# Практическая работа №17

Проектирование информационной модели предметной области в нотации IDEF1X с помощью MS Visio

# Цель работы

Целью работы является освоение технологии построения информационной модели логического и физического уровней в нотации IDEF1X с использованием пакета Microsoft Visio.

# Задачи

Основными задачами практической работы являются:

* приобретение студентами навыков построения информационной модели логического уровня;
* нормализации полученной модели;
* построения информационной модели физического уровня.

# Краткие теоретические сведения

## Понятие информационной модели. Уровни информационной модели

Методология IDEF1X – язык для семантического моделирования данных, основанный на концепции

«сущность-связь».

Различают два уровня информационной модели: **логический** и **физический**.

**Логическая модель** позволяет понять суть проектируемой системы, отражая логические взаимосвязи между сущностями.

Различают 3 подуровня логического уровня модели данных, отличающиеся по глубине представления информации о данных:

* диаграмма сущность-связь (Entity-Relationship Diagram (ERD);
* модель данных, основанная на ключах (Key Based Model (KB);
* полная атрибутивная модель (Fully Attributed Model (FA).

**Физическая модель** отражает физические свойства проектируемой базы данных (типы данных, размер полей, индексы). Параметры физической информационной модели зависят от выбранной системы управления базами данных (СУБД).

## Основные элементы информационной модели логического уровня

* + 1. *Сущности и атрибуты*

**Сущность** – это множество **реальных** или **абстрактных объектов** (людей, предметов, документов и т.п.), **обладающих общими атрибутами или характеристиками**. Любой объект системы может быть представлен только одной сущностью, которая должна быть уникально идентифицирована. *Именование сущности* осуществляется с помощью *существительного в единственном числе*. При этом имя сущности должно отражать **тип** или **класс** объекта, а не его **конкретный экземпляр** (например, **Студент**, а не **Петров**) (рис. 1).

Студент

Фамилия Имя Отчество

Дата поступления Номер билета

ID студента

Рисунок 1 – Графическое представление сущности «Студент» в MS Visio

Любая сущность характеризуется набором атрибутов (**свойств**).

**Атрибут сущности** – характеристика сущности, то есть свойство реального объекта. Например, на рис. 1 атрибутами сущности **«Студент»** являются **«ID студента»**, **«Фамилия»**, **«Имя»**, **«Отчество»**, **«Дата поступления»** и **«Номер билета»**.

В свою очередь, *атрибуты сущности* делятся на 2 вида: *собственные* и *наследуемые. Собственные* атрибуты являются уникальными в рамках модели. *Наследуемые* атрибуты передаются от сущности-родителя при установке связи с другими сущностями.

**Первичный ключ (Primary Key, PK).** Каждая сущность должна обладать *атрибутом* или *комбинацией атрибутов*, чьи значения *однозначно определяют* каждый *экземпляр сущности*. Эти атрибуты образуют *первичный ключ* сущности.

**Внешний ключ (Foreign Key, FK)**. Если между двумя сущностями *имеется специфическое отношение* связи или *категоризации*, то *атрибуты*, входящие в *первичный ключ родительской* или *общей сущности*, *наследуются* в качестве *атрибутов сущностью-потомком* или *категориальной сущностью* соответственно. Эти атрибуты и

называются внешними ключами. Наследуемый атрибут может использоваться в сущности в качестве части или целого первичного ключа, альтернативного ключа или не ключевого атрибута.

* + 1. *Отношения в IDEF1X-модели*

При построении информационной модели различают следующие типы отношений между сущностями:

*идентифицирующее*, *не идентифицирующее*, *не специфическое (многие-ко-многим) и отношения категоризации*.

**Мощность отношения** служит для обозначения отношения числа экземпляров родительской сущности к числу экземпляров дочерней.

## Нормализация данных

*Нормализация* – это процесс проверки и реорганизации сущностей и атрибутов с целью удовлетворения требований к реляционной модели данных. Процесс нормализации сводится к последовательному приведению структур данных к нормальным формам – формализованным требованиям к организации данных.

Первая нормальная форма (1НФ). Сущность находится в первой нормальной форме тогда и только тогда, когда все атрибуты содержат атомарные значения. Среди атрибутов не должно встречаться повторяющихся групп, т.е. несколько значений для каждого экземпляра.

Вторая нормальная форма (2НФ). Сущность находится во второй нормальной форме, если она находится в первой нормальной форме, и каждый не ключевой атрибут полностью зависит от первичного ключа (не может быть зависимости от части ключа).

Третья нормальная форма (3 НФ). Сущность находится в третьей нормальной форме, если она находится во второй нормальной форме и никакой не ключевой атрибут не зависит от другого не ключевого атрибута (не должно быть зависимости между не ключевыми атрибутами).

