

ПМ02

Разработка и администрирование баз данных

Поиск по сайту

НАВИГАЦИЯ

КОНЦЕПЦИИ БАЗ ДАННЫХ

ОСНОВНЫЕ ПОНЯТИЯ БАЗ ДАННЫХ

ЛР СОЗДАНИЕ БАЗЫ ДАННЫХ

ТЕСТ 0

ВВЕДЕНИЕ В MYSQL

ТЕСТ ВВЕДЕНИЕ В MYSQL

ДЕНВЕР

ССЫЛКА ДЛЯ СКАЧИВАНИЯ ДЕНВЕР

СОЗДАНИЕ БД И ТАБЛИЦ С ПОМОЩЬЮ ЯЗЫКА ЗАПРОСОВ

ЛР СОЗДАНИЕ БД И ТАБЛИЦ В MYSQL

ТЕСТ MYSQL СОЗДАНИЕ БД И ТАБЛИЦ

Основные понятия баз данных

Развитие вычислительной техники осуществлялось по двум основным направлениям:

- применение вычислительной техники для выполнения численных расчетов;
- использование средств вычислительной техники в информационных системах.

Информационная система — это совокупность программно-аппаратных средств, способов и людей, которые обеспечивают сбор, хранение, обработку и выдачу информации для решения поставленных задач. На ранних стадиях использования информационных систем применялась файловая модель обработки. В дальнейшем в информационных системах стали применяться базы данных. Базы данных являются современной формой организации, хранения и доступа к информации. Примерами крупных информационных систем являются банковские системы, системы заказов железнодорожных билетов и т.д.

Базы данных находят свое применение в информационных системах

СОЗДАНИЕ БАЗЫ ДАННЫХ В PHPMYADMIN

ЛР_СОЗДАНИЕ БД В PHPMYADMIN

ТЕСТ СОЗДАНИЕ БД И ТАБЛИЦ В PHPMYADMIN

СОЗДАНИЕ ПРИЛОЖЕНИЯ ДЛЯ РАБОТЫ С БД MYSQL В LAZARUS

ЛР_СОЕДИНЕНИЕ С БД

ТЕСТ ПОДКЛЮЧЕНИЕ БД К ПРИЛОЖЕНИЮ LAZARUS

ЛР_РАБОТА С ЗАПИСЯМИ ТАБЛИЦЫ MYSQL В LAZARUS

ТЕСТ_LAZARUS РАБОТА С ЗАПИСЯМИ ТАБЛИЦЫ

ИНФОРМАЦИОННЫЕ СИСТЕМЫ

ТЕСТ 1

ЖИЗНЕННЫЙ ЦИКЛ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ ИС

ТЕСТ2

КНИГА ЖАЛОБ И ПРЕДЛОЖЕНИЙ

КАРТА САЙТА

База данных – это интегрированная совокупность структурированных и взаимосвязанных данных, организованная по определенным правилам, которые предусматривают общие принципы описания, хранения и обработки данных. Обычно база данных создается для предметной области.

Предметная область – это часть реального мира, подлежащая изучению с целью создания базы данных для автоматизации процесса управления.

Наборы принципов, которые определяют организацию логической структуры хранения данных в базе, называются **моделями данных**.

Существуют **4 основные модели данных** – списки (плоские таблицы), реляционные базы данных, иерархические и сетевые структуры.

В течение многих лет преимущественно использовались плоские таблицы (плоские БД) типа списков в Excel. В настоящее время наибольшее распространение при разработке БД получили реляционные модели данных. **Реляционная модель данных** является совокупностью простейших двумерных таблиц – отношений (англ. relation), т.е. ~~простейшая двумерная таблица~~ определяется как отношение (множество однотипных записей объединенных одной темой).

~~От термина relation (отношение) происходит название реляционная модель данных.~~ В реляционных БД используется несколько двумерных таблиц, в которых строки называются записями, а столбцы полями, между записями которых устанавливаются связи. Этот способ организации данных позволяет данные (записи) в одной таблице связывать с данными (записями) в других таблицах через уникальные идентификаторы (ключи) или ключевые поля.

Основные понятия реляционных БД: нормализация, связи и ключи

1. Принципы нормализации:

1. без повторов полей
2. в каждой таблице - уникальный ID
3. Каждому значению ключа - достаточная информация об объекте таблицы
4. Изменение значения в одном поле не изменяет значения в других полях

- В каждой таблице БД не должно быть повторяющихся полей;
- В каждой таблице должен быть уникальный идентификатор (первичный ключ);
- Каждому значению первичного ключа должна соответствовать достаточная информация о типе сущности или об объекте таблицы (например, информация об успеваемости, о группе или студентах);
- Изменение значений в полях таблицы не должно влиять на информацию в других полях (кроме изменений в полях ключа).

