

Основы базы данных SQL

Урок 2

Часть 1

2020 г.



Summary of first week

1. CREATE DATABASE
2. CREATE TABLE
3. PRIMARY KEY & FOREIGN KEY
4. AUTO INCREMENT (IDENTITY)
5. ALTER TABLE
6. UPDATE
7. DELETE & DROP
8. BEGIN TRAN & ROLLBACK & COMMIT
9. COMMENT

```
CREATE TABLE Orders (  
    OrderID int NOT NULL PRIMARY KEY,  
    OrderNumber int NOT NULL,  
    PersonID int FOREIGN KEY REFERENCES Persons(PersonID)  
);
```

SELECT

Оператор SELECT используется для выбора данных из базы данных.

Возвращенные данные хранятся в таблице результатов, которая называется набором результатов. Если вы хотите выбрать все поля, доступные в таблице, используйте следующий синтаксис:

```
SELECT * FROM table_name;
```

Если же, вам нужно выгрузить только некоторые поля, нужно использовать следующий синтаксис

```
SELECT column1, column2, ...  
FROM table_name;
```

Здесь *column1*, *column2*, ... являются именами полей таблицы, из которой вы хотите выбрать данные.

SELECT DISTINCT

Оператор **SELECT DISTINCT** используется для возврата только различных (разных) значений.

```
SELECT DISTINCT column1, column2, ...  
FROM table_name;
```

SELECT TOP (LIMIT)

Предложение SELECT TOP используется для указания количества возвращаемых записей (позволяет ограничить выборку числа записей до заданного числа)

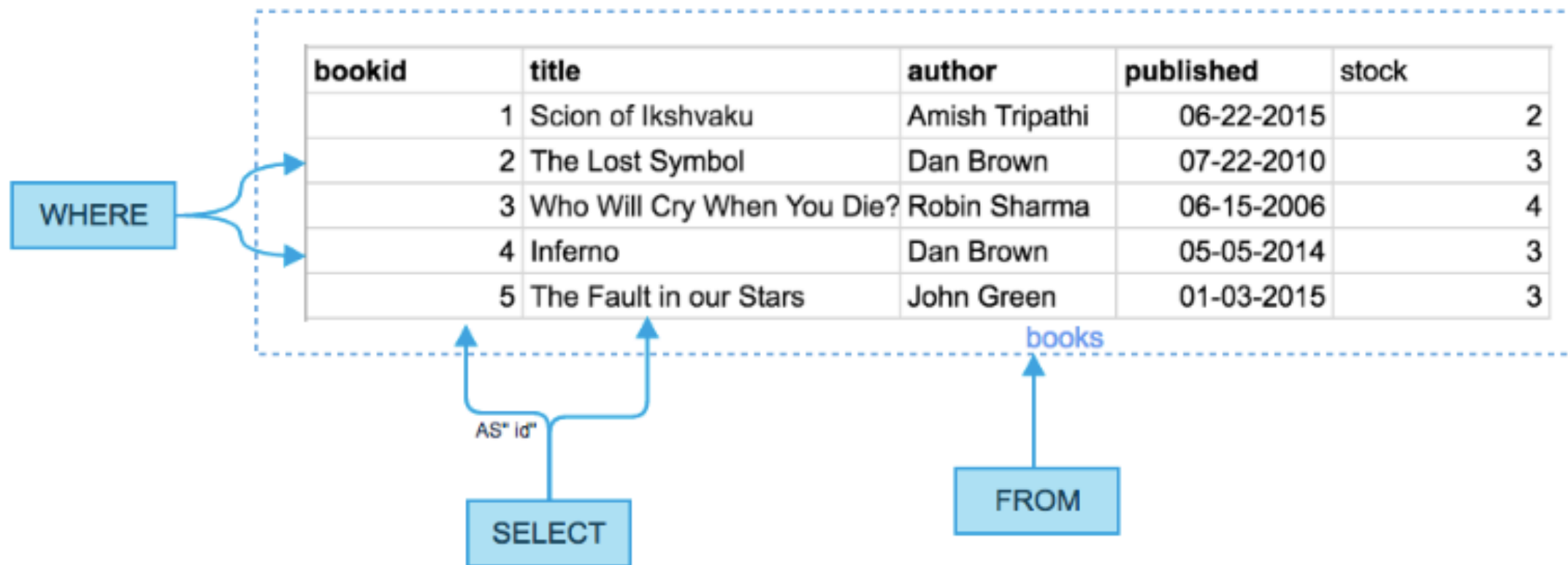
```
SELECT TOP number|percent column_name(s)  
FROM table_name
```

Предложение *SELECT TOP* полезно для больших таблиц с тысячами записей. Возврат большого количества записей может повлиять на производительность.

