

Контрольная работа № 4

По теме № 4: «Информационный поиск и информационно-поисковые системы (ИПС)»

Понятие о языке реляционных баз данных SQL (*Structured Query Language*)

Один из наиболее популярных языков для работы с реляционными базами данных. Создан фирмой IBM в результате многолетних и серьезных исследований.

Мы рассматриваем лишь шесть основных конструкций/«предложений» стандарта языка SQL, а для каждого из предложений лишь наиболее простые варианты использования.

Это:

SELECT - непосредственно запрос к базе данных, обеспечивает выборку данных;

INSERT - вставка новой строки данных в таблицу;

UPDATE - обновление значения данных в существующей строке;

DELETE - удаление строки из таблицы;

CREATE - создание таблицы;

DROP - уничтожение таблицы.

<SQL-предложение> ::= <SELECT-предложение> | <INSERT-предложение> | <UPDATE-предложение> | <DELETE-предложение> | <CREATE-предложение> | <DROP-предложение>

<SELECT-предложение> ::= SELECT <список полей> FROM <имя таблицы> <WHERE-клауза>

<список полей> ::= <имя поля> { , <имя поля> } | *

<имя таблицы> ::= <имя>

<имя поля> ::= <имя>

Получив **SELECT**-предложение, интерпретатор выбирает из таблицы <имя таблицы> перечисленные поля в тех строках, которые удовлетворяют **WHERE**-клаузе.

* - обозначает все поля таблицы.

Результатом выполнения **SELECT**-предложения является новая, в общем случае меньшая по размерам (возможно пустая) таблица, состоящая из отобранных строк и столбцов (полей).

<WHERE-клауза> ::=

WHERE <имя поля типа TEXT> [NOT] LIKE <строка-образец> |

WHERE <выражение> [NOT] IN (<список констант>) |

WHERE <логическое выражение> |

WHERE ALL

WHERE-клауза при выполнении предложений **SELECT**, **UPDATE** и **DELETE** играет роль фильтра: требуемые действия производятся не со всеми строками заданной таблицы, а только с теми из них, поля которых удовлетворяют условиям **WHERE**-клаузы.

Первая альтернатива **WHERE**-клаузы - **LIKE-альтернатива** - позволяет выбрать строки, поля которых (они должны быть текстового типа) соответствуют (не соответствуют в случае с **NOT**) строке-образцу. В строке-образце наряду с обычными символами выделены специальные символы:

% - обозначает любую последовательность из нуля или более символов;

_ - обозначает любой одиночный символ;

[] - обозначает любой одиночный символ из перечисленных в квадратных скобках. Например, [abcdef]. В квадратных скобках можно задавать и диапазон допустимых символов. Например, [a-f].

[^] - обозначает любой одиночный символ, не принадлежащий перечисленным в квадратных скобках. Например, [^abcdef] или [^a-f].

Вторая альтернатива **WHERE**-клаузы - **IN-альтернатива** - содержит выражение текстового или целого типа. Роль переменных в этом выражении играют поля строки таблицы. **IN**-альтернатива позволяет выбрать те строки, для которых выражение принимает (не принимает в случае с **NOT**) одно из значений, перечисленных в списке констант <список констант>. Тип констант из списка должен совпадать с типом выражения, а, следовательно, и с типом имен полей, в него входящих.

Третья альтернатива **WHERE**-клаузы - **BOOL-альтернатива** - содержит выражение логического типа, содержащее, как и в **IN**-альтернативе, имена полей строки в качестве переменных. Эта альтернатива позволяет выбрать строки, для которых <логическое выражение> истинно.

Логические операции **NOT**, **AND**, **OR** имеют обычный смысл.

Четвертая альтернатива **WHERE**-клаузы - **ALL-альтернатива** - говорит о том, что фильтрацию строк проводить не нужно.

Один из основных вариантов операции ПОИСК/ВЫБОРКА:

SELECT <что> - проекция (вертикальная выборка) 2 шаг
FROM <где>
WHERE <условия выборки> - горизонтальная выборка 1 шаг

Результат работы – (реляционная) таблица.

Пример 1.

ПОИСК (SEARCH): найти ЗЧ, заказавших Т2

```
SELECT З, ФИО, АДР, ГОР
FROM ЗЧ
WHERE З IN
      SELECT З
      FROM ЗЗ
      WHERE Т = 'Т2'
```

Результат:

31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В

Пример 2.

