|  |
| --- |
|  |

|  |
| --- |
| **Лабораторная работа «Проектирование базы данных»** |
| *Лабораторный практикум по базам данных* |
|  |
| 1. Концептуальное проектирование 2. Логическое проектирование 3. Физическое проектирование |
|  |
| **ЧувГУ, каф. Вычислительной техники** |
| **23.02.2021** |
|  |

# *Значительная часть проектов в области информационных технологий направлена на разработку и создание информационных систем, в рамках которых осуществляется обработка данных различной сложности. Целью таких проектов является разработка и создание информационной системы с базами данных. Разрабатываемая информационная система должна удовлетворять заданным функциональным и информационным требованиям с учетом заданных ограничений. Проектирование базы данных - это поиск способов удовлетворения функциональных требований средствами имеющейся компьютерной технологии с учетом заданных ограничений.*

Основные задачи проектирования:

* Обеспечение хранения в БД всей необходимой информации.
* Обеспечение возможности получения данных по всем необходимым запросам.
* Сокращение избыточности и дублирования данных.
* Обеспечение [целостности базы данных](https://ru.wikipedia.org/wiki/%D0%A6%D0%B5%D0%BB%D0%BE%D1%81%D1%82%D0%BD%D0%BE%D1%81%D1%82%D1%8C_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85).

# 

# ЗАДАНИЕ

Спроектировать централизованную реляционную базу данных по индивидуальному заданию, взяв за основу предлагаемый ниже порядок разработки. Проектирование включает в себя следующие этапы:

1. *Концептуальное проектирование*
2. *Логическое проектирование*
3. *Физическое проектирование*

## Концептуальное проектирование

**Концептуальное (инфологическое) проектирование** — построение семантической модели предметной области, то есть информационной модели наиболее высокого уровня абстракции. Такая модель создаётся без ориентации на какую-либо конкретную [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94) и [модель данных](https://ru.wikipedia.org/wiki/%D0%9C%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85). Термины «семантическая модель», «концептуальная модель» и «инфологическая модель» являются синонимами.

Кроме того, в этом контексте равноправно могут использоваться слова «модель базы данных» и «модель предметной области» (например, «концептуальная модель базы данных» и «концептуальная модель предметной области»), поскольку такая модель является как образом реальности, так и образом проектируемой базы данных для этой реальности.

Конкретный вид и содержание концептуальной модели базы данных определяется выбранным для этого формальным аппаратом. Обычно используются графические нотации, подобные [ER-диаграммам](https://ru.wikipedia.org/wiki/ER-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%8C_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85#%D0%9D%D0%BE%D1%82%D0%B0%D1%86%D0%B8%D0%B8) (ERD).



Пример концептуальной схемы (*нотация Мартина*)

Основные элементы ER-моделей:

* **объекты (сущности);**
* **атрибуты объектов;**
* **связи между объектами.**

Сущность — объект предметной области, имеющий атрибуты.

Связь между сущностями характеризуется:

* **типом связи** (1:1, 1:N, N:М);
* **классом принадлежности**. Класс может быть обязательным и необязательным. Если каждый экземпляр сущности участвует в связи, то класс принадлежности — обязательный, иначе — необязательный.

***Порядок разработки концептуальной схемы***

1. Определение типов сущностей
2. Определение типов связей
3. Определение атрибутов и связывание их с типами сущностей и связей
4. Определение доменов атрибутов
5. Определение потенциальных и выбор первичных ключей
6. Специализация или генерализация типов сущностей (необязательный шаг)
7. Создание диаграммы «сущность-связь» (ER-диаграммы)
8. **2.1 Первый шаг моделирования** - извлечение информации (из интервью, диаграмм, документов) и **выделение сущностей**

Грубый план проектирования:

* Выпишите из описания предметной области все существительные
* Удалите явно лишние
* Проведите обобщение!!!
* Выявите сущности
* Выделите центральную сущность
  1. **Второй шаг моделирования** — идентификация связей.

|  |  |  |  |
| --- | --- | --- | --- |
| **Сущность 1** | **Описание связи** | **Кратность** | **Сущность 2** |
| ГРУППА | Группа **должна** включать в себя **много** студентов | N, М | СТУДЕНТ |
| Студент **может** входить в состав ровно **одной** группы | 0,1 |
| ЗАКАЗ | Заказ **должен** включать **одну** и **более** строк заказа | 1,М | СТРОКА ЗАКАЗА |
| Строка заказа **должна** относиться ровно к **одному** заказу | 1,1 |
| ВЕДОМОСТЬ | Ведомость **может** содержать **много** строк | 0,М | СТРОКА ВЕДОМОСТИ |
| Строка ведомости **должна** относиться ровно к **одной** ведомости | 1,1 |

* 1. **Третий шаг моделирования** — идентификация атрибутов.