# Рекомендации по выполнению практического занятия

Практическая работа выполняется группой студентов (2-3 человека) в пакете Microsoft Visio.

Данное занятие может выполняться на основе результатов функционального моделирования предметной области.

Отчет по практическому занятию выполняется в формате MS Word, который содержит экранные формы моделей согласно заданию.

# Методика выполнения практического занятия

**Упражнение 1. Построение логической информационной модели уровня «сущность-связь»**

## Составление пула – списка потенциальных сущностей

Информационная модель может быть построена на основе функциональной модели или без нее. Использование функциональной модели в качестве основы для информационного моделирования позволяет создать структуру базы данных, полностью соответствующей функциям предприятия. Названия всех интерфейсных дуг функциональной модели (выполненной в нотации IDEF0) заносятся в *пул* – список потенциальных сущностей. Только в данном случае информационная модель будет адекватна выполняемым функциям. Функциональная модель для рассматриваемого примера представлена в *приложении А*.

Список потенциальных сущностей (при использовании программного продукта MS Office Visio для функционального моделирования) должен быть составлен вручную. В случае использования CASE-средства *AllFusion Process Modeler* отчет по интерфейсным дугам генерируется автоматически. Список потенциальных сущностей для рассматриваемого примера будет представлен таблицей вида (рис. 2).

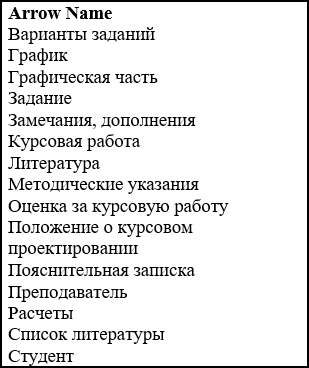


Рисунок 2 – Пул – список потенциальных сущностей

Теперь из этого списка необходимо выделить сущности, остальные интерфейсные дуги будут преобразованы в атрибуты сущностей.

В качестве сущностей выделим следующие:

1. задание;
2. пояснительная записка;
3. курсовая работа;
4. положение о курсовом проектировании;
5. студент;
6. преподаватель;
7. график;
8. методические указания.

## Создание логической модели «сущность-связь»

1. Запустите MS Visio.
2. На закладке выбора шаблона выберите категорию *Программное обеспечение и базы данных* и в ней элемент

*Схема модели базы данных*. Нажмите кнопку *Создать* в правой части экрана.

1. Установите необходимые параметры страницы (масштаб, ориентация страницы).
2. MS Visio поддерживает различные нотации моделей баз данных. Для того чтобы задать нотацию IDEF1X, необходимо выбрать пункты меню База данных → Параметры → Документ. В открывшемся окне на вкладке Общие установить переключатель в меню Набор символов на IDEF1X. Меню Имена, видимые на схеме, позволяет указать, какие имена атрибутов сущности будут отображены на диаграмме (концептуальные, физические или оба варианта одновременно). В данном случае для логического представления информационной модели необходимо выбрать пункт *Концептуальные имена* (рис. 3).

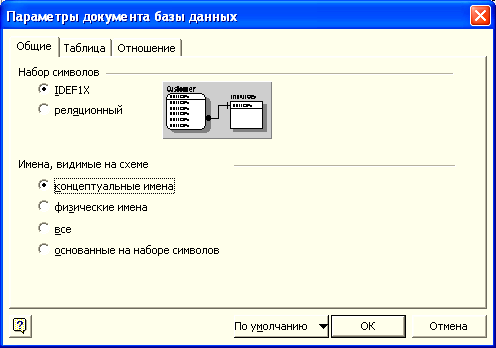


Рисунок 3 – Настройка параметров модели

В закладке *Отношение* окна *Параметры документа базы данных* в меню *Показывать* отметить галочкой пункт *Мощность*, в меню *Отображение вида* выбрать пункт *Показывать вербальную фразу*, снять галочку в пункте *Обратный текст* (рис. 4). Данные настройки позволят отобразить имя и мощность связи в модели.

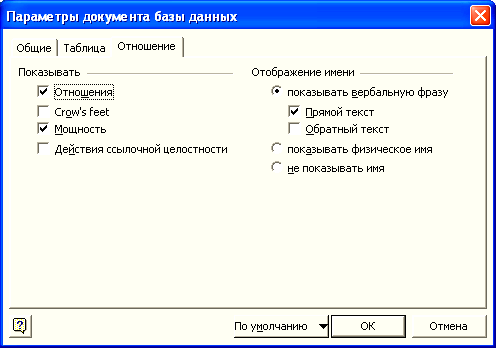


Рисунок 4– Настройка вида отношений информационной модели

1. Для того чтобы создать сущность, необходимо перетащить элемент на рабочее поле. Переход в режим редактирования сущности осуществляется двойным щелчком по сущности или по нажатию правой кнопки мыши и выбора пункта меню *Свойства базы данных*.



