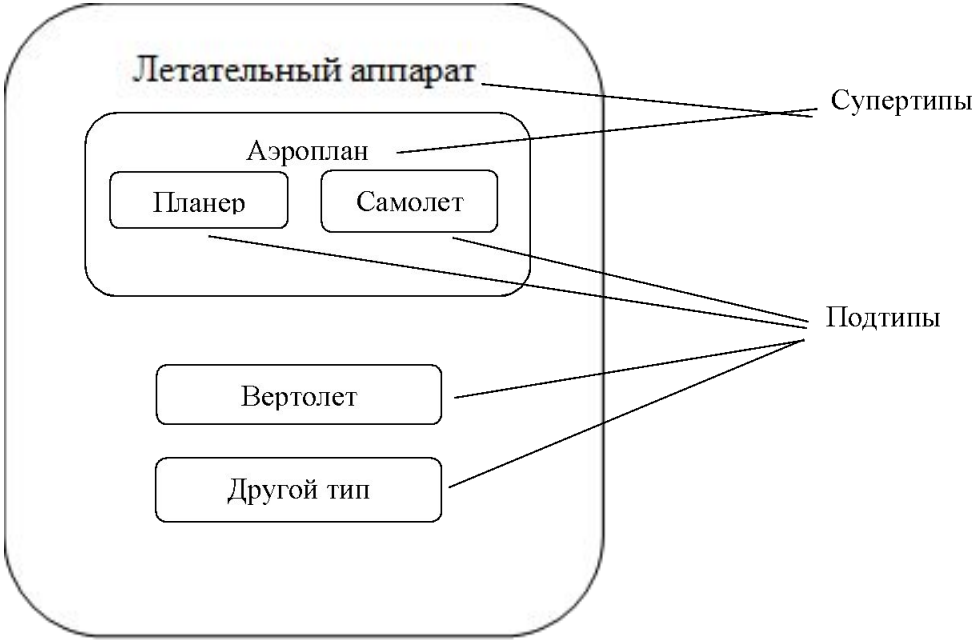
Метод Баркера

Диаграмма «сущность-связь» с атрибутами



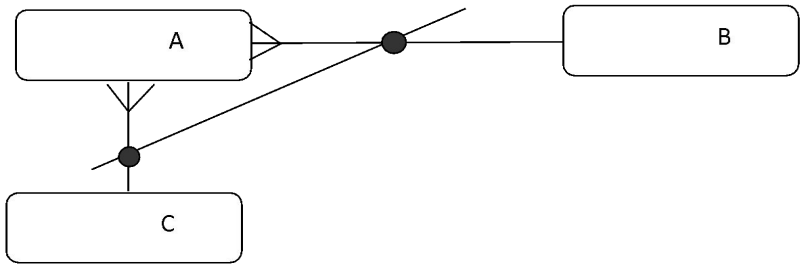
Метод Баркера

Супертипы и подтипы: одна сущность является обобщающим понятием для группы подобных сущностей

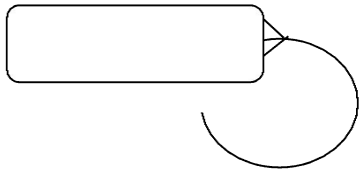


Метод Баркера

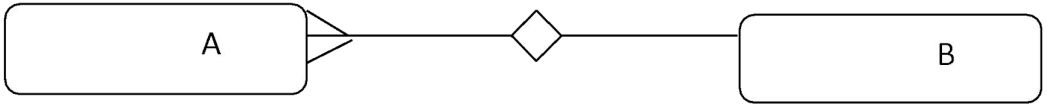
Взаимно исключающие связи: каждый экземпляр сущности участвует только в одной связи из группы взаимно исключающих связей



Рекурсивная связь: сущность может быть связана сама с собой



Неперемещаемые (non-transferrable) связи: экземпляр сущности не может быть перенесен из одного экземпляра связи в другой



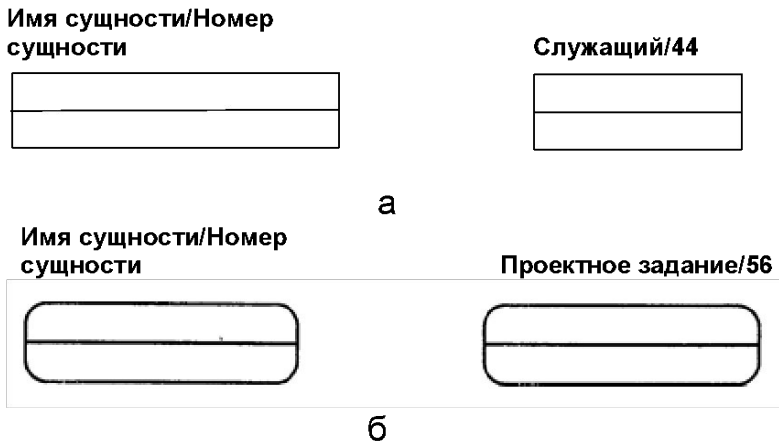
Метод IDEF1X

Метод IDEF1X основан на подходе Чена, позволяет построить модель данных, эквивалентную реляционной модели в третьей нормальной форме.

Сущность является не зависимой от идентификаторов, если каждый экземпляр сущности может быть однозначно идентифицирован без определения его отношений с другими сущностями.

Сущность является зависимой от идентификаторов, если однозначная идентификация экземпляра сущности зависит от его отношения к другой сущности.

Независимые** (а) **и зависимые** (б) **от идентификатора сущности

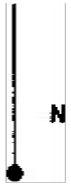


Метод IDEF1X

Степень/мощность связи - количество экземпляров сущности-потомка, которое может существовать для каждого экземпляра сущности-родителя.

Мощность связи может принимать следующие значения:

N - ноль, один или более,
Z - ноль или один,
P - один или более,
фиксированное число.



