

*МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«САМАРСКИЙ ГОСУДАРСТВЕННЫЙ АЭРОКОСМИЧЕСКИЙ
УНИВЕРСИТЕТ имени академика С.П. КОРОЛЕВА»
(Национальный исследовательский университет
СГАУ)*

**СИСТЕМЫ УПРАВЛЕНИЯ
БАЗАМИ ДАННЫХ ПРИ
ТЕХНОЛОГИЧЕСКОЙ
ПОДГОТОВКЕ
ПРОИЗВОДСТВА**

Электронное методическое пособие

САМАРА
2010

Составители: МЕЩЕРЯКОВ Александр Викторович
ГАЛУЗИНА Татьяна Викторовна
РАМЗАЕВА Елена Анатольевна
СМЕЛОВ Виталий Геннадьевич
ЧЕМПИНСКИЙ Леонид Андреевич
ШУЛЕПОВ Александр Павлович

Методические указания предназначены для студентов обучающихся по специальности: 160301 Авиационные двигатели и энергетические установки, изучающих курсы «Технология производства АД и ЭУ», «Технология машиностроения», «Технологические методы обеспечения надежности деталей ГТД», «Информационные технологии» и магистерской программы «Энергетика, экология и двигательные установки ракетных и космических систем» по направлению 160700.68 «Двигатели летательных аппаратов»

Методические указания разработаны на кафедре производства двигателей летательных аппаратов.

© Самарский государственный
аэрокосмический университет, 2010

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. СТРУКТУРА ДАННЫХ В СУБД	5
2. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ	6
3. СЕМАНТИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РМД	8
4. МОДЕЛЬ СУЩНОСТЬ-СВЯЗЬ	11
5. ПРЕОБРАЗОВАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ В ЛОГИЧЕСКУЮ СХЕМУ	15
6. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СХЕМЫ БД	16
7. ДОПОЛНИТЕЛЬНЫЕ ЭЛЕМЕНТЫ ER-МОДЕЛЕЙ	17
8. ПРЕОБРАЗОВАНИЕ СОСТАВНЫХ ОБЪЕКТОВ	21
9. СОДЕРЖАНИЕ РАБОТЫ	22
10. ВАРИАНТЫ ЗАДАНИЙ ДЛЯ АНАЛИЗА	23
ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ	24
БИБЛИОГРАФИЧЕСКИЙ СПИСОК	24

ВВЕДЕНИЕ

Комплексы средств информационной и компьютерной поддержки, реализующие специальные задачи автоматизированного проектирования ТП (САПР ТП) (например, индивидуальное проектирование маршрутно-операционных ТП или проектирование на основе типового маршрута, решение задач выбора оборудования, оснастки, технологических переходов и других элементов ТП), используют значительный объем справочной информации.

Для сохранения и использования справочной информации о наименованиях технологических операций и переходов, видах оборудования, оснастки, приспособлений, сведений о режущем инструменте, нормативно-справочной информации (допуски, припуски и др.) могут применяться современные технологии проектирования и создания баз данных.

При реализации концепции единого информационного пространства предприятия (ЕИП) обеспечивается создание, хранение и обработка информации в единой базе данных с помощью системы управления базами данных (СУБД).

Справочные данные специальных приложений САПР ТП должны включаться в виде разделов единой базы данных ЕИП. Для интеграции баз данных необходимо понимание технологии структурирования и организации данных средствами современных СУБД.

Современные СУБД позволяют структурировать, систематизировать и организовывать данные для их компьютерного хранения и обработки.

СУБД выполняет функции посредника между системами автоматизированного проектирования (или другими информационными системами) и базами данными.

Основные функции СУБД:

- физическое размещение данных и их описаний в памяти;
- обеспечение механизмов поиска запрашиваемых данных;
- разрешение проблем, возникающих при одновременном запросе одних и тех же данных многими пользователями;

обеспечение защиты данных от некорректных обновлений или не-санкционированного доступа.

При выполнении основных этих функций СУБД использует описание структуры хранимых данных. Основная особенность сохранения данных с помощью СУБД, в отличие от файловых данных, – это наличие процедур для ввода и хранения не только самих данных, но и описания структуры этих данных.

Кроме того, СУБД имеет язык манипулирования данными, используемый для формулирования запросов в программных приложениях и непосредственно пользователями.

1. СТРУКТУРА ДАННЫХ В СУБД

Исторически технологии баз данных и СУБД разрабатывались и совершенствовались при создании информационных систем, использующих сохраняемые во внешней памяти массивы фактов. Такие базы данных имеют наименование фактографических.

