

Тема:

БАЗА

ДАННЫХ

SQLITE

Вопрос 1 - Особенности базы данных

При работе с базой данных пользователя, как правило, интересует не все ее содержимое, а некоторая конкретная информация.

Найти нужные сведения можно последовательным просмотром записей. Однако такой способ поиска неудобен и малоэффективен.

Большинство систем управления базами данных позволяют произвести выборку нужной информации путем выполнения запросов.

Пользователь в соответствии с определенными правилами формулирует запрос, указывая, каким критериям должна удовлетворять интересующая его информация, а система выводит записи, удовлетворяющие запросу.



Введение

*Большинство современных СУБД
построено на реляционной модели
данных.*

*Для получения информации из
отношений (таблиц) базы данных
в качестве языка манипулирования
данными в теоретическом плане
используется язык *SQL**

Реляционная модель данных

- Реляционная модель ориентирована на организацию данных в виде двумерных таблиц.
- Реляционная таблица представляет двумерный массив и обладает свойствами:
 - Каждый элемент таблицы – один элемент данных
 - Все столбцы в таблице однородные, т.е. все элементы в столбце имеют одинаковый тип (числовой, символьный и т.д.) и длину
 - Каждый столбец имеет уникальное имя
 - Одинаковые строки в таблице отсутствуют
 - Порядок следования строк и столбцов может быть произвольным

Пример реляционной таблицы

№ личного дела	Фамилия	Имя	Отчество	Дата рождения	Группа
16493	Сергеев	Петр	Михайлович	01.01.76	111
16593	Петрова	Анна	Владимировна	15.03.75	112
16693	Анохин	Андрей	Борисович	14.04.76	113

Вопрос 2 - Язык разработки баз данных SQL



***SQL** – структурированный язык запросов, предназначенный для работы с БД реляционного типа.*

SQL является интерактивным языком запросов, который обеспечивает пользователю быстрый доступ к данным.

SQL является также языком программирования баз данных.

Программисты могут вставить SQL-запросы в свои программы, чтобы получить доступ к базам данных

SQL – язык распределения базы данных, служит для распределения данных взаимодействующих систем, для распределенной обработки баз данных

Функции языка SQL:

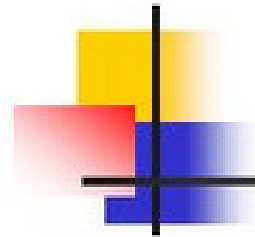
- *Организация данных – создание и изменение структуры баз данных*
- *Чтение данных*
- *Обработка данных – удаление, добавление и корректировка данных*
- *Управление доступа к данным – предоставление привилегий (ограничение возможностей) пользователю для чтения и изменения данных*
- *Совместное использование данных-координация общего пользования данных многими пользователями*
- *Целостность данных – защита данных от разрушения при сбое системы или других обстоятельствах*



SQL в компьютерной сети

*Сервер базы данных выполняет
SQL – запрос и возвращает
пользователю только ту
информацию из базы данных,
которая соответствует этому
SQL - запросу*

Элементы языка SQL



- *Ключевые слова*
- *Имена*
- *Константы*
- *Типы данных*
- *Встроенные функции*
- *выражения*

Ключевые слова. Имена. Константы

Ключевые слова – это



фиксированный набор английских слов, которые определяют тип запроса и необходимую информацию для выполнения этого запроса

Имена используются для

обозначения (присвоения имени) таблиц, столбцов в таблице, а также владельцев таблиц (баз данных)

Константы служат для явного

указания величин – чисел, строк, дату и время – в командах SQL

Типы данных. Выражения.



Типы данных служат для представления информации в базах данных.

В SQL определен набор типов данных (char, varchar , integer, smallint...)

Выражения в SQL представляют собой имена, константы, встроенные функции, связанные между собой знаками арифметических операций.

В сложных выражениях для изменения порядка вычислений применяются круглые скобки.

Встроенные функции

в основном предназначены для преобразования типов данных и для обработки строк

Некоторые встроенные функции

Current_date()- возвращает текущую дату

Current_time(точность) - возвращает текущее время

Char_length(строка) – возвращает длину строки

Extract – возвращает значение части day, hour и т.д. даты

Lower(строка) - возвращает строку, преобразованную к нижнему регистру

Upper (строка) - возвращает строку, преобразованную к верхнему регистру

Month(дата) – возвращает значение месяца из указанной даты в виде целого числа

Year(дата) – возвращает значение года из указанной даты в виде целого числа

Вопрос 3 – Операторы языка SQL

Чтение данных.

Оператор Select. Предложение Select



Оператор Select читает данные из базы данных и возвращает их в виде таблицы результата запроса

Предложение Select, с которого начинается оператор Select, содержит элементы данных, которые будут возвращены в виде результирующей таблицы. Указанные в предложении Select элементы данных будут составлять столбцы возвращаемой таблицы. В качестве возвращаемых столбцов могут быть указаны:

- 1) Имя столбца некоторой таблицы базы данных;*
- 2) Константа, которая будет содержаться в соответствующем столбце возвращаемой таблицы*
- 3) Выражение, которое будет вычисляться для каждой строки возвращаемой таблицы, и помещаться в соответствующем столбце этой таблицы*

Предложение FROM



Предложение FROM начинается с ключевого слова FROM, за которым следует в простом случае список спецификаций таблиц, разделенных запятыми. В общем случае за ключевым словом FROM указывается операция соединения (JOIN) исходных таблиц.

