

Практическая работа 40

Создание ER-диаграмм

Цель работы: Получить практический опыт создания концептуальной модели

Перечень оборудования и программного обеспечения

Персональный компьютер
Microsoft Office (Word, Visio, Access)

Краткие теоретические сведения

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач. В результате её решения должны быть определены содержание БД, эффективный для всех её будущих пользователей способ организации данных и инструментальные средства управления данными.

Основная цель процесса проектирования БД состоит в получении такого проекта, который удовлетворяет следующим требованиям:

1. Корректность схемы БД, т.е. база отображать моделируемую предметную область (ПО), где каждому объекту предметной области соответствуют данные в памяти ЭВМ, а каждому процессу – адекватные процедуры обработки данных.
2. Обеспечение ограничений (на объёмы внешней и оперативной памяти и другие ресурсы вычислительной системы).
3. Эффективность функционирования (соблюдение ограничений на время реакции системы на запрос и обновление данных).
4. Защита данных (от аппаратных и программных сбоев и несанкционированного доступа).
5. Простота и удобство эксплуатации.
6. Гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей.

3.1 Этапы проектирования базы данных

Процесс проектирования включает в себя следующие этапы:

1. Концептуальное проектирование.
2. Определение требований к операционной обстановке, в которой будет функционировать информационная система.
3. Выбор системы управления базой данных (СУБД) и других инструментальных программных средств.
4. Логическое проектирование БД.
5. Физическое проектирование БД.

Концептуальный подход не предоставляет формальных способов моделирования реальности, но он закладывает основы методологии проектирования баз данных.

Концептуальное проектирование

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций а будущих пользователей БД. Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависящих от реализации БД. Это описание выражается в терминах не отдельных объектов ПО и связей между ними, а их типов, связанных с ними ограничений целостности и тех процессов, которые приводят к переходу предметной области из одного состояния в другое.

Рассмотрим основные подходы к созданию концептуальной модели предметной области.

Функциональный подход к проектированию БД

Этот метод реализует принцип "от задач" и применяется тогда, когда известны функции некоторой группы лиц и/или комплекса задач, для обслуживания информационных потребностей которых создаётся рассматриваемая БД.

Предметный подход к проектированию БД

Предметный подход к проектированию БД применяется в тех случаях, когда у разработчиков есть чёткое представление о самой ПО и о том, какую именно информацию они хотели бы хранить в БД, а структура запросов не определена или определена не полностью. Тогда основное внимание уделяется исследованию ПО и наиболее адекватному её отображению в БД с учётом самого широкого спектра информационных запросов к ней.

Проектирование с использованием метода "сущность-связь"

Метод "сущность-связь" (entity-relation, ER-method) является комбинацией двух предыдущих и обладает достоинствами обоих. Этап инфологического проектирования начинается с моделирования ПО. Проектировщик разбивает её на ряд локальных областей, каждая из которых (в идеале) включает в себя информацию, достаточную для обеспечения запросов отдельной группы будущих пользователей или решения отдельной задачи (подзадачи). Каждое локальное представление моделируется отдельно, затем они объединяются.

Выбор локального представления зависит от масштабов ПО. Обычно она разбивается на локальные области таким образом, чтобы каждая из них соответствовала отдельному внешнему приложению и содержала 6-7 сущностей.

Сущность – это объект, о котором в системе будет накапливаться информация. Сущности бывают как физически существующие (например, СОТРУДНИК или АВТОМОБИЛЬ), так и абстрактные (например, ЭКЗАМЕН или ДИАГНОЗ).

Для сущностей различают тип сущности и экземпляр. Тип характеризуется именем и списком свойств, а экземпляр – конкретными значениями свойств.

Типы сущностей можно классифицировать как сильные и слабые. Сильные сущности существуют сами по себе, а существование слабых сущностей зависит от существования сильных. Например, читатель библиотеки – сильная сущность, а абонемент этого читателя – слабая, которая зависит от наличия соответствующего читателя. Слабые сущности называют подчинёнными (дочерними), а сильные – базовыми (основными, родительскими).