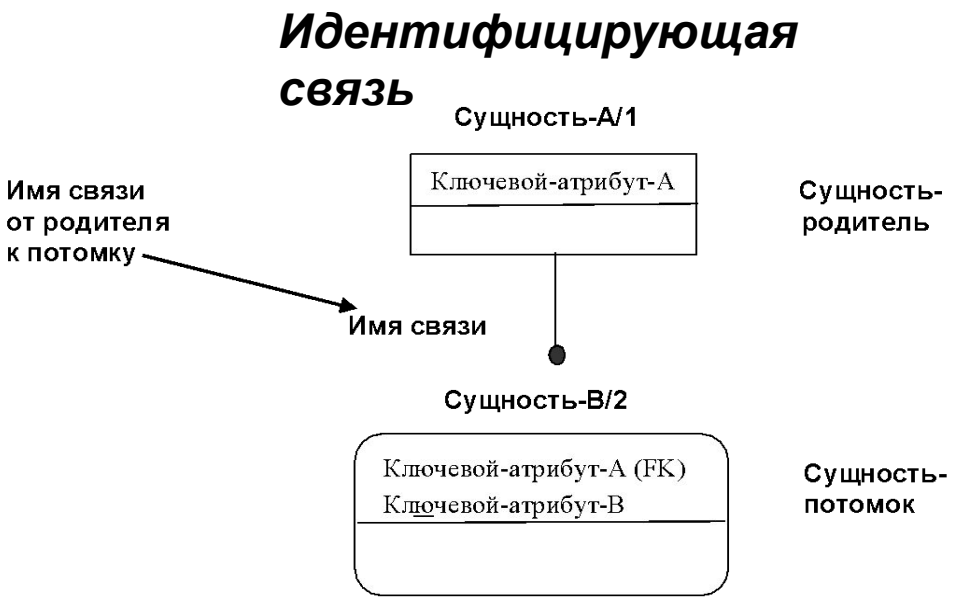
По умолчанию мощность связи принимается равной N:

Метод IDEF1X

Идентифицирующая связь - если экземпляр сущности-потомка однозначно определяется своей связью с сущностью-родителем.

Сущность-потомок в идентифицирующей связи является зависимой от идентификатора сущностью.

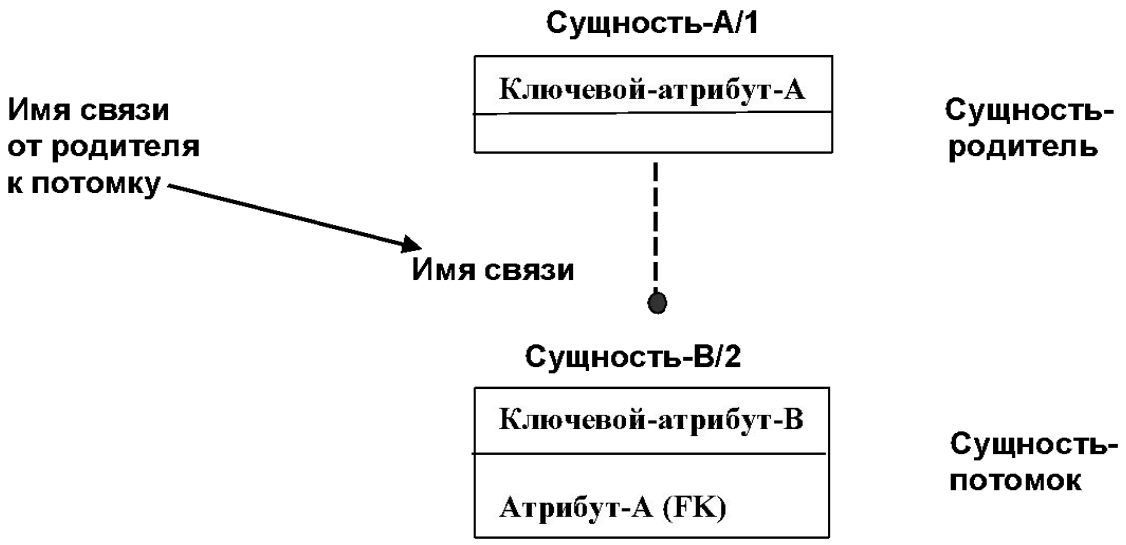
Сущность-родитель в идентифицирующей связи может быть, как независимой, так и зависимой от идентификатора сущностью.



Метод IDEF1X

Неидентифицирующая связь - если экземпляр сущности-потомка не определяется однозначно своей связью с сущностью-родителем

Неидентифицирующая связь



Подход, используемый в CASE-средстве Silverrun

Вариант нотации Чена используется для концептуального моделирования данных на стадии формирования требований

ERD-диаграмма

Подход, используемый в CASE-средстве Silverrun

Графическое представление сущности

Подход, используемый в CASE-средстве Silverrun

Виды идентификаторов:

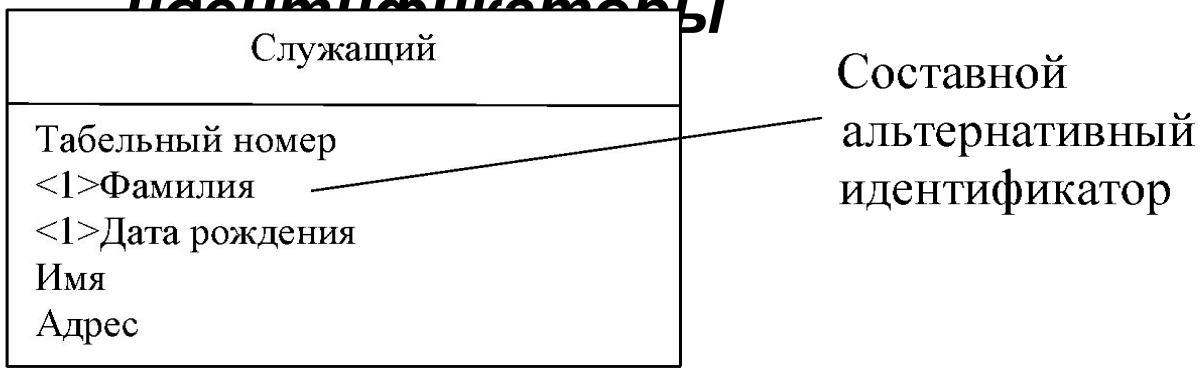
- *первичный/альтернативный:*

Первичный (основной) идентификатор – один, на диаграмме подчеркивается.

Альтернативные идентификаторы предваряются символами <1> для первого альтернативного идентификатора, <2> для второго и т. д.