Каждому выявленному атрибуту следует присвоить осмысленное имя, понятное пользователям. О каждом атрибуте в документацию помещаются следующие сведения:

* имя атрибута и его описание
* любые синонимы (алиасы), имеющиеся для данного атрибута
* тип данных и размерность их значения
* значение, принимаемое для данного атрибута по умолчанию (если имеется)
* является ли атрибут обязательным (то есть может ли он отсутствовать или иметь значение NULL)
* является ли атрибут составным и из каких простых атрибутов он состоит
* является ли атрибут производным, и какой метод следует использовать для вычисления его значения.
* является ли данный атрибут множественным.

Каждый атрибут из перечня атрибутов свяжите с некоторой сущностью.

* 1. **Четвертый шаг моделирования -** определение доменов атрибутов. Этот шаг для ACCESS можно пропустить.
  2. **Пятый шаг моделирования.** Для каждой сущности установите потенциальные ключи**,** после чего выберите *первичный* ключ.
  3. **Шестой шаг моделирования.** Если есть необходимость, определите суперклассы и подклассы типов сущностей.
  4. **Седьмой шаг моделирования.** Создайте диаграмму «сущность-связь»

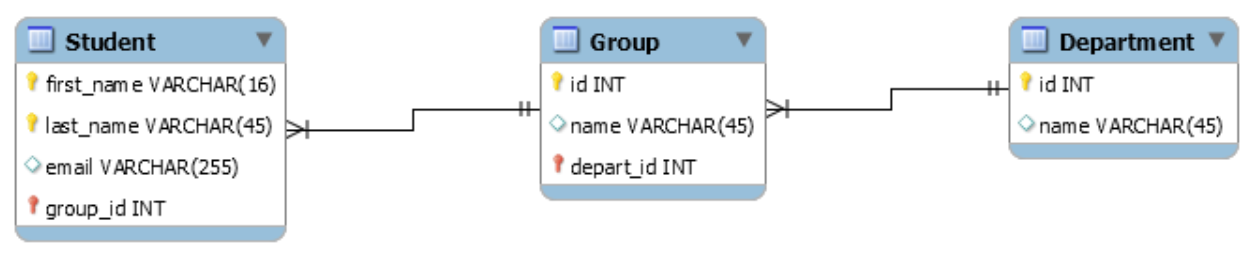
**По окончании данного этапа получаем концептуальную модель, инвариантную к структуре базы данных. Чаще всего она представляется в виде модели "сущность-связь".**

## Логическое проектирование

**Логическое (даталогическое) проектирование** — создание схемы базы данных на основе конкретной модели данных, например, реляционной модели данных. Для реляционной модели данных логическая модель — набор схем отношений, обычно с указанием первичных ключей, а также «связей» между отношениями, представляющих собой внешние ключи.

Преобразование концептуальной модели в логическую модель, как правило, осуществляется по формальным правилам. Этот этап может быть в значительной степени автоматизирован.

На этапе логического проектирования учитывается специфика конкретной модели данных, но может не учитываться специфика конкретной СУБД.

****

*Пример логической схемы для реляционной модели данных*  
(*нотация Мартина*)

* 1. Предварительно необходимо решить вопрос – однопользовательская или многопользовательская БД создается? При разработке многопользовательской БД требуется предусмотреть создание подсхем.
  2. Создайте поэтапно логическую модель базы данных:
     1. Преобразуйте концептуальную модель в логическую модель.
     2. Определите набор отношений исходя из структуры логической модели.
     3. Проверьте модель с помощью правил нормализации.
     4. Проверьте модель в отношении транзакций пользователя. Для этого используйте перечень запросов, сформированных на этапе анализа предметной области. Если какую-либо транзакцию выполнить вручную не удастся, значит составленная модель данных не адекватна и содержит ошибки, которые потребуется устранить.
     5. Создайте диаграмму «сущность-связь». Отобразите на этой диаграмме миграцию первичных ключей.
     6. Определите требования поддержки целостности данных. Поместите сведения обо всех установленных ограничениях целостности данных в словарь данных.