Чтобы задать имя сущности, в окне *Свойства базы данных* нужно выбрать категорию *Определение*, снять галочку в пункте *Синхронизация имен при вводе* (в противном случае, физическое и логическое имя сущности будут совпадать, что по практическим соображениям не всегда удобно) и задать концептуальное имя сущности. Руководствуясь данным алгоритмом, создадим 8 сущностей, определенных в пункте 5.1 (см. рис. 5).

Студент Преподаватель

Задание Курсовая работа Пояснительная записка

Методические указания Положение о курсовом проектировании

График

Рисунок 5– Сущности информационной модели логического уровня

1. Далее необходимо установить связи между сущностями.

Сначала составим описание предметной области на естественном языке. Любой студент должен выполнить одну или несколько курсовых работ. Каждая курсовая работа должна выполняться одним студентом (в идеале).

Каждая курсовая работа выполняется в соответствии с методическими указаниями и положением о курсовом проектировании.

Курсовая работа сдается по графику.

Курсовая работа оформляется в виде пояснительной записки.

Преподаватель проводит консультации, проверяет и ставит оценку за курсовую работу. Таким образом, сформулируем имена связей:

***СТУДЕНТ*** выполняет ***КУРСОВУЮ РАБОТУ***.

***ПРЕПОДАВАТЕЛЬ*** проверяет ***КУРСОВУЮ РАБОТУ***.

***КУРСОВАЯ РАБОТА*** выполняется в соответствии с ***ЗАДАНИЕМ***.

***КУРСОВАЯ РАБОТА*** оформляется в виде ***ПОЯСНИТЕЛЬНОЙ ЗАПИСКИ***. ***МЕТОДИЧЕСКИЕ УКАЗАНИЯ*** определяют требования к ***КУРСОВОЙ РАБОТЕ***.

***КУРСОВАЯ РАБОТА*** организуется согласно ***ПОЛОЖЕНИЮ ПО КУРСОВОМУ ПРОЕКТИРОВАНИЮ***. ***КУРСОВАЯ РАБОТА*** сдается по ***ГРАФИКУ***.

Во всех случаях сущность *Курсовая работа* является дочерней, за исключением связи с сущностью *Пояснительная записка*. Определим типы связей и построим модель (см. рис. 6). В дальнейшем можно будет подкорректировать связи между сущностями.

Чтобы установить **связи** между сущностями, необходимо перетащить на рабочую область элемент , поднести один конец стрелки к родительской сущности, другой – к дочерней.

Примечание. При правильном связывании каждая сущность будет подсвечена красным цветом.

В MS Office Visio по умолчанию используется *не идентифицирующее* отношение. Чтобы изменить **тип связи**, необходимо двойным щелчком по связи открыть окно *Свойства базы данных* и в категории в категории *Прочее* указать тип отношения (идентифицирующее, не идентифицирующее). В этой же категории указывается мощность связи (см. рис. 6).

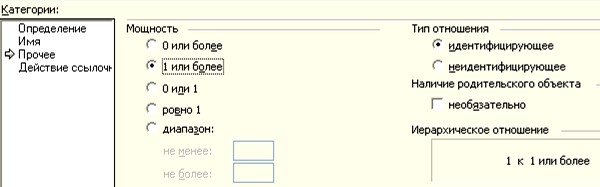


Рисунок 6 – Определение типа связи и мощности

Примечание. Кроме того, при не идентифицирующем отношении нужно указать, является ли наличие родительской сущности обязательным (т.е. может ли существовать экземпляр дочерней сущности, если не существует экземпляра родительской). Если наличие родительского объекта является необязательным, графически это отобразится в виде не закрашенного ромба со стороны родительской сущности.

Следующий шаг – в категории *Имя* в поле *Вербальная фраза* нужно указать имя отношения (рис. 7). Также можно указать имя связи в поле *Обратная фраза* для спецификации отношения потомок-родитель (в нашем случае обратная фраза отображаться не будет).

Примечание. Все изменения при закрытии окна свойств сохраняются автоматически.

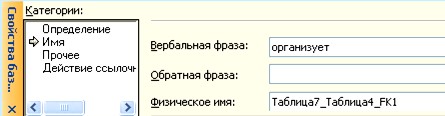


Рисунок 7 – Определение имени отношения

После определения имен, типов связей и задания мощностей получим информационную модель, представленную на рис. 8.

