

УПРАВЛЕНИЕ ДАННЫМИ

Лекция 4 **ПРОЕКТИРОВАНИЕ БАЗЫ ДАННЫХ**
ЧАСТЬ 1

Цель проектирования: создать точную и защищенную БД, на основе которой можно гарантировать эффективное построение прикладных программ.

При изучении и разработке БД рассматриваются следующие два аспекта:

- инфологический
- датологический.

Этапы проектирования базы данных

I. Этап: Инфологический

II. Этап: Логическое проектирование

1. Построение внешней модели

2. Построение концептуальной модели

III. Этап: физическое проектирование.

3. Построение внутренней модели



Рисунок 1 -- Процесс проектирования БД:
ИПП – информационные потребности пользователя;
ПрО – предметная область

Инфологический аспект

связан со смысловым содержанием данных независимо от способа их представления в памяти систем.

При инфологическом проектировании решаются вопросы:

- 1) о каких объектах или явлениях реального мира требуется накапливать и обрабатывать информацию в БД;
- 2) какие их основные характеристики и взаимосвязи между собой следует учитывать;
- 3) уточнения вводимых в БД понятий об объекте и явлениях, их характеристиках и взаимосвязях.

На этапе инфологического проектирования выделяется предметная область (ПрО) – информация об объектах, процессах и явлениях окружающего мира, которая с точки зрения потенциальных пользователей должна храниться и обрабатываться в информационной системе.

Даталогический аспект

связан с представлением данных в памяти БД.

При даталогическом проектировании разрабатываются формы представления информации посредством данных, приводятся модели и методы преобразования данных, формулируются правила смысловой интерпретации данных.

Инфологическое проектирование

Обеспечение *информационных потребностей (запросов) пользователей* имеет 2 аспекта:

- 1) определение границ конкретной предметной области и разработка описания соответствующей информационной модели;
- 2) разработка банка данных, ориентированного на эффективное обслуживание запросов различных категорий пользователей.

С точки зрения целевой направленности профессиональной деятельности принято выделять 5 основных категорий пользователей:

- 1) аналитики;
- 2) системные программисты;
- 3) прикладные программисты;
- 4) администраторы;
- 5) конечные пользователи.

Пользователей можно разделить на пользователей-людей и пользователей-задач; пользователей с различным уровнем компетентности или приоритетом и т.д.

Уровень сложности и важности задач информационного обеспечения информационных систем в рамках рассматриваемой технологии определяет ряд **основных требований к банкам данных**:

- адекватность информации состоянию предметной области;
- быстродействие и производительность;
- массовость использования;
- защита информации;
- возможность расширения круга решаемых задач.

На этапе инфологического (информационно-логического) проектирования осуществляется построение *семантической модели*, описывающей сведения из предметной области, которые могут заинтересовать пользователей баз данных.

Сначала из объективной реальности выделяется предметная область.

Анализ информационных потребностей потенциальных пользователей имеет 3 аспекта:

- определение собственно сведений об объектах предметной области;
- анализ возможных запросов к базам данных;
- выявление ограничений целостности.

Предметная область

– это часть реального мира, данные о которой мы хотим отразить в базе данных.

Например, в качестве предметной области можно выбрать бухгалтерию какого-либо предприятия, отдел кадров, банк, магазин и т.д.

Предметная область бесконечна и содержит как существенно важные понятия и данные, так и малозначащие или вообще не значащие данные.

Важность данных зависит от выбора предметной области и представлений пользователей о ней.

Все инфологические модели представляют собой описательные модели предметной области – **концептуальные схемы**. В качестве представления концептуальной схемы будем использовать диаграмму «Сущность-связь» или ER-диаграмму «*Entity-Relationship*».

В основе построения инфологической модели предметной области лежит **анализ представлений пользователей**.

1 шаг построения БД: АНАЛИЗ ПРЕДСТАВЛЕНИЙ ПОЛЬЗОВАТЕЛЕЙ

Одной из важнейших форм контактов с пользователями является «**интервью**».

В результате интервью удастся выяснить **элементы данных**, которыми потенциальный пользователь пользуется или намерен пользоваться в своей работе.

Опросив всех потенциальных пользователей, проектировщик получает набор их представлений о предметной области.

Для дальнейшего построения базы данных требуется объединить разрозненные представления в логическую схему базы данных.

Алгоритм объединения представлений пользователей

Решаются три задачи:

- выявление сущностей,
- установление связей между ними
- распределение атрибутов по сущностям или связям.

