

Основы SQL. Практическое применение.

Джавадов Бахтияр

Аналитик

Финансовый департамент ВТБ

Структура

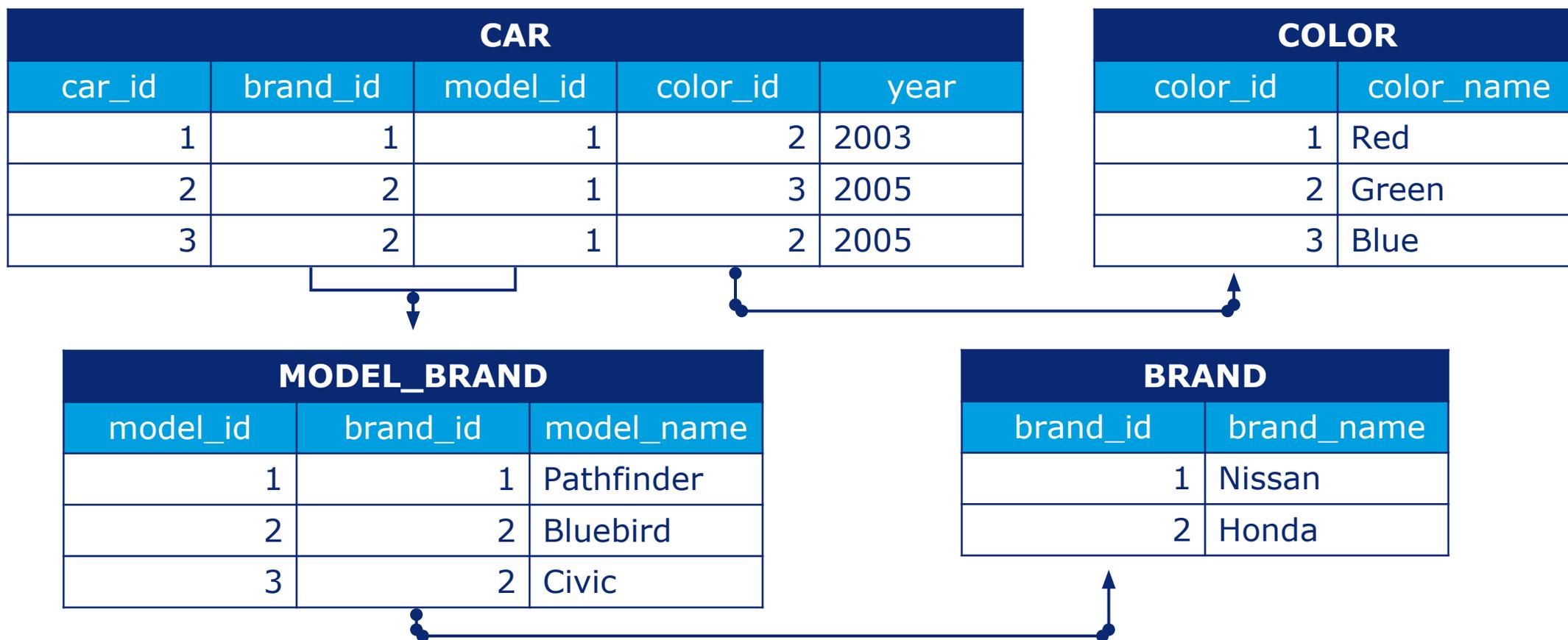
1. Введение в SQL
2. Типы данных
3. Запросы SELECT
4. Запросы с условиями
5. Отбор и исключение NULL
6. Фильтрация и сортировка запросов
7. Простые SELECT запросы

1. Введение в SQL

SQL (Structured Query Language – «Структурированный язык запросов») – это универсальный компьютерный язык, применяемый для создания, модификации и управления данными в реляционных базах данных.

1. Введение в SQL

Пример реляционной базы данных



2. Типы данных

Типы данных SQL делятся на 3 группы:

- Числовые типы данных. Целые (INT, NUMBER) и дробные числа (FLOAT, DECIMAL).
- Строковые (строковые) типы данных. (VARCHAR, TEXT)
- Календарные типы данных. Дата (DATE) и время (TIME). (DATETIME, TIMESTAMP)

3. Запросы SELECT

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>;
```

```
SELECT *  
FROM <table_name>;
```

Запросы SQL не чувствительны к пробелам и табуляциям, а также к строчным или прописным буквам.

3. Запросы **SELECT**

Примеры:

```
SELECT firstname  
FROM customer;
```

```
SELECT firstname, lastname  
FROM customer;
```

```
SELECT *  
FROM customer;
```

4. Запросы с условиями

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <condition1>  
      AND / OR <condition2>  
      AND / OR ... ;
```

Для более сложных логических операций можно добавить к условию операторы **AND** и **OR**.

Оператор **AND** показывает результат при выполнении одновременно нескольких условий.

Оператор **OR** показывает результат при выполнении хотя бы одного из нескольких условий.