Факт рассматривается как элементарная неделимая единица информации и представляет собой совокупность: реквизит (обозначение, наименование факта, признака) и соответствующее ему значение реквизита. Например, реквизиты: наименование, марка материала, тип заготовки – могут иметь значения – диск, Сталь45, штамповка 5 кл. точности.

Общая структура фактографической информации может включать различные комбинации реквизитов, обладающие определенным содержанием. Например, Изделие А, изготовлено 25.10.2005 бригадой № 1, обнаружен дефект - неполное прилегание фланца, возвращено для доработки 27.10.2005, дефектные испытания провел контролер А.И. Сидоров.

Структура данных определяется особенностями предметной области, задачи которой автоматизируются. Структура данных это неизменяемая в течение длительного времени часть информации.

В БД должны сохраняться не только реквизиты и их значения, но и логически возможные в предметной области комбинации между ними. То есть в БД должна быть описана логическая структура данных, соответствующая логической структуре данных в предметной области.

В любой СУБД определен набор правил компоновки данных и описания возможных связей между ними – правила структурирования данных (логической модели данных). Эти правила определяют тип логической модели данных.

В истории развития СУБД применялись логические модели трех типов – иерархическая, сетевая, реляционная.

В настоящее время наиболее распространены БД реляционной модели данных (реализующие логическую модель реляционного типа). Практически все широко применяемые СУБД реализуют реляционную модель структурирования данных (относятся к реляционному типу).

Представление структуры данных технической и технологической информации в виде реляционной модели часто сложнее, чем построение модели такого типа в экономических задачах. Более гибким и соответствующим задачам САПР ТП является описание объектной модели данных. Но при физической реализации даже базы данных ЕИП используют СУБД реляционного типа. Например, Oracle, SQL Server.

Реляционные СУБД обладают рядом известных преимуществ, это: возможность поддержки уже реализованных систем, использующих традиционные решения;

простота использования технологии, основанной на понятной табличной модели и математически строгой теории реляционной алгебры;

широкое распространение и тщательная многолетняя апробация предлагаемых на рынке продуктов.

2. РЕЛЯЦИОННАЯ МОДЕЛЬ ДАННЫХ

Реляционная модель придумана в 1970 г. математиком Коддом и поэтому имеет второе название - модель Кодда.

В реляционных базах данных информация хранится в виде взаимосвязанных двумерных таблиц. В таблицу объединяются данные об информационных объектах, имеющих одинаковый набор реквизитов.

Каждая строка таблицы содержит реквизиты одного информационного объекта.

Значения одноименных реквизитов размещаются в столбцах таблицы. Конкретное значение хранится в поле строки

Например, таблица «Типы элементарных поверхностей» (ЭП):

Код ЭП	Наименование ЭП	Точность	Шероховатость
ТН	Торец наружный	12 квалитет	Rz 20
ЦВ	Цилиндр внутренний	9 квалитет	Ra 2.5
Ш	Шлицы	10 квалитет	Ra 2.5

Связывание данных различных таблиц друг с другом выполняется с помощью значений реквизитов при их совпадении.

Столбец «Код ЭП» таблицы «Методы обработки» содержит ключ элементарной поверхности. Если нужно узнать параметры поверхности, соответствующие этим строкам из таблицы «Методы обработки», то следует поискать этот ключ в таблице «Типы элементарных поверхностей».

Подобное взаимоотношение между таблицами называется связью (relationship).

Пример:

Т а б л и ц а 1

Методы обработки

Код ЭП	Порядок применения	Метод обработки	Достижимая точность	Достижимая шероховатость
ТН	1	Точение черновое	12	Rz 20
ТН	2	Термообработка	NULL	NULL
ЦВ	1	Сверление	12	Rz 40
ЦВ	2	Расверливание	12	Rz 20
ЦВ	3	Растачивание	9	Ra 2.50

Различают три типа связей между таблицами реляционной базы данных:

один к одному — каждой записи одной таблицы соответствует только одна запись другой таблицы;

один ко многим (многие к одному) — одной записи главной таблицы могут соответствовать несколько записей подчиненной таблицы.

Связывание таблиц реляционной БД производится по совпадению значений в связываемых столбцах. Для связи можно выделять любые столбцы, лишь бы хранимые в них значения могли быть сравнимы. Нельзя сопоставлять текстовую информацию и числовую, т.е. данные

в связываемых полях должны быть одного типа. Такая свобода связывания позволяет получать любые комбинации данных, которые возможны в области реального мира.

