

SQL

Создание таблицы

Создание пустой таблицы

```
CREATE TABLE имя_таблицы (column_id INT PRIMARY KEY AUTO_INCREMENT, column VARCHAR(30));
```

На основе другой

```
CREATE TABLE имя_таблицы AS SELECT ...
```

с внешними ключами

- FOREIGN KEY (связанное_поле_зависимой_таблицы)
- REFERENCES главная_таблица (связанное_поле_главной_таблицы)

ON DELETE

- можно установить действия, которые выполняются для записей подчиненной таблицы при удалении связанной строки из главной таблицы
 - CASCADE: автоматически удаляет строки из зависимой таблицы при удалении связанных строк в главной таблице
 - SET NULL: при удалении связанной строки из главной таблицы устанавливает для столбца внешнего ключа значение NULL. (В этом случае столбец внешнего ключа должен поддерживать установку NULL).
 - SET DEFAULT: похоже на SET NULL за тем исключением, что значение внешнего ключа устанавливается не в NULL, а в значение по умолчанию для данного столбца.
 - RESTRICT: отклоняет удаление строк в главной таблице при наличии связанных строк в зависимой таблице.

Типы данных полей

- INT: Целое число, могут принимать значения от -2 147 483 648 до 2 147 483 647
- DECIMAL / NUMERIC: Вещественное число, в скобках указывается максимальная длина числа (исключает символы слева и справа от десятичной запятой) и количество знаков после запятой.
- DATE: Дата в формате ГГГГ-ММ-ДД
- VARCHAR: Строка длиной не более 255 символов, в скобках указывается максимальная длина строки, которая может храниться в поле

Добавление новых строк

Простое добавление

```
INSERT INTO таблица(поле1, поле2) VALUES (значение1, значение2);
```

С помощью запроса

```
INSERT INTO table (columns) SELECT columns FROM table2;
```

Использование вложенных запросов

Обновление таблицы

Одного столбца

```
UPDATE таблица SET поле = выражение
```

Нескольких столбцов

```
UPDATE таблица SET поле1 = выражение1, поле2 = выражение2
```

Нескольких таблиц

```
UPDATE book, supply SET book.amount = book.amount + supply.amount WHERE book.title = supply.title AND book.author = supply.author;
```

порядок выполнения SQL запроса на выборку на СЕРВЕРЕ:

FROM
WHERE
GROUP BY
HAVING
SELECT
ORDER BY

Удаление таблицы

все записи

```
DELETE FROM таблица;
```

удв. условию

```
DELETE FROM таблица WHERE условие;
```

Операторы присоединения

INNER JOIN

```
SELECT ... FROM таблица_1 INNER JOIN таблица_2 ON условие
```

каждая строка одной таблицы сопоставляется с каждой строкой второй таблицы; для полученной «соединённой» строки проверяется условие соединения; если условие истинно, в таблицу результата добавляется соответствующая «соединённая» строка;

LEFT/RIGHT OUTER JOIN

```
SELECT ... FROM таблица_1 LEFT JOIN таблица_2 ON условие
```

в результат включается внутреннее соединение (INNER JOIN) первой и второй таблицы в соответствии с условием; затем в результат добавляются те записи первой таблицы, которые не вошли во внутреннее соединение на шаге 1, для таких записей соответствующие поля второй таблицы заполняются значениями NULL

CROSS JOIN

```
SELECT ... FROM таблица_1 CROSS JOIN таблица_2
```

соединяет две таблицы

```
SELECT ... FROM таблица_1, таблица_2
```

Из нескольких таблиц

```
SELECT FROM first INNER JOIN second ON first.first_id = second.first_id INNER JOIN third ON second.second_id = third.second_id
```

Вложенные запросы в операторах соединения

```
SELECT ( SELECT ... ) имя_вложенного_запроса ON условие
```

USING()

Можно использовать вместо ON

(Structured Query Language, язык структурированных запросов)

Вложенные запросы

Возвращающий одно значение

- Может использоваться в условии отбора записей WHERE как обычное значение совместно с операциями =, <>, >, <=, >=, <, >
- может использоваться в выражениях как обычный операнд, например, к нему можно что-то прибавить, вычесть и пр.

Возвращает несколько значений одного столбца.

- NOT IN: в результирующую таблицу будут включены все записи, для которых выражение со знаком отношения верно хотя бы для одного элемента результирующего запроса
- ANY: в результирующую таблицу будут включены все записи, для которых выражение со знаком отношения верно для всех элементов результирующего запроса
- ALL: в результирующую таблицу будут включены все записи, для которых выражение со знаком отношения верно для всех элементов результирующего запроса

Можно использовать в разделе WHERE совместно с оператором IN.

WHERE имя_столбца IN (вложенный запрос, возвращающий один столбец)

результат выполнения запроса выводится в отдельном столбце результирующей таблицы. При этом результатом запроса может быть только одно значение, тогда оно будет повторяться во всех строках.

после SELECT

Простые запросы

Запрос на вывод всех данных таблицы

```
SELECT * FROM table
```

Запрос на отдельные столбцы

```
SELECT name, column FROM table
```

Выборка новых столбцов и присвоение имен

```
SELECT column AS Название FROM table;
```

Логические функции

```
IF(логическое_выражение, выражение_1, выражение_2)
```

Функция вычисляет логическое выражение, если оно истинно – в поле заносится значение выражения_1, в противном случае – значение выражения_2. Все три параметра IF() являются обязательными.

Запрос с условием

Логическое выражение может включать операторы сравнения (равно «=», не равно «<>», больше «>», меньше «<», больше или равно «>=», меньше или равно «<=») и выражения, допустимые в SQL.

может включать логические операции (И «and», ИЛИ «or», НЕ «not») и круглые скобки, изменяющие приоритеты выполнения операций

SELECT column FROM table WHERE column < 600;

BETWEEN и IN. Приоритет у этих операторов такой же как у операторов сравнения, то есть они выполняются раньше, чем NOT, AND, OR.

Оператор LIKE

позволяет сравнивать строки не на полное совпадение (не совпадение), а в соответствии с шаблоном.

Любая строка, содержащая ноль или более символов

%

(подчеркивание) Любой одиночный символ

_

Запрос с сортировкой

ORDER BY

- ASC (по возрастанию)
- DESC (по убыванию).

Названием столбца

Номером столбца

Именем столбца (указанным после AS)

Столбцы после ключевого слова ORDER BY можно задавать:

Выборка уникальных элементов

DISTINCT

```
SELECT DISTINCT column FROM table;
```

GROUP BY

```
SELECT column FROM table GROUP BY column
```

Размещается после GROUP BY

HAVING

Запросы с групповыми функциями

Групповые функции

- COUNT()
- SUM()
- MIN()
- MAX()
- AVG()

Вычисляемые столбцы

- CEILING(x): возвращает наименьшее целое число, большее или равное x (округляет до целого числа в большую сторону)
- ROUND(x,k): округляет значение x до k знаков после запятой, если k не указано – x округляется до целого
- FLOOR(x): возвращает наибольшее целое число, меньшее или равное x (округляет до целого числа в меньшую сторону)
- POWER(x, y): возведение x в степень y
- SQRT(x): квадратный корень из x
- DEGREES(x): конвертирует значение x из радиан в градусы
- RADIANS(x): конвертирует значение x из градусов в радианы
- ABS(x): модуль числа x
- PI(): pi = 3.1415926...