Входные данные: представления пользователей

Выходные данные: концептуальная схема предметной области

Метод:

1. Анализируем всю систему представлений пользователей.

1.1. Составляем список элементов данных, подлежащих включению в базу данных.

1.2. Унифицируем терминологию.

1.3. Придаем рассматриваемому элементу данных наиболее информативное имя.

1.4. Добавляем недостающие (уточняющие) элементы данных

2. Выявляем элементы данных, которые можно рассматривать как сущности.

2.1. Среди элементов данных выбираем существительные, которые имеют атрибуты, явно к ним относящиеся.

Для формализации процесса строим квадратную таблицу (В), назовем ее таблицей совместимости, в которой каждому элементу данных из списка соответствует строка и столбец.

Элемент b_{ij} таблицы принимает значения 1 в том и только том случае, если элемент данных, соответствующий j -му столбцу очевидным образом является атрибутом элемента, соответствующего i -ой строке (пример таблицы совместимости представлен на рис. 2).

2.2. Составляем список элементов, которым в таблице (В) соответствуют ненулевые строки – эти элементы образуют сущности

2.3. Для каждой сущности составляем список атрибутов (элементы, которым в строке сущности соответствует 1) – они образуют исходное множество атрибутов.

2.4. Выбираем первичный ключ для каждой сущности.

3. Выявляем связи между найденными сущностями.

3.1. Находим пары сущностей, имеющих одинаковые атрибуты.

3.2. Устанавливаем остальные связи между сущностями. Это делается, исходя из семантики элементов данных, например, если одна сущность оказывает какое-либо действие на другую и т.д.

4. Из таблицы (B) удаляем строки и столбцы, соответствующие уже рассмотренным элементам данных.

В таблице останутся элементы, из которых могут быть получены дополнительные сущности и дополнительные атрибуты.

5. Строим ER-диаграмму, используя выявленные сущности и связи, и распределяя атрибуты между ними согласно таблице совместимости.

Рассмотрим работу этого алгоритма на примере.

Пусть необходимо спроектировать базу данных для отдела контейнеров компании, осуществляющей морские перевозки.

Допустим, что в результате интервью со служащими компании получены представленные на рис.1 наборы элементов данных, соответствующие представлениям актуальных и потенциальных пользователей

Пример

представление первого пользователя

судно	грузоподъемность	особенности
-------	------------------	-------------

представление второго пользователя

судно		
↓		
порт	дата прибытия	дата отплытия

представление третьего пользователя

грузополучатель

↓

контейнер#	порт назначения	судно	дата прибытия	агент
------------	-----------------	-------	---------------	-------

представление четвертого пользователя

агент

↓

контейнер#

накладная#

порт отправления

грузополучатель

порт назначения

судно

дата доставки

Рисунок

представление пятого пользователя

накладная#	порт отправления	порт назначения	грузополучатель	дата отплытия
------------	------------------	-----------------	-----------------	---------------

дата доставки	судно	агент	контейнер#	содержимое	категория разгрузки	размер
---------------	-------	-------	------------	------------	---------------------	--------

представление шестого пользователя

судно	порт	контейнер#
↓		

представление седьмого пользователя

судно	порт	контейнер#	размер контейнера	категория разгрузки	порт назначения
-------	------	------------	-------------------	---------------------	-----------------

Рисунок 1 -- Представления пользователей

Выполняем последовательность шагов, соответствующую представленному алгоритму

1. Анализируем всю систему представлений.

Унифицируем терминологию. В результате дополнительного опроса выясняется, что «размер» и «размер контейнера» -- имена одного и того же атрибута. Поэтому включаем только одно из этих имен – «размер контейнера», поскольку это имя более содержательно.

В результате получаем следующий набор элементов данных, который должен быть отображен в проектируемой базе:

{судно, грузоподъемность, особенности, порт, контейнер#, дата прибытия, дата отправки, дата отплытия, дата доставки, порт назначения, порт отправления, агент, грузополучатель, содержимое, категория разгрузки, размер контейнера, накладная#, контейнер, имя судна, накладная, порт название}. (1)

2. Выявляем элементы данных, которые можно рассматривать как сущности.

1) Для этого строим квадратную таблицу (B). Таблица (B) для рассматриваемого примера представлена на рис.2.