ВКЛЮЧЕНИЕ (INSERT): появился новый ЗЧ – 34

```
INSERT INTO ЗЧ '34', 'Козлов', '. . .', 'М'
```

Результат:

ЗЧ

З	ФИО	АДР	ГОР
31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В
34	Козлов	<a4>	М

Пример 3.

УДАЛЕНИЕ (DELETE): 33 отказался от заказа на Т2

```
DELETE З
WHERE З = '33' AND Т = 'Т2'
```

Результат:

ЗЗ

З	Т	К
31	Т1	1
31	Т2	2
31	Т3	4
32	Т1	3
32	Т2	4

Пример 4.

УДАЛЕНИЕ (DELETE): 33 выбыл из числа заказчиков

```
DELETE ЗЧ
WHERE З = '33'
```

Результат:

ЗЧ

З	ФИО	АДР	ГОР
31	Иванов	<a1>	М
32	Петров	<a2>	М

Пример 5.

ОБНОВЛЕНИЕ (UPDATE): изменился адрес 31

```
UPDATE ЗЧ
SET АДР = 'Новая ул. ,2, кв.22'
WHERE З = '31'
```

Результат:

ЗЧ

З	ФИО	АДР	ГОР
31	Иванов	<a11>	М
32	Петров	<a2>	М
33	Серов	<a3>	В

Пример 6.

СОЗДАНИЕ ТАБЛИЦЫ (CREATE) создание таблицы ЗЗ (пока пустой)
с именами столбцов: З (строка длиной 2 байта)
Т (строка длиной 2 байта)
К (целое число)

```
CREATE TABLE ЗЗ (З TEXT(2), Т TEXT(2), К LONG)
```

Результат:

ЗЗ

З	Т	К
---	---	---

Пример 7.

УНИЧТОЖЕНИЕ ТАБЛИЦЫ (DROP) уничтожение таблицы ЗЗ

```
DROP TABLE ЗЗ
```

Результат:

Таблица ЗЗ удаляется из базы данных

Еще несколько примеров операций поиска/выборки:

Пример 8.

```
SELECT З - найти идентификаторы заказчиков, заказавших Т2
FROM ЗЗ
WHERE Т = 'Т2'
```

Результат:

31
32
33

Пример 9.

```
SELECT ФИО, АДР - найти ФИО и адреса заказчиков, живущих в Москве
FROM ЗЧ
WHERE ГОР = 'М'
```

Результат:

Иванов	<a1>
Петров	<a2>

Пример 10.

```
SELECT [UNIQUE] Т - найти неповторяющиеся идентификаторы заказанных товаров
FROM ЗЗ
```

Результат:

Т1
Т2
Т3

Пример 11.

SELECT * - выдать полную (всю) информацию о заказчиках

FROM ЗЧ

Результат:

31	Иванов	<a1>	М
32	Петров	<a2>	М
33	Серов	<a3>	В

Пример 12.

SELECT Т - найти идентификаторы товаров, которые производятся

FROM ТР в Москве и стоят более 2500 рублей

WHERE ГОР = 'М' AND ЦЕНА > 2500

Результат:

Т4

Пример 13.

SELECT НАИМ, МАРКА - найти наименования и марки товаров, которые

FROM ТР производятся в Москве и стоят более 2500 рублей

WHERE ГОР = 'М' AND ЦЕНА > 2500

Результат:

Тв	Руб
----	-----

Пример 14.

SELECT НАИМ, МАРКА - найти наименования и марки товаров, которые

FROM ТР стоят 1500 или 2500 рублей

WHERE ЦЕНА IN (1500, 2500)

Результат:

Хол	Мор
Тв	Рек

Пример 15.

SELECT НАИМ, МАРКА - найти наименования и марки товаров, цена которых

FROM ТР оканчивается на 500 (рублей)

WHERE ЦЕНА LIKE '%-500'

Результат:

Хол	Мор
Тв	Рек
Тв	Руб

Пример 16.

SELECT ФИО, АДР - найти фамилии и адреса заказчиков, фамилии которых

FROM ЗЧ начинаются на 'П', 'Р' или 'С', вторые буквы – любые,

WHERE ФИО LIKE '[ПРС]_[^Т]' третья – не 'Т'.

Результат:

ФИО	АДР
Серов	<a3>