



**МЧС РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ УНИВЕРСИТЕТ
ГОСУДАРСТВЕННОЙ ПРОТИВОПОЖАРНОЙ СЛУЖБЫ**



Кафедра прикладной математики и информационных технологий

**Дисциплина
«БАЗЫ ДАННЫХ»
направление подготовки
231300.62 «Прикладная математика»**

Тема № 2 «Проектирование баз данных»

Занятие № 2.1 «Технология создания баз данных»

Учебные цели

Дидактические

- обобщить и систематизировать знания о порядке и содержании процесса проектирования баз данных;

Учебные

- изложить основные положения по нормализации отношений;
- раскрыть сущность методов нормализации.

Воспитательные

- воспитывать у обучающихся стремление к углубленному изучению учебного материала.

Содержание и порядок проведения занятия	Время, мин
ВСТУПИТЕЛЬНАЯ ЧАСТЬ	5
ОСНОВНАЯ ЧАСТЬ	170
Учебные вопросы:	
1.Общая схема проектирования баз данных	25
2.Цели и методы нормализации отношений	25
3.Обеспечение целостности данных	35
4.Характеристики CASE-технологии и ее применение для разработки логической структуры базы данных	35
5.Существенные свойства и показатели качества баз данных.	25
6.Методы расчета показателей качества .	25
ЗАКЛЮЧИТЕЛЬНАЯ ЧАСТЬ	5

Литература

Основная:

1. Иванов, Александр Юрьевич. Базы данных: учебное пособие: [гриф МЧС] / А.Ю. Иванов; МЧС России. – СПб.: СПбУ ГПС МЧС России, 2010. – 204 с.

Дополнительная:

1. Кузин, Александр Владимирович. Базы данных: учебное пособие: [гриф УМО] / А.В. Кузин, С.В. Левонисова. – 4-е изд., стер. – М. : Академия, 2010. – 320 с.

2. Мамаев Е.В. SQL Server 2000. – СПб.: БХВ-Петербург, 2004. – 1280 с.

3. Грэй П. Логика, алгебра и базы данных / Пер. с англ.- М.: Машиностроение, 1989. – 368 с.

Нормативно-правовая:

Доклад «О долгосрочных перспективах развития системы МЧС России (МЧС России - 2030) Доклад Министра РФ по делам гражданской обороны, чрезвычайным ситуациям и ликвидации последствий стихийных бедствий. М.: МЧС России, 2012».

Государственный доклад «О состоянии защиты населения и территорий Российской Федерации от чрезвычайных ситуаций природного и техногенного характера в 2012 году».

Основы единой государственной политики РФ в области ГО на период 2020 года (утверждена Президентом РФ от 03.09.2011, № ПР-2613).

Стратегия инновационного развития РФ на период до 2020 года (утверждена распоряжением Правительства РФ от 08.12.2011 года, №2227-р).

Федеральный закон от 22.07.2008 г. №123 – ФЗ (ред.от 10.07.2012) «Технический регламент о требованиях пожарной безопасности».

Закон РФ от 29 декабря 2012 года №273-ФЗ «Об образовании в Российской Федерации» с изменениями и дополнениями на 2013 год.

Организационно-методические указания по подготовке территориальных органов, спасательных воинских формирований, подразделений федеральной противопожарной службы, военизированных горноспасательных частей, образовательных учреждений и организаций МЧС России в области гражданской обороны, предупреждения и ликвидации чрезвычайных ситуаций, обеспечения пожарной безопасности и безопасности людей на водных объектах на 2014-2016 годы.

Федеральный закон № 149 «Об информации, информационных технологиях и о защите информации» от 27 июля 2006 г.

1. Общая схема проектирования базы данных

В проектировании баз данных выделяются следующие этапы:

1. анализ информационных потребностей;
2. инфологическое моделирование;
3. логическое проектирование;
4. физическая реализация.

Анализ информационных потребностей. На этом этапе разработчик базы данных анализирует информацию, циркулирующую в органе управления, для которого разрабатывается база данных, и определяет, какие задачи будут решаться основными пользователями — должностными лицами — с помощью базы данных. В обязательном порядке на этом этапе определяется, с какими запросами пользователи будут обращаться к базе и какие выходные документы (отчеты) они будут получать. При этом для запросов определяются перечень выдаваемой информации, условия поиска, частота выполнения запроса, кто из должностных лиц его будет выполнять. Для отчетов дополнительно определяется форма документа, выводимого на печать. Полученные на этом этапе решения оформляются в произвольном виде (обычно табличном).

Инфологическое моделирование. Основной задачей этого этапа является разработка инфологической модели (ИЛМ) предметной области. Исходными данными для этого являются результаты, полученные на предыдущем этапе, а также знания о специфике предметной области. ИЛМ должна отображать объекты (сущности) предметной области, их учитываемые характеристики – атрибуты, а также взаимные связи между объектами и атрибутами. ИЛМ представляется чаще всего в виде графической диаграммы.

Кроме формирования ИЛМ, на данном этапе дается характеристика всех атрибутов, учитываемых в базе данных. Она предполагает указание принадлежности атрибута (к объекту или к связи), типа данных, к которому относятся значения атрибутов, их длины, области допустимых значений, ограничений целостности, выводимости из значений других атрибутов и ряд других характеристик. Результаты оформляются в табличном виде.

