**Лабораторная работа №1. Проектирование базы данных.**

**Цель работы:**

– получить практические навыки создания концептуальной и логической моделей базы данных.

**Теоретические сведения**

## 

Проектирование базы данных (БД) – одна из наиболее сложных и ответственных задач, связанных с созданием информационной системы (ИС). 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:

* корректность схемы БД, т.е. база должна быть образом моделируемой предметной области (ПО);
* обеспечение ограничений ;
* эффективность функционирования ;
* защита данных (от аппаратных и программных сбоев и несанкционированного доступа);
* простота и удобство эксплуатации;
* гибкость, т.е. возможность развития и адаптации к изменениям предметной области и/или требований пользователей;

## *1. Этапы проектирования базы данных*

Процесс проектирования включает в себя следующие этапы:

Концептуальное проектирование – это процедура конструирования информационной модели, не зависящей от каких-либо физических условий реализации.

Логическое проектирование – это процесс конструирования информационной модели на основе существующих моделей данных, не зависимо от используемой СУБД и других условий физической реализации.

Физическое проектирование – это процедура создания описания конкретной реализации БД с описанием структуры хранения данных, методов доступа к данным.

### 1.1. Концептуальное проектирование

Основными задачами концептуального проектирования являются определение предметной области системы и формирование взгляда на ПО с позиций сообщества будущих пользователей БД, т.е. инфологической модели ПО.

Концептуальная модель ПО представляет собой описание структуры и динамики ПО, характера информационных потребностей пользователей в терминах, понятных пользователю и не зависимых 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.

1. 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.

1. 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.

1. 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:

* 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. Остальные атрибуты называются описательными и заключают в себе интересующие свойства сущности.
* 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.
* Однозначные и многозначные атрибуты (могут иметь соответственно одно или много значений для каждого экземпляра сущности).
* 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.

По типу различают множественные связи "один к одному" (1:1), "один ко многим" (1:N) и "многие ко многим" (M:N).

Степень связи определяется количеством сущностей, которые охвачены данной связью. Пример бинарной связи – связь между отделом и сотрудниками, которые в нём работают. Примером тернарной связи является связь типа экзамен между сущностями ДИСЦИПЛИНА, СТУДЕНТ, ПРЕПОДАВАТЕЛЬ. Из последнего примера видно, что связь также может иметь атрибуты (в данном случае это Дата проведения и Оценка). Пример ER–диаграммы с указанием сущностей, их атрибутов и связей приведен на рис. 1.

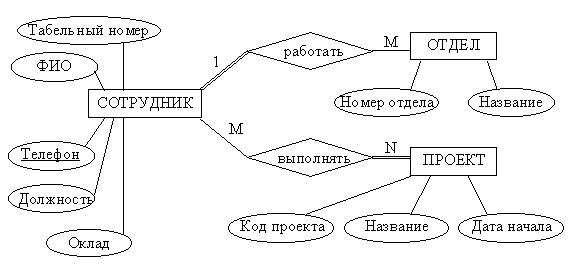


Рисунок 1 - Пример ER–диаграммы с однозначными и многозначными атрибутами

### 1.2. Логическое проектирование БД

На этапе логического проектирования разрабатывается логическая структура БД, соответствующая логической модели ПО. Решение этой задачи существенно зависит от модели данных, поддерживаемой выбранной СУБД. Будем рассматривать логическое проектирование БД для реляционной модели данных, так как современные СУБД – реляционные.

Проектирование реляционной базы данных проходит в том же порядке, что и проектирование БД других моделей данных, но имеет свои особенности.

Проектирование схемы БД должно решать задачи минимизации дублирования данных и упрощения процедур их обработки и обновления. При неправильно спроектированной схеме БД могут возникнуть аномалии модификации данных. Они обусловлены отсутствием средств явного представления типов множественных связей между объектами ПО и неразвитостью средств описания ограничений целостности на уровне модели данных.

На этом этапе выполняются следующие действия :

* Удаление связей M:N;
* Удаление рекурсивных связей;
* Удаление связей с атрибутами;
* Удаление множественных атрибутов;
* Перепроверка связей типа 1:1;
* Удаление избыточных связей.

Далее выполняется нормализация отношений.

Нормализация отношений выполняется на основе анализа первичных ключей и существования функциональных зависимостей между атрибутами. Как правило нормализация выполняется в несколько этапов. Каждый этап соответствует определенной нормальной форме (НФ). При проектировании реляционных баз данных требование первой нормальной формы (1НФ) должны выполняться всегда, остальные по желанию проектировщика. Однако, чтобы исключить аномалии обновления и избыточность данных рекомендуется приводить отношение к третьей нормальной форме 3НФ.

***Требование 1НФ*** : все атрибуты должны быть атомарными.

Ненормализованное отношение приводится к 1НФ следующими способами:

* Выравнивание таблиц или добавление строк;
* Один атрибут или группа атрибутов, которые назначены ключом отношения повторяющейся группы, помещается в отдельные отношения. Во вновь созданных отношениях устанавливаются свои первичные ключи.

***Требование 2НФ***: отношение удовлетворяет 1НФ и каждый атрибут , который не входит в состав первичного ключа, функционально полно зависит от первичного ключа.

Функциональная зависимость описывает связь между атрибутами отношения R(A,B) и обзначается. Атрибут (группа атрибутов) А называется ***детерминантом***.

Полная функциональная зависимость означает, что если атрибут В функционально зависит от первичного ключа, то зависит от полного его значения, а не какого-то подмножества. 2НФ применяется к отношениям с составными ключами.

