

Программное обеспечение работы с современными базами данных

Проблемы создания и ведения реляционных баз данных

Основные задачи программного обеспечения баз данных

При работе с реляционными базами данных можно условно выделить две основные задачи:

- собственно работа с базой данных , включающая создание и ведение базы данных (создание структур таблиц, добавление записи в таблицу, удаление записи, обновление, выборка нужной записи);
- создание пользовательских приложений, включающих разработку пользовательского интерфейса по работе с базой данных.

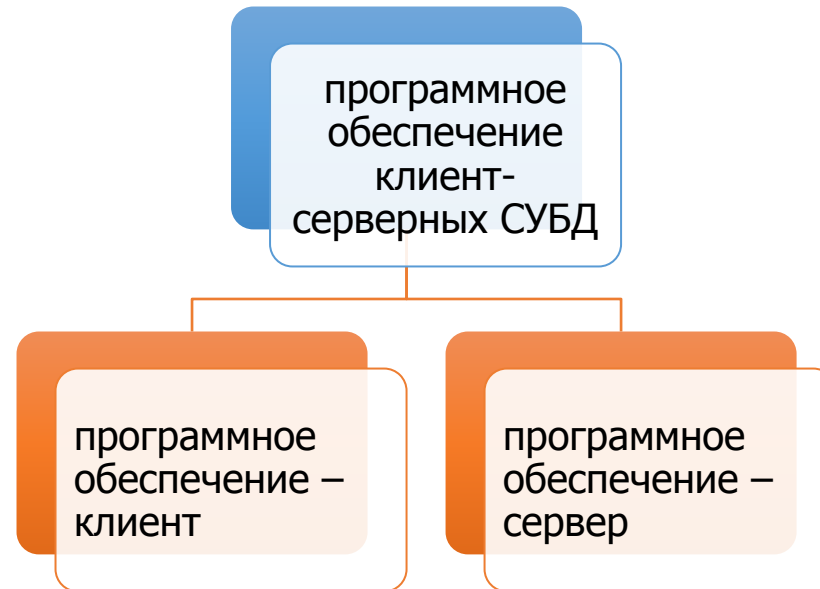
Для решения указанных задач современные СУБД в своем составе могут содержать следующие программные средства:

- языки процедурного пошагового программирования,
- средства визуального программирования (графический интерфейс, диспетчер проекта, мастера и построители),
- средства создания объектно-ориентированных приложений.

Кроме этого, при разработке пользовательских программ во многих СУБД допускается использование других языков программирования, а также использование библиотек разного рода.

В работе клиент-серверных систем участвуют два типа компьютеров (сервер и клиент).

Соответственно, различают клиентское и серверное программное обеспечение.



Возможные варианты использования программного обеспечения в СУБД MS SQL Server

Средства ведения баз данных на
сервере MS SQL

Службы SQL-сервер (MS SQL Server и др.)

Средства разработки клиентских
приложений

- Программное обеспечение клиента SQL-сервер (Transact SQL, SQL Server Query Analyzer и др.)
- MS Access (ODBC)
- MS Visual Basic
- MS Visual Studio
- MS Visual FoxPro
- Java (JDBC)
- Borland Delphi
- Borland C++Builder и др.
- Библиотеки функций (API, ODBC и др.)

Проблемы создания и ведения реляционных баз данных

При создании базы данных и организации работы с ней возникают три основные проблемы:

- собственно создание базы данных (создание таблиц, индексов, ограничений целостности);
- обеспечение безопасности и разграничения доступа;
- организация доступа к элементам таблицы (выборка, редактирование, удаление, добавление).

Организация доступа к базе данных является важнейшей функцией информационной системы.

Рассмотрим простой пример.

Пусть у нас есть таблица СТУДЕНТ, хранящая информацию следующего рода:
СТУДЕНТ (Код студента, Фамилия, Имя, Отчество, Дата поступления).

Теперь мы хотим выполнить некоторый запрос к базе данных, результатом которого должны стать те строки таблицы СТУДЕНТ, для которых дата поступления окажется больше 01.06.2016.

Рассмотрим последовательность действий для реализации данного запроса.

- Получаем доступ к таблице СТУДЕНТ и устанавливаем указатель текущей строки на первую строку таблицы.
- Анализируем поле "Дата поступления" в текущей строке.
- Если значение "Дата поступления" > "01.06.2016", распечатываем на экране данные об абитуриенте.
- Если таблица не кончилась, перемещаем указатель текущей строки на следующую строку и переходим к шагу 2, иначе заканчиваем работу.

Пример реализации на Object Pascal

```
Table.First;
while (not Table.Eof) do
begin
    if FieldByName("Дата поступления").Value >
        "01.06.2016"
    then List.Add(FieldByName("Фамилия").AsString);
    Table.Next;
end;
```


Запрос SQL

```
SELECT Фамилия  
FROM Студент  
WHERE Дата поступления > "01.06.2016"
```

SQL (Structured Query Language)

Структурированный язык запросов

Состоит из трех частей:

- DDL (Data Definition Language) – язык определения данных. Предназначен для создания базы данных (таблиц, индексов и т.д.) и редактирования ее схемы.
- DCL (Data Control Language) – язык управления данными. Содержит операторы для разграничения доступа пользователей к объектам базы данных.
- DML (Data Manipulation Language) – язык обработки данных. Содержит операторы для внесения изменений в содержимое таблиц базы данных.