Логическое проектирование. Исходными данными являются результаты, полученные на этапе инфологического моделирования. Основным результатом этого этапа является логическая структура базы, называемая *реляционной схемой* базы данных.

Проектирование реляционных схем является одним из самых сложных и ответственных этапов всего процесса проектирования. Одной из ключевых задач здесь является нормализация отношений, т.е. приведение схем отношений к нормальной форме, обеспечивающей целостность БД в ходе работы с ней на этапе эксплуатации. Для нормализации баз данных разработаны специальные методы.

Кроме проектирования реляционной схемы, на этапе логического проектирования осуществляется оценка качества будущей базы данных по таким показателям, как ее предполагаемый объем и оперативность выполнения запросов.

Физическая реализация. На этом этапе в среде выбранной СУБД формируются структуры таблиц, осуществляется их первоначальное заведение, разрабатываются запросы и отчеты в соответствии с решениями, полученными на первом этапе. Таким образом, промежуточным результатом, полученным на этом этапе, является демонстрационный прототип (макет) базы данных, показывающий возможность (или невозможность) реализации всех предъявляемых к ней требований.

В случае если макет удовлетворяет предъявляемым требованиям, база данных заполняется до полного объема и передается в опытную эксплуатацию. В противном случае осуществляется возврат на один из предыдущих этапов с целью уточнения полученных на нем результатов.

2. Цели проектирования и методы нормализации отношений

а) Цели проектирования отношений

Основными целями проектирования реляционных БД являются:

- 1 достижение полноты хранения необходимых данных;
- 2 исключение избыточности данных;
- 3 сведение числа хранимых в БД отношений к минимуму;
- 4 нормализация отношений для упрощения решения проблем, связанных с обновлением данных.

Достижение полноты хранения необходимых данных является вполне очевидной целью. Предполагается, что БД должна содержать все данные, представляющие интерес для пользователя. Таким образом, следует предусмотреть возможность размещения в БД всех необходимых данных. Поэтому одним из первых шагов проектирования является определение всех атрибутов, которые впоследствии будут помещены в БД (это выполняется на этапе инфологического проектирования БД). После определения состава атрибутов решается вопрос, сколько отношений необходимо и какие атрибуты включать в какие отношения.

Исключение избыточности данных:

П_Д

ПРЕП#	ДЕНЬ_НЕД
133	Пн.
133	Вт.
199	Пн.
199	Ср.
255	Вт.

а)

П_Д

ПРЕП#	ДЕНЬ_НЕД
133	Пн.
133	—
199	—
199	Ср.
255	Вт.

б)

Рис.1

А)

Ф_П_Д

ПРЕП#	ФАМ	ПАУД	ДЕНЬ_НЕД
133	Иванов	119	Пн.
133	Иванов	119	Вт.
199	Кузнецов	121	Пн.
199	Кузнецов	121	Ср.
255	Смирнов	119	Вт.

Б)

Ф_П_Д

ПРЕП#	ФАМ	ПАУД	ДЕНЬ_НЕД
133	Иванов	119	Пн.
133	—	—	Вт.
199	Кузнецов	121	Пн.
199	—	—	Ср.
255	Смирнов	119	Вт.

Рис.2

Лучшим решением данной проблемы является исключение избыточности путем декомпозиции исходного отношения, что лежит в основе рассматриваемого ниже метода проектирования.

Необходимость минимизации числа хранимых в БД отношений обуславливается тем, что разбиение одного отношения на два или более меньших желательно для исключения определенных проблем, но является неудобным для пользователя. Таким образом, нельзя допускать неограниченный рост числа отношений. Для некоторых отношений очень важна проблема обновления (например, для отношения **Ф_П_Д**). Необходимо уметь обнаруживать такие отношения и «нормализовать» их посредством разбиения предписанным образом. Такое разбиение одного отношения на два или более в соответствии со специальной процедурой разбиения называется **нормализацией**. На нормализации основывается большинство методов проектирования схем реляционных БД. Последние две цели противоречат друг другу, поэтому в реальности определяется взаимный компромисс между ними, что является частью завершающего этапа проектирования.

Проблемы использования единственного отношения. Самым простым вариантом организации схемы БД является составление единственного отношения, в которое вносятся все учитываемые в БД атрибуты. Такое отношение носит название *универсального*.

Примером универсального отношения является отношение **РАСПИСАНИЕ** (рис.3), в которое входят следующие атрибуты:

условный номер преподавателя – **ПРЕП#**;

фамилия преподавателя – **ФАМ**;

номер преподавательской – **ПАУД**;

номер телефона в преподавательской – **ТЕЛ**;

место проведения занятия – **МЕСТО**;

день недели - **ДЕНЬ_НЕД**;

время проведения занятия – **ЧАСЫ**;

учебная группа – **ГРУППА**;

шифр дисциплины – **ШИФР**.