4. Запросы с условиями

Операторы для работы с числовыми значениями

Операторы	Определение / условие
=, <>, <, >, <=, >=	Стандартные числовые операторы
BETWEEN ... AND ...	Число входит в числовой диапазон (включая диапазон, указанный в операторе)
NOT BETWEEN ... AND ...	Число не входит в числовой диапазон (включая диапазон, указанный в операторе)
IN (...)	Включение числа в числовой список
NOT IN (...)	Исключение числа из числового списка

4. Запросы с условиями

Примеры:

```
SELECT firstname, lastname, city, country  
FROM customer  
WHERE customerid = 20;
```

```
SELECT *  
FROM track  
WHERE bytes BETWEEN 5000000 AND 6000000  
AND milliseconds >= 300;
```

```
SELECT *  
FROM track  
WHERE albumid IN (40,37,15,222)  
OR bytes >= 11000000;
```

4. Запросы с условиями

Операторы для работы со строковыми значениями

Операторы	Определение / условие
=, <>	Равно и не равно значению
LIKE, %, _	Равно значению (не чувствителен к регистрам)
NOT LIKE, %, _	Не равно значению (не чувствителен к регистрам)
%	Используется для поиска символов в значениях (для любого количества символов)
_	Используется для поиска символов в значениях (для одного символа)
IN (...)	Включение значения в числовой список
NOT IN (...)	Исключение значения из числового списка

4. Запросы с условиями

Примеры:

```
SELECT *  
FROM track  
WHERE composer = 'AC/DC';
```

```
SELECT *  
FROM track  
WHERE composer LIKE '_o_ %'  
AND composer NOT LIKE '%f%';
```

```
SELECT *  
FROM track  
WHERE albumid IN ('Bob Dylan', 'Roy Z')  
AND milliseconds >= 400000;
```

5. Отбор и исключение NULL

Ячейке, в которой запись данных не производилась либо значение данных не присваивалась, дается значение по умолчанию – **NULL**.

Проще: значение **NULL** – пустая ячейка.

Отбор и исключение пустых строк производится следующим образом:

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <column_name1> IS/IS NOT NULL  
      AND / OR <condition2>  
      AND / OR ... ;
```

5. Отбор и исключение NULL

Примеры:

```
SELECT *  
FROM track  
WHERE composer IS NULL  
AND bytes > 100000  
AND bytes < 1000000 ;
```

```
SELECT *  
FROM track  
WHERE composer IS NOT NULL  
AND bytes > 100000  
AND bytes < 1000000 ;
```

6. Фильтрация и сортировка

```
SELECT DISTINCT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <conditions>;
```

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <conditions>  
ORDER BY <column_name1> ASC/DESC, <column_name2> ASC/DESC, ...;
```

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <conditions>  
ORDER BY <column_name1> ASC/DESC, <column_name2> ASC/DESC, ...  
LIMIT <number_of_records_in>;
```

* если доп. условие: { **LIMIT** <number_of_records_not_in>, <number_of_records_in> }
 { **LIMIT** <number_of_records_in> **OFFSET** <number_of_records_not_in> }

6. Фильтрация и сортировка

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <conditions>  
ORDER BY <column_name1> ASC/DESC, <column_name2> ASC/DESC, ...  
LIMIT <number_of_records_in> ;
```

- Оператор **DISTINCT** отбирает уникальные записи.
- Оператор **ORDER BY** сортирует записи.
- Оператор **LIMIT** отбирает указанную часть из запроса. Оператор **OFFSET** используется с оператором **LIMIT**, и исключает указанную часть, сдвигая нашу выборку ниже по таблице.

6. Фильтрация и сортировка

Примеры:

```
SELECT DISTINCT composer
FROM track
WHERE composer LIKE '_o_ %'
AND composer NOT LIKE '%f%';
```

```
SELECT *
FROM track
WHERE composer IS NOT NULL
ORDER BY bytes DESC;
```

```
SELECT firstname, lastname, city, country, phone
FROM customer
WHERE country IN ('USA','Canada')
ORDER BY FirstName, LastName
LIMIT 10 OFFSET 5;
```

7. Простые SELECT запросы

Важно соблюдать порядок прописывания запроса:

```
SELECT <column_name1>, <column_name2>, ...  
FROM <table_name>  
WHERE <conditions>  
ORDER BY <column_name1> ASC/DESC, <column_name2> ASC/DESC, ...  
LIMIT <number_of_records_in> OFFSET <number_of_records_not_in>;
```

В простых запросах **LIMIT** (фильтрация) указывается в конце, и лишь после идет **ORDER BY** (сортировка).

7. Простые SELECT запросы

Задания:

1. Из таблицы клиентов отобразить всех, кто живет во Франции и Германии.
2. Из таблицы клиентов отобразить топ 5 городов в США и в Канаде, отсортировав их по убыванию фамилии клиента.
3. Из таблицы музыкальных произведений отобразить одним запросом те, у которых размер находится от 1,5 мб до 2 мб и от 20 мб до 21 мб.
4. Из таблицы музыкальных произведений отобразить одним запросом те, у которых есть композитор. Длительность которого составляет от 5 до 5,5 минут и размер до 9 мб. Отсортируйте по композитору и наименованию произведения.

СПАСИБО ЗА ВНИМАНИЕ!