2. Виды логической связи.

Связь устанавливается между двумя общими полями (столбцами) двух таблиц. Существуют связи с отношением «один-к-одному», «один-ко-многим» и «многие-ко-многим».

Отношения, которые могут существовать между записями двух таблиц:

- один — к — одному, каждой записи из одной таблицы соответствует одна запись в другой таблице;
- один — ко — многим, каждой записи из одной таблицы соответствует несколько записей другой таблице;
- многие — к — одному, множеству записей из одной таблице соответствует одна запись в другой таблице;
- многие — ко — многим, множеству записей из одной таблицы соответствует несколько записей в другой таблице.

Тип отношения в создаваемой связи зависит от способа определения связываемых полей:

Что и когда

1. Отношение «один-ко-многим» создается в том случае, когда только одно из полей является полем первичного ключа или уникального индекса.

2. Отношение «**один-к-одному**» создается в том случае, когда оба связываемых поля являются ключевыми или имеют уникальные индексы.
3. Отношение «**многие-ко-многим**» фактически является двумя отношениями «один-ко-многим» с третьей таблицей, первичный ключ которой состоит из полей внешнего ключа двух других таблиц

3. Ключи. **Ключ** – это столбец (может быть несколько столбцов), добавляемый к таблице и позволяющий установить связь с записями в другой таблице. Существуют **ключи двух типов: первичные и вторичные или внешние.**

Первичный ключ – это одно или несколько полей (столбцов), комбинация значений которых однозначно определяет каждую запись в таблице. Первичный ключ не допускает значений Null и всегда должен иметь уникальный индекс. Первичный ключ используется для связывания таблицы с внешними ключами в других таблицах.

Внешний (вторичный) ключ - это одно или несколько полей (столбцов) в таблице, содержащих ссылку на поле или поля первичного ключа в другой таблице. Внешний ключ определяет способ объединения таблиц.

Из двух логически связанных таблиц одну называют таблицей первичного ключа или главной таблицей, а другую таблицей вторичного (внешнего) ключа или подчиненной таблицей. СУБД позволяют сопоставить родственные записи из обеих таблиц и совместно вывести их в форме, отчете или запросе.

Существует **три типа первичных ключей:** ключевые поля счетчика (счетчик), простой ключ и составной ключ.

Поле счетчика (Тип данных «Счетчик»). Тип данных поля в базе данных, в котором для каждой добавляемой в таблицу записи в поле

Индексирование записей (как я понимаю)

автоматически заносится уникальное числовое значение.

Любой простой (одно поле) идентификационный ключ

Простой ключ. Если поле содержит уникальные значения, такие как коды или инвентарные номера, то это поле можно определить как первичный ключ. В качестве ключа можно определить любое поле, содержащее данные, если это поле не содержит повторяющиеся значения или значения **Null**.

Составной ключ. В случаях, когда невозможно гарантировать уникальность значений каждого поля, существует возможность создать ключ, состоящий из нескольких полей. Чаще всего такая ситуация возникает для таблицы, используемой для связывания двух таблиц многие - ко - многим.

Необходимо еще раз отметить, что в поле первичного ключа должны быть только уникальные значения в каждой строке таблицы, т.е. совпадение не допускается, а в поле вторичного или внешнего ключа совпадение значений в строках таблицы допускается.

Если возникают затруднения с выбором подходящего типа первичного ключа, то в качестве ключа целесообразно выбрать поле счетчика.

Программы, которые предназначены для структурирования информации, размещения ее в таблицах и манипулирования данными называются **системами управления базами данных (СУБД)**. Другими словами СУБД предназначены как для создания и ведения базы данных, так и для доступа к данным. В настоящее время насчитывается более 50 типов СУБД для персональных компьютеров. К наиболее распространенным типам СУБД относятся: MS SQL Server, Oracle, Informix, Sybase, MS Access и т. д.

Создание БД. Этапы проектирования

Создание БД начинается с проектирования.

Прежде всего **Этапы проектирования БД:**

1. Предметная область
2. Анализ данных
3. Определение отношений

- исследование предметной области;
- анализ данных (сущностей и их атрибутов);
- определение отношений между сущностями и определение первичных и вторичных (внешних) ключей.

В процессе проектирования определяется структура реляционной БД (состав таблиц, их структура и логические связи). **Структура таблицы** определяется составом столбцов, типом данных и размерами столбцов, ключами таблицы.