РАСПИСАНИЕ

ПРЕП#	ФАМ	ПАУД	ТЕЛ	МЕСТ О	ДЕНЬ_НЕ Д	ЧАСЫ	ГРУПП А	ШИФР
133	Иванов	119	9597	541	Пн.	1-2	121	ОИ242
133	Иванов	119	9597	542	Вт.	3-4	122	ОИ242
199	Кузнецов	121	9742	731	Пн.	1-2	131	ОК142
199	Кузнецов	121	9742	731	Пн.	1-2	132	ОК142
199	Кузнецов	121	9742	731	Ср.	5-6	132	ОК142
255	Смирнов	119	9597	544	Вт.	1-2	122	ОИ241

Рис.3

Для малых БД, включающих небольшое число атрибутов, универсальное отношение используется в качестве отправной точки при проектировании БД. Однако существует ряд причин, почему не следует использовать данное отношение в качестве единственного в БД. Это обусловлено тем, как будет использоваться БД и какое воздействие на данные в отношении **РАСПИСАНИЕ** будут оказывать определенные операции. Различаются три специфичные проблемы: проблема, связанная с обновлением (модификацией) данных в БД; проблема, обусловленная необходимостью удаления кортежей; проблема, обусловленная необходимостью вставки новых кортежей. Эти проблемы обычно называют *аномалиями* вставки, удаления и обновления, понимая под аномалией отклонение от нормы.

Аномалия вставки проявляется следующим образом. Если появляется новый преподаватель, который еще не проводит занятия, то для него необходимо включить в БД кортеж с нулевыми (пустыми) значениями для атрибутов **МЕСТО**, **ДЕНЬ_НЕД**, **ЧАСЫ**, **ГРУППА** и **ШИФР**. Пример такого включения представлен на рис.4. Пустые значения в полях **МЕСТО** и **ГРУППА** интерпретируются СУБД как 0 (ноль). Если теперь требуется получить ответ на запрос: «Выдать список преподавателей, которые проводят занятия в группах, номера которых меньше 130», то в этот список попадет и преподаватель **Семенов**, хотя он вообще не проводит ни одного занятия.

РАСПИСАНИЕ

ПРЕП#	ФАМ	ПАУД	ТЕЛ	МЕСТО	ДЕНЬ_НЕД	ЧАСЫ	ГРУППА	ШИФР
133	Иванов	119	9597	541	Пн.	1-2	121	ОИ242
133	Иванов	119	9597	542	Вт.	3-4	122	ОИ242
199	Кузнецов	121	9742	731	Пн.	1-2	131	ОК142
199	Кузнецов	121	9742	731	Пн.	1-2	132	ОК142
199	Кузнецов	121	9742	731	Ср.	5-6	132	ОК142
255	Смирнов	119	9597	544	Вт.	1-2	122	ОИ241
341	Семенов	116	9642	0	—	—	0	—

Рис.4

Аномалия удаления для отношения **РАСПИСАНИЕ** проявляется в случае удаления из БД единственного кортежа, в котором **ПРЕП#=255**. Он соответствует преподавателю **Смирнову**. Предположим, что для **Смирнова** перенесены его занятия на следующую неделю. Поскольку это единственный кортеж с информацией о **Смирнове**, то его удаление приведет к потере информации о номере преподавательской аудитории и номере телефона. Если затем запросить список всех преподавателей на кафедре, то **Смирнов** в него не попадет.

Аномалия обновления для отношения **РАСПИСАНИЕ** проявляется в нескольких случаях. Во-первых, если какой-либо преподаватель (например, **Иванов**) изменил свою преподавательскую, то в БД нужно проследить это изменение во всех кортежах, описывающих **Иванова**. Вместе с тем за каждой преподавательской закреплен свой номер телефона. Поэтому следует изменить еще и номер телефона у **Иванова**.

Во-вторых, если вдруг изменился номер телефона в какой-то преподавательской (например, в ПАУД=119), то это изменение следует проследить не только для **Иванова**, находящегося в ней, но и для других преподавателей этой аудитории, например, **Смирнова**). Если не произвести эти обновления в полном объеме, то в результате получается противоречивая БД.

Понятие нормальной формы схемы БД. Выше было отмечено, что одной из задач проектирования реляционной БД является разбиение отношений с нежелательными аномалиями. Вопросы, которые необходимо решить при этом (т.е. в ходе нормализации), сводятся к следующим:

- 1) как получить исходные отношения (до начала разбиения)?
- 2) как распознать «плохие» отношения?
- 3) как выполнить разбиение?
- 4) что является признаком завершения разбиения?

На первый вопрос можно дать такой ответ. Для БД, общее число хранимых атрибутов в которой невелико, исходным отношением служит универсальное. При большем числе атрибутов для получения исходных отношений следует воспользоваться существующими методами (например, рассматриваемым ниже методом «сущность-связь»). Если определено исходное отношение, то оно должно иметь форму, в которой все поля заполнены атомарными значениями.

Определение. Отношение находится в **первой нормальной форме (1НФ)**, если каждый его элемент имеет, и всегда будет иметь атомарное значение.

Определение. Если даны два атрибута A и B и каждому значению A в отношении соответствует ровно одно значение B , то говорят, что **B функционально зависит от A (B F -зависит от A , или $A \rightarrow B$)**. В определении F -зависимости для атрибутов БД просматривается аналогия с определением понятия функция в математическом анализе (здесь A выступает в роли аргумента, а B – в роли функции). Если между тремя атрибутами A , B и C установлены F -зависимости $A \rightarrow B$ и $B \rightarrow C$, то говорят, что между A и C существует **транзитивная F -зависимость**.