Студент Преподаватель



проверяет

Задание

выполняет

P

определяет

P

P Пояснительная записка Курсовая работа

~~оформ~~ляется

P

1

определяет требованияP

P

организует

определяет дату сдачи

Положение о курсовом проектировании

Методические указания

График

Рисунок 8 – Информационная модель уровня «сущность-связь»

# Упражнение 2. Разработка логической модели данных, основанной на ключах

1. Необходимо определить ключевые атрибуты для каждой сущности, обращая внимание на то, что дочерние сущности наследуют ключевые атрибуты от родительских (см. рис. 10).

Для этого двойным щелчком мыши по сущности откроем окно редактирования ее свойств, перейдем в категорию *Столбцы*, по нажатию кнопки *Добавить* введем имя поля (например, для сущности *Задание* ключевым атрибутом будет являться *Вариант задания*). Чтобы сделать атрибут ключевым, необходимо отметить галочкой пункт *РК* (рис. 9). Данное поле становится обязательным автоматически.



Рисунок 9 – Определение ключевого атрибута

Аналогичным образом зададим ключевые атрибуты для всех сущностей информационной модели. Результат представлен на рис. 10.

Как видно из рисунка 10 по сравнению с информационной моделью уровня «сущность-связь», был изменен тип связи между сущностями *Методические указания* и *Курсовая работа*, поскольку ключевые атрибуты сущности *Методические указания* для сущности *Курсовая работа* будут являться избыточными (зная номер зачетной книжки, можно узнать специальность и курс, на котором учится студент).

Студент

Преподаватель



Курсовая работа

P

Вариант задания (FK) проверяет

№ зачетной книжки (FK)

выполняет

Задание

P

Дисциплина (FK,FK)

табельный номер (FK)

Вуз (FK)

Год издания (FK)

Группа (FK)

~~оформляется~~

1

Пояснительная записка

Вариант задания (FK) Дисциплина (FK)

№ зачетной книжки (FK)

Код дисциплины (FK) Специальность (FK)

Курс (FK)

P

Положение о курсовом проектировании

организует

определяет

P

Методические указания

имеет

P

Определяет дату сдачи

График

Группа Дисциплина

Дисциплина Специальность

Курс

Вуз Год издания

Вариант задания Дисциплина

№ зачетной книжки

табельный номер

Рисунок 10 – Информационная модель с ключевыми атрибутами

1. Кроме того, отметим, что три сущности (Задание, График, Методические указания) содержат одинаковые атрибуты Дисциплина. Это является некорректным. Чтобы устранить данную ошибку, выделим одноименную сущность и свяжем ее идентифицирующими связями с вышеуказанными сущностями (рис. 11).

Студент

Преподаватель

проверяет

выполняет

P

P

Курсовая работа

Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

Пояснительная записка

Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

оформляется

1

График

Код дисциплины (FK)

Группа

определяет дату сдачи P

табельный номер (FK)

Вуз (FK)

Год издания (FK)

Группа (FK) Специальность (FK)

Курс (FK)

Положение о курсовом проектировании

P

организует

определяет

P

Дисциплина

определяет

имее

т

Задание

Код дисциплины (FK) Вариант задания

определяет

Методические указания

имеет

Специальность

Курс Код дисциплины (FK)

Код дисциплины

Вуз Год издания

табельный номер

№ зачетной книжки

Рисунок 11 – Скорректированная информационная модель, основанная на ключах

# Упражнение 3. Создание полной атрибутивной модели

Для того чтобы получить полную атрибутивную модель, необходимо дополнить сущности не ключевыми атрибутами. Дополненная модель представлена на рисунке 11.

Примечание. Если атрибут не является обязательным, нужно убедиться, что в окне *Свойства базы данных* в категории *Столбцы* в пункте *Обязательное* не стоит галочка. Не обязательные к заполнению атрибуты справа от имени имеют пометку *(О)*.

# Упражнение 4. Нормализация полной атрибутивной модели

1. Проверим, все ли атрибуты имеют атомарные значения, т.е. среди атрибутов не должно встречаться повторяющихся групп, нескольких значений для каждого экземпляра (например, номер телефона\_1, номер телефона\_2). Видим, что атрибут *Авторы* в сущности *Методические указания* не удовлетворяет требованиям 1 НФ (у методических указаний может быть несколько авторов). Необходимо выделить сущность, которая будет содержать сведения об авторах методических указаний. Поскольку авторами всегда являются преподаватели вузов, новую сущность выделять не имеет смысла, свяжем сущности *Методические указания* и *Преподаватель*, предварительно удалив атрибут *Авторы*. Остальные атрибуты соответствуют 1 НФ. Атрибутивная модель, приведенная к 1 НФ, представлена на рис. 12.