Для каждой сущности выбираются свойства (атрибуты). Различают:

1. Идентифицирующие и описательные атрибуты. Идентифицирующие атрибуты имеют уникальное значение для сущностей данного типа и являются потенциальными ключами. Они позволяют однозначно распознавать экземпляры сущности. Из потенциальных ключей выбирается один первичный ключ (ПК). В качестве ПК обычно выбирается потенциальный ключ, по которому чаще происходит обращение к экземплярам записи. Кроме того, ПК должен включать в свой состав минимально необходимое для идентификации количество атрибутов.

2. Составные и простые атрибуты. Простой атрибут состоит из одного компонента, его значение неделимо. Составной атрибут является комбинацией нескольких компонентов, возможно, принадлежащих разным типам данных (например, ФИО или адрес). Решение о том, использовать составной атрибут или разбивать его на компоненты, зависит от характера его обработки и формата пользовательского представления этого атрибута.

3. Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).

4. Основные и производные атрибуты. Значение основного атрибута не зависит от других атрибутов. Значение производного атрибута вычисляется на основе значений других атрибутов (например, возраст студента вычисляется на основе даты его рождения и текущей даты).





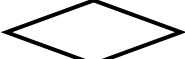
Спецификация атрибута состоит из его названия, указания типа данных и описания ограничений целостности – множества значений (или домена), которые может принимать данный атрибут.

Далее осуществляется спецификация связей внутри локального представления. Связи могут иметь различный содержательный смысл (семантику). Различают связи типа "сущность-сущность", "сущность-атрибут" и "атрибут-атрибут" для отношений между атрибутами, которые характеризуют одну и ту же сущность или одну и ту же связь типа "сущность-сущность".

Каждая связь характеризуется именем, обязательностью, типом и степенью. Различают факультативные и обязательные связи. Если вновь порождённый объект одного типа оказывается по необходимости связанным с объектом другого типа, то между этими типами объектов существует

обязательная связь (обозначается двойной линией). Иначе связь является факультативной.

Анализ предметной области выражается с помощью диаграммы "сущность-связь". Очень важным свойством модели "сущность-связь" является то, что она может быть представлена в виде графической схемы. Существует несколько вариантов обозначения элементов диаграммы "сущность-связь", каждый из которых имеет свои положительные черты. Воспользуемся неким гибридом нотаций Чена (обозначение сущностей, связей и атрибутов) и Мартина (обозначение степеней и кардинальностей связей). В таблице приводится список используемых здесь обозначений.

| Обозначение | Значение |
|------------------------------------------------------------------------------------|-----------------------------|
|  | Набор независимых сущностей |
|  | Набор зависимых сущностей |
|  | Атрибут |
|  | Ключевой атрибут |
|  | Набор связей |

Атрибуты с сущностями и сущности со связями соединяются прямыми линиями.

В процессе построения диаграммы можно выделить несколько очевидных этапов:

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:n) и "многие ко многим" (m:n). ER-диаграмма приведена на рис. 1.

1. Идентификация представляющих интерес сущностей и связей.
2. Идентификация семантической информации в наборах связей (например, является ли некоторый набор связей отображением $1:n$).
3. Определение атрибутов и наборов их значений (доменов).
4. Организация данных в виде отношений "сущность-связь".