Определение. Атрибут A является **ВОЗМОЖНЫМ КЛЮЧОМ** отношения, если A выступает в роли *первичного ключа*, т.е. по значению, которое принимает атрибут A , можно однозначным образом выбрать из отношения единичную запись.

Определение. Если в отношении существует $A \rightarrow B$, причем A является составным атрибутом, но ни для какого подмножества $A' = A$ не выполняется условие $A' \rightarrow B$, то говорят, что A является **детерминантом** (или **определителем**) B .

Рассмотрим введенные определения на примере отношения **РАСПИСАНИЕ** (см. рис.3). Условно обозначим имена атрибутов буквами A, B, C, D, E, F, G, H и I в таком порядке, как они представлены в отношении (т.е. **ПРЕП#** обозначим A , **ФАМ** – B и т.д.).

Тогда в отношении **РАСПИСАНИЕ** можно выделить F-зависимости:

$A \rightarrow B$

$A \rightarrow C$

$A \rightarrow D$

$C \rightarrow D$

$D \rightarrow C$

$F, G, H \rightarrow A$

$F, G, H \rightarrow E$

$F, G, H \rightarrow I$.

Отношение имеет единственный возможный ключ $\langle F, G, H \rangle$ и четыре детерминанта: $\langle A \rangle$, $\langle C \rangle$, $\langle D \rangle$ и $\langle F, G, H \rangle$.

Одним из самых важных результатов, полученных в теории реляционных БД, является утверждение о том, что большинство потенциальных аномалий в БД устраняется в случае преобразования каждого отношения в нормальную форму Бойса-Кодда.

Определение. Отношение находится в **нормальной форме Бойса-Кодда (НФБК)**, если и только если каждый детерминант отношения является его возможным ключом. В нашем примере имеются детерминанты, которые не являются возможным ключом. Следовательно, отношение **РАСПИСАНИЕ** не находится в НФБК и должно быть подвергнуто нормализации.

Кроме упомянутых 1НФ и НФБК в теории реляционных баз данных существуют и другие нормальные формы, наиболее известными из которых являются 2НФ, 3НФ и 4НФ. Между всеми этими формами существует определенное взаимное соответствие. Приведем его, опуская комментарии, чтобы показать место НФБК в ряду нормальных форм:

2НФ – это 1НФ, в которой каждый непервичный атрибут полностью зависит от каждого первичного ключа;

3НФ – это 2НФ, в которой ни один из непервичных атрибутов не является транзитивно зависимым от первичного ключа;

НФБК – это 3НФ, в которой никакой атрибут не зависит транзитивно ни от одного ключа (это другое определение НФБК);

4НФ – это НФБК, в которой устранены так называемые многозначные F-зависимости.

Ненормализованное отношение

Удаление повторяющихся групп

Первая нормальная форма (1НФ)

Удаление зависимости от части ключа

Вторая нормальная форма (2НФ)

Удаление транзитивных зависимостей

Третья нормальная форма (3НФ)

Удаление из функциональных зависимостей оставшихся аномалий

Нормальная форма Бойса-Кодда (НФБК)

Удаление многозначных функциональных зависимостей

Четвертая нормальная форма (4НФ)



Метод декомпозиции без потерь

Является одним из самых простых методов нормализации отношений. Сущность его заключается в следующем. Пусть имеется отношение $R(A,B,C,D,E,...)$, которое не находится в НФБК. Декомпозиция без потерь производится в два шага.

Шаг 1. Выделяется F-зависимость, которая является причиной отсутствия НФБК. Предположим, такой F-зависимостью является $C \rightarrow D$.

Шаг 2. Отношение $R(A,B,C,D,E,...)$ разделяется на следующие два: $R_1(A,B,C,E,...)$ и $R_2(C,D)$. Отношение R_2 иначе называется проекцией R по атрибутам F-зависимости $C \rightarrow D$. Декомпозиция называется «без потерь», т.к. исходное отношение R восстанавливается из отношений R_1 и R_2 путем применения к последним реляционной операции **соединения по общему полю**. Простым правилом выбора функциональной зависимости для проекции на первом шаге декомпозиции является поиск транзитивной зависимости вида $A \rightarrow C \rightarrow D$ с последующим использованием $C \rightarrow D$ для проекции.

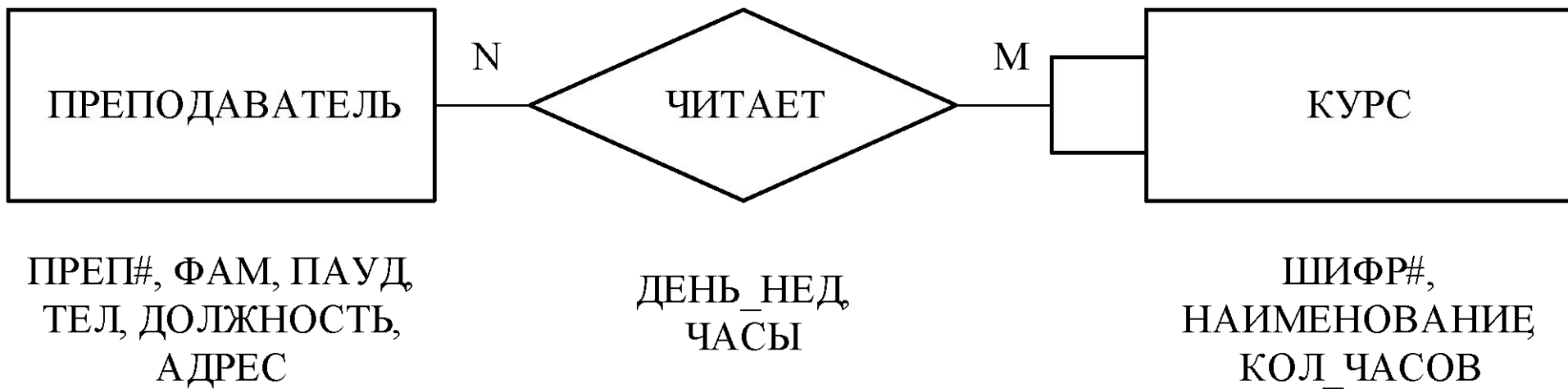
Метод «сущность-связь»