Таблицы в БД это удобное для человека представление математического понятия отношение (relation – отсюда и название логической модели). Именно простота представления данных в виде взаимосвязанных таблиц привела к столь широкому распространению реляционных СУБД.

Реляционные таблицы имеют ряд важных ограничений.

В каждой ячейке таблицы может быть записано только одно значение реквизита, и таблица не должна содержать повторяющихся столбцов. Это правило атомарности данных. Иначе говоря, внутри ячейки таблицы не должно содержаться подтаблицы.

Каждая таблица имеет первичный ключ – один столбец или комбинация столбцов, которая уникальна для данной таблицы. То есть каждая строка таблицы и соответствующий ей объект отличаются от других.

Ограничения реляционной модели позволяют реализовать на множестве таблиц операции математически строгой алгебры отношений. Алгебра отношений обеспечивает мощный аппарат манипулирования данными в таблицах.

3. СЕМАНТИЧЕСКОЕ ПРОЕКТИРОВАНИЕ РМД

При проектировании схемы данных простой информационной системы достаточно представления данных в виде системы таблиц.

Однако ограничение атомарности данных в ячейках таблицы часто усложняет задачу представления реальных данных в виде взаимосвязанных таблиц.

Например, документ «Операционная технология» уже имеет «табличный вид» и может быть представлен в виде таблицы переходов. Реквизитами информационного объекта ПЕРЕХОД могут быть: «номер перехода», «номер операции», «содержание перехода», «оснастка», «режущий инструмент», «вспомогательный инструмент» и т.д. Такие данные в РМД не могут быть представлены в виде одной табли-

цы, т.к. такая таблица может иметь неатомарные значения в столбцах «оснастка», «инструмент» и т.п.

Объект ПЕРЕХОД будет представлен в виде трех связанных таблиц.

В данном примере предполагаем, что оснастка имеет набор сохраняемых реквизитов, отличный от набора реквизитов, характеризующих инструменты. Инструменты же, вне зависимости от их типа, имеют одинаковый набор реквизитов. В реальной БД инструменты даже одного типа могут иметь разные наборы реквизитов-характеристик, т.е. должны размещаться в разных таблицах с разными наборами столбцов.

Итак, задача представления данных реальной предметной области в РМД не всегда тривиальна.

При создании логической схемы данных в более крупных системах необходима промежуточная стадия моделирования данных – концептуальное проектирование, при котором конкретная предметная область описывается гораздо ближе к интуитивному пониманию и в то же время достаточно формальным образом.

При концептуальном проектировании логическая модель базы данных представляется на более высоком уровне абстракции. Такая модель называется концептуальной моделью базы данных

ПЕРЕХОД

№опер	№ пер	Содержание
5	1	Установить и закрепить в приспособлении
5	2	Подрезать торец 1, выдерживая размер 50
5	3	Сверлить отверстие 2, диаметром 15

ОСНАСТКА

№огр	№пер	Наименование
5	1	Трех кулачковый патрон

ИНСТРУМЕНТ

№огр	№пер	Наименование	Тип
5	2	резец	режущий
5	2	штангенциркуль	измерит.
5	3	Сверло	режущий
5	3	Пробка	измерит
5	3	Глубиномер	измерит

(концепция – понятие). Она создается с помощью методов семантического (смыслового) моделирования.

Семантическое моделирование занимается созданием моделей, представляющих смысловое значение данных. Семантика баз данных занимается связями между множеством данных и элементами реального мира, которые оно представляет. Цель семантики базы данных – обеспечить абстрактную модель, которая позволяет зафиксировать все элементы реального мира, интересующие пользователя базы данных.

Разработка крупной базы данных, как правило, является плодом коллективных усилий и в идеале должна проводиться с привлечением конечных пользователей системы, а также специалистов, которые будут заниматься ее реализацией. Поэтому необходимо, чтобы проект БД был документирован в форме, понятной всем заинтересованным сторонам. Концептуальные модели передают гораздо больше смысловой информации. Кроме того, жизненный цикл крупной базы данных не заканчивается ее реализацией, поэтому так важно документировать

ее проект в понятной форме. Отсутствие такой документации значительно осложняет выполнение даже небольших изменений в схеме существующей реляционной БД.

Таким образом, процесс проектирования крупной базы данных можно разбить на две основные стадии.

1. Фиксация требований пользователей базы данных и их представление в виде концептуальной модели.

2. Преобразование концептуальной модели в логическую схему, которую можно реализовать в заданной СУБД.

Графическое представление взаимосвязи объектов предметной области для человека более информативно, чем текстовое описание, поэтому при создании концептуальной модели используются диаграммы, состоящие из графических примитивов (прямоугольников, овалов или других фигур). Используемый в диаграмме набор примитивов называется нотацией диаграммы.