- *простой/составной:* идентификатор, состоящий из одного атрибута, является простым, из нескольких атрибутов - составным;

Альтернативные идентификаторы



Подход, используемый в CASE-средстве Silverrun

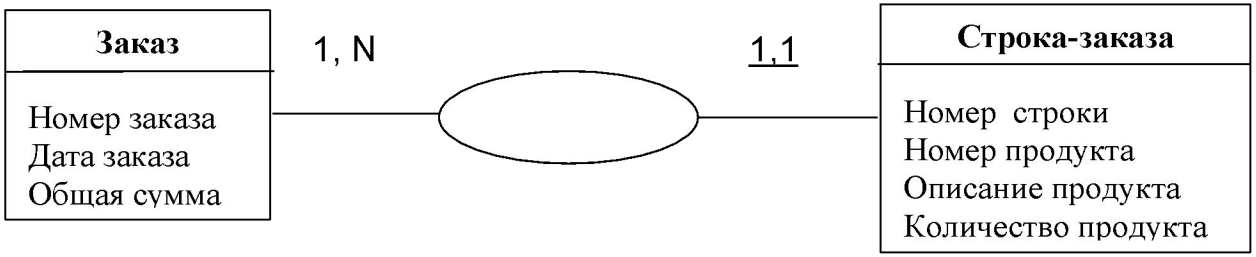
Виды идентификаторов:

- *абсолютный/относительный:*

  – абсолютный идентификатор - если все атрибуты, составляющие идентификатор, принадлежат сущности;

  – относительный идентификатор - если один или более атрибутов идентификатора принадлежат другой сущности.

Относительный идентификатор



Подход, используемый в CASE-средстве Silverrun

Атрибуты связи

Связь между сущностями в концептуальной модели данных является типом, который представляет множество экземпляров связи между экземплярами сущностей.

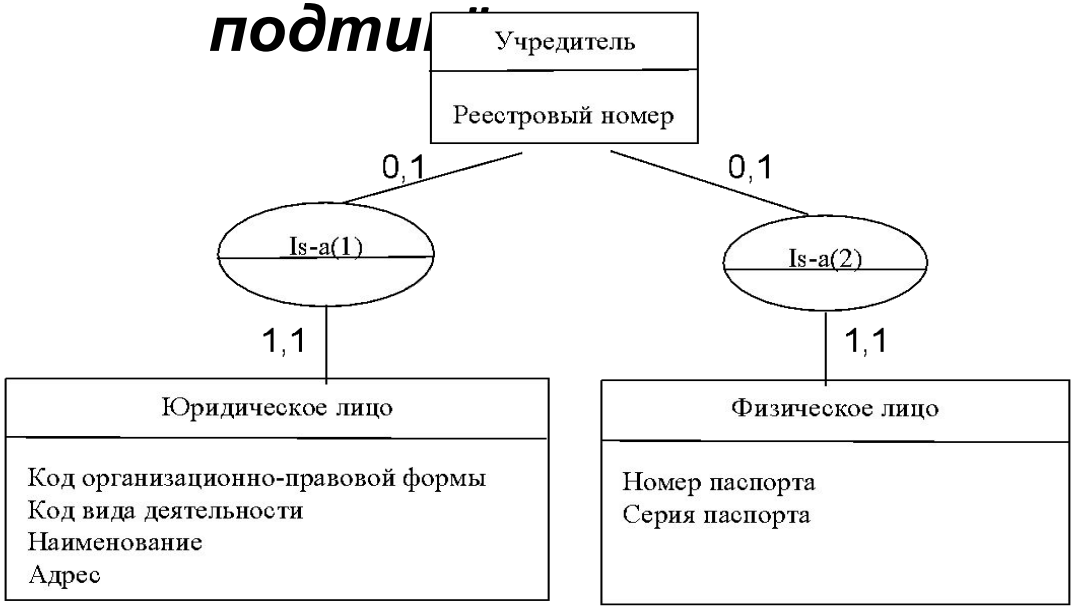
Идентификатор связи

Подход, используемый в CASE-средстве Silverrun

Атрибуты связи

Связь "супертип - подтип" - общие атрибуты типа определяются в сущности - супертипе, сущность-подтип наследует все атрибуты супертипа

Связь "супертип-подти



Алгоритм перехода от ER–модели к реляционной схеме данных

Шаг 1. Каждая простая сущность превращается в таблицу. Имя сущности становится именем таблицы.

Шаг 2. Каждый атрибут становится возможным столбцом с тем же именем. Столбцы, соответствующие необязательным атрибутам, могут содержать неопределенные значения; столбцы, соответствующие обязательным атрибутам, - не могут.

Шаг 3. Компоненты уникального идентификатора сущности превращаются в первичный ключ таблицы. Если в состав уникального идентификатора входят связи, к числу столбцов первичного ключа добавляется копия уникального идентификатора сущности, находящейся на дальнем конце связи.

Алгоритм перехода от ER–модели к реляционной схеме данных

Шаг 4. Связи многие-к-одному и один-к-одному становятся внешними ключами. Необязательные связи соответствуют столбцам, допускающим неопределенные значения; обязательные связи - столбцам, не допускающим неопределенные значения.

Шаг 5. Индексы создаются для первичного ключа (уникальный индекс), внешних ключей и тех атрибутов, на которых предполагается базировать запросы.

Шаг 6. Если в концептуальной схеме присутствовали подтипы, то возможны два способа:

- все подтипы в одной таблице (а);
- для каждого подтипа - отдельная таблица (б) .

Алгоритм перехода от ER–модели к реляционной схеме данных

Все в одной таблице	Таблица - на подтип
Преимущества	
Все хранится вместе Легкий доступ к супертипу и подтипам Требуется меньше таблиц	Более ясны правила подтипов Программы работают только с нужными таблицами
Недостатки	
Слишком общее решение Требуется дополнительная логика работы с разными наборами столбцов и разными ограничениями Потенциальное узкое место (в связи с блокировками) Столбцы подтипов должны быть необязательными В некоторых СУБД для хранения неопределенных значений требуется дополнительная память	Слишком много таблиц Смущающие столбцы в представлении UNION Потенциальная потеря производительности при работе через UNION Над супертипом невозможны модификации

Алгоритм перехода от ER–модели к реляционной схеме данных

Шаг 7. Имеется два способа работы при наличии исключающих связей:

- общий домен (а)
- явные внешние ключи (б)

Если остающиеся внешние ключи все в одном домене (способ (а)) - создаются два столбца:

- идентификатор связи
- и идентификатор сущности.

Если результирующие внешние ключи не относятся к одному домену - для каждой связи создаются явные столбцы внешних ключей.

ФИЗИЧЕСКОЕ ПРОЕКТИРОВАНИЕ БАЗ ДАННЫХ

Особенности построения физической модели базы данных

Физический уровень –отображение логической модели на модель данных конкретной СУБД.

Физическое проектирование - преобразование логической схемы с учетом синтаксиса, семантики и возможностей выбранной целевой СУБД.

Проблемы проектирования базы данных

- *Проблема логического проектирования баз данных*: Каким образом отобразить объекты предметной области в абстрактные объекты модели данных?
- *Проблема физического проектирования баз данных:* Как обеспечить эффективность выполнения запросов к базе данных?

Особенности построения физической модели базы данных

Основные определения элементов физической модели

Физический тип данных – тип данных, характеризующий столбец с данными.