Для того чтобы привести отношение ко 2НФ, нужно ислючить из отношения частичную зависимость и поместить ее в новое отношение вместе с копией их детерминанта.

***Требование 3НФ:*** Отношение находится находится во 2НФ и каждый неключевой атрибут нетранзитивно зависит от первичного ключа.

Если в отношении R(A,B,C) имеют место следующие функциональные зависимости:

А -> B и B -> C, то говорят, что атрибут С транзитивно зависит от атрибута А через атрибут В.

Для того чтобы привести отношение к 3НФ, нужно исключить из отношения транзитивную зависимость, поместив ее с новое отношение вместе с копией детерминанта.

Процесс нормализации заключается в декомпозиции отношения посредством выполнения последовательных операций проекции.

На этапе логического проектирования необходимо определить требования поддержки целостности данных. Ограничения целостности представляют собой ограничения, которые вводятся с целью предотвращения ввода в базу данных противоречивых данных. Различают следующие пять типов ограничений целостности :

* обязательные данные;
* ограничения для доменов атрибутов;
* целостность сущностей;
* ссылочная целостность;
* требования данного пользователя.

### 

### 1.3. Физическое проектирование БД

Этап физического проектирования заключается в увязке логической структуры БД и физической среды хранения с целью наиболее эффективного размещения данных, т.е. отображении логической структуры БД в структуру хранения. Решается вопрос размещения хранимых данных в пространстве памяти, выбора эффективных методов доступа к различным компонентам "физической" БД. Результаты этого этапа документируются в форме схемы хранения на языке определения данных (DDL). Принятые на этом этапе решения оказывают определяющее влияние на производительность системы. Между физическим и логическим проектированием существует обратная связь, так как иногда с целью повышения эффективности приходится менять структуру БД. Это возникает , если полгая нормализация данных не позволяет достичь требуемой эффективности обработки информации. Принимается решение о денормализации отношений. При этом следует учитывать следующее:

* денормализация усложняет реализацию БД,
* денормализация снижает гибкость системы,
* денормализация ускоряет выборку данных, но снижается скорость обновления данных.

Иногда денормализацию называют оптимизацией исполнения.

Одной из важнейших составляющих проекта базы данных является разработка средств защиты БД. Защита данных имеет два аспекта: защита от сбоев и защита от несанкционированного доступа. Для защиты от сбоев разрабатывается стратегия резервного копирования. Для защиты от несанкционированного доступа каждому пользователю доступ к данным предоставляется только в соответствии с его правами доступа.

**Задание на лабораторную работу**

Разработать концептуальную и логическую модели базы данных для формирования информационной системы в предметной области в соответствии с вариантом.

В базе данных предусмотреть:

1. Не менее 5 таблиц.
2. Представить таблицы в 3НФ.
3. Предусмотреть средства обеспечения целостности базы данных.
4. Не менее 3 хранимых процедур.
5. Не менее 3 представлений.
6. Не менее 3 функций пользователя.
7. Не менее 3 триггеров.

**Варианты предметных областей.**

1. Кафедра – Информационная система учета оценок студентов по дисциплинам.
2. Автобаза – Информационная система учета работы транспорта и водителей.
3. Касса театра – Информационная система бронирования и продажи билетов.
4. Турфирма – Информационная система продажи туров.
5. Гостиница –Информационная система учета заселения и освобождения номеров.
6. Поликлиника – Информационная система регистратуры
7. Больница – Информационная система приемного отделения.
8. Отдел кадров – Информационная система начальника отдела.
9. Аэропорт – Информационная система прибытия и убытия самолетов
10. Авиакомпания – Информационная система продажи билетов.
11. ГАИ – Информационная система регистрации автомобилей.
12. Интернет-магазин – Информационная система учета продаж.
13. Интернет-магазин – Информационная система учета закупок
14. Ресторан – Информационная система учета продаж.
15. Ресторан – Информационная система учета закупок продуктов.
16. Банк – Информационная система учета вкладов
17. Банк – Информационная система учета кредитов
18. Отделение милиции – Информационная система учета происшествий.
19. ЗАГС – Информационная система учета оказания услуг.
20. Библиотека – Информационная система учета книжного фонда.
21. Рекламное агентство – Информационное агентство учета заказов и услуг
22. Компьютерная фирма – Информационная система учета закупки комплектующих.
23. Компьютерная фирма – Информационная система учета продажи товаров и услуг.
24. Парикмахерская – Информационная система учета закупок товаров и продаж услуг
25. Кондитерская – Информационная система учета закупок и продаж.
26. СТО – Информационная система учета закупок и продаж.
27. Автосалон – Информационная система учета доходов и расходов.
28. Общежитие – Информационная система расселения студентов.
29. Автозаправка – Информационная система учета продаж и поступления топлива.
30. Музей– Информационная система учета поступлений и отгрузки экспонатов.

**Содержание отчета**

1. Тема и цель работы
2. Задание на лабораторную работу
3. Концептуальная модель базы данных.
4. Логическая модель базы данных.
5. Перечень средств обеспечения целостности базы данных.
6. Перечень представлений, хранимых процедур, функций пользователя и

триггеров с указанием их назначения.

1. Выводы

**Контрольные вопросы**

1. Что такое концептуальная модель базы данных?
2. Что такое логическая модель базы данных?
3. Какие ограничения используются в базах данных?
4. Что такое первичный ключ и для чего он используется?
5. Что такое внешний ключ и для чего он используется?
6. Какие требования предъявляются к таблице в 1НФ (2НФ, 3НФ)?
7. Что такое сущность в предметной области?
8. Какие возможны виды связей между сущностями?
9. Какие существуют ограничения целостности базы данных?