*Описание декларативных правил поддержки целостности БД*

***Для полей***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **№** | **Отношение** | **Атрибут** | **Условие на значение** | **Сообщение об ошибке** |
| 1 | Репертуар | Дата | <Date() |  |
| 2 | Служащий | Дата рождения | >=#01.01.1930# And <#01.01.2021# | Ошибка! Очень молод или очень стар! |
|  |  |  |  |  |

***Для записей***

|  |  |  |  |
| --- | --- | --- | --- |
| **№** | **Отношение** | **Условие на значение** | **Сообщение об ошибке** |
| 1 | Актер | Not ([Разряд]<14 And Not ([Звание]="")) | Разряд актера не соответствует его званию |
|  |  |  |  |
|  |  |  |  |

* 1. На примере разработанной БД покажите ER-диаграмму, не приведенную
* ко 2–й нормальной форме (2НФ)
* к 3–й нормальной форме (3НФ)

## Физическое проектирование

Физическая модель данных описывает данные средствами конкретной СУБД. Мы будем считать, что физическая модель данных реализована средствами именно реляционной СУБД, хотя это необязательно. Отношения, разработанные на стадии формирования логической модели данных, преобразуются в таблицы, атрибуты становятся столбцами таблиц, для ключевых атрибутов создаются уникальные индексы, домены преображаются в типы данных, принятые в конкретной СУБД.

Ограничения, имеющиеся в логической модели данных, реализуются различными средствами СУБД, например, при помощи индексов, декларативных ограничений целостности, триггеров, хранимых процедур.

**Физическое проектирование —** создание [схемы базы данных](https://ru.wikipedia.org/wiki/%D0%A1%D1%85%D0%B5%D0%BC%D0%B0_%D0%B1%D0%B0%D0%B7%D1%8B_%D0%B4%D0%B0%D0%BD%D0%BD%D1%8B%D1%85) для конкретной [СУБД](https://ru.wikipedia.org/wiki/%D0%A1%D0%A3%D0%91%D0%94). Специфика конкретной СУБД может включать в себя ограничения на именование объектов базы данных, ограничения на поддерживаемые типы данных и т. п. Кроме того, специфика конкретной СУБД при физическом проектировании включает выбор решений, связанных с физической средой хранения данных (выбор методов управления дисковой памятью, разделение БД по файлам и устройствам, методов доступа к данным), создание индексов и т. д. Результатом физического проектирования логической схемы на языке SQL может являться **скрипт**.

***Порядок разработки физической схемы***

1. Перенос глобальной логической модели данных в среду целевой СУБД

* Проектирование основных таблиц в среде целевой СУБД
* Разработка способов получения производных данных
* Реализация бизнес-правил в среде целевой СУБД

1. Проектирование физического представления базы данных

* Анализ транзакций
* Выбор файловой структуры
* Выбор индексов
* Оценка потребности в дисковой памяти

1. Проектирование пользовательских представлений
2. Разработка механизмов защиты
3. Организация мониторинга и настройка функционирования системы

В рамках данной лабораторной работы мы не будем подробно останавливаться на всех этапах физического проектирования.

После того, как проект базы данных создан, наступает **этап реализации проекта**.

**ТРЕБОВАНИЯ К ОТЧЕТУ**

1. Концептуальная модель данных
2. Логическая модель данных
3. Диаграмма функциональных зависимостей универсального отношения (выборочно)
4. Пример ER-диаграммы, не приведенной ко 2–й нормальной форме (2НФ)
5. Пример ER-диаграммы, не приведенной к 3–й нормальной форме (3НФ)

