

Database

Условные конструкции в SQL: операторы CASE и IF



Преподаватель

Портрет

Имя Фамилия

Текущая должность

Количество лет опыта

Какой у Вас опыт - ключевые кейсы

Самые яркие проекты

Дополнительная информация по вашему усмотрению

Корпоративный e-mail

Социальные сети (по желанию)

Важно

- 

Камера должна быть включена на протяжении всего занятия
- 

В течение занятия вопросы задавать в чате или когда преподаватель спрашивает, есть ли у Вас вопросы
- 

Вести себя уважительно и этично по отношению к остальным участникам занятия
- 

Организационные вопросы по обучению решаются с кураторами, а не на тематических занятиях
- 

Во время занятия будут интерактивные задания, будьте готовы включить камеру или демонстрацию экрана по просьбе преподавателя

Повторение



Введение WHERE



Оператор поиска по маске



Операторы множественного выбора



Значение NULL

План занятия

- Оператор CASE
- Алиасы
- Оператор IF
- Другие способы для создания столбцов на основе имеющихся
- Строковые функции
- Конкатенация строк



ОСНОВНОЙ БЛОК





Оператор CASE



Оператор CASE

Это инструмент для выполнения условной логики в SQL.

Оператор CASE



Определение

Оператор CASE работает аналогично оператору IF-THEN-ELSE в языках программирования, позволяя возвращать разные значения на основе заданных условий.

Код

```
CASE
    WHEN условие1 THEN результат1
    WHEN условие2 THEN результат2
    ...
    ELSE результатN
END
```

Оператор CASE

WHEN

Задаёт условие, которое проверяется. Условий может быть несколько

THEN

Определяет результат, который возвращается, если условие выполнено

ELSE

(Необязательно) задаёт значение, которое будет возвращено, если ни одно из условий не выполнено

END

Завершает блок CASE

Оператор CASE



Создает новый столбец в выборке, но не влияет на исходные данные в таблице



Алиасы



Алиасы (псевдонимы)

Это функции, которые позволяют сделать результат запроса более понятным и удобным для использования.

Алиасы



Алиасы задаются с помощью ключевого слова `AS`, но его можно и опустить.



Алиасы полезны при работе с вычисляемыми значениями или с функциями.



Выражение

Это имя столбца, арифметическое выражение, функция или подзапрос.



Имя

Это алиас (псевдоним), который вы присваиваете результату выражения.

Алиас



Определение

- Алиасы задаются с помощью ключевого слова AS, но его можно и опустить.
- Алиасы особенно полезны, когда вы работаете с вычисляемыми значениями или используете функции, и хотите дать результату понятное имя.

Код

```
SELECT
    выражение1 AS имя1,
    выражение2 AS имя2,
    ...
FROM
    таблица;
```



ВОПРОСЫ





ЗАДАНИЕ



Задание 1



Составьте запрос чтобы: классифицировать товары таблицы **products** по их стоимости **standard_cost**, присваивая каждому из них категорию:

- "Дорогой" от **50**
- "Средний" от **20** до **50** включая **50**
- "Дешевый" до **20** включительно

Решение 1

```
SELECT product_name, standard_cost,
CASE WHEN standard_cost > 50 THEN 'Дорогой'
WHEN standard_cost > 20 AND standard_cost <= 50 THEN 'Средний'
ELSE 'Дешевый' END AS price_category FROM products;
```

Задание 2



Подумайте как можно оптимизировать запрос Всегда ли стоит указывать все интервалы. Помните, что SQL как и Питон читает код сверху вниз. Составьте этот запрос.

Решение 2

```
SELECT product_name, standard_cost,
CASE WHEN standard_cost > 50 THEN 'Дорогой'
WHEN standard_cost > 20 THEN 'Средний'
ELSE 'Дешевый' END AS price_category FROM products;
```

Задание 3



Предположим, что вы хотите предоставить разные скидки **discount** клиентам таблица **customers** в зависимости от их региона. Составьте для этого запрос.

- 'WA', 'CA' - 5% скидка
- 'ID', 'OR' - 7% скидка
- 'UT', 'NV' - 13% скидка
- Остальные - без скидки

Решение 3

```
SELECT *,
CASE WHEN state_province IN ('WA', 'CA') THEN 5
WHEN state_province IN ('ID', 'OR') THEN 7
WHEN state_province IN ('UT', 'NV') THEN 13
ELSE 0 END as discount
FROM customers
```

Задание 4

Вы хотите установить статус для заказов таблица **orders** в зависимости от даты отправки **shipped_date** и даты заказа **order_date**. Составьте этот запрос.

- Если нет данных в shipped_date то статус 'Ожидание отправки',
- Если **shipped_date = order_date**, то 'Отправлено в день заказа'.
- В остальных случаях - 'Отправлено'.

Решение 4

```

SELECT
    id,
    order_date,
    shipped_date,
    CASE
        WHEN shipped_date IS NULL THEN 'Ожидание отправки'
        WHEN shipped_date = order_date THEN 'Отправлено в день заказа'
        ELSE 'Отправлено'
    END AS order_status
FROM
    orders;

```



ВОПРОСЫ





Оператор IF



Оператор IF

Это функция, которая является простым способом выполнения условной логики, однако он поддерживается только My SQL.