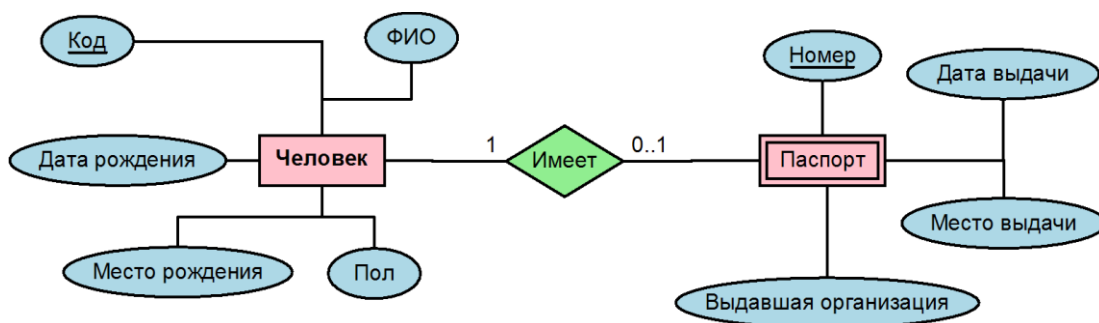


Рис.1. ER-диаграмма

Степень связи определяется количеством сущностей, которые охвачены данной связью.

Пример бинарной связи – связь между ОТДЕЛ и СОТРУДНИК, которые в нём работают.

Примером тернарной связи является связь между сущностями СОТРУДНИК и ПРОЕКТ. Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 2.

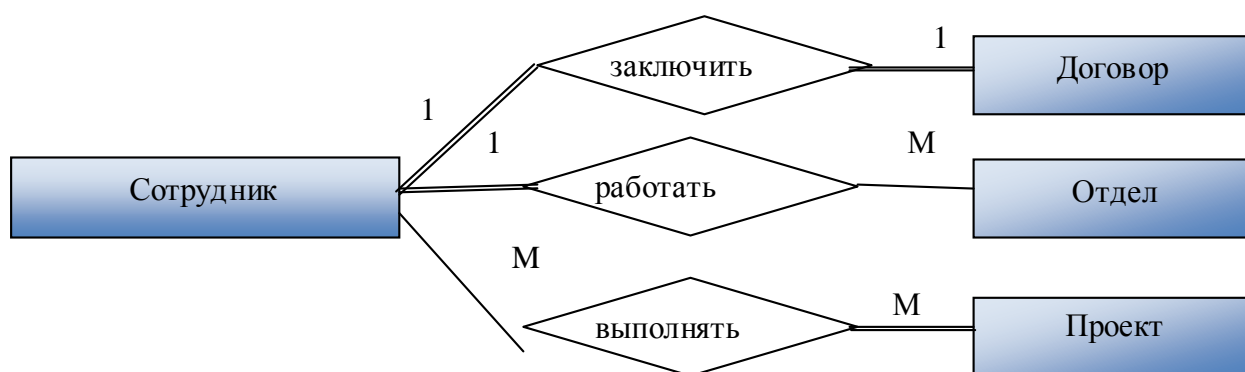


Рис.2. Пример ER–диаграммы с однозначными и многозначными атрибутами

По завершении объединения результаты проектирования являют собой концептуальную модель предметной области.

Типы взаимосвязей в модели. Обеспечение непротиворечивости и целостности данных в базе

На практике часто используются связи, устанавливающие различные виды соответствия между объектами «связанных» типов, — это один к одному (1:1), один ко многим (1:M), многие ко многим (M:M).

Связь один к одному означает, что каждому экземпляру первого объекта (A) соответствует только один экземпляр второго объекта (B) и, наоборот, каждому экземпляру второго объекта (B) соответствует только один экземпляр первого объекта (A).

Связь один ко многим означает, что каждому экземпляру одного объекта (A) может соответствовать несколько экземпляров другого объекта (B), а каждому экземпляру второго объекта (B) может соответствовать только один экземпляр первого объекта (A).

Связь многие ко многим означает, что каждому экземпляру одного объекта (A) могут соответствовать несколько экземпляров второго объекта (B) и, наоборот, каждому экземпляру второго объекта (B) могут соответствовать тоже несколько экземпляров первого объекта (A).

Пример 1.1. Рассмотрим совокупность следующих информационных объектов:

СТУДЕНТ (Номер студента, ФИО, Дата рождения, Номер группы);
СТИПЕНДИЯ (Номер студента, Размер стипендии);
ГРУППА (Номер группы, Специальность);
ПРЕПОДАВАТЕЛЬ (Код преподавателя, ФИО, Должность).