	судно	грузоподъемность	особенности	порт	Контейнер#	Дата прибытия	Дата отправки	Дата отплытия	Дата доставки	Порт назначения	Порт отправления	агент	грузополучатель	содержимое	Категория разгрузки	размер	Накладная#	контейнер	Имя судна	накладная	Порт название
судно	-	1	1			1		1											1		
грузоподъемность		-																			
особенности			-																		
порт				-		1		1											1		1
Контейнер#					-																
Дата прибытия						-															
Дата отправки							-														
Дата отплытия								-													
Дата доставки									-												
Порт назначения										-											
Порт отправления											-										
агент												-									
грузополучатель													-								
содержимое														-							
Категория разгрузки															-						
размер																-					
Накладная#																	-				
контейнер					1		1		1	1	1		1	1	1	1	1	-			
Имя судна																			-		
накладная					1		1		1	1	1		1				1			-	
портназвание																					-

Рисунок 2 -- Таблица совместимости (В)

2). Составляем список элементов, которым в таблице (В) соответствуют ненулевые строки и находим соответствующие им элементы, которым в строке соответствует 1 – они образуют исходное множество сущностей и их атрибутов.

По нашему примеру получаем следующий список

СУДНО (грузоподъемность, особенности, дата прибытия, дата отплытия, имя-судна)

ПОРТ (дата прибытия, дата отплытия, порт-название, имя-судна)

КОНТЕЙНЕР (контейнер#, содержимое, категория разгрузки, размер, дата отправки, дата доставки, порт назначения, накладная#, грузополучатель, порт отправления)

НАКЛАДНАЯ (накладная#, грузополучатель, контейнер#, дата отправки, дата доставки, порт назначения, порт отправления).

Выберем первичные ключи для нашего примера:

для сущности СУДНО выберем в качестве ключа атрибут «имя-судна»;

для сущности ПОРТ – «порт-название»;

для сущности КОНТЕЙНЕР -- «контейнер#»;

для сущности НАКЛАДНАЯ – «накладная#».

На ER-диаграмме первичные ключи выделяются подчеркиванием (рис. 3).

3. Выявляем связи между найденными сущностями.

Находим пары сущностей, имеющих одинаковые атрибуты.

1) Сущности СУДНО и ПОРТ имеют три общих атрибута – *дата прибытия, дата отплытия и имя-судна.*

Поскольку эти атрибуты можно отнести как к сущности СУДНО, так и к сущности ПОРТ, то наилучшим решением является отнесение этих атрибутов к связи между сущностью СУДНО и сущностью ПОРТ, которая определяет заходы судна в порты, и дать этой связи имя, например, **РЕЙС**.

2) пара сущностей, имеющих общие атрибуты – КОНТЕЙНЕР и НАКЛАДНАЯ: их общие атрибуты -- *накладная#, контейнер#, грузополучатель, дата отправки, дата доставки, порт-назначения, порт-отправления.*

Накладная – это документ, сопровождающий несколько контейнеров одного грузополучателя. Аналогично общие атрибуты совершенно равнозначно могут быть отнесены как к сущности КОНТЕЙНЕР, так и к сущности НАКЛАДНАЯ.

Поэтому наиболее целесообразное решение состоит в отнесении общих атрибутов *к связи* между сущностями КОНТЕЙНЕР и НАКЛАДНАЯ. Назовем эту связь – **СОПРОВОЖДАЕТ**.

Если среди общих атрибутов встречаются уникальные --идентификаторы сущностей (первичные ключи), например, *накладная#* и *контейнер#*, то их надо оставить в качестве атрибутов к соответствующим сущностям (рис. 3).

3) Выявим другие связи между выделенными сущностями.

Рассмотрение элементов данных, используемых *третьим, четвертым, шестым и седьмым* пользователями показывает, что сущность **КОНТЕЙНЕР** связана с сущностью **СУДНО**.

Степень связи N:1, класс принадлежности обязательный для сущности КОНТЕЙНЕР, потому, что суда не перевозящие контейнеры нашу организацию не интересуют.

Связи СУДНО-НАКЛАДНАЯ, ПОРТ-НАКЛАДНАЯ, ПОРТ-КОНТЕЙНЕР можно не рассматривать: при необходимости они могут быть образованы из уже выделенных связей.

Такая необходимость может возникнуть, если появятся новые пользователи со своими наборами тех же элементов данных.

4. Из таблицы (В) удаляем строки и столбцы, соответствующие уже рассмотренным элементам данных.

После удаления из таблицы (В) уже рассмотренных элементов, остается один - единственный элемент – **АГЕНТ**.

Делаем его сущностью АГЕНТ с единственным атрибутом – имя-агента.