4. МОДЕЛЬ СУЩНОСТЬ-СВЯЗЬ

Существует много разных подходов к семантическому моделированию баз данных. В последние 10 лет одним из наиболее популярных языков семантического моделирования является UML. Проектирование реляционных БД – только одна и не слишком большая область применения этого языка, его возможности гораздо шире.

Значительно раньше в практике проектирования БД появилась модель Сущность–Связь. Эта модель проще UML, в ней меньше понятий, терминов. Модель Сущность–Связь разрабатывались именно для поддержки проектирования реляционных БД.

Модель Сущность–Связь была впервые предложена Ченом (Chen, 1976) и с тех пор подверглась значительным уточнениям. Модель Сущность–Связь позволяет создавать концептуальные модели, описывая сущности и связи между ними и представляя их в виде диаграмм.

Именно различные вариации этой модели чаще всего используются при моделировании баз данных. Общеупотребительное сокращение модели Сущность–Связь - ER-моделирование (Entity – Relation).

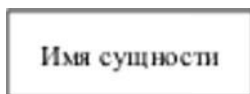
UML может использоваться для унифицированного объектно-ориентированного моделирования всего, чего угодно, в нем содержится масса различных понятий, терминов и вариантов использования,

совершенно избыточных с точки зрения проектирования реляционных БД. Если вычленишь из общего механизма UML то, что действительно требуется для проектирования реляционных БД, то мы получим в точности ER-диаграммы с другой нотацией и терминологией

ER-модель базы данных состоит из следующих трех основных примитивов.

1. Сущность. Это множество элементов предметной области, обладающих общими атрибутами или характеристиками. Это могут быть предметы, процессы, документы, понятия, используемые в моделируемой предметной области. Сущность должна представлять собой множество экземпляров сущности, отличимых между собой. Например сущность ДЕТАЛИ имеет экземпляры сущности – Деталь 1, Деталь 2, Деталь 3.

Сущность представляется в ER-диаграмме прямоугольником, в котором указано ее имя.



Имя сущности должно отражать тип предметов, моделируемых сущностью, а не конкретное название экземпляра сущности.

2. Атрибуты. Они описывают сущность. Атрибуты представляются овалами, которые прикреплены к сущности. Внутри овала указывается имя атрибута.

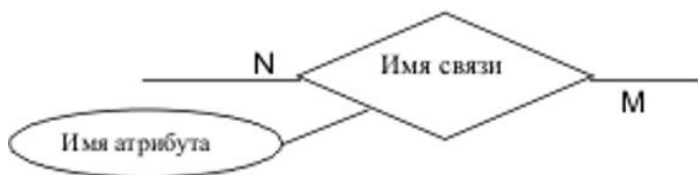


Очень важным является понятие ключевой атрибут — это та часть сущности, которая уникальным образом определяет каждый экземпляр сущности. Ключевые атрибуты выделяются подчеркиванием, они не обязательно должны быть простыми. Иногда приходится образовывать составной ключевой атрибут, т.к. только комбинация нескольких атрибутов отличает один экземпляр сущности от других.

Иногда некоторые реквизиты, характеризующие сущность, являются самостоятельными сущностями, связанными с данной сущностью, а не только ее атрибутами (сущность «замаскировавшаяся по

атрибут»). В первом примере связанных таблиц можно выделить атрибут «метод обработки» как самостоятельную сущность. Одинаковые методы обработки применимы к разным типам элементарных поверхностей. Список методов обработки может быть изменен вне зависимости от изменения списка элементарных поверхностей (появился новый метод). Каждый из методов обработки может характеризоваться предельными значениями точности и шероховатости, достижимыми при его применении (имеются собственные атрибуты).

3. Связи. Связь представляет взаимодействие между сущностями (смысловую ассоциацию двух или более сущностей). На диаграмме она изображается ромбом, который соединяет сущности, участвующие в связи.



Связь характеризуется степенью, которая показывает, сколько сущностей участвует в связи. Связь может иметь атрибуты, соответствующие ассоциации двух или нескольких сущностей.

Типы связей:

1 : 1 - один к одному (один экземпляр сущности связан с одним и не более экземпляром другой сущности). Например, Деталь и Чертеж или Комплект документации.