Уникальный индекс первичного ключа – индекс, передающий столбцу в таблице все свойства первичного ключа.

Хранимая процедура - объект базы данных, представляющий собой набор SQL-инструкций, который компилируется один раз и хранится на сервере.

Триггер – хранимая процедура, запускаемая СУБД автоматически, при наступлении определенного в коде хранимой процедуры события.

Внешний ключ – подмножество столбцов некоторой переменной таблицы R2, значения которых должны совпадать со значениями

Особенности построения физической модели базы данных

Термины физической модели

Сущность (концептуальная или логическая модель)	Таблица (физическая модель)
Зависимая сущность	Первичный ключ родителя, как часть первичного ключа потомка
Независимая сущность	Первичный ключ родителя, как неключевой атрибут потомка
Атрибут	Столбец
Логический тип данных (text, number, clob)	Физический тип данных (зависит от СУБД)
Домен (логический)	Домен (физический)
Первичный ключ	Первичный ключ, уникальный кластеризованный индекс первичного ключа
Внешний ключ	Внешний ключ, уникальный некластеризованный индекс внешнего ключа
Альтернативный ключ	Альтернативный ключ, уникальный некластеризованный индекс альтернативного ключа
Бизнес правило	Триггер или хранимая процедура
Связь	Связь, поддерживаемая внешними ключами
Идентифицирующая связь	Первичный ключ родителя становится частью первичного ключа потомка
Неидентифицирующая связь	Первичный ключ родителя становится неключевым атрибутом потомка

Особенности построения физической модели базы данных

Этапы физического проектирования баз данных

1. Проектирование таблиц базы данных с учетом специфики выбранной СУБД.
2. Реализация бизнес-правил в выбранной СУБД.
3. Дальнейшая оптимизация физической модели базы данных.
4. Разработка стратегии обеспечения безопасности информации.
5. Осуществление постоянного мониторинга базы данных и СУБД.

Особенности построения физической модели базы данных

Анализ необходимости введения контролируемой избыточности

Денормализация – снижение требований к уровню нормализации отношений.

Виды денормализации, повышающие производительность системы

1. Использование производных данных:
   - дополнительная стоимость хранения производных данных и поддержки согласованности с текущими значениями исходных данных;
   - издержки на выполнение вычислений значений производных атрибутов при каждом обращении к ним.
2. Дублирование атрибутов.

Особенности построения физической модели базы данных

Анализ необходимости введения контролируемой избыточности

Дублирование атрибутов

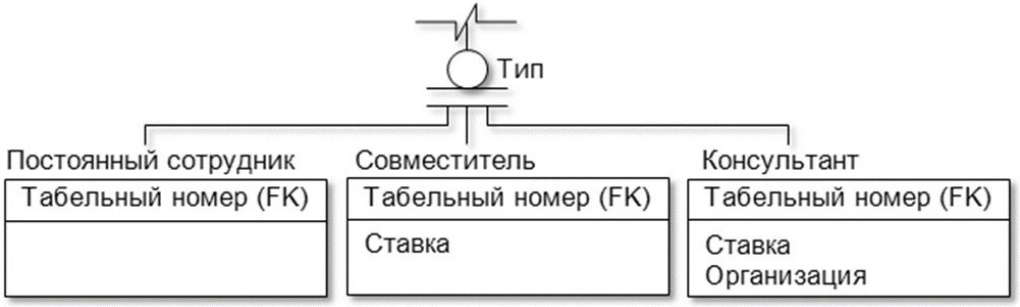
2.1. Объединение отношений, связанных 1:1.

Иерархия наследования

(неполная категория)

Иерархия наследования

(полная категория)



Особенности построения физической модели базы данных

Анализ необходимости введения контролируемой избыточности

Дублирование атрибутов

2.2. Дублирование атрибутов в связях типа 1:M:

- возможность включения атрибута одной таблицы в другую таблицу.

2.3. Использование служебных таблиц:

- значительно снижается вероятность ошибки при указании значений для атрибутов;
- уменьшается размер исходной таблицы;
- при изменении описания параметра значительно проще изменить одно значение в служебной таблице, чем корректировать множество записей в исходной.

2.4. Введение повторяющихся (многозначных) атрибутов.

2.5. Создание сводных таблиц.

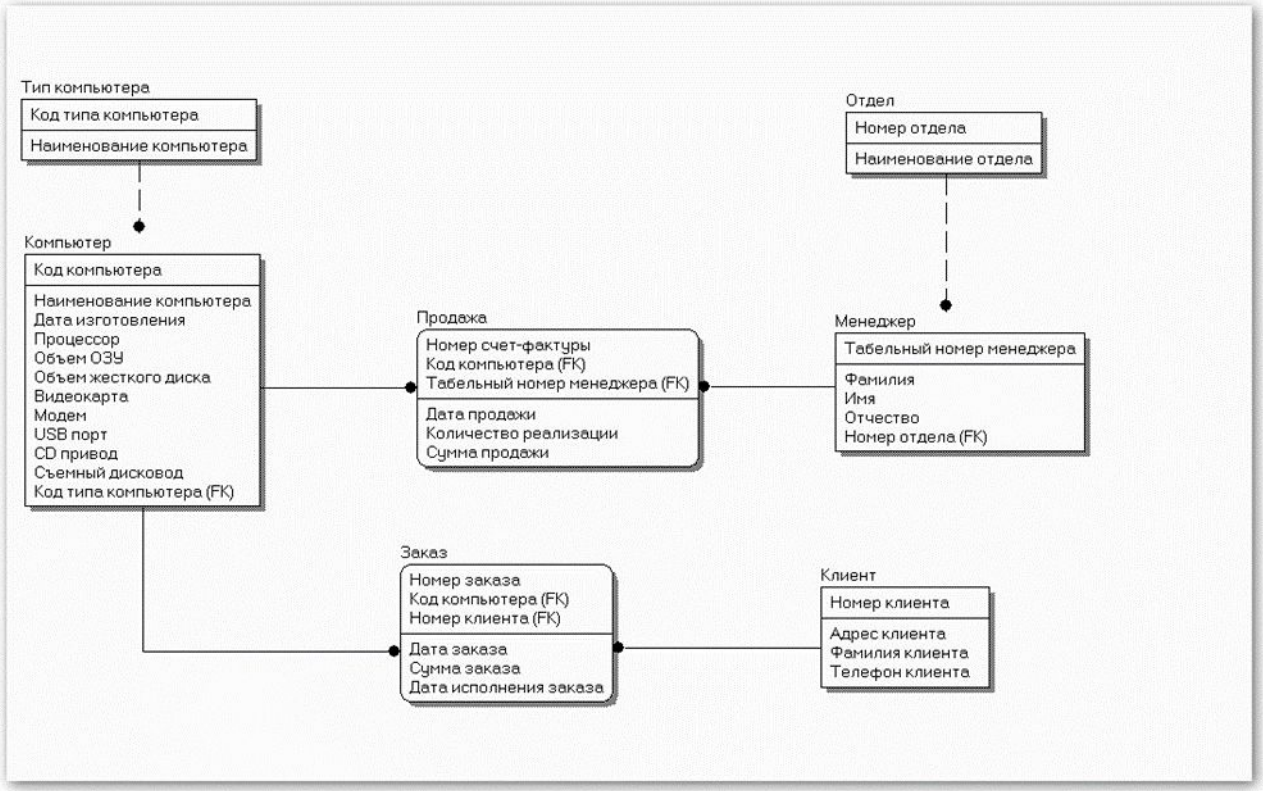
Особенности построения физической модели базы данных