Студент

P

проверяет Курсовая работа



Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

оформляется

выполняет

определяет

P

P

табельный номер (FK)

Вуз (FK)

Год издания (FK)

Группа (FK) Специальность (FK)

Курс (FK) Оценка Литература

P

1

Задание

Код дисциплины (FK) Вариант задания

Пояснительная записка

Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

Теория (O) Расчеты (O) Графическая часть (O)

P

определяет дату

теор\_часть (O) расчеты (O) Графич\_часть (O) кол-во страниц

имеет организует

Положение о курсовом проектировании

определяет

Дисциплина

Методические указания

Специальность

Курс Код дисциплины (FK)

имеет

Авторы

График

Код дисциплины (FK)

Группа

определяет

Дата Аудитория

Название\_дисц

Код дисциплины

Кол-во страниц

Вуз Год издания

ФИО

Должность Кафедра

табельный номер



|  |  |  |
| --- | --- | --- |
| № зачетной книжки | |  |
| ФИО  Группа | |
|  |  | |

Преподаватель

Рисунок 12 – Полная атрибутивная модель

1. Приведем модель ко 2 НФ. Проверим, все ли атрибуты зависят от составного ключа, а не от его части. Проверка показала, что все не ключевые атрибуты сущностей полностью зависят от составного ключа. Значит, модель удовлетворяет требованиям 2 НФ.
2. Проверим, есть ли транзитивная зависимость между не ключевыми атрибутами. Проверка показала, что взаимозависимости между не ключевыми атрибутами нет. Таким образом, модель, представленная на рисунке 13, приведена к 3 НФ.

Примечание. К нормализации относились также действия, выполненные в п. 2 упражнения 2.

Студент

Курсовая работа

|  |  |
| --- | --- |
| № зачетной книжки | |
| ФИО  Группа | |
| выпо | лняет |

Преподаватель табельный номер

ФИО



Должность Кафедра

Задание

P

~~определяет~~



Код дисциплины (FK)

Вариант задания

Теория (O) Расчеты (O) Графическая часть (O)

определяет

Дисциплина

P

Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

табельный номер (FK)

Вуз (FK)

Год издания (FK)

Группа (FK) Специальность (FK)

Курс (FK) Оценка

Литература

P

проверяет

оформляется

1

P

организует составляет

Пояснительная записка

Код дисциплины (FK) Вариант задания (FK)

№ зачетной книжки (FK)

теор\_часть (O) расчеты (O) Графич\_часть (O) кол-во страниц

P

имеет

определяет дату сдачи

Положение о курсовом проектировании

Вуз Год издания

Кол-во страниц

имеет

Название\_дисц

Код дисциплины

Методические указания

Специальность

Курс Код дисциплины (FK)

табельный номер (FK) определяет

График Код дисциплины (FK)

Группа

Дата Аудитория

Рисунок 13 – Информационная модель, приведенная к 1 НФ

# Упражнение 5. Создание физической модели

1. Необходимо переключиться на физический уровень представления информационной модели. Для этого нужно выбрать пункты меню *База данных* → *Параметры* → *Документ*. В открывшемся окне на вкладке *Общие* установить переключатель в меню *Имена, видимые на схеме*. В данном случае для физического представления информационной модели необходимо выбрать пункт *Физические имена* (рис. 14).
2. В закладке *Таблица* окна *Параметры документа базы данных* в меню *Отображать* выбрать пункт *Вертикальные линии,* в меню *Типы данных* – *Показывать физические* и в меню *Порядок – Физический порядок* (рис. 15).

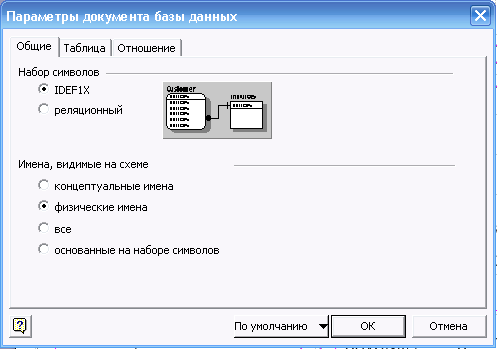


Рисунок 14 – Настройка параметров модели

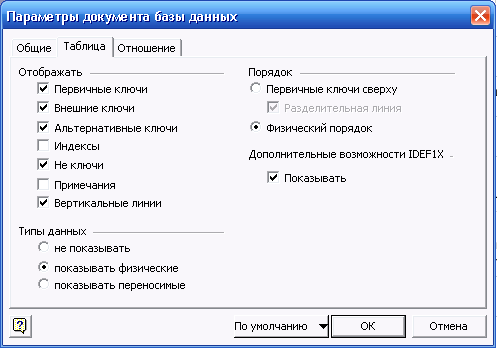


Рисунок 15 – Настройка параметров отображения сущности

1. В закладке Отношение окна Параметры документа базы данных в меню Отображение вида выбрать пункт Показывать физическое имя (рис. 16).