1 : M - или M : 1 - один ко многим (один экземпляр первой сущности соответствует нескольким экземплярам второй сущности, но каждому экземпляру второй сущности соответствует только один экземпляр первой сущности). Например, Изделие может иметь несколько Модификаций



Рис. 1. Пример ER-диаграммы

$M : N$ - многие ко многим (каждому экземпляру одной сущности соответствует несколько экземпляров другой сущности и наоборот). Например, для изготовления детали применяются инструменты разного типа, но инструмент одного типа может использоваться для обработки разных деталей

Кроме того, связь между сущностями может быть обязательной или необязательной. В первом случае говорят, что класс принадлежности сущности обязательный. Иначе, имеет место необязательный класс принадлежности сущности.

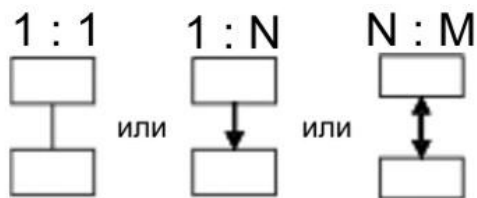
Если каждому экземпляру сущности А обязательно соответствуют один или более экземпляров сущности В, то класс принадлежности связи АВ обязательный. Например, связь сущностей Родители - Дети (БД детского сада).

Если же в сущности А имеются экземпляры, которым не требуется соответствие экземпляра сущности В, - имеет место необязательный класс принадлежности связи. Например, сущности Сотрудники и Дети сотрудников (В БД отдела кадров не у каждого сотрудника есть дети).

Пример. Концептуальная модель данных о механической обработке деталей, используемых при сборке изделий:

Возможны другие варианты нотаций ER-диаграмм.

Типы связей:



Нотация Баркера для связи один ко многим:



Ту или иную форму **графического** представления выбирает по своему усмотрению проектировщик базы данных, но обязательно в модели должны присутствовать сущности, атрибуты (ключевые элементы подчеркнуты), связи и их степени.

5. ПРЕОБРАЗОВАНИЕ КОНЦЕПТУАЛЬНОЙ МОДЕЛИ В ЛОГИЧЕСКУЮ СХЕМУ

Определить логическую схему реляционной БД – это значит определить названия всех входящих в нее таблиц.

Для каждой таблицы нужно определить список наименований полей (заголовков столбцов таблицы).

Для каждой таблицы должен быть определен ключевой столбец или комбинация ключевых столбцов - первичный ключ таблицы. В столбцах первичного ключа таблицы комбинация значений не может повторяться в пределах данной таблицы. Значения первичного ключа уникальны в пределах строк таблицы. Это свойство позволяет однозначно определить каждую строку таблицы.

Для таблиц должны быть определены связи, ассоциации двух или более таблиц между собой. С помощью связывания таблиц в БД будет возможен поиск по значениям данных одной таблицы, соответствующим

щих им строк (значений) в другой таблице. Для связывания таблиц должны быть предусмотрены дополнительные столбцы.

Реляционная схема данных допускает только два типа связей 1:N (N:1) или 1:1. Связь типа N:M, допускаемая в ER-диаграмме, в реляционной схеме она не возможна.

Перечисленные параметры логической схемы БД определяются в процессе анализа концептуальной модели, сформированной на этапе семантического проектирования.

Концептуальная модель преобразуется в логическую схему БД на основе четкого алгоритма, и это позволило разработать специальные программные приложения, реализующие такой алгоритм автоматически, так называемые CASE-системы (Erwin, S-Designor и т.д.).

При построении схемы данных на основе ER-диаграммы анализируются тип связи между сущностями (степень связи) и класс принадлежности связи.

6. ГРАФИЧЕСКОЕ ПРЕДСТАВЛЕНИЕ СХЕМЫ БД

Графическая диаграмма логической схемы БД состоит из прямоугольников, каждый из которых обозначает отдельную таблицу схемы. Название таблицы указывается над прямоугольником или в заголовке, отделяемом линией от остальной части прямоугольника.

Внутри прямоугольника перечисляются имена столбцов таблицы.

Ключевые столбцы подчеркиваются.

Линии связи проводятся строго между связываемыми именами столбцов. Пример графического изображения схемы реляционной БД приведен на рис. 2.

Реляционная схема может быть описана более компактно, с помощью скобочных выражений. Для данного примера это выглядит так:

ДЕТАЛЬ(КодДетали,Наименование)

ИЗДЕЛИЕ(КодИзделия,Наименование)

ВХОДИТ(КодИзделия,КодДетали,Количество)

ОБРАБОТКА(КодДетали,КодМетода,Точность,Время)

МЕТОДЫ(КодМетода,Наименование,Точность)

Графическая схема наглядней для определения маршрута поиска информации, если по данным одной таблицы требуется определить значения, хранящиеся в другой.