Оператор IF



Определение

- **condition:** Условие, которое проверяется.
- **true_result:** Возвращаемое значение, если условие истинно.
- **false_result:** Возвращаемое значение, если условие ложно.

Код

```
SELECT IF(condition, true_result,
false_result) AS new_column_name
FROM table_name;
```

Использование оператора CASE вместо IF

Стандарт SQL

Удобство работы с множественными условиями

CASE поддерживается во всех основных СУБД (например, MySQL, SQL Server, PostgreSQL, Oracle).

CASE обеспечивает более читабельный и логичный подход по сравнению с вложенными IF.



ВОПРОСЫ





ЗАДАНИЕ



Задание

Составьте запрос чтобы: вывести имя продукта из таблицы products и его категорию.

- `standard_cost > 20` - 'Expensive'.
- В обратном случае - 'Affordable' с применением оператора IF.

Решение

```
SELECT
    product_name,
    IF(standard_cost > 20, 'Expensive', 'Affordable') AS PriceCategory
FROM products;
```



ВОПРОСЫ





Способы создания столбцов на основе имеющихся

Математические операции



Эти операции могут быть использованы для создания новых столбцов.



Важно применять все математические операции только к числовым столбцам.

Вычислить общую стоимость каждого товара



```
SELECT
    product_id,
    unit_price,
    quantity,
    (unit_price * quantity) AS total_price
FROM order_details;
```




Строковые функции

Это функции, которые позволяют манипулировать строками данных, например, объединять строки, извлекать подстроки, преобразовывать регистр символов и т.д.

Строковые функции



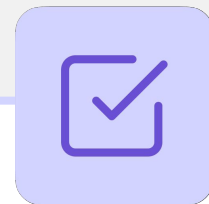
Функция LEFT

Извлекает определенное количество символов с начала строки.

Функция RIGHT

Извлекает определенное количество символов с конца строки.

Выбрать название компании



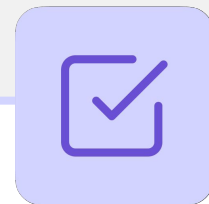
```
SELECT company,  
RIGHT (company, 2) AS ShortName  
FROM customers;
```



Функция SUBSTRING

Это функция, которая извлекает подстроку, начиная с определенной позиции и на определенную длину.

Выбрать три цифры рабочего телефона



```
SELECT  
* , SUBSTRING(business_phone, 6, 3)  
FROM customers;
```



ВОПРОСЫ





Конкатенация строк



Оператор CONCAT

Это оператор, который может объединить несколько строковых столбцов в один.

Задание



Составьте запрос чтобы: выбрать полное имя состоящее из имени и фамилии.

Таблица **employees**.

Решение

```
SELECT
    CONCAT(first_name, ' ', last_name) AS full_name
FROM employees;
```



Функция COALESCE

Это функция, которая возвращает первое ненулевое (NOT NULL) значение из списка аргументов.

Задание

В таблице **employees** замените все пропуски в столбце **notes** на значение **'Not filled'**.

Выведите полученный столбец и столбец **notes**.

Решение

```
SELECT notes, COALESCE(notes, 'Not filled')
FROM employees;
```

Функция	Описание	Пример использования	Результат
CONCAT	Объединяет строки вместе.	<code>SELECT CONCAT(first_name, ' ', last_name) AS full_name FROM employees;</code>	Полное имя, объединенное из имени и фамилии.
SUBSTRING / SUBSTR	Извлекает подстроку из строки, начиная с определенной позиции.	<code>SELECT SUBSTRING(product_name, 1, 3) AS short_name FROM products;</code>	Первые три символа из названия продукта.
LEFT	Возвращает указанное количество символов с начала строки.	<code>SELECT LEFT(product_name, 5) AS short_name FROM products;</code>	Первые пять символов из названия продукта.
RIGHT	Возвращает указанное количество символов с конца строки.	<code>SELECT RIGHT(product_name, 4) AS end_chars FROM products;</code>	Последние четыре символа из названия продукта.

Функция	Описание	Пример использования	Результат
LENGTH	Возвращает длину строки в байтах.	<code>SELECT LENGTH(product_name) AS name_length FROM products;</code>	Длина названия продукта.
TRIM	Удаляет пробелы с начала и конца строки.	<code>SELECT TRIM(' Sample Text ') AS trimmed_text;</code>	"Sample Text" без пробелов в начале и конце.
LTRIM	Удаляет пробелы с начала строки.	<code>SELECT LTRIM(' Sample Text ') AS trimmed_text;</code>	"Sample Text " без пробелов в начале.
RTRIM	Удаляет пробелы с конца строки.	<code>SELECT RTRIM(' Sample Text ') AS trimmed_text;</code>	" Sample Text" без пробелов в конце.
REPLACE	Заменяет все вхождения подстроки на другую подстроку.	<code>SELECT REPLACE(product_name, 'Pen', 'Marker') AS new_name FROM products;</code>	Замена "Pen" на "Marker" в названии продукта.

Функция	Описание	Пример использования	Результат
UPPER	Преобразует все символы строкового выражения в верхний регистр (заглавные буквы).	SELECT UPPER('sql функции');	"SQL ФУНКЦИИ"
LOWER	Преобразует все символы строкового выражения в нижний регистр (строчные буквы).	SELECT LOWER('SQL ФУНКЦИИ');	"sql функции"



ВОПРОСЫ



Заключение