**КОНРОЛЬНЫЕ ВОПРОСЫ**

1. Дайте определение 1НФ, 2НФ, 3НФ, НФБК, 4НФ
2. Назовите основные правила перехода от концептуальной к логической реляционной схеме.
3. Как в реляционной модели реализуются супертипы и подтипы?
4. Как в реляционной модели реализуются рекурсивные связи?
5. Как в реляционной модели реализуются взаимно исключающие связи?
6. Какие шаблоны проектирования относятся к шаблонам 1:М?
7. Какие шаблоны проектирования относятся к темпоральным шаблонам?
8. Назовите шаблоны проектирования иерархических структур.
9. Как реализуется семантическая целостность?
10. Приведите примеры нарушения семантическая целостность на уровне отношений.
11. Какие связи недопустимы для реляционной логической модели? По-другому: от каких связей избавляются при переходе от концептуальной модели к реляционной логической?

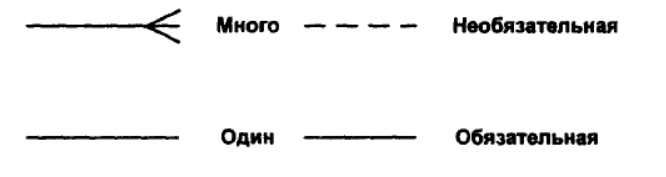
**ИСТОЧНИКИ (рекомендуемая литература)**

1. **Википедия** <https://ru.wikipedia.org/wiki/Проектирование_баз_данных>
2. [**https://studfile.net/preview/1644616/page:3/**](https://studfile.net/preview/1644616/page:3/)

# ПРИЛОЖЕНИE

### Приложение А. Нотация Баркера

Степень и обязательность связи можно показать графически



*Степень и обязательность связи*

*Изображение выглядит как стол

Автоматически созданное описание*

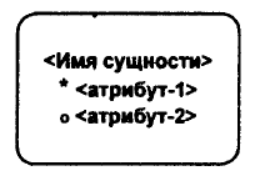
*Кратность или кардинальность связи*

Имя связи всегда формируется с точки зрения родителя, так что может быть образовано предложение соединением имени сущности-родителя, имени связи, выражения степени и имени сущности-потомка.

Например, связь продавца с контрактом может быть выражена следующим образом:

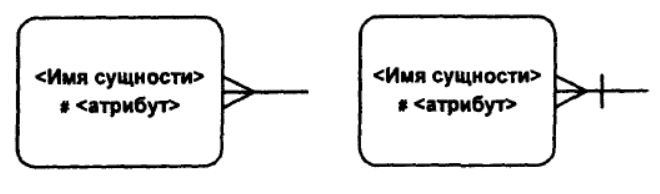
* продавец **может** получить вознаграждение за **один** контракт **или более**;
* контракт **должен** быть инициирован ровно **одним** продавцом

Атрибут может быть либо обязательным, либо необязательным. Обязательность означает, что атрибут не может принимать неопределенных значений (*null values*).



*Обязательный (помечен звездочкой) и необязательный (помечен кружком) атрибуты*

Атрибут может быть либо описательным (т. е. обычным дескриптором сущности), либо входить в состав уникального идентификатора (первичного ключа). **Уникальный идентификатор** — это атрибут или совокупность атрибутов и/или связей, предназначенная для уникальной идентификации каждого экземпляра данного типа сущности. Сущность, как правило, может быть полностью идентифицирована своим первичным ключом. Однако, могут быть случаи, когда в идентификации данной сущности участвуют также атрибуты другой сущности-родителя. В случае полной идентификации каждый экземпляр данного типа сущности полностью идентифицируется своими собственными ключевыми атрибутами (рис. а ), в противном случае в его идентификации участвуют также атрибуты другой сущности-родителя (рис. б ). Идентифицирующая связь изображается поперечной линий на стороне вороньей лапки.