Спецификатор таблицы определяет таблицу, из которой запрашиваются исходные данные для формирования возвращаемой таблицы. Спецификатор таблицы представляет собой либо имя исходной таблицы, либо имя исходной таблицы вместе с псевдонимом, указываемым после имени через пробел.

Предложения SQL

1.Выбор всех данных из таблицы XXX

SELECT * FROM XXX

2.Выбор Name, Address, GPA из таблицы XXX

SELECT Name, Address, GPA FROM XXX

3.Выбор Name, Price, Price*1.5 данных из таблицы XXX

SELECT Name, Price, Price*1.5 FROM XXX

4.Выбор всех данных из таблицы XXX
с однократным повторением

SELECT DISTINCT Name FROM XXX

**// Процедура «Показ данных основной
таблицы «oborud_db»».**

begin

SQLQuery1.Close;

***SQLQuery1.SQL.Text:='SELECT *
FROM oborud_db';***

SQLQuery1.Open;

end.

Предложение *WHERE*

Предложение *WHERE* осуществляется отбор нужных строк из таблицы, получаемой в предложении *FROM*. Отбор строк производится в соответствии с условием поиска, которое указывается в предложении за ключевым словом *WHERE*. Условие поиска представляет собой выражение, которое вычисляется для каждой строки возвращаемой таблицы. При вычислении выражения данные соответствующей строки берутся из столбцов, указанных в выражении. Для каждой проверяемой строки условие поиска может иметь одно из трех значений – *TRUE*, *FALSE* и *NULL*. Отбираются только те строки, для которой условие поиска имеют значение *TRUE*.

В SQL существует пять простых условий поиска:

- ✓ Сравнение;
- ✓ Проверка на принадлежность диапазону;
- ✓ Проверка на принадлежность множеству;
- ✓ Проверка на соответствие шаблону;
- ✓ Проверка на равенство значению *NULL*.

Предложение *WHERE*

Сравнение. При сравнении вычисляются и сравниваются значения двух выражений. Существует шесть операций сравнения, которые показаны на следующей синтаксической диаграмме (рис 4)

Результатом сравнения будет значения *TRUE*, если значения выражений сравниваются и *FALSE* – если не сравниваются. В случае, если значение хотя бы одного выражения будет *NULL*, то результат сравнения будет *NULL*.

Проверка на принадлежность диапазону. Данная проверка определяет находится или нет значение проверяемого выражения в диапазоне, указанном за ключевым словом *BETWEEN*, как показано на следующей диаграмме (рис 5)

Проверка на принадлежность диапазону эквивалента следующему логическому выражению

$(A \geq B) \text{ AND } (A \leq C)$, где *A* – проверяемое выражение, а *B* и *C* – соответственно нижнее и верхнее границы диапазона.

Проверка на принадлежность множеству. Данной проверке соответствует следующая синтаксическая диаграмма (рис 6)

Проверка на принадлежность множеству определяет, равняется или нет значение проверяемого выражения одному из констант, указанных за ключевым словом *IN*. Проверка на принадлежность множеству эквивалентна следующему условию поиска сравнение

$(A=C1) \text{ OR } (A=C2) \dots (A=Cn)$, где *A* – проверяемое выражение, а *C_i* – константы.

Выбор всех данных из таблицы XXX по условию <, >, =

```
SELECT * FROM XXX WHERE CENA>1300
```

Выбор всех данных из таблицы XXX по условию похожих на ...

```
SELECT * FROM XXX WHERE NAME LIKE "N_"
```

Выбор всех данных из таблицы XXX по условию похожих на ...

```
SELECT * FROM XXX WHERE MARKA LIKE "N%"
```

Выбор всех данных из таблицы XXX по условию соответствующих интервалу

```
SELECT * FROM XXX WHERE CENA BETWEEN 1300 AND 4000
```

Выбор всех данных из таблицы XXX по условию содержащих сочетания букв

```
SELECT * FROM XXX WHERE MARKA IN ("BO", "MA", "SH")
```

Группировка данных таблицы XXX

```
SELECT NAME, CENA FROM XXX  
GROUP BY NAME
```

Сортировка данных таблицы XXX

```
SELECT NAME FROM XXX ORDER BY  
NAME
```

Выбор всех данных из таблиц XXX и
YYY

```
SELECT NAME FROM XXX, YYY
```

Вставка новой записи в таблицу

**INSERT INTO XXX (POLE1, POLE2, POLE 3)
VALUES (“1954”, “ОКТЯБРЬ”, “12”)**

SQLQuery1.SQL.Text:=‘INSERT INTO XXX (POLE1, POLE2, POLE 3) VALUES (“1954”, “10”, “12”);

Для повышения эффективности использования запроса затем используется оператор **PREPARE**

SQLQuery1.PREPARE

Для фактического выполнения запроса используется затем оператор **EXECSQL**

SQLQuery1.EXECSQL

Модификация записи существующей в таблице XXX

UPDATE XXX

SET MARKA=’ГАЗ-3307’

WHERE NOMER AVTO=’12345’

Удаление записи существующей в таблице XXX

DELETE FROM XXX

WHERE NOMER AVTO=’12345’