Здесь информационные объекты СТУДЕНТ и СТИПЕНДИЯ связаны отношением один к одному, так как каждый студент может иметь только одну стипендию и каждая стипендия может быть назначена только одному студенту.

Информационные объекты ГРУППА и СТУДЕНТ связаны отношением один ко многим, так как одна группа может включать в себя много студентов, в то время как каждый студент может обучаться только в одной группе.

Информационные объекты СТУДЕНТ и ПРЕПОДАВАТЕЛЬ связаны отношением многие ко многим, так как один студент может обучаться у многих преподавателей и один преподаватель может обучать многих студентов.

Для пользователей АИС важно, чтобы база данных отображала предметную область однозначно и непротиворечиво, т.е. чтобы она удовлетворяла условию целостности.

Выделяют два основных типа ограничений по условию целостности данных в базе.

1. Каждая строка таблицы должна отличаться от остальных ее строк значением хотя бы одного столбца.

Пример 1.2. Сотрудники одного отдела могут оказаться полными тезками, иметь одинаковые должность и телефон.

2. Внешний ключ не может быть указателем на несуществующую строку той таблицы, на которую он ссылается. Это ограничение называется ограничением целостности данных в базе по ссылкам.

Пример 1.3. В столбце Название отдела таблицы СОТРУДНИК хранятся сведения о принадлежности сотрудников к отделу, т. е. этот столбец является внешним ключом для ссылки на таблицу ОТДЕЛ. Для обеспечения ограничения целостности данных по ссылкам каждое название отдела из таблицы СОТРУДНИК должно принадлежать конкретному столбцу из таблицы ОТДЕЛ.

В реальных базах данных названия не делают ключевыми из-за их длины, замедляющей процесс поиска, и возможности изменения, создающей сложности с сопровождением системы.

Задания

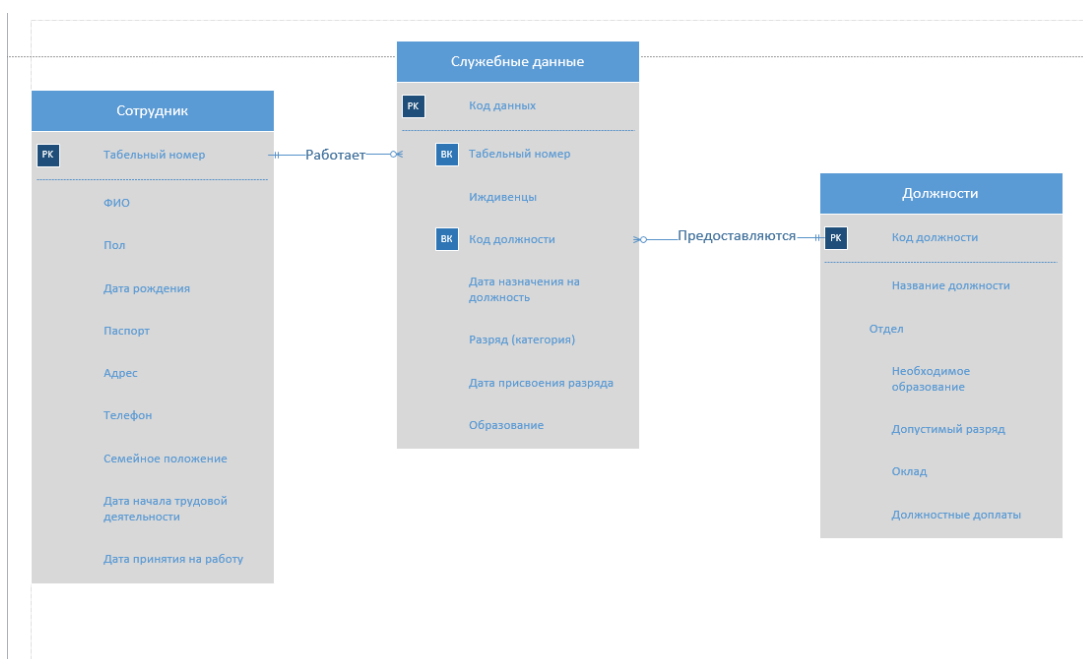
1. Изучить теоретические сведения.
2. В соответствии с вариантом задания создать ER-диаграмму. Оформить отчет по выполнению практической работы.

