

Занятие 2

QA.

START UP

QA.

Выборка и модификация данных
Ограничение и фильтрация данных
с помощью предикатов

QA.

START UP

Темы занятия

1

Понятие предиката. Логические операторы

A

Понятие NULL значений. Оператор IS [NOT] NULL

B

Операторы BETWEEN, IN, LIKE

C

Объединение выборок (оператор UNION[ALL])

3

Добавление новых данных. Оператор INSERT

4

Модификация данных с помощью оператора UPDATE

5

Удаление данных с помощью оператора DELETE

1. Понятие предиката. Логические операторы

Понятие предиката

Предикат - любое выражение, результатом которого являются значения:

TRUE (истина), **FALSE** (ложь) или **UNKNOWN** (неизвестно)

- Используются с булевыми операторами **AND**, **OR** или **NOT**
- Используются в условиях поиска предложений **WHERE** и **HAVING**
- Исключение: предикаты **NULL** (отсутствие значения), **EXISTS** (существование) не могут принимать значение UNKNOWN

Виды предикатов

- **Предикат сравнения:** =, <>, !=, >, >=, !>, <, <=, !<

Определяет два выражения, соединяемых оператором сравнения

- **BETWEEN**

Определяет диапазон значений для проверки

- **IN**

Определяет, совпадает ли указанное значение с одним из значений, содержащихся во вложенном запросе или списке

- **LIKE**

Определяет, совпадает ли указанная символьная строка с заданным шаблоном

- **IS [NOT] NULL**

Позволяет проверить отсутствие [наличие] значения в полях.
Определяет, может ли указанное выражение быть NULL

BETWEEN

Проверяет, попадают ли значения проверяемого выражения в диапазон, задаваемый граничными выражениями, соединяемыми служебным словом AND

Синтаксис **BETWEEN::=**

```
test_expression [NOT] BETWEEN begin_expression AND  
end_expression
```

Предикат

```
exp1 BETWEEN exp2 AND exp3
```

равносилен предикату

```
exp1 >= exp2 AND exp1 <= exp3
```

Предикат

```
exp1 NOT BETWEEN exp2 AND exp3
```

равносилен предикату

```
NOT (exp1 BETWEEN exp2 AND exp3)
```

IN

Определяет, будет ли значение проверяемого выражения обнаружено в наборе значений:

- диапазон явно определен
- диапазон получен с помощью табличного подзапроса

Синтаксис **IN**::=

```
test_expression [NOT] IN ( subquery | expression [  
                        , ...n ] )
```

LIKE

Определяет, совпадает ли указанная символьная строка с заданным шаблоном

Синтаксис **LIKE**::=

`match_expression` [**NOT**] **LIKE** `pattern` [**ESCAPE** `escape_character`]

- **%** Любая строка длиной от нуля и более символов
- **_** (подчеркивание) Любой одиночный символ
- **[]** Любой одиночный символ, содержащийся в диапазоне ([... -...]) или наборе ([abcdef...])
- **[^]** Любой одиночный символ, не содержащийся в диапазоне ([^...-...]) или наборе ([^abcdef...])

Объединение выборок. UNION и UNION ALL

- Объединяет результаты двух или более запросов в один результирующий набор
- Количество и порядок столбцов должны быть одинаковыми во всех запросах
- Типы данных должны быть совместимыми
- **ALL** объединяет в результирующий набор все строки (вместе с дубликатами)

Синтаксис **UNION [ALL]::=**

```
{<query_specification> | (<query_expression>)}  
  UNION [ALL]  
  <query_specification | (<query_expression>)  
  [UNION [ALL] <query_specification> |  
  (<query_expression>)  
  [ ...n ]]
```

Сортировка набора данных и оператор TOP

Предложение **ORDER BY** сортирует данные, возвращаемые запросом

- Упорядочение результирующего набора запроса
- Определение порядка сортировки

Синтаксис **ORDER BY::=**

```
ORDER BY order_by_expression [COLLATE  
collation_name] [ASC | DESC] [, ...n]
```

Оператор **TOP** ограничивает число строк, возвращаемых в результирующем наборе запроса до заданного числа или процентного значения. Используется совместно с предложением **ORDER BY**

Синтаксис **TOP::=**

```
[TOP (expression) [PERCENT] [WITH TIES]]
```

3. Добавление новых данных. Оператор INSERT

Добавление новых данных. Оператор INSERT

```
INSERT INTO <object> ([<column_name>, ... ])  
VALUES (<data_value>, ...)
```

```
INSERT INTO <object> ([<column_name>, ... ])  
SELECT <select_statement>, ...
```

- Добавляет одну строку в таблицу
- Вставка одной и более строк в таблицу с использованием оператора SELECT

4. Модификация данных с помощью оператора UPDATE

Модификация данных. Оператор UPDATE

```
UPDATE [TOP ( expression )] <object>  
SET <column_name1 = { expression | DEFAULT |  
NULL } [, column_name2 = ...]>  
[WHERE <search_condition>]
```

- Изменяет существующие данные в таблице или представлении
- Модифицирует **все** записи, которые **удовлетворяют условию**

5. Удаление данных с помощью оператора DELETE

Удаление данных. Оператор DELETE

```
DELETE [ TOP ( expression ) [ PERCENT ] ]  
FROM <object>  
[WHERE <search_condition>]
```

- Удаляет существующие данные в таблице или представлении
- Удаляет все записи, которые удовлетворяют условию

СПАСИБО