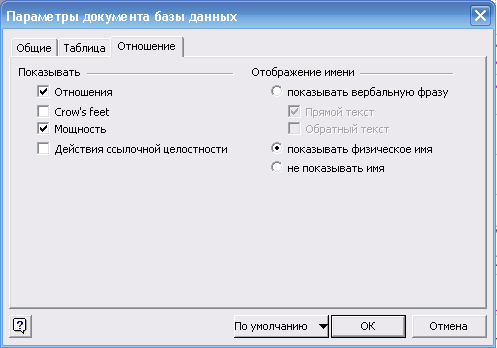


Рисунок 16 – Настройка вида отношений информационной модели

По окончании настройки документа информационная модель будет выглядеть, как представленная на рис. 17.

1. Для каждого атрибута (поля) необходимо определить тип данных.

Таблица2

Таблица1

Таблица5

CHAR(10) табельный номер

CHAR(10) Вариант задания (FK) CHAR(10) № зачетной книжки (FK)

CHAR(10) ФИО

CHAR(10) Уч\_степень (O) CHAR(10) теор\_часть (O)

CHAR(10) Должность CHAR(10) расчеты (O)

CHAR(10) Кафедра CHAR(10) Графич\_часть (O) CHAR(10) кол-во страниц

1

Таблица4\_Таблица5\_FK1

Таблица1\_Таблица4\_FK1 Таблица2\_Таблица4\_FK1

Таблица3\_Таблица4\_FK1

Таблица9

CHAR(10) Код дисциплины

Таблица4

INTEGER Вариант задания (FK)

P

CHAR(10) № зачетной книжки (FK)

Таблица3\_FK1

P

P

CHAR(10) Название\_дисц

Таблица9\_FK1

CHAR(10) Вуз (FK)

CHAR(10) Год издания (FK)

CHAR(10) Группа (FK)

CHAR(10) Специальность (FK) CHAR(10) Курс (FK)

CHAR(10) Оценка

CHAR(10) Литература

Таблица6\_Таблица4\_FK1

Таблица9\_Таблица6\_FK1

P

P

CHAR(10)

Таблица3

Вариант задания

CHAR(10) CHAR(10)

Теория (O) Расчеты (O)

CHAR(10) CHAR(10)

Таблица6

Специальность

Курс

Таблица8\_Таблица4\_FK1

CHAR(10) Графическая часть (O)

CHAR(10) Код дисциплины (FK)

Таблица7\_Таблица4\_FK

Таблица9\_Таблица8\_FK1

Таблица8

CHAR(10) Код дисциплины (FK) CHAR(10) Группа

CHAR(10) CHAR(10)

Дата Аудитория

Таблица7

CHAR(10) Вуз

CHAR(10) Год издания

CHAR(10) Кол-во страниц

ФИО

Группа

CHAR(10) CHAR(10)

№ зачетной книжки

CHAR(10)

Рисунок 17 – Вид физической модели

Типы данных можно представить в виде правил, ограничивающих вид сведений, которые могут быть введены в каждый столбец таблицы базы данных. Например, чтобы в поле, которое предназначено только для дат, нельзя было ввести имя, этому полю назначается тип данных «Дата».

**Примечание** *(Выбор между переносимыми и физическими типами данных).*

*Переносимые типы данных* — это обобщенные типы данных, соответствующие в разных системах баз данных простым, совместимым между собой физическим типам.

*Физические типы данных* — это типы данных, поддерживаемые целевой базой данных. Щелкните сущность, содержащую атрибуты, для которых требуется установить типы данных. В окне *Свойства базы данных* в списке *Категории* выберите вариант *Столбцы*.

Под списком столбцов установите переключатель в положение *Физический тип данных*.

В группе *Тип данных* для каждого атрибута выберите необходимый вариант из множества альтернатив (рис.

1. Описание типов данных приведено в *Приложении Б*.

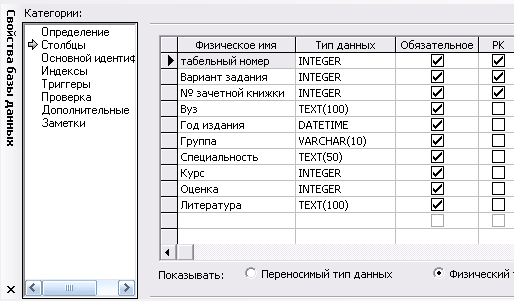


Рисунок 18– Определение типа данных атрибутов сущности

После того, как будут выполнены все действия, физическая модель будет выглядеть, как показано на рис. 19.

Таким образом, проделав все вышеперечисленные действия, получим информационную модель физического уровня, на основе которой может быть сгенерирована схема БД (в нашем случае в MS Office Access).