Перенос логической схемы данных в среду целевой СУБД

1. Проектирование таблиц и связей.
2. Задание:
   - доменов;
   - первичных, альтернативных и внешних ключей;
   - неопределенных (NULL) и обязательных (NOT NULL) значений;
   - значений по умолчанию (DEFAULT);
   - правил контроля целостности;
   - хранимых процедур и триггеров.
3. Модификация логической схемы с учетом семантики и синтаксиса, принятой в целевой СУБД.

Особенности построения физической модели базы данных

Логическая модель данных системы "Реализация средств вычислительной



Особенности построения физической модели базы данных

Перенос логической схемы данных в среду целевой СУБД

Правила ссылочной целостности

Правило целостности внешних ключей:

- для каждого значения внешнего ключа должно существовать соответствующее значение первичного ключа в родительском отношении.

Ссылочная целостность может быть нарушена при выполнении операций:

1) обновление кортежа в родительском отношении;
2) удаление кортежа в родительском отношении;
3) вставка кортежа в дочернее отношение;
4) обновление кортежа в дочернем отношении.

Особенности построения физической модели базы данных

Перенос логической схемы данных в среду целевой СУБД

Основные стратегии поддержания ссылочной целостности:

1. RESTRICT – не разрешать выполнение операции, приводящей к нарушению ссылочной целостности.
2. CASCADE – разрешить выполнение требуемой операции, но внести при этом необходимые поправки в других кортежах отношений так, чтобы не допустить нарушения ссылочной целостности и сохранить все имеющиеся связи.

Дополнительные стратегии поддержания ссылочной целостности:

1. NONE – никаких операций по поддержке ссылочной целостности не выполняется.
2. SET NULL – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей заменять на неопределенные значения (null-значения).
3. SET DEFAULT – разрешить выполнение требуемой операции, но все возникающие некорректные значения внешних ключей изменять на

Особенности построения физической модели базы данных

Реализация бизнес-правил и анализ транзакций

После реализации бизнес-правил необходимо проверить выполнимость и эффективность всех транзакций.

Разработка механизмов защиты

Разработка пользовательских представлений

Представление в БД – динамический результат одной или более операций, выполненных над таблицами БД с целью получения новой сводной таблицы. Представление является виртуальной таблицей, которая реально в БД не существует, но создается по запросу (SELECT) определенного пользователя в результате выполнения этого запроса.

Определение прав доступа

Каждый пользователь обладает строго определенным набором прав (привилегий) в отношении конкретной таблицы или представления.

Особенности построения физической модели базы данных

Организация мониторинга и настройка функционирования системы

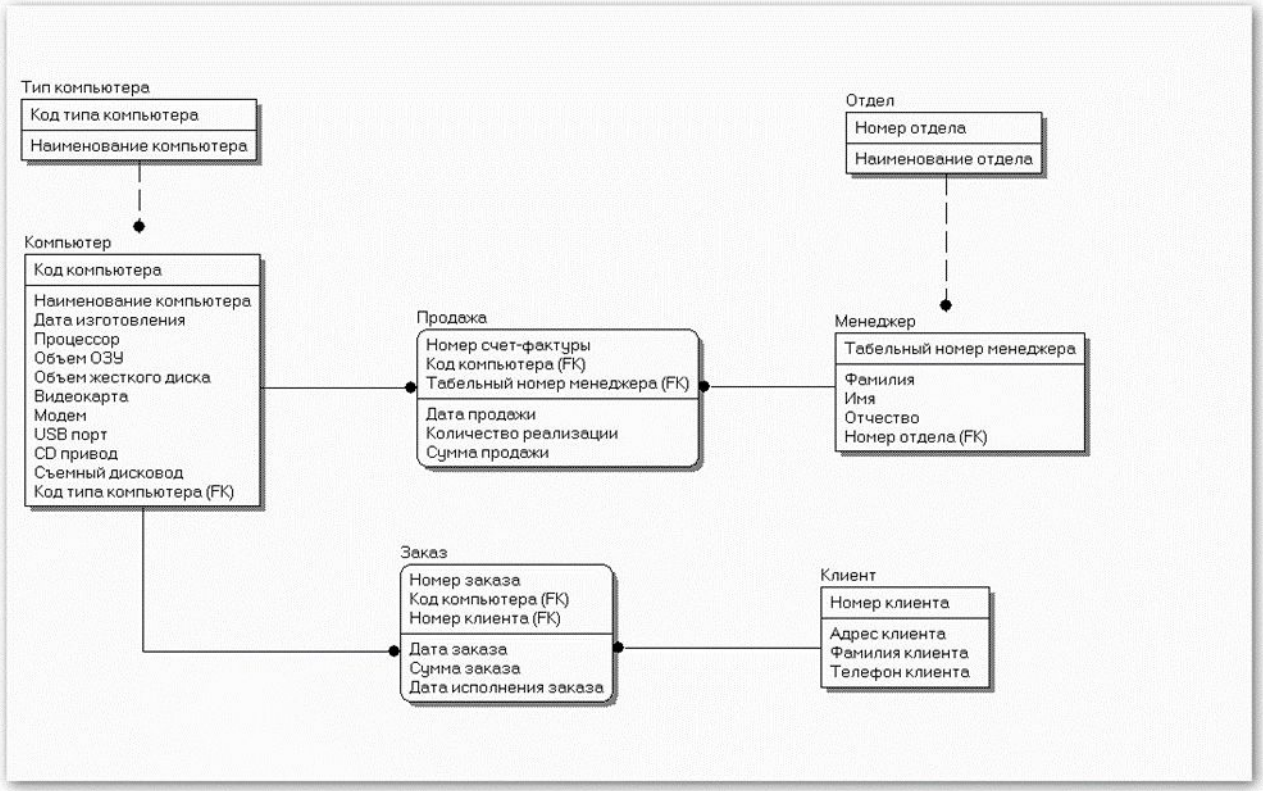
Мониторинг необходим с целью устранения ошибочных проектных решений или изменения требований к системе.