В случае если количество атрибутов, хранимых в БД, очень велико (считается, что достаточным пороговым числом является число 20), возникают определенные трудности применения метода декомпозиции, равно как и других методов, в которых анализ F-зависимостей производится на начальном этапе (метода синтеза, например). Одним из методов, широко используемым в настоящее время для нормализации БД с большим числом атрибутов, является метод «сущность-связь». В этом методе F-зависимости анализируются на конечном этапе, когда уже получен набор отношений путем преобразования по заранее известным правилам инфологической модели (ИЛМ), представленной в форме диаграммы «сущность-связь» (*entity-relation diagram*), иначе обозначаемой как ER-диаграмма.

Проектирование схемы реляционной БД в данном методе производится в три этапа:

- 1) формирование ИЛМ в виде ER-диаграммы (осуществляется на этапе инфологического моделирования);
- 2) преобразование ER-диаграммы в первоначальную схему БД;
- 3) анализ каждого отношения первоначальной схемы БД на предмет нахождения в НФБК с последующей декомпозицией найденных ненормализованных отношений.

Третий этап чаще всего не выполняется, т.к. преобразование ER-диаграммы на втором этапе осуществляется по правилам, позволяющим практически всегда получать отношения в НФБК. Приведем основные понятия ER-диаграммы на основе примера, приведенного на рис.6.



Определение. **Сущностью** называется некоторый объект предметной области, имеющий экземпляры, обладающие одинаковым набором свойств, но отличающиеся друг от друга значениями этих свойств. Экземпляры одной сущности должны допускать однозначную идентификацию. На ER-диаграмме сущности обозначаются прямоугольниками. Сущностями, представленными на рис.6, являются **ПРЕПОДАВАТЕЛЬ** и **КУРС**. *Определение.* **Связь** представляет собой логическое соединение между двумя или более сущностями. Связь имеет экземпляры. Экземпляры связи соединяют экземпляры сущностей. На ER-диаграмме связи обозначаются ромбами, связанными с соответствующими сущностями. На рис.6 представлена одна связь – **ЧИТАЕТ**, соединяющая сущности **ПРЕПОДАВАТЕЛЬ** и **КУРС**.

Определение. **Атрибут** есть свойство сущности или связи. Атрибут (набор атрибутов), по значению(ям) которого(ых) можно однозначно идентифицировать экземпляр сущности, называется **ключом сущности**. **Ключом связи**, позволяющим однозначно идентифицировать экземпляр связи, является набор ключей связанных сущностей. На ER-диаграмме атрибуты подписываются под сущностями или связями, которые они определяют. Ключи подчеркиваются или помечаются знаком #. На рис.6 сущность **ПРЕПОДАВАТЕЛЬ** имеет атрибуты **ПРЕП#** (ключ), **ФАМ**, **ПАУД**, **ТЕЛ**, **ДОЛЖНОСТЬ**, **АДРЕС**. Сущность **КУРС** имеет атрибуты **ШИФР#** (ключ), **НАИМЕНОВАНИЕ**, **КОЛ_ЧАСОВ**. Ключом для связи **ЧИТАЕТ** является пара **ПРЕП#**, **ШИФР#**. Неключевыми атрибутами связи **ЧИТАЕТ** являются **ДЕНЬ_НЕД** и **ЧАСЫ**.

Определение. Степенью связи называется соотношение 1:1 (один-к-одному), 1:М (один-ко-многим), М:1 (многие-к-одному) или М:N (многие-ко-многим), которое указывает на возможность или невозможность одновременной связи одного экземпляра одной сущности со многими экземплярами другой (рис.7).



Рис.7

На рис.6 между сущностями **ПРЕПОДАВАТЕЛЬ** и **КУРС** ER-диаграммы показана степень связи M:N. *Определение.* **Класс принадлежности сущности к связи называется обязательным**, если каждый экземпляр сущности охвачен этой связью. **Класс принадлежности сущности к связи называется необязательным**, если могут существовать экземпляры сущности, не охваченные этой связью. Данное определение иллюстрирует рис.8.

