<!--

Л.р. №1. Создание и заполнение отношений

1. Отношение "Поставщики" (Suppliers)

Таблица для хранения информации о поставщиках, включая идентификатор, название и телефон.

sql

CREATE TABLE Suppliers (

supplier\_id CHAR(8) PRIMARY KEY, -- Идентификатор поставщика

supplier\_name VARCHAR(30) NOT NULL, -- Название поставщика

phone\_number VARCHAR(15) -- Телефон поставщика

);

2. Отношение "Поставки" (Deliveries)

Таблица для хранения информации о поставках, включая шифр поставки, название товара, индекс

поставщика (внешний ключ), единицу измерения, вес единицы товара, количество товара, номер склада,

номер линии и дату поставки.

sql

CREATE TABLE Deliveries (

delivery\_code CHAR(6) PRIMARY KEY, -- Шифр поставки (первичный ключ)

product\_name VARCHAR(30), -- Название товара

supplier\_id CHAR(8), -- Индекс поставщика (внешний ключ)

unit\_of\_measure VARCHAR(3) DEFAULT 'шт', -- Единица измерения (по умолчанию 'шт')

unit\_weight DECIMAL(7, 2), -- Вес единицы товара

quantity DECIMAL(7, 2) NOT NULL, -- Количество товара

warehouse\_number INT, -- Номер склада

line\_number CHAR(1), -- Номер линии

delivery\_date DATE, -- Дата поставки

FOREIGN KEY (supplier\_id) REFERENCES Suppliers(supplier\_id) -- Внешний ключ на поставщика

);

3. Отношение "Заказы" (Orders)

Таблица для хранения информации о заказах, включая идентификатор заказа, индекс товара (внешний ключ),

единицу измерения и количество товара.

sql

CREATE TABLE Orders (

order\_id INT PRIMARY KEY, -- Идентификатор заказа

product\_code CHAR(6), -- Индекс товара (внешний ключ)

unit\_of\_measure VARCHAR(3), -- Единица измерения

quantity DECIMAL(7, 2), -- Количество товара

FOREIGN KEY (product\_code) REFERENCES Deliveries(delivery\_code) -- Внешний ключ на поставку

);

Л.р. №2. Выборка данных

1. Создать упорядоченный список товаров с наименованиями и телефонами поставщиков

Для того чтобы создать упорядоченный список товаров с наименованиями и телефонами поставщиков, нужно

объединить таблицы Deliveries и Suppliers:

sql

SELECT d.product\_name, s.supplier\_name, s.phone\_number

FROM Deliveries d

JOIN Suppliers s ON d.supplier\_id = s.supplier\_id

ORDER BY d.product\_name;

Этот запрос возвращает список товаров, наименования поставщиков и их телефоны, отсортированный по

наименованию товара.

2. Для конкретного заказа выдать список местоположения заказанных товаров на складах (с учётом

количества товара в заказе и на складе)

Для того чтобы получить информацию о местоположении товаров, нужно объединить таблицы Orders и

Deliveries, а также учитывать количество товара в заказе и на складе:

sql

SELECT o.order\_id, d.product\_name, d.warehouse\_number, d.quantity AS warehouse\_quantity,

o.quantity AS order\_quantity,

(d.quantity - o.quantity) AS remaining\_quantity

FROM Orders o

JOIN Deliveries d ON o.product\_code = d.delivery\_code

WHERE o.order\_id = ? -- Здесь нужно подставить конкретный ID заказа

ORDER BY d.warehouse\_number;

Этот запрос возвращает местоположение заказанных товаров на складе, включая количество товара на

складе, количество в заказе и остаток товара, если таковой есть.

3. Посчитать общий вес конкретного заказа

Для того чтобы посчитать общий вес заказа, нужно умножить количество товара в заказе на вес единицы

товара для каждой поставки, а затем суммировать результаты:

sql

SELECT o.order\_id, SUM(o.quantity \* d.unit\_weight) AS total\_weight

FROM Orders o

JOIN Deliveries d ON o.product\_code = d.delivery\_code

WHERE o.order\_id = ? -- Здесь нужно подставить конкретный ID заказа

GROUP BY o.order\_id;

Этот запрос возвращает общий вес конкретного заказа, учитывая количество каждого товара и вес

единицы товара.-->