а б

*Виды идентификации: а - полная идентификация;*

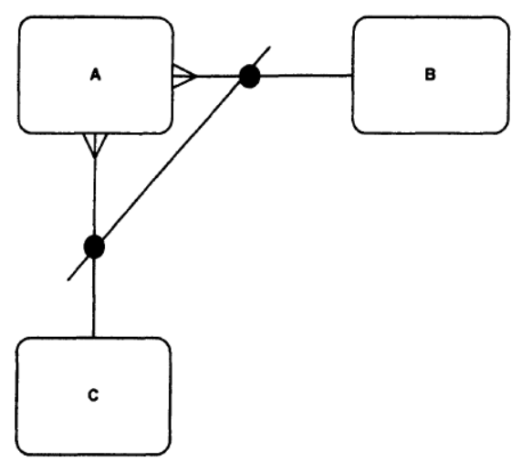
*б — идентификация посредством другой сущности*

Понятия категории и общей сущности Чена заменяются Баркером на эквивалентные понятия *подтипа* и *супертипа* соответственно. Подтипы наследуют все свойства супертипа (в том числе и первичный ключ!), а также участвуют во всех отношениях, в которых участвует супертип. Обратное не верно.

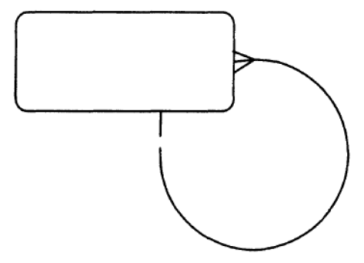
**Супертипы и подтипы**: одна сущность является обобщающим понятием для группы подобных сущностей.

|  |  |
| --- | --- |
|  |  |

**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



**Ассоциированная сущность**



### [Изображение выглядит как текст Автоматически созданное описание](#_Системный_анализ_предметной)

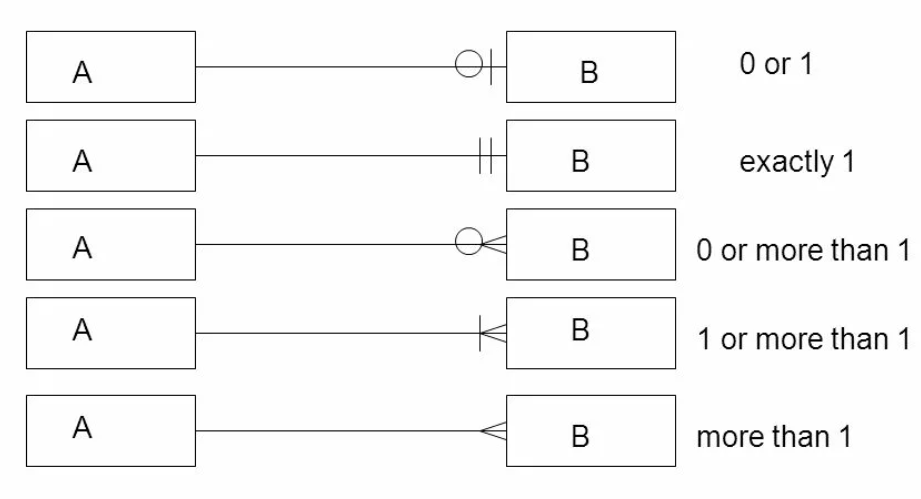
### Приложение Б. Нотация Мартина

Три символа используются для представления количества элементов:

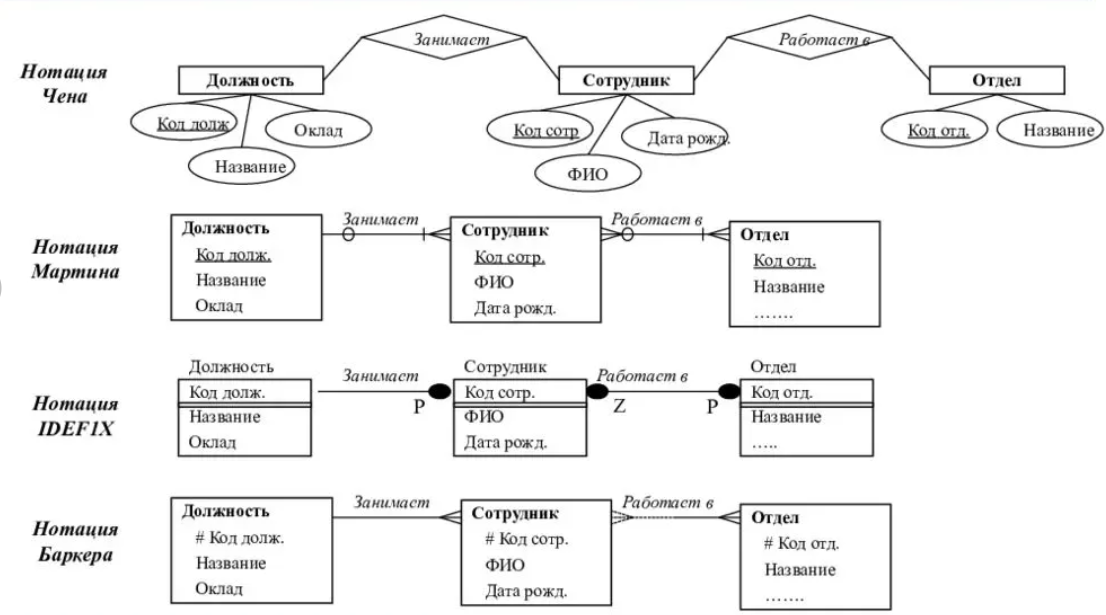
1. кольцо представляет «нулевой»
2. прочерк представляет собой «один»
3. в лапке вороньей представляет «много» или «бесконечные»