<table border="0"> <thead> <tr> <th>ПРЕПОДАВАТЕЛЬ</th> <th>ЧИТАЕТ</th> <th>КУРС</th> </tr> </thead> <tbody> <tr> <td>ИВАНОВ</td> <td>_____</td> <td>ОИ242</td> </tr> <tr> <td>КУЗНЕЦОВ</td> <td>_____</td> <td>ОК142</td> </tr> <tr> <td>СМИРНОВ</td> <td>_____</td> <td>ОИ241</td> </tr> </tbody> </table>	ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС	ИВАНОВ	_____	ОИ242	КУЗНЕЦОВ	_____	ОК142	СМИРНОВ	_____	ОИ241	<table border="0"> <thead> <tr> <th>ПРЕПОДАВАТЕЛЬ</th> <th>ЧИТАЕТ</th> <th>КУРС</th> </tr> </thead> <tbody> <tr> <td>ИВАНОВ</td> <td>_____</td> <td>ОИ242</td> </tr> <tr> <td>КУЗНЕЦОВ</td> <td>_____</td> <td>ОК142</td> </tr> <tr> <td>СМИРНОВ</td> <td>_____</td> <td>• ОИ241</td> </tr> </tbody> </table>	ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС	ИВАНОВ	_____	ОИ242	КУЗНЕЦОВ	_____	ОК142	СМИРНОВ	_____	• ОИ241
ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС																							
ИВАНОВ	_____	ОИ242																							
КУЗНЕЦОВ	_____	ОК142																							
СМИРНОВ	_____	ОИ241																							
ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС																							
ИВАНОВ	_____	ОИ242																							
КУЗНЕЦОВ	_____	ОК142																							
СМИРНОВ	_____	• ОИ241																							
<p>а) классы принадлежности: обязательный-обязательный</p>	<p>б) классы принадлежности: обязательный-необязательный</p>																								
<table border="0"> <thead> <tr> <th>ПРЕПОДАВАТЕЛЬ</th> <th>ЧИТАЕТ</th> <th>КУРС</th> </tr> </thead> <tbody> <tr> <td>ИВАНОВ</td> <td>• _____</td> <td>ОИ242</td> </tr> <tr> <td>КУЗНЕЦОВ</td> <td>_____</td> <td>ОК142</td> </tr> <tr> <td>СМИРНОВ</td> <td>_____</td> <td>ОИ241</td> </tr> </tbody> </table>	ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС	ИВАНОВ	• _____	ОИ242	КУЗНЕЦОВ	_____	ОК142	СМИРНОВ	_____	ОИ241	<table border="0"> <thead> <tr> <th>ПРЕПОДАВАТЕЛЬ</th> <th>ЧИТАЕТ</th> <th>КУРС</th> </tr> </thead> <tbody> <tr> <td>ИВАНОВ</td> <td>• _____</td> <td>ОИ242</td> </tr> <tr> <td>КУЗНЕЦОВ</td> <td>_____</td> <td>ОК142</td> </tr> <tr> <td>СМИРНОВ</td> <td>_____</td> <td>• ОИ241</td> </tr> </tbody> </table>	ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС	ИВАНОВ	• _____	ОИ242	КУЗНЕЦОВ	_____	ОК142	СМИРНОВ	_____	• ОИ241
ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС																							
ИВАНОВ	• _____	ОИ242																							
КУЗНЕЦОВ	_____	ОК142																							
СМИРНОВ	_____	ОИ241																							
ПРЕПОДАВАТЕЛЬ	ЧИТАЕТ	КУРС																							
ИВАНОВ	• _____	ОИ242																							
КУЗНЕЦОВ	_____	ОК142																							
СМИРНОВ	_____	• ОИ241																							
<p>в) классы принадлежности: необязательный-обязательный</p>	<p>г) классы принадлежности: необязательный-необязательный</p>																								

Рис.8

На ER-диаграмме обязательный класс принадлежности сущности обозначается точкой, взятой в рамку, а необязательный – точкой без рамки, установленной на линии связи. На рис.6 сущность **ПРЕПОДАВАТЕЛЬ** имеет необязательный класс принадлежности (т.е. полагается, что может быть преподаватель, не читающий никакой курс), а сущность **КУРС** имеет обязательный класс принадлежности (все курсы должны читаться каким-либо преподавателем). Построением ER-диаграммы предметной области завершается первый этап проектирования схемы БД.

В зависимости от степени связи, классов принадлежностей сущностей, наличия атрибутов связи и арности связи каждая связь на ER-диаграмме на втором этапе преобразуется в одно или несколько отношений схемы БД. *Определение.* Обычным отношением, описывающим сущность, или просто **обычным отношением сущности** будем называть отношение, в которое входят все атрибуты данной сущности и только они одни. *Определение.* **Расширенным отношением сущности** будем называть отношение, в которое входят все атрибуты обычного отношения сущности, а также ключ связанной сущности. *Определение.* **Обычным отношением связи** будем называть отношение, в которое входят все атрибуты данной связи и ключ связи и только они одни. *Определение.* **Расширенным отношением связи** будем называть отношение, в которое входят все атрибуты связи и все атрибуты связанных сущностей.

Правила преобразования связей ER-диаграммы приведены в табл.1. Дадим краткий комментарий к правилам. Если связь бинарная, у нее нет неключевых атрибутов и она имеет степень 1:1, то существует четыре варианта ее преобразования в схему БД.