Таблица2

Таблица1 Таблица5

INTEGER табельный номер (FK) INTEGER Вариант задания (FK) INTEGER № зачетной книжки (FK) TEXT(100) теор\_часть (O)

TEXT(100) расчеты (O)

TEXT(100) Графич\_часть (O)

INTEGER кол-во страниц

1

Таблица4\_Таблица5\_FK1

Таблица1\_Таблица4\_FK1

Таблица2\_Таблица4\_FK1 Таблица3\_Таблица4\_FK1

Таблица2\_Таблица3\_FK1

P

P

Таблица4

INTEGER табельный номер (FK) INTEGER Вариант задания (FK) INTEGER № зачетной книжки (FK)

Таблица9

P

\_Таблица3\_FK1

TEXT(100) Вуз (FK)

DATETIME Год издания (FK)

VARCHAR(10) Группа (FK)

TEXT(50) Специальность (FK) INTEGER Курс (FK)

INTEGER Оценка

Таблица6\_Таблица4\_FK1 Таблица9\_Таблица6\_FK1

P

P

INTEGER INTEGER TEXT(100) TEXT(100)

Таблица3

табельный номер (FK) Вариант задания

Теория (O) Расчеты (O)

Таблица6

TEXT(100) Графическая часть (O)

TEXT(50) INTEGER

Специальность Таблица8\_Таблица4\_FK1

Курс

VARCHAR(10) Код дисциплины (FK)

Таблица7\_Таблица4\_FK

Таблица9\_Таблица8\_FK1

Таблица8

VARCHAR(10) Код дисциплины (FK) VARCHAR(10) Группа

DATETIME Дата

TEXT(5) Аудитория

Таблица7

Вуз Год издания Кол-во страниц

TEXT(100) DATETIME INTEGER

№ зачетной книжки

ФИО

Группа

INTEGER TEXT(50) VARCHAR(10)

Код дисциплины Название\_дисц

VARCHAR(10)

TEXT(50)

табельный номер

ФИО

Уч\_степень (O) Должность Кафедра

INTEGER TEXT(50) TEXT(50) TEXT(50) TEXT(50)

Рисунок 19 – Физическая модель базы данных

# Задание

В соответствии с вариантом задания, определенным преподавателем, последовательно выполнить следующие действия:

1. создать информационную модель логического уровня (выполнить упражнения 1-3). Минимальное количество сущностей – 4;
2. провести нормализацию полученной модели (упражнение 4);
3. на основе нормализованной логической модели построить информационную модель физического уровня (упражнение 5).

# Варианты

1. Проектирование ИС «Отдел кадров»;
2. Проектирование ИС «Агентство аренды»;
3. Проектирование ИС «Аптека»;
4. Проектирование ИС «Ателье»;
5. Проектирование ИС «Аэропорт»;
6. Проектирование ИС «Библиотека»;
7. Проектирование ИС «Кинотеатр»;
8. Проектирование ИС «Поликлиника»;
9. Проектирование ИС «Автосалон»; 10.Проектирование ИС «Таксопарк».

# Порядок выполнения работы

Для выполнения работы необходимо:

а) повторить правила техники безопасности при работе с вычислительной техникой;

б) изучить соответствующий раздел лекционного курса, а также теоретическую часть настоящего методического указания;

в) выполнить лабораторную работу согласно описанной в пункте 5 методике в соответствии с вариантом задания;

г) в соответствии с требованиями, приведенными в разделе 8 практикума, оформить отчет по лабораторной работе;

е) защитить лабораторную работу в соответствии с требованиями преподавателя.

# Требования к содержанию и оформлению отчета

Отчет должен содержать:

1. титульный лист;
2. название практического занятия, цель;
3. пул – список потенциальных сущностей;
4. нормализованную информационную модель логического уровня;
5. информационную модель физического уровня;
6. выводы по проделанной работе.

Отчет должен быть представлен в бумажном виде.

# Контрольные вопросы

1. Какие диаграммы позволяет строить нотация IDEF1X?
2. Для чего предназначена диаграмма «сущность-связь»?
3. Чем отличается полная атрибутивная модель от диаграммы «сущность-связь»?
4. Какие виды отношений существуют и чем они отличаются?
5. Что представляет собой нормализация?
6. В чем разница между логическим уровнем модели данных и физическим?

# 11.СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Вендров А. М. Практикум по проектированию программного обеспечения экономических информационных систем. : - М.: Финансы и статистика, 2004.-190 с.
2. Вендров А. М. Проектирование программного обеспечения экономических информационных систем -М.: Финансы и статистика, 2006.-543 с.
3. Маклаков С.В. Создание информационных систем с AllFusion Modeling Suite. – М.: ДИАЛОГ-МИФИ, 2005

– 432с.