К базовым понятиями модели БД «сущность – связь» относятся: сущности, связи между ними и их атрибуты (свойства).

Сущность – любой конкретный или абстрактный объект в рассматриваемой предметной области. Сущности – это базовые типы информации, которые хранятся в БД (в реляционной БД каждой сущности назначается таблица). К сущностям могут относиться: студенты, клиенты, подразделения и т.д. **Экземпляр сущности** и **тип сущности** – это разные понятия. Понятие тип сущности относится к набору однородных личностей, предметов или событий, выступающих как целое (например, студент, клиент и т.д.). Экземпляр сущности относится, например, к конкретной личности в наборе. Типом сущности может быть студент, а экземпляром – Петров, Сидоров и т. д.

Атрибут – это свойство сущности в предметной области. Его наименование должно быть уникальным для конкретного типа сущности. Например, для сущности студент могут быть использованы следующие атрибуты: фамилия, имя, отчество, дата и место рождения, паспортные данные и т.д. В реляционной БД атрибуты хранятся в полях таблиц.

Связь – взаимосвязь между сущностями в предметной области. Связи представляют собой соединения между частями БД (в реляционной БД – это соединение между записями таблиц).

Сущности – это данные, которые классифицируются по типу, а связи показывают, как эти типы данных соотносятся один с другим. Если описать некоторую предметную область в терминах сущности – связь, то получим **модель сущность - связь** для этой БД.

Пример: Рассмотрим предметную область: **Деканат (Успеваемость студентов)**.

В БД «Деканат» должны храниться данные о студентах, группах студентов, об оценках студентов по различным дисциплинам, о преподавателях, о стипендиях и т.д. Ограничимся данными о студентах, группах студентов и об оценках студентов по различным дисциплинам. Определим сущности, атрибуты сущностей и основные требования к функциям БД с ограниченными данными.

Основными предметно-значимыми **сущностями** БД «Деканат» являются: Студенты, Группы студентов, Дисциплины, Успеваемость.

Основные предметно-значимые **атрибуты** сущностей:

- студенты – фамилия, имя, отчество, пол, дата и место рождения, группа студентов;
- группы студентов – название, курс, семестр;
- дисциплины – название, количество часов;
- успеваемость – оценка, вид контроля.

Основные требования к функциям БД:

- выбрать успеваемость студента по дисциплинам с указанием общего количества часов и вида контроля;
- выбрать успеваемость студентов по группам и дисциплинам;

- выбрать дисциплины, изучаемые группой студентов на определенном курсе или определенном семестре.

Из анализа данных предметной области следует, что каждой сущности необходимо назначить простейшую двумерную таблицу (отношения). Далее необходимо установить логические связи между таблицами. Между таблицами Студенты и Успеваемость необходимо установить такую связь, чтобы каждой записи из таблицы Студенты соответствовало несколько записей в таблице Успеваемость, т.е. один – ко – многим, так как у каждого студента может быть несколько оценок.

Логическая связь между сущностями Группы – Студенты определена как один – ко – многим исходя из того, что в группе имеется много студентов, а каждый студент входит в состав одной группы. Логическая связь между сущностями Дисциплины – Успеваемость определена как один – ко – многим, потому что по каждой дисциплине может быть поставлено несколько оценок различным студентам.

На основе вышеизложенного составляем модель сущность – связь для БД «Деканат»

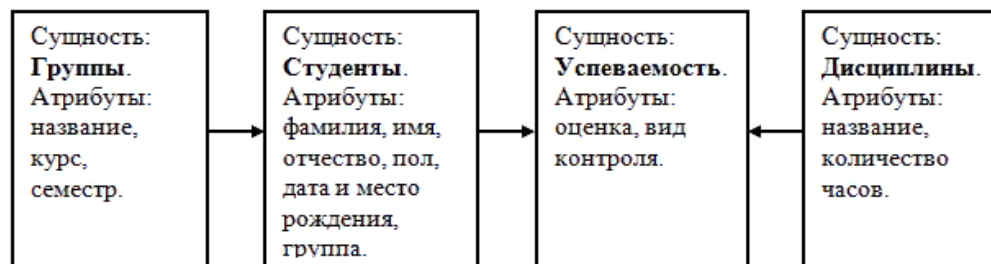


Рис. 1.

→ - стрелка является условным обозначением связи: один – ко – многим.

Для создания БД необходимо применить одну из известных СУБД, например СУБД Access.

Комментарии

У вас нет прав для добавления комментариев.

[Войти](#) | [Последняя активность сайта](#) | [Пожаловаться](#) | [Печать страницы](#) | На основе **Google Сайтов**