Таблица 1

Правила преобразования связей ER-диаграммы в отношения БД

№ пп	Характеристика связи	Степень связи	Класс принадлежности	Отношение сущности 1	Отношение связи	Отношение сущности 2
1	Бинарная, нет атрибутов связи	1:1	обяз.-обяз.	–	расширенное	–
2			обяз.-необяз.	расширенное	–	обычное
3			необяз.-обяз.	обычное	–	расширенное
4			необяз.-необяз.	обычное	обычное	обычное
5		1:M	обяз.-обяз.	обычное	–	расширенное
6			обяз.-необяз.	обычное	обычное	обычное
7			необяз.-обяз.	обычное	–	расширенное
8			необяз.-необяз.	обычное	обычное	обычное
9			M:N	любые	обычное	обычное
10	Бинарная, есть атрибуты связи	любая	любые	обычное	обычное	обычное
11	N-арная	любая	любые	обычное	обычное	обычное

1. Если обе сущности имеют обязательные классы принадлежности, то связь преобразуется в одно отношение – расширенное отношение связи, включающее все атрибуты связываемых сущностей; 2. Если первая сущность имеет обязательный, а вторая – необязательные классы принадлежности, то связь преобразуется в два отношения: первое – расширенное отношение первой сущности, второе – обычное отношение второй сущности. 3. Третий вариант аналогичен второму с изменением номеров сущностей. 4. Если обе сущности имеют необязательные классы принадлежности, то связь преобразуется в три отношения: обычное отношение первой сущности, обычное отношение связи и обычное отношение второй сущности. Остальные варианты, приведенные в табл. 1 для других степеней и арностей связи, расшифровываются аналогичным образом.

3. Обеспечение целостности данных

Информация в базе данных не должна быть противоречивой. Это достигается выполнением различных правил, называемых ограничениями целостности. Такие правила проявляются в различных формах.

Контроль значений атрибутов отношения. Здесь могут выполняться различные виды проверок:

- 1) проверка типа данных (не должно быть возможности ввода текстовой информации в поле, предназначенное для хранения чисел);
- 2) проверка интервала (вводимое значение не должно выпадать из указанных границ значений);

3) сравнение полей, которое проверяет значение одного атрибута по значению другого (например, дата приема сотрудника на работу не должна быть меньше даты его рождения).

Целостность ключа. Означает требование, чтобы поля первичного ключа не были пустыми. Кроме того, значения первичного ключа в отношении не должны повторяться.

Ссылочная целостность. Требует, чтобы значению внешнего ключа некоторого кортежа некоторого отношения обязательно соответствовало бы такое же значение первичного ключа другого отношения.

4. Характеристика CASE-технологии и ее применение для разработки логической структуры базы данных

CASE-технология представляет собой методологию проектирования ИС, а также набор инструментальных средств, позволяющих в наглядной форме моделировать предметную область, анализировать эту модель на всех этапах разработки и сопровождения ИС и разрабатывать приложения в соответствии с информационными потребностями пользователей. Большинство существующих CASE-средств основано на методологиях структурного (в основном) или объектно-ориентированного анализа и проектирования, использующих спецификации в виде диаграмм или текстов для описания внешних требований, связей между моделями системы, динамики поведения системы и архитектуры программных средств.

Согласно обзору передовых технологий (Survey of Advanced Technology), составленному фирмой Systems Development Inc. в 1996 г. по результатам анкетирования более 1000 американских фирм, CASE-технология в настоящее время попала в разряд наиболее стабильных информационных технологий. Ее использовала половина всех опрошенных пользователей более чем в трети своих проектов, из них 85% завершились успешно.

Однако, несмотря на все потенциальные возможности CASE-средств, существует множество примеров их неудачного внедрения, в результате чего CASE-средства становятся «полочным» ПО (shelfware). В связи с этим необходимо отметить следующее:

- CASE-средства не обязательно дают немедленный эффект; он может быть получен только спустя какое-то время;

- реальные затраты на внедрение CASE-средств обычно намного превышают затраты на их приобретение;

- CASE-средства обеспечивают возможности для получения существенной выгоды только после успешного завершения процесса их внедрения.

Ввиду разнообразной природы CASE-средств было бы ошибочно делать безоговорочные утверждения относительно реального удовлетворения тех или иных ожиданий от их внедрения. Отметим факторы, усложняющие определение возможного эффекта от использования CASE-средств:

- широкое разнообразие качества и возможностей CASE-средств;
- относительно небольшое время использования CASE-средств в различных организациях и недостаток опыта их применения;
- разнообразии практики внедрения CASE-средств в различных организациях;
- отсутствие детальных метрик и данных для уже выполненных и текущих проектов;
- широкий диапазон предметных областей проектов;
- различная степень интеграции CASE-средств в различных проектах.

5. Существенные свойства и показатели качества баз данных

Оценка качества структур баз данных является необходимой составляющей этапа логического проектирования БД.

Качество БД как сложных информационных структур представляет собой совокупность многих свойств. В этой связи его можно характеризовать разными показателями, отражающими эти свойства. Однако наиболее существенными свойствами, которые отражают аспект практической применимости БД, следует считать объем БД и оперативность выполнения заданных запросов к БД.

Поэтому далее рассматриваются вопросы оценки объема проектируемых БД и оценки оперативности запросов к БД.

6. Методы расчета показателей качества

а) Оценка объема проектируемых баз данных

Оценка объема БД является неотъемлемой частью общей оценки качества проектированной БД. Она предполагает определение объема, который будут занимать во внешней памяти файлы БД.

