МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ**

**«СЕВАСТОПОЛЬСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»**

Институт информационных технологий и управления в технических системах

Кафедра «Информационные системы»

**ОТЧЁТ**

по лабораторной работе №5

по дисциплине

«Теория баз данных»

на тему: «Язык SQL. Коррелированные вложенные подзапросы»

Выполнил: обучающийся

группы ИС/б-18-2-о

Пузырёв Д. В.

Проверил:

Гончаренко Д. Г.

Севастополь

2020

**1. ЦЕЛЬ РАБОТЫ**

Ознакомиться с принципом работы коррелированных подзапросов.

**2. ХОД РАБОТЫ**

**2.1. Соотнесение таблицы со своей копией**

Пусть необходимо найти все заказы со значениями количества деталей выше среднего значения для каждого поставщика (рис. 2.1).

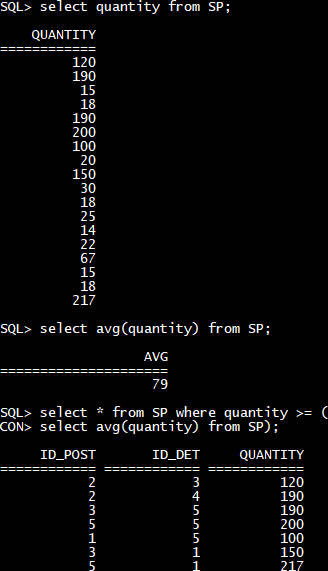


Рисунок 2.1 – Соотнесение таблицы со своей копией

**2.2. Коррелированный запрос с использованием двух таблиц**

Пусть требуется найти все детали со значением поля «city» таблицы с поставщиками равным “Kiev” (рис. 2.2).

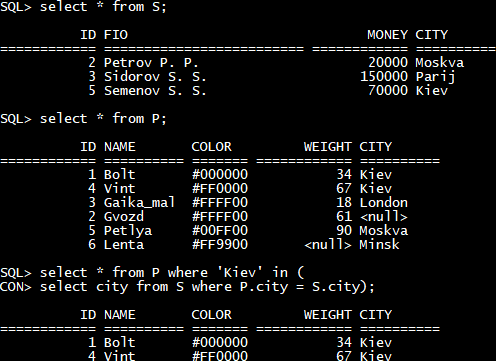


Рисунок 2.2 – Использование двух таблиц

**2.3. Специальные операторы SQL**

Допустим, необходимо узнать номера всех деталей, поставляемых более чем одним поставщиком. Для каждой строки-кандидата из внешнего запроса внутренний запрос находит строки, у которых совпадают значения полей «id\_det», но не совпадают значения полей «id\_post» (рис. 2.3).

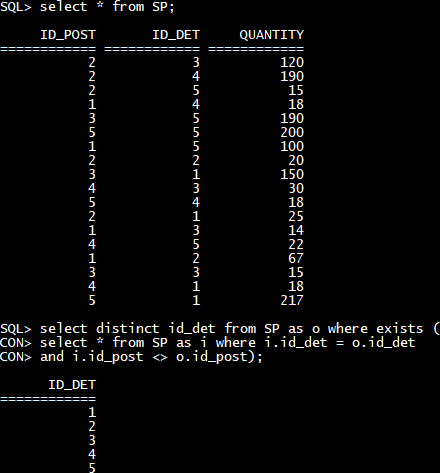


Рисунок 2.3 – Оператора «exists»

Если необходимо найти все детали, у которых город равен “Kiev”, то запрос будет выглядеть следующим образом (рис. 2.4).

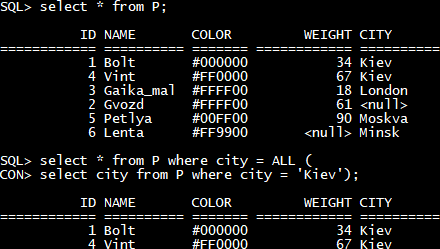


Рисунок 2.4 – Оператор «all»

Если нужно найти всех поставщиков, которые живут в любых городах, кроме Москвы, то запрос будет выглядеть следующим образом (рис. 2.5).

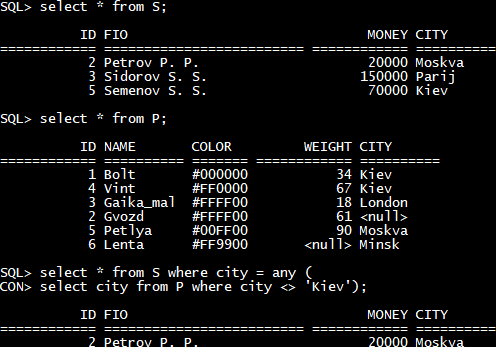


Рисунок 2.5 – Оператор «any»

**3. КОНТРОЛЬ**

**3.1. Коррелированные вложенные подзапросы**

Коррелированный подзапрос использует псевдоним таблицы, определённый не во вложенном запросе, а во внешнем. При этом, вложенный запрос выполняется много раз – для каждой строки внешней таблицы.

**3.2. Назначение псевдонима. Пример**

Пусть необходимо вывести имена и номера всех продавцов, которые имеют более одного заказчика:

SELECT snum, sname

FROM Salespeople main

WHERE 1 <

(SELECT COUNT (\*)

FROM Customers

WHERE snum = main.snum);

В случае, если перед именем поля не указывается имя псевдонима, то данное поле относится к текущему подзапросу. Таким образом, поле «snum» во вложенном подзапросе относится к таблице «Customers», а не к «Salespeople». В случае, если такого поля в соответствующей таблице нет, оно ищется в подзапросе верхнего уровня. Так, если бы поле «snum» отсутствовало в таблице «Customers», оно относилось бы к «Salespeople».

**3.3. Назначение и правила использования операторов «exists», «all», «any»**

Специальные операторы SQL имеют смысл только для подзапросов. Они принимают подзапрос как аргумент – точно так же, как это делает «in».

«Exists(X)» – булевский оператор. Он получит значение «истина», если запрос X вернёт хоть одну строку.

Оператор «any» становится верным, если значение из верхнего подзапроса совпадает, по крайней мере, с одним значением из вложенного подзапроса. «Any» используется в сочетании со знаками равенства.

Оператор «all» становится верным, если все значения из вложенного подзапроса равны значению-кандидату из внешнего запроса. «All» используется в сочетании со знаками равенства.

**3.4. Альтернатива операторам «exists», «all», «any»**

Операторы «any» и «all» можно выразить через «exists» в коррелированном подзапросе, в явном виде они нужны лишь для упрощения записи запроса. Обратное утверждение не верно.

Действие оператора «all» противоположно оператору «any».

**ВЫВОД**

Ознакомлены с принципом работы коррелированных подзапросов.