Для формирования этой логической схемы применялся алгоритм, правила которого представлены в табл. 2.

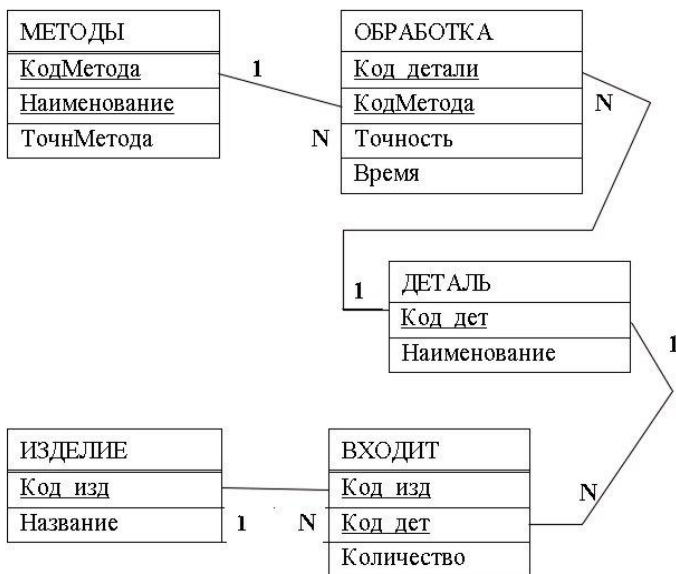


Рис. 2. Реляционная схема, соответствующая ER-диаграмме (рис.1)

7. ДОПОЛНИТЕЛЬНЫЕ ЭЛЕМЕНТЫ ER-МОДЕЛЕЙ

1. Возможно связывание экземпляров одной и той же сущности между собой. Например, сборочные изделия могут использоваться для сборки других изделий. В этом случае говорят о рекурсивной связи сущности Изделие.

2. Категоризация сущностей. Сущность может быть разделена и представлена в виде двух или более сущностей-категорий, каждая из которых имеет общие атрибуты и/или связи. Сущности-категории могут иметь и свои собственные атрибуты и/или связи, а также, в свою очередь, могут быть разделены своими сущностями-категориями на следующем уровне. Расщепляемая на категории сущность получила название «общей сущности».

Например, переходы операционной технологии в соответствии со своим предназначением можно разделить на группы: установочные, вспомогательные, формообразующие и т.п. Переходы всех групп имеют ряд одинаковых атрибутов, но каждая группа может иметь дополнительные атрибуты или иметь связи с сущностями, с которыми другие группы связи не имеют (Оснастка, Режущий Инструмент и т.п.).

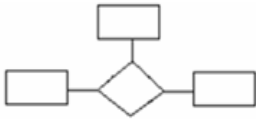
Для демонстрации декомпозиции сущности на категории используются диаграммы категоризации. Такая диаграмма содержит общую сущность, две и более сущности-категории и специальный узел-дискриминатор, который описывает способы декомпозиции сущностей (см. рис. 3).

При переходе к реляционной модели для отражения категоризации сущностей возможны два способа представления сущностей-категорий: все категории в одной таблице или для каждой категории – отдельная таблица.

При первом способе в таблицу включают все атрибуты всех под-типов. При этом для некоторых экземпляров данных ряд атрибутов не будет иметь смысла (поля будут содержать неопределенное значение Null). В эту таблицу добавляется по крайней мере один столбец, содержащий код ТИПА, он становится частью первичного ключа.

Алгоритм преобразования ER-модели в реляционную схему БД

Степень связи	Класс принадлежности	Решение
1 : 1	<i>Для обеих сущностей обязательный</i>	Требуется только одна таблица, содержащая атрибуты обеих сущностей. Первичным ключом этой таблицы может быть ключ любой из двух сущностей.
	<i>Для одной сущности обязательный, для другой – необязательный</i>	Необходимо построение двух таблиц, по одной на каждую сущность. Ключи сущностей будут служить первичными ключами для данных таблиц. Кроме того, ключ сущности, для которого класс принадлежности является необязательным, добавляется в качестве атрибута в таблицу, выделенную для сущности с обязательным классом принадлежности.
	<i>Для обеих сущностей необязательный</i>	Необходимо использовать три таблицы: по одной для каждой сущности и одна таблица для связи. Ключевые атрибуты сущностей служат в качестве первичных ключей в соответствующих таблицах, Таблица связи в качестве своих атрибутов будет иметь по одному ключу от каждой сущности. Эти атрибуты являются ключевыми и для таблицы связи (составной первичный ключ)
1 : N	<i>Для N-связной сущности обязательный</i>	Достаточно использовать две таблицы. Для каждой сущности организуется таблица.
		Ключ каждой сущности используется как первичный ключ соответствующей таблицы. Дополнительно, ключ 1-связной сущности должен быть добавлен как атрибут в таблицу, отводимую для N-связной сущности. Если связь двух сущностей имеет дополнительные атрибуты, то они тоже добавляются в таблицу N-связной сущности.