SQL WHERE

WHERE используется для фильтрации записей (для извлечения только тех записей, которые удовлетворяют указанному условию).

SELECT *column1, column2, ...*
FROM *table_name*
WHERE *condition;*



WHERE используется не только в операторе SELECT, но и в UPDATE, DELETE и т.д.

SQL AND, OR and NOT Operators

WHERE (условие) можно комбинировать с операторами AND, OR и NOT. Операторы AND и OR используются для фильтрации записей на основе более чем одного условия:

- Оператор **AND** отображает запись, если все условия, разделенные символом AND, имеют значение **TRUE**.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 AND condition2 AND condition3 ...;
```

- Оператор **OR** отображает запись, если любое из условий, разделенных OR, является **TRUE**.

```
SELECT column1, column2, ...  
FROM table_name  
WHERE condition1 OR condition2 OR condition3 ...;
```


SQL AND, OR and NOT Operators

first_expression	last_expression	AND	OR
TRUE	TRUE	TRUE	TRUE
TRUE	FALSE	FALSE	TRUE
FALSE	TRUE	FALSE	TRUE
FALSE	FALSE	FALSE	FALSE

- Оператор **NOT** отображает запись, если условие(я) **NOT TRUE**

```
SELECT column1, column2, ...  
FROM table_name  
WHERE NOT condition;
```

Аналогично арифметическим операторам, логические операторы также выполняются в строго определенной последовательности. Если в выражении используется несколько логических операторов, они выполняются в следующей последовательности: сначала *NOT*, затем *AND* и, наконец, *OR*

NULL Values

Поле со значением **NULL** - это поле без значения.

Если поле в таблице является необязательным, можно вставить новую запись или обновить запись без добавления значения в это поле. Затем поле будет сохранено со значением NULL.

Невозможно проверить значения NULL с помощью операторов сравнения =, <, <>.

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NULL;
```

Оператор **IS NOT NULL** используется для проверки не пустых значений (значений NOT NULL).

```
SELECT column_names  
FROM table_name  
WHERE column_name IS NOT NULL;
```

Значение NULL отличается от нулевого значения или поля, содержащего пробелы. Поле со значением NULL - это поле, которое было оставлено пустым во время создания записи!

LIKE Operator

Оператор **LIKE** используется в предложении **WHERE** для поиска указанного шаблона в столбце.

В сочетании с оператором LIKE часто используются два знака:

% - знак процента представляет ноль, один или несколько СИМВОЛОВ.

_ - подчеркивание представляет собой один СИМВОЛ

```
SELECT column1, column2, ...  
FROM table_name  
WHERE columnN LIKE pattern;
```

LIKE Operator	Description
WHERE CustomerName LIKE 'a%'	Finds any values that start with "a"
WHERE CustomerName LIKE '%a'	Finds any values that end with "a"
WHERE CustomerName LIKE '%or%'	Finds any values that have "or" in any position
WHERE CustomerName LIKE '_r%'	Finds any values that have "r" in the second position
WHERE CustomerName LIKE 'a_%'	Finds any values that start with "a" and are at least 2 characters in length
WHERE CustomerName LIKE 'a__%'	Finds any values that start with "a" and are at least 3 characters in length
WHERE ContactName LIKE 'a%o'	Finds any values that start with "a" and ends with "o"

LIKE Operator

Wildcards character (подстановочный знак) используется для замены одного или нескольких символов в строке

Symbol	Description	Example
%	Represents zero or more characters	bl% finds bl, black, blue, and blob
_	Represents a single character	h_t finds hot, hat, and hit
[]	Represents any single character within the brackets	h[oa]t finds hot and hat, but not hit
^	Represents any character not in the brackets	h[^oa]t finds hit, but not hot and hat
-	Represents a range of characters	c[a-b]t finds cat and cbt

```
SELECT * FROM table_name
WHERE column_name LIKE '[abc]%'
```

Questions?