# ПРИЛОЖЕНИЕ А

Функциональная модель процесса «Выполнить курсовую работу»

Методические указания

Положение о курсовом

проектировании

График

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  | Курсовая работа |
|  | Выполнить |  |  |
|  |
|  | курсовую |  |  |
|  | работу |  | Оценка за курсовую работу |
| 0р. |  | 0 |  |

Варианты заданий

Список

литературы

Purpose: Моделировать процесс выполнения курсовой работы View point: Студент

Студент

Рисунок А.1 – Контекстная диаграмма процесса выполнения курсовой работы

C1

Методические указания



Задание

C2

Положение о курсов ом проектиров ании

График 1

Полу чить задание

0р. 1

2

Варианты заданий

3

Список литерату ры

Подобрать литерату ру 0р. 2

Литерату ра

Сделать расчеты

Расчеты

Графическая часть

0р.

3

Сделать графическу ю

часть

0р. 4

Оформить

Пояснительная записка

пояснительну ю записку

0р. 5

Полу чить консу льтацию

0р.

Замечания, дополнения

6 Защитить

Ку рсов ая работа

ку рсов ую O1

O2

Преподав атель

M1 Сту дент

0р.

работу

7 Оценка за

ку рсов ую работу

Рисунок А.2 – Диаграмма декомпозиция блока А0

# ПРИЛОЖЕНИЕ Б

Типы данных Microsoft Office Access

## Символьные типы

Символьные типы используются для представления как строк символов, так и отдельных символов. Таблица Б.1 – Перечень символьных типов

|  |  |  |
| --- | --- | --- |
| **Тип данных** | **Назначение** | **Размер** |
| CHAR | Строковый тип | до 32767 байт. по умолчанию 1 байт |
| VARCHAR | Тоже, что и CHAR |  |
| LONG VARCHAR | Символьный тип произвольной длины.  Аналог МЕМО-полям в dBase, FoxPro, Access | Длина произвольная. Ограничена максимальным размером файлов базы данных (2 гигобайта) |
| TEXT | Тоже, что и LONG VARCHAR |  |

## Числовые типы

Числовые типы предназначены для обозначения целых, вещественных и денежных типов. Таблица Б.2 – Перечень числовых типов

|  |  |  |  |
| --- | --- | --- | --- |
| **Тип данных** | **Диапазон значений** | **Точность - число знаков после запятой** | **Размер** |
| INTEGER | от -2 147 483 648 до +2 147 483 647 | 0 | 4 байта |
| SMALLINT | от -32 768 до +32 767 | 0 | 2 байта |
| REAL | от -3.4 e-38 до 3.4 e+38 | до 6 | 4 байта |
| DOUBLE | от -1.797 e-308 до +1.797 e+308 | до 15 | 8 байт |
| DECIMAL | числа, состоящие из N цифр c M цифрами в дробной части. По умолчанию  N=30, M=6 | M | сколько требуется |
| NUMERIC | Тоже, что и DECIMAL |  |  |

## Типы дата/время

Типы дата/время предназначены для хранения времени, дат и дат совместно с временем. Таблица Б.3 – Форматы представления данных типа дата/время, определяемые по умолчанию

|  |  |
| --- | --- |
| **Тип данных** | **Формат, используемый по умолчанию** |
| DATETIME | 'YYYY-MM-DD HH:NN:ss.SSS' |

* **YYYY** – четыре цифры, обозначающие год:
* **MM** – две цифры, обозначающие месяц:
* **DD** – две цифры, обозначающие день:
* **HH** – две цифры, обозначающие часы:
* **NN** – две цифры, обозначающие минуты:
* **ss** – две цифры, обозначающие секунды:
* **SSS** – три цифры, обозначающие доли секунд.

По умолчанию составляющие времени HH, NN, ss, SSS принимаются равными нулю, а DD - единице.

## Двоичные типы

Двоичные типы предназначены для представления двоичных данных, включая изображения и другую информацию, не обрабатываемую собственными средствами СУБД.

Таблица Б4 -

Таблица Б.4 – Двоичные типы

|  |  |  |
| --- | --- | --- |
| **Тип данных** | **Назначение** | **Размер** |
| BIT | Тип для представления значений 0 и 1. Аналог полей типа Logical в dBase, FoxPro | 1 байт |
| BINARY | Тоже, что и CHAR, за исключением операций сравнения. В отличии от CHAR, данные этогопо умолчанию 1 байт типа сравниваются на полное  совпадение двоичных кодов байтов | до 32767 байт |
| LONG | Тип для представления двоичных данных произвольной длины | Длина произвольная.  Ограничена максимальным размером файлов базы данных (2 гиго- байта) |