На протяжении всего жизненного цикла системы необходимо постоянно вести наблюдение за уровнем ее производительности, что позволит своевременно реагировать на изменения, происходящие в окружающей среде.

Внесение любых изменений в БД должно проводиться с обязательным их тестированием.

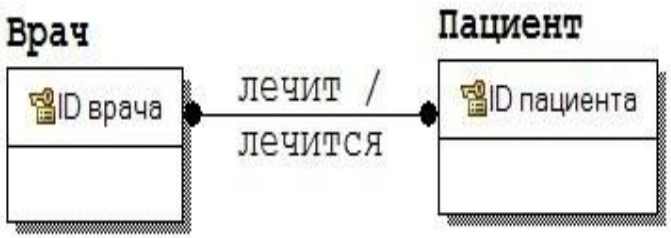
Особенности построения физической модели базы данных

Логическая модель данных системы "Реализация средств вычислительной



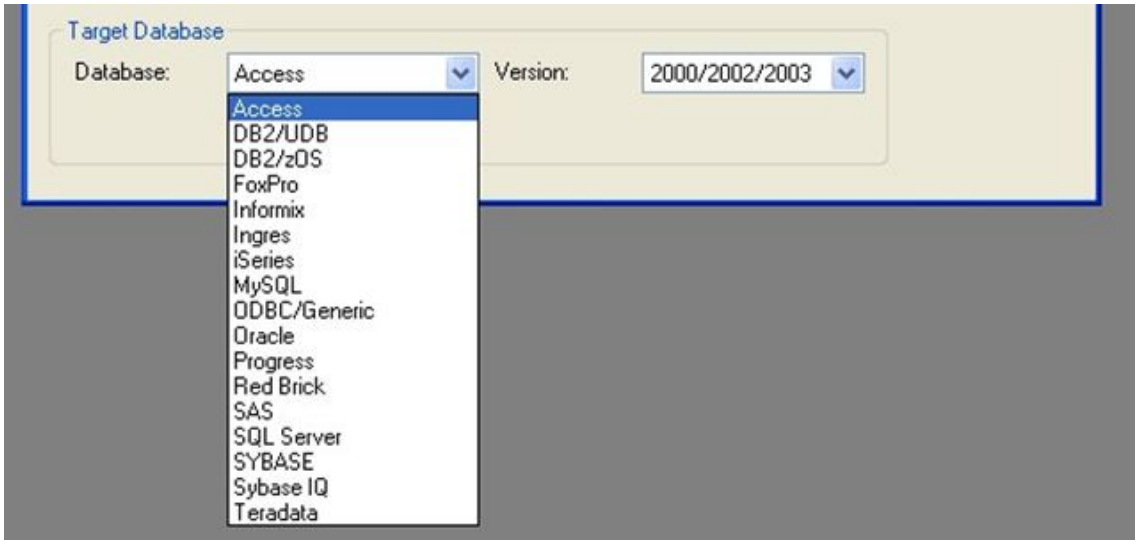
Особенности построения физической модели базы данных

Логическая модель данных системы



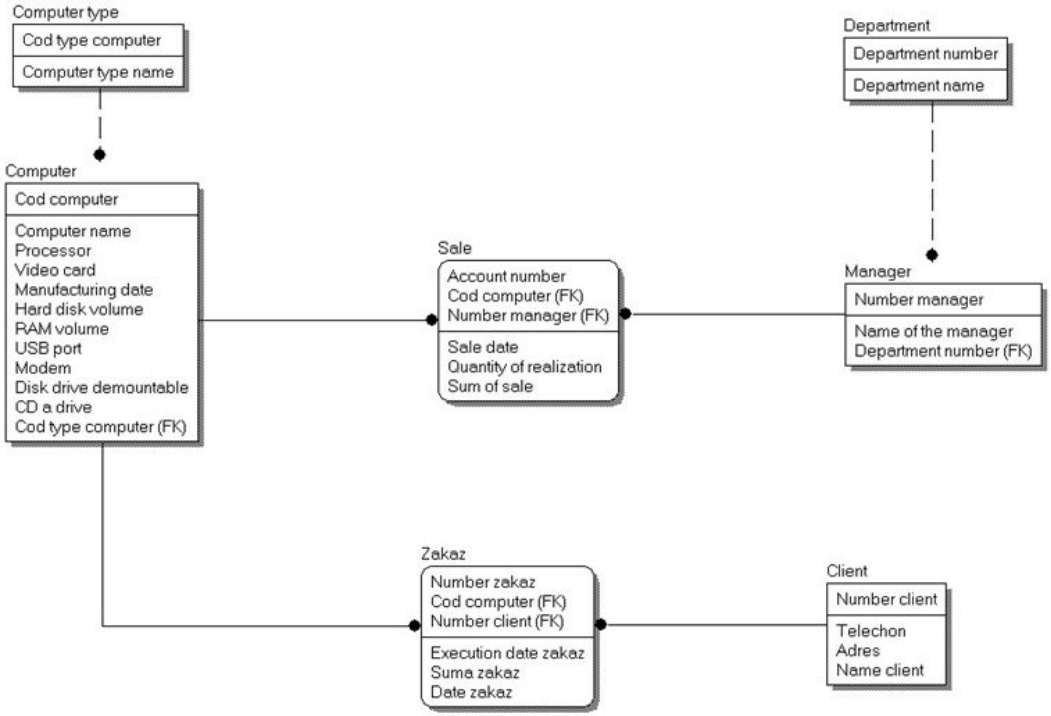
Особенности построения физической модели базы данных