1 : N	<p><i>Для N-связной сущности необязательный</i></p>	<p>Необходимо формирование трех таблиц: по одной для каждой сущности и таблица для связи. Ключ каждой сущности служит первичным ключом соответствующей таблицы. -</p>
	<p>Таблица связи должна иметь среди своих атрибутов ключи от каждой сущности. Эти атрибуты составляют первичный ключ таблицы связи (составной первичный ключ). Если связь двух сущностей имеет дополнительные атрибуты, то они добавляются в таблицу для связи, как неключевые поля.</p>	
M : N		<p>Для хранения данных необходимо три таблицы: по одной для каждой сущности и одна таблица для связи. Ключ каждой сущности используется в качестве первичного ключа соответствующей таблицы. Таблица для связи среди своих атрибутов должна иметь ключ каждой сущности. Эти атрибуты составляют первичный ключ таблицы связи (составной первичный ключ). Если связь сущностей имеет дополнительные атрибуты, то они добавляются в таблицу для связи, как неключевые поля.</p>
	<p>В случае трехсторонней связи необходимо использовать четыре таблицы, по одной для каждой сущности, и еще одна для связи. Ключ каждой сущности должен служить в качестве первичного ключа для соответствующей таблицы.</p>	
	<p>Таблица для связи будет иметь среди своих атрибутов ключевые атрибуты от каждой сущности (составной первичный ключ). Если связь сущностей имеет дополнительные атрибуты, то они добавляются в таблицу для связи, как неключевые поля. Аналогично, когда связь n-сторонняя, требуется (n+1)-таблица.</p>	

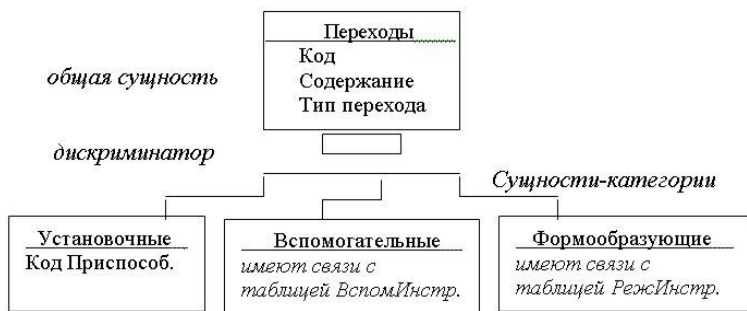


Рис. 3. Диаграмма категоризации

Если каждый экземпляр общей сущности обязательно должен быть членом одной и только одной категории, то целесообразным является представление каждой категории в виде отдельной таблицы, содержащей копию первичного ключа таблицы общей сущности. Недостатком такого решения является то, что создается много таблиц, но они содержат только значимые атрибуты подтипа.

Последняя ситуация очень характерна при организации сохранения данных для САПР: всевозможные типы и виды оборудования, инструмента, оснастки и т.д. приводят к созданию сотен таблиц (отдельная таблица для каждого подтипа/подвида).

8. ПРЕОБРАЗОВАНИЕ СОСТАВНЫХ ОБЪЕКТОВ

При построении концептуальной модели иногда удобно использовать составные объекты.

На рис. 4 приведена диаграмма, в которой используются составные объекты, иерархически вложенные друг в друга.

Внутренний составной объект ВЫПОЛНЯЕТ имеет тип связи "многие ко многим". В соответствии с изложенными выше правилами, преобразование этой модели в реляционную модель требует генерации трех таблиц.

Таблица 1 «Специалист»: ключевое поле – ТабНомер, неключевые поля – ФИО, Должность.

Таблица 2 «Вид работы»: ключевое поле – КодВида, неключевое поле – Характеристика.

