**ТЕМА 3. СОЕДИНЕНИЯ (JOIN)**

1. **Введение в SQL-джойны**

SQL-джойны позволяют комбинировать данные из разных таблиц. Это полезно, когда есть несколько таблиц с данными, которые нужно объединить, чтобы получить полезную информацию.

1. **Внутреннее соединение (INNER JOIN)**

INNER JOIN объединяет строки из двух таблиц на основе условия, определенного в предложении ON. В результате выбираются только те строки, для которых условие соответствия выполняется.

Пример:

-- Запрос для получения информации о продуктах, их поставщиках и количестве в наличии

SELECT product\_name, suppliers.company\_name, units\_in\_stock

FROM products

INNER JOIN suppliers ON products.supplier\_id = suppliers.supplier\_id

ORDER BY units\_in\_stock DESC;

-- Запрос для получения суммы количества товаров по категориям

SELECT category\_name, SUM(units\_in\_stock)

FROM products

INNER JOIN categories ON products.category\_id = categories.category\_id

GROUP BY category\_name

ORDER BY SUM(units\_in\_stock) DESC

LIMIT 5;

-- Запрос для получения суммы стоимости товаров по категориям с условием

SELECT category\_name, SUM(unit\_price \* units\_in\_stock)

FROM products

INNER JOIN categories ON products.category\_id = categories.category\_id

WHERE discontinued <> 1

GROUP BY category\_name

HAVING SUM(unit\_price \* units\_in\_stock) > 5000

ORDER BY SUM(unit\_price \* units\_in\_stock) DESC;

-- Запрос для получения информации о заказах и сотрудниках, обслуживающих заказы

SELECT order\_id, customer\_id, first\_name, last\_name, title

FROM orders

INNER JOIN employees ON orders.employee\_id = employees.employee\_id;

-- Запрос для получения информации о заказах, продуктах и деталях заказа

SELECT order\_date, product\_name, ship\_country, unit\_price, quantity, discount

FROM orders

INNER JOIN order\_details ON orders.order\_id = order\_details.order\_id

INNER JOIN products ON order\_details.product\_id = products.product\_id;

-- Запрос для получения информации о заказах, клиентах, продуктах и сотрудниках

SELECT contact\_name, company\_name, phone, first\_name, last\_name, title, order\_date, product\_name, ship\_country, products.unit\_price, quantity, discount

FROM orders

JOIN order\_details ON orders.order\_id = order\_details.order\_id

JOIN products ON order\_details.product\_id = products.product\_id

JOIN customers ON orders.customer\_id = customers.customer\_id

JOIN employees ON orders.employee\_id = employees.employee\_id

WHERE ship\_country = 'USA';

1. **Внешние соединения (LEFT, RIGHT JOIN)**

LEFT JOIN и RIGHT JOIN также объединяют две таблицы на основе условия, но они включают в результат все строки из одной из таблицы и соответствующие строки из другой таблицы.

Пример LEFT, RIGHT JOIN:

-- Запрос для получения списка поставщиков и их продуктов (включая тех, у которых нет продуктов) SELECT company\_name, product\_name

FROM suppliers

LEFT JOIN products ON suppliers.supplier\_id = products.supplier\_id;

-- Запрос для получения списка клиентов и их заказов (включая тех, у кого нет заказов)

SELECT company\_name, order\_id

FROM customers

LEFT JOIN orders ON orders.customer\_id = customers.customer\_id

WHERE order\_id IS NULL;

-- Запрос для получения списка сотрудников и их заказов (включая тех, у кого нет заказов) SELECT last\_name, order\_id

FROM employees

LEFT JOIN orders ON orders.employee\_id = employees.employee\_id

WHERE order\_id IS NULL;

-- Запрос для получения списка заказов и клиентов (включая заказы без клиентов)

SELECT company\_name, order\_id

FROM orders

RIGHT JOIN customers ON orders.customer\_id = customers.customer\_id

WHERE order\_id IS NULL;

1. **Рекурсивное соединение (SELF JOIN)**

SELF JOIN используется, когда вам нужно объединить таблицу с самой собой. Это может быть полезно, например, для построения иерархических структур данных, таких как организационные деревья.

Пример:

-- Создание таблицы сотрудников и выполнение SELF JOIN запроса для получения информации о менеджерах CREATE TABLE employee (

employee\_id int PRIMARY KEY,

first\_name varchar(256) NOT NULL,

last\_name varchar(256) NOT NULL,

manager\_id int,

FOREIGN KEY (manager\_id) REFERENCES employee(employee\_id);

);

INSERT INTO employee

(employee\_id, first\_name, last\_name, manager\_id)

VALUES

(1, 'Windy', 'Hays', NULL),

(2, 'Ava', 'Christensen', 1),

(3, 'Hassan', 'Conner', 1),

(4, 'Anna', 'Reeves', 2),

(5, 'Sau', 'Norman', 2),

(6, 'Kelsie', 'Hays', 3),

(7, 'Tory', 'Goff', 3),

(8, 'Salley', 'Lester', 3);

-- Запрос для получения информации о сотрудниках и их менеджерах

SELECT e.first\_name || ' ' || e.last\_name AS employee,

m.first\_name || ' ' || m.last\_name AS manager

FROM employee e

LEFT JOIN employee m ON m.employee\_id = e.manager\_id

ORDER BY manager;

1. **USING & NATURAL JOIN**

USING и NATURAL JOIN - это сокращенные способы объединения таблиц, когда имена столбцов в обеих таблицах совпадают.

Пример USING & NATURAL JOIN:

-- Запрос для получения информации о заказах, клиентах, продуктах и сотрудниках с использованием USING SELECT contact\_name, company\_name, phone, first\_name, last\_name, title, order\_date, product\_name, ship\_country, products.unit\_price, quantity, discount

FROM orders

JOIN order\_details USING(order\_id)

JOIN products ON

USING(product\_id)

JOIN customers ON USING(customer\_id)

JOIN employees ON USING(employee\_id)

WHERE ship\_country = 'USA';

-- Запрос для получения информации о заказах и сотрудниках с использованием NATURAL JOIN SELECT order\_id, customer\_id, first\_name, last\_name, title

FROM orders

NATURAL JOIN employees;

1. **Псевдонимы с помощью AS**

**AS** используется для создания псевдонимов для столбцов и таблиц в запросе. Псевдонимы позволяют дать более понятные имена столбцам или временно переименовать таблицы.

Пример:

-- Запрос для получения количества сотрудников с использованием псевдонима

SELECT COUNT(\*) AS employees\_count

FROM employees;

-- Запрос для получения количества уникальных стран с использованием псевдонима

SELECT COUNT(DISTINCT country) AS country

FROM employees;

-- Запрос для получения суммы количества товаров по категориям с использованием псевдонима SELECT category\_id, SUM(units\_in\_stock) AS units\_in\_stock

FROM products

GROUP BY category\_id

ORDER BY units\_in\_stock DESC

LIMIT 5;

-- Запрос для получения суммы стоимости товаров по категориям с использованием псевдонима SELECT category\_id, SUM(unit\_price \* units\_in\_stock) AS total\_price

FROM products

WHERE discontinued <> 1

GROUP BY category\_id

HAVING SUM(unit\_price \* units\_in\_stock) > 5000

ORDER BY total\_price DESC;