Выбор СУБД Target Database



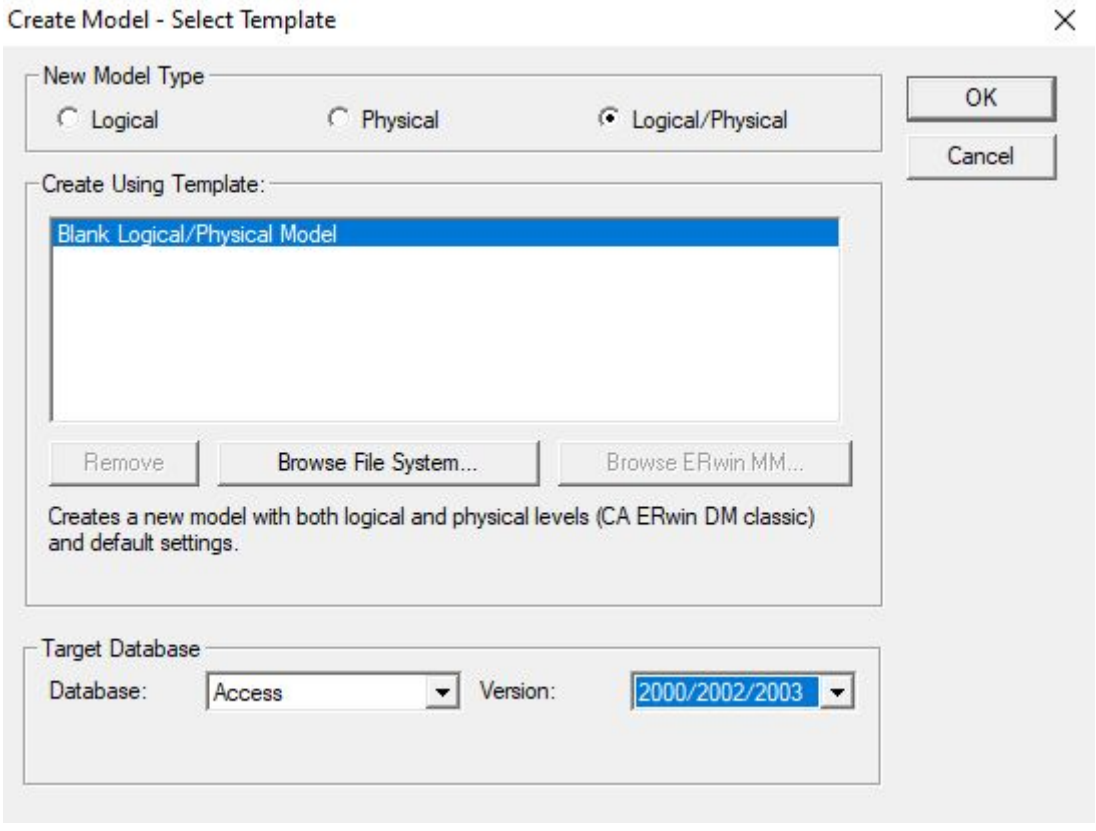
Особенности построения физической модели базы данных

Физическая модель данных системы "Реализация средств вычислительной техники"



Пример построения модели базы данных

Логическая модель данных системы «Деятельность автосалона"



Пример построения модели базы данных

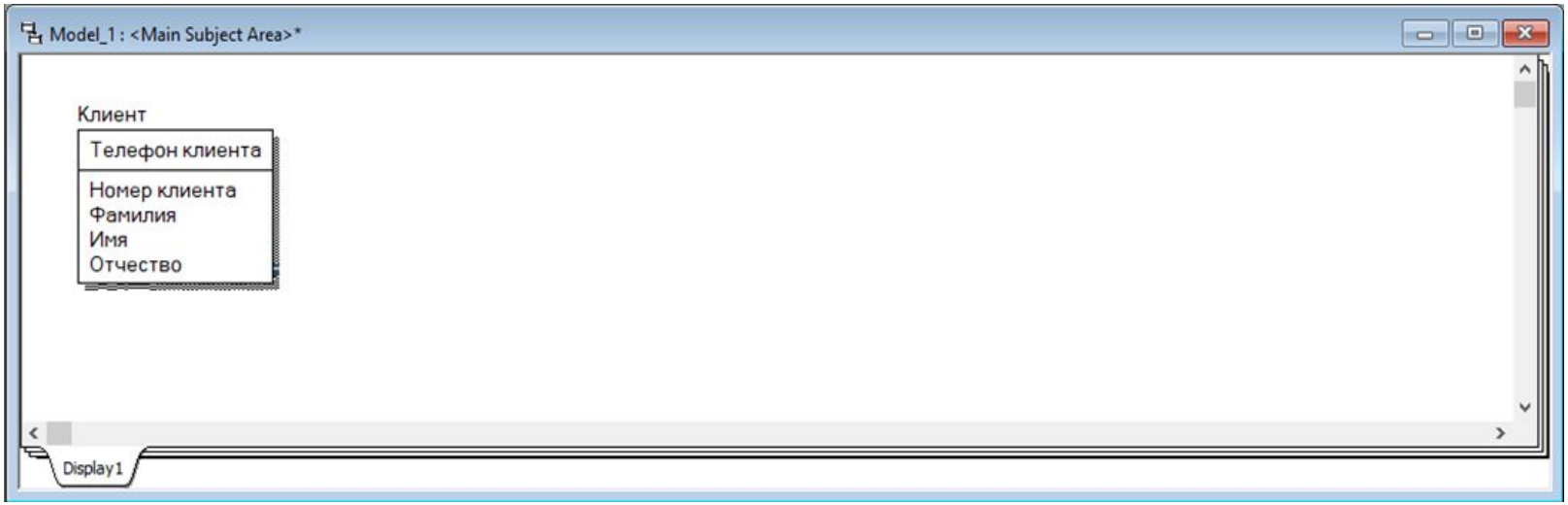
Логическая модель данных системы «Деятельность автосалона"

Создадим новую сущность «Клиент» со следующими атрибутами:

- Номер клиента (Primary Key, Number);
- Фамилия (String);
- Имя (String);
- Отчество (String);
- Телефон клиента (String).