Из анализа представлений следует, что мы можем связать эту сущность либо с сущностью КОНТЕЙНЕР, либо с сущностью НАКЛАДНАЯ.

Поскольку агент сопровождает не один контейнер, а может сопровождать несколько партий контейнеров, на которые выписаны накладные, то более логично установить связь между сущностями НАКЛАДНАЯ и АГЕНТ.

Назовем установленную связь – **КОНТРОЛИРУЕТ**.

На рис. 3 представлена ER-диаграмма, являющаяся концептуальной моделью рассмотренной предметной области.

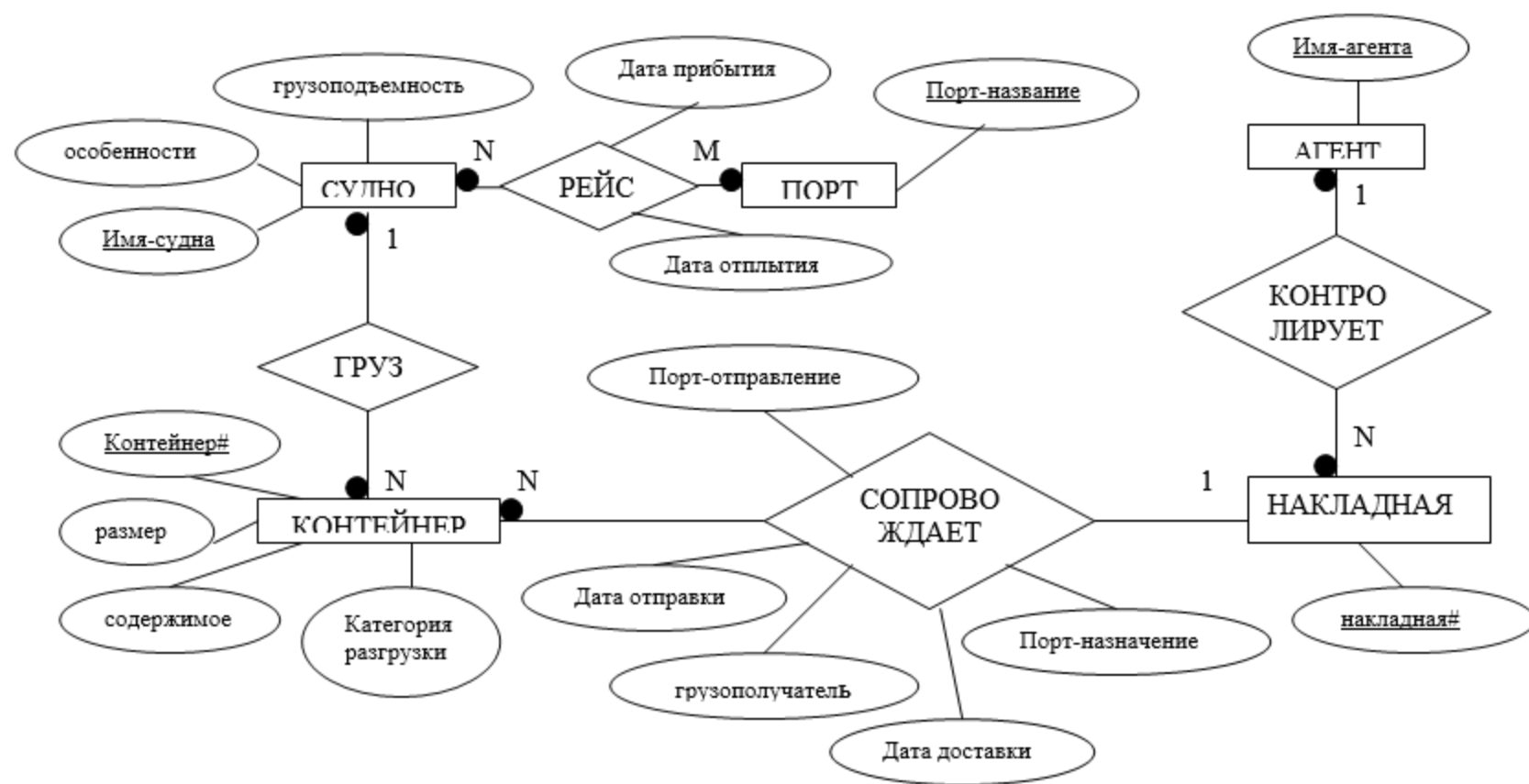


Рисунок 3 -- ER –диаграмма рассматриваемой предметной области

СПАСИБО ЗА ВНИМАНИЕ!