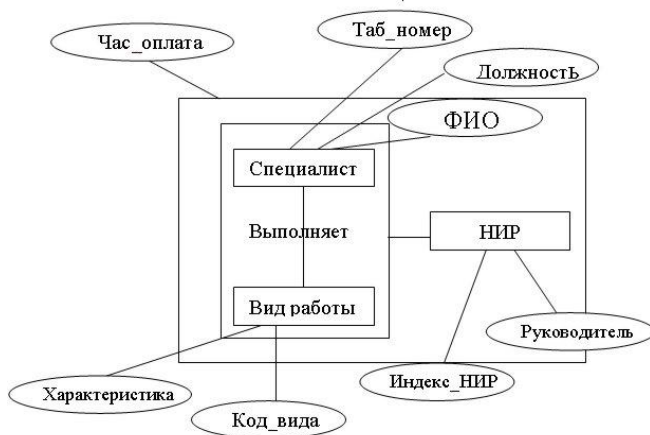


Рис. 4. Пример ER-диаграммы с составными элементами

Таблица 3 «Выполняет» имеет два ключевых поля: КодВида, Таб_Номер.

Таблица «Выполняет» имеет тип связи «один ко многим» и обязательный класс принадлежности с сущностью НИР. В соответствии с алгоритмом, в схему БД добавляется таблица 4 – «НИР», ключевое поле ИндексНир, неключевое поле – Руководитель. Кроме того, в таблицу «Выполняет» добавляется неключевое поле – ИндексНир. Связь таблицы «Выполняет» и сущности НИР имеет атрибут «Почасовая оплата», значит, в таблицу «Выполняет» нужно добавить еще одно неключевое поле – ОплатаЧас.

9. СОДЕРЖАНИЕ РАБОТЫ

1. Проанализировать вариант с описанием одной из задач, решаемых при автоматизированном проектировании ТП с целью определить набор реквизитов, используемых при решении этой задачи.

2. В процессе анализа задачи построить концептуальную модель данных в виде ER-диаграммы в любой нотации.

3. Построение ER-диаграммы начинается с выявления сущностей предметной области задачи. Затем в ER-диаграмме обозначаются связи, которые возможны между выделенными сущностями. Для связей важно указать их тип и обязательность. Далее ER-диаграмма дополня-

ются атрибутами сущностей. При этом некоторые атрибуты могут быть выделены в самостоятельные сущности (множественные атрибуты) и для них тоже должны быть указаны возможные связи. Некоторые сущности могут иметь соотношения типа категоризации

4. Используя правила алгоритма, приведенные в табл. 1, построить реляционную схему размещения данных, соответствующую сформированной ER-диаграмме.

10. ВАРИАНТЫ ЗАДАНИЙ ДЛЯ АНАЛИЗА

1. Реализация алгоритма автоматизированного проектирования маршрута обработки детали, с использованием справочных данных о возможных маршрутах обработки элементарных поверхностей. Необходимо составить реляционную схему данных, содержащую данные о деталях, состоящих из элементарных поверхностей со всеми необходимыми характеристиками. Кроме того, БД должна сохранять маршруты обработки каждой поверхности детали и маршрут обработки детали в целом. И кроме того, БД должна содержать справочную информацию обо всех типах элементарных поверхностей, обо всех возможных для их получения методах и маршрутах обработки. Задание может быть поделено между подгруппами студентов.

2. Автоматизированная система подготовки технологической документации.

3. Автоматизированная система расчета операционных размеров. Необходимо составить реляционную схему данных, содержащую данные о конструкторских и операционных размерах размерной схемы. Кроме того, БД должна сохранять последовательность операций и переходов и соответствие им операционных размеров. И кроме того, БД должна содержать справочную информацию о наименованиях операций и переходов. Операции характеризуются средней экономичной точностью и величиной припуска.

4. БД для сохранения нормативно справочной информации, используемой при расчете РТХ операционных размеров (таблицы допусков, припусков и т.п.).

5. Автоматизированная система проектирования ТП по методу типового проектирования.

ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

1. Для чего при создании информационных систем применяются СУБД? Какими возможностями обладает современная СУБД?
2. Что такое база данных и что такое СУБД. В чем различие этих понятий?
3. Что такое логическая модель данных БД?
4. Принципы организации данных при реляционной модели БД.
5. Порядок действий при семантическом методе проектирования БД.
6. Состав примитивов ER-диаграммы. Типы возможных связей между сущностями.
7. Определите понятия: сущность, атрибут, ключ сущности.
8. Что значит определить схему реляционной БД?
9. Какие типы связей между структурными элементами существуют в реляционной БД?
10. Каким образом можно преобразовать концептуальную модель данных в реляционные структуры?

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ролланд Ф.Л. Основные концепции баз данных. – М.: Вильямс, 2002. – 251 с.
2. Глушаков С.В., Ломотько Д.В. Базы данных. – Харьков: Фолио, 2002. – 504 с.