Эти символы используются в парах для представления **четырех типов кратности**, которые могут быть у сущности в отношениях. Внутренний компонент обозначения представляет минимум, а внешний компонент представляет максимум.

* **кольцо и тире** → минимум ноль, максимум один (необязательно)
* **тире и тире** → минимум один, максимум один (обязательно)
* **кольцо и гусиная лапка** → минимум ноль, максимум много (необязательно)
* **тире и гусиная лапка →** минимум один, максимум много (обязательно)



**Пример использования разных нотаций**



[Изображение выглядит как текст

Автоматически созданное описание](#_Системный_анализ_предметной)

### Приложение В. Диаграммы и нотации

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Стадии** | **Модели** | | **Диаграммы** | **Методологии (нотации)** | |
| **Анализ  предметной области** | ***Информационная модель предметной области*** | | | | |
| *Модели  данных* | Диаграммы "*сущность -связь*" (ERD) | | CHEF, Martin, Bachman, IDEFX1X, Shlaer & Mellor, Merise, IEM | |
| Диаграммы модели данных (DMD) | | Martin, Bachman | |
| Диаграммы структур данных (DSD) | | Jackson | |
| Диаграммы логических структур данных (LDS) | | SSADM | |
| Диаграммы UML | | OOA&D | |
| ***Функциональные модели предметной области*** | | | | |
| *Модели  процессов* | Диаграммы модели бизнес-процессов (*контекстная диаграмма*, *диаграмма декомпозиции*, *диаграмма дерева узлов* | | | IDEF0, IDEF3 |
| Диаграммы потоков данных  (DFD) | | | Yuordan/DeMarco, Gane & Sarson, SSADM |
| Графы преобразований | | | Ward & Mellor, Gane & Sarson, Hatley |
| Диаграммы UML | | | OOA&D |
| *Модели  состояний* | Диаграммы состояний (STD) | | | Ward & Mellor, Hatley |
| Диаграммы жизненного цикла (ELH) | | | SSADM |
| Диаграммы UML | | | OOA&D |
| **Проектирование** | *Модели  процессов проектирования* | Структурные схемы (STC) | | | Youtdan/Constantine Page-Jones |
| Диаграммы UML | | | OOA&D |
| **Реализация** |  | Диаграммы UML | | | OOA&D |

Как видно из таблицы, не существует единой модели, в рамках которой можно представить весь жизненный цикл системы. Так, стадия анализа предметной области системы представляется с помощью трех типов моделей:

1. **модели данных**, которые описывают внутренние, логические взаимоотношения между данными в системе;
2. **модели процессов**, которые описывают процессы обработки или преобразования данных в системе;
3. **модели состояний**, которые описывают поведение или отклики системы на события.

Важно понимать, что информационная и функциональная модели предметной области создаются на этапе анализа требований к базе данных и не содержат предположений о технологии реализации базы данных. Они строятся независимо от выбираемой модели данных (сетевой, иерархической, реляционной, объектно-ориентированной, многомерной и т.д.), поддерживаемой СУБД, модели вычислений, программно-аппаратной платформы для базы данных. **Информационная и функциональная модели предметной области являются входными данными для процесса проектирования базы данных**.

[Изображение выглядит как текст

Автоматически созданное описание](#_Системный_анализ_предметной)