При реализации БД в среде конкретной СУБД обычно создаются следующие типы файлов:

файлы-таблицы,

файлы-индексы,

файлы-запросы,

файлы-отчеты и прочие файлы.

В файлах-таблицах содержатся непосредственно данные, т.е. кортежи значений атрибутов соответствующего отношения. Файлы-таблицы содержат служебную часть с описанием структуры отношения и область данных, содержащую кортежи значений атрибутов.

Файлы-индексы содержат первичные или вторичные индексы для быстрого доступа к данным файлов-таблиц.

Остальные файлы (файлы-запросы, файлы-отчеты и прочие файлы) предназначены для реализации predetermined запросов пользователей, отчетов и прочих документов.

Расчет объема проектируемой БД основывается на расчете объема, отводимого для областей данных файлов-таблиц. Объем остальной части БД полагается не превышающим 10-20% от этого значения.

Как известно, мощностью отношения, или файла-таблицы, называется максимально возможное число кортежей (записей) в нем.

Используя это понятие, объем области данных i -го файла-таблицы рассчитывается по формуле

$$, \quad (1) \quad V_i = N_i D_i$$

где N_i – мощность i -го файла-таблицы; D_i – длина кортежа i -го файла-таблицы.

Длина кортежа i -го файла равняется сумме длин (размеров) всех атрибутов, составляющих схему отношения этого файла, т.е.

$$, \quad (2) \quad D_i = \sum_{j=1}^{n_i} d_{ij}$$

где d_{ij} – размер j -го поля (атрибута) в i -м файле БД ($j=1, \dots, n_i$).

Итоговая оценка объема БД, содержащей M файлов, определяется выражением

$$V = 1,2 \sum_{i=1}^M D_i = 1,2 \sum_{i=1}^M \sum_{j=1}^{n_i} d_{ij}$$

Мощности отношений БД определяются, исходя из анализа инфологической модели БД (заданной, например, в виде ER-диаграммы) и ее параметров (заданных мощностей сущностей, количественных значений параметров степеней связи и т.д.).

Длины кортежей отношений БД определяются, исходя из заданных размеров полей каждого из атрибутов, что также осуществляется на этапе инфологического проектирования БД.

Для оценки мощностей отношений БД можно воспользоваться следующими соображениями.

Связываемой мощностью сущности относительно заданной связи называется максимально возможное число экземпляров сущности, связанных этой связью.

Очевидно, что для сущностей с обязательным классом принадлежности связываемая мощность равна непосредственно мощности сущности, а для сущностей с необязательным классом принадлежности – меньше мощности сущности.

Мощностью связи на ER-диаграмме называется максимально возможное число экземпляров этой связи.

Если степень связи равна 1:1, то мощность связи равна связываемым мощностям охватываемых ею сущностей.

Если степень связи равна 1:M, то мощность связи равна связываемой мощности второй сущности или произведению связываемой мощности первой сущности на количественное значение параметра M степени связи.

Если степень связи равна M:N, то мощность связи равна произведению связываемой мощности одной сущности на количественное значение параметра M (или N) степени связи другой сущности.

б) Оценка оперативности выполнения запросов к базе данных

При оценке оперативности выполнения запросов в качестве основного показателя чаще всего используется среднее время, выраженное в условных единицах.

Анализ составляющих времени выполнения запросов к БД (времени поиска требуемого блока на диске, времени считывания с диска, времени передачи в ОП, времени процессорной обработки данных и т.п.) позволяет сделать вывод о том, что с большой степенью точности можно полагать это время пропорциональным числу операций обмена между ВЗУ (диск) и ОП, осуществляемых в ходе выполнения запроса. Так как это число не зависит от производительности ЭВМ и во многом определяется структурными решениями по построению БД, именно его представляется целесообразным использовать в качестве условных единиц для оценки оперативности выполнения запросов.

Для оценки числа операций обмена между ВЗУ и ОП (в дальнейшем по тексту – просто обменов) используется следующая модель обработки запросов.

Полагается, что данные на физическом уровне в файлах реляционной БД хранятся последовательно по кортежам одинаковой длины (т.е. за первым кортежем в файле хранится второй, за вторым – третий и т.д.). Длина кортежа равняется сумме длин всех полей, входящих в схему отношения. Объем файла оценивается по формуле (3.3).

В ОП данные из файла считываются блоками (страницами) одинакового размера. Размер блока устанавливается в СУБД (обычно он лежит в диапазоне от 512 байт до 4 Кбайт для различных СУБД).

В ОП данные из файла считываются блоками (страницами) одинакового размера. Размер блока устанавливается в СУБД (обычно он лежит в диапазоне от 512 байт до 4 Кбайт для различных СУБД). В результате получается, что число обменов, необходимое для считывания всего файла, равняется числу блоков, на которое можно разбить этот файл. Однако не всегда при выполнении запроса файл считывается целиком.

Поиск по запросу в реляционных БД осуществляется одним из двух способов: последовательным или прямым (индексным). Второй случай предполагает обязательное наличие для поискового атрибута (дескриптора) индексного файла. В первом случае индексный файл отсутствует. Так как механизмы поиска в этих двух случаях различны, необходимо каждый из них рассмотреть отдельно.