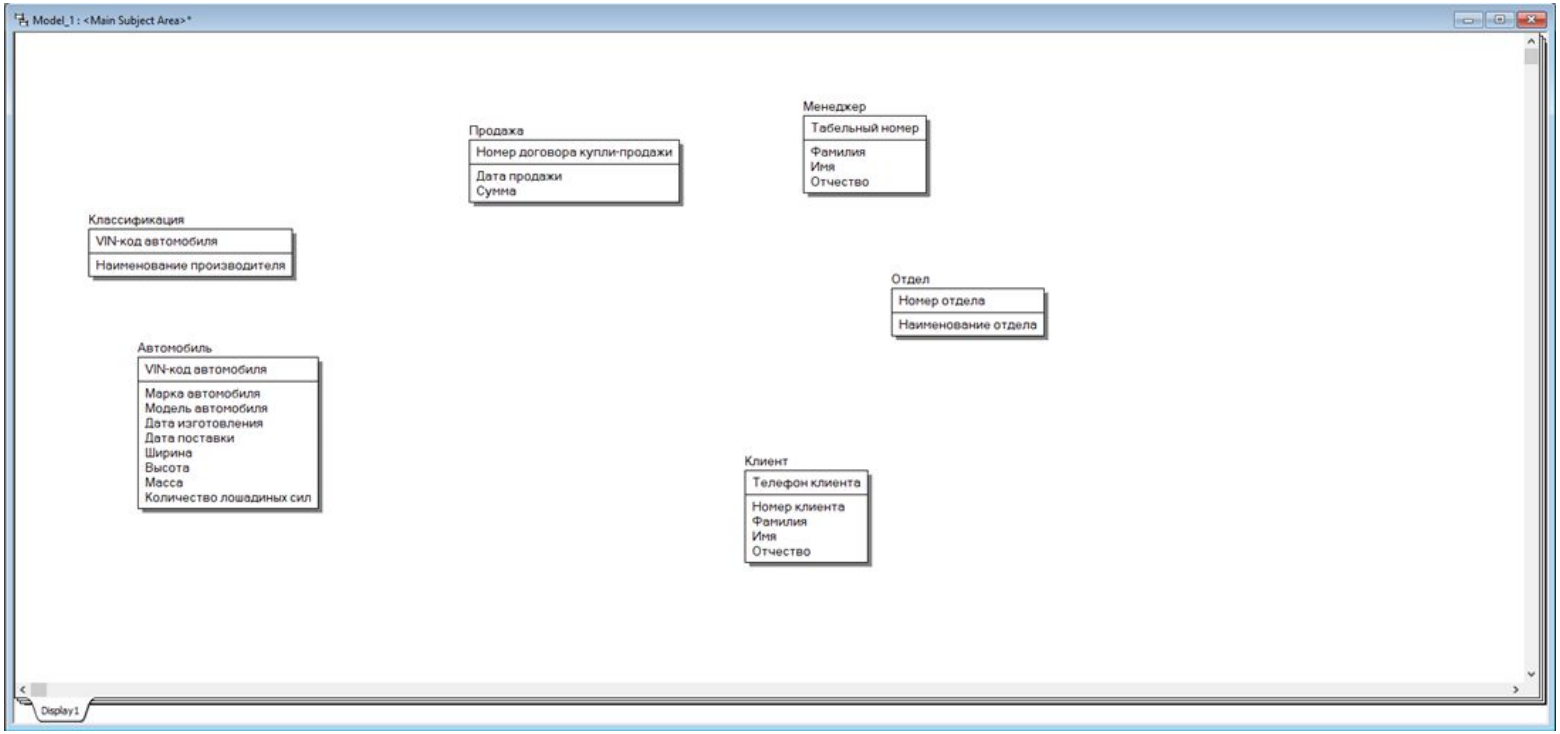
Пример построения модели базы данных

Логическая модель данных системы «Деятельность автосалона"



Пример построения модели базы данных

Логическая модель данных системы «Деятельность автосалона"



Пример построения модели базы данных

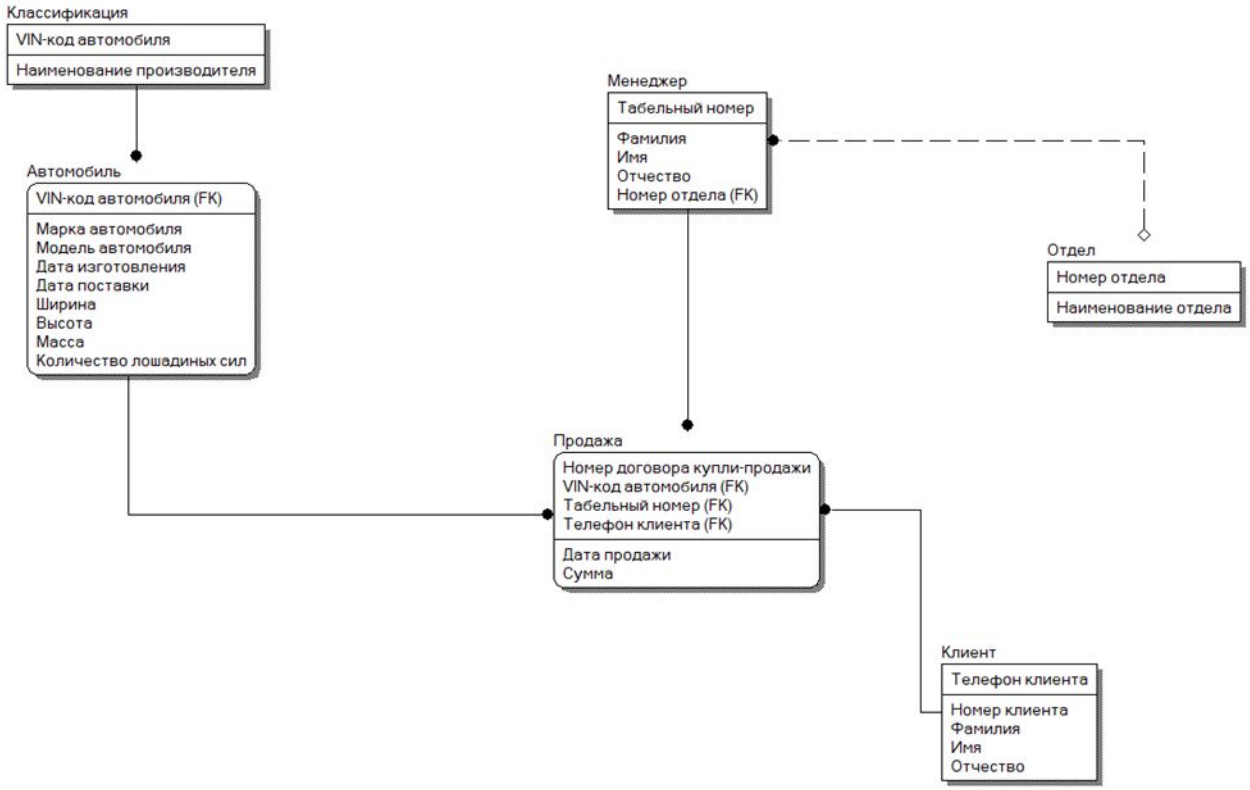
Логическая модель данных системы «Деятельность автосалона"

Установим связи между сущностями:

- классификация → автомобиль (идентифицирующая связь);
- автомобиль → продажа (идентифицирующая связь);
- отдел → менеджер (неидентифицирующая связь);
- менеджер → продажа (идентифицирующая связь);
- клиент → продажа (идентифицирующая связь).

Пример построения модели базы данных

Логическая модель данных системы «Деятельность автосалона"



Спасибо за внимание!