Порядок выполнения работы

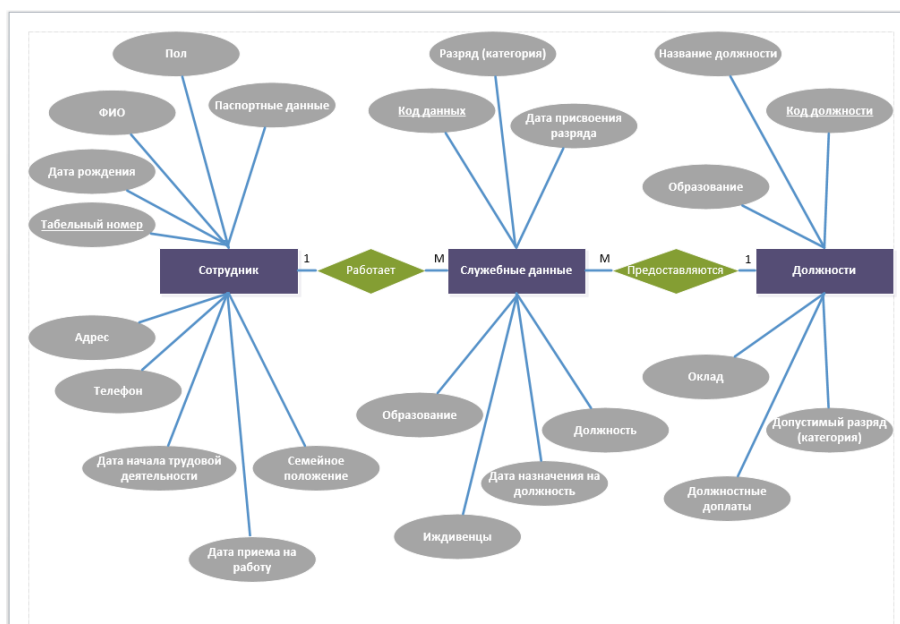
Воспользуемся результатом нормализации данных предыдущей практической (№39):

| Сотрудник | Служебные данные | Должности |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------|
| Табельный номер ФИО пол дата рождения паспортные данные адрес телефон семейное положение дата принятия на работу дата начала трудовой деятельности | Код данных Табельный номер Код должности дата назначения на должность должность разряд (категория) дата присвоения разряда образование иждивенцы | Код должности отдел должность (название) должностные доплаты образование допустимый разряд оклад |

Свяжем таблицы базы данных. Связи таблиц «один ко многим». Одной записи в таблице личных данных может соответствовать несколько записей таблицы служебных данных, также одной записи сущности Должности соответствует несколько служебных данных:



2 Создадим ER-диаграмму



Содержание отчета

1. Название работы.
2. Цель работы.
3. Перечень оборудования и программного обеспечения.
4. Порядок выполнения задания в соответствии с вариантом.
5. Вывод.

Варианты заданий

Варианты заданий представлены в практической работе 39

Контрольные вопросы:

- 1 Для чего нужны связи между таблицами БД?
- 2 Для какой цели предназначена схема данных?
- 3 Какие виды связей имеются в БД?
- 4 Когда используется отношение «один ко многим»?
- 5 В каких случаях устанавливается отношение «один к одному»?
- 6 Какие этапы проектирования БД принято выделять?
- 7 В чем назначение системного анализа?

Используемая литература

- Э. В. Фуфаев, Д. Э. Фуфаев Базы данных. М.: Академия, 2013
 Э. В. Фуфаев, Д. Э. Фуфаев Разработка и эксплуатация удаленных баз данных. М.: Академия, 2012