**Практическое занятие № 8**

**Язык SQL. Соединение таблиц**

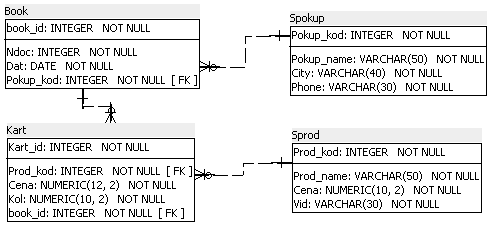
**Цель занятия**. Научиться выполнять соединение таблиц, освоить различные варианты соединения: естественное (внутреннее), неестественное, внешнее.

В реляционной БД взаимосвязанная информация в ходе нормализации распределяется по отдельным таблицам. Соединение позволяет объединить данные связанных таблиц.

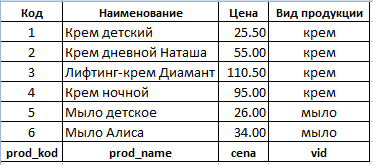
Различают следующие виды соединения:

* естественное;
* неестественное;
* внешнее соединение

Виды соединений рассмотрим на примере БД Продажи (рис.1), с заполненными таблицами (рис.2-5).



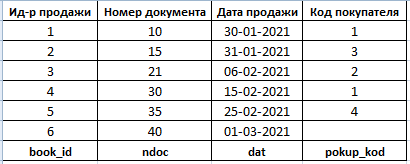
**Рис. 1 База данных Продажи**



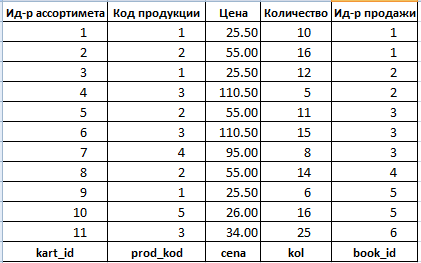
**Рис. 2 Данные для таблицы sprod (Продукция)**



**Рис. 3 Данные для таблицы spokup (Покупатели)**

****

**Рис. 4 Данные для таблицы book (Книга продаж)**



**Рис. 5 Данные для таблицы kart (Ассортимент продаж)**

Для иллюстрации всех видов соединений в БД были внесены изменения. В таблице book (Книга продаж) изменена колонка Код покупателя (разрешено содержать значения null). Это изменение можно внести оператором

ALTER TABLE book

alter pokup\_kod DROP NOT NULL

**1) Естественное соединение (внутреннее)**

Естественное соединение использует существующие в БД связи таблиц Primary Key(PK) - Foreign Key (FK).

***Пример 1.*** Выберем все продажи (номер документа, дату продажи) с указанием наименования покупателя. Для получения указанных данных необходимо выполнить соединение двух таблиц: *Книги продаж* (*book*) и *Покупателей* (spokup), которые имеют связь PK (spokup.pokup\_kod) – FK(book.pokup\_kod).

Составим оператор запроса и включим в раздел WHERE предикат сравнения:

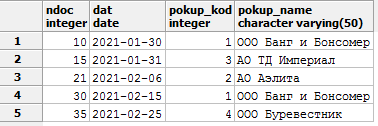
SELECT ndoc, dat, b.pokup\_kod, pokup\_name

FROM spokup sp, book b

WHERE sp.pokup\_kod=b.pokup\_kod

*Алгоритм выполнения:* После прямого произведения таблиц в разделе FROM из всех возможных комбинаций строк таблиц предикатом сравнения раздела WHERE отбираются те, которые совпадают по колонке *Код покупателя* (pokup\_kod). Предикат сравнения задает условие естественного соединения таблиц.

Результат запроса



Предикат сравнения в разделе WHERE можно заменить оператором JOIN в разделе FROM:

*Таблица1 JOIN Таблица2 ON Таблица1.FK=Таблица2.PK*

Часто указывают полный вариант записи INNER JOIN, которое дословно переводится как внутреннее соединение, в отличие от внешних соединений, рассматриваемых далее.

Изменим оператор запроса примера 1:

SELECT ndoc, dat, b.pokup\_kod, pokup\_name

FROM spokup sp JOIN book b ON sp.pokup\_kod=b.pokup\_kod

Условие соединения таблиц может сочетаться с другими условиями.

***Пример 2.*** Выбрать все продажи продукции с кодом 1: покупатель, дата продажи, количество, цена, стоимость.

В данном примере нам необходимо соединить уже 3 таблицы: Покупателя (spokup) с Книгой продаж (book) по колонке Код покупателя (pokup\_kod) и полученные строки соединить с Ассортиментом продажи (kart) по колонке Ид-р продажи (book\_id). И предикатом сравнения отобрать строки относящиеся к заданной продукции.

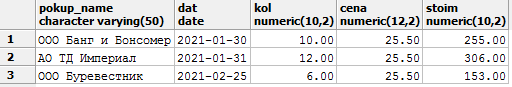
SELECT pokup\_name, dat, kol, cena, cast (kol\*cena as Numeric(10,2)) stoim

FROM spokup sp JOIN book b ON sp.pokup\_kod=b.pokup\_kod

JOIN kart k ON b.book\_id=k.book\_id

WHERE prod\_kod=1

Результат запроса:



***Упражнение 1****.* Выбрать продажи за январь 2021 г.: покупатель, дата продажи, код продукции, наименование продукции, цена, количество

**2) Неестественное соединение**

Этот вид соединения основывается на данных, а не на структуре БД (связях таблиц).

***Пример 3***. Отобрать все пары покупателей таким образом, чтобы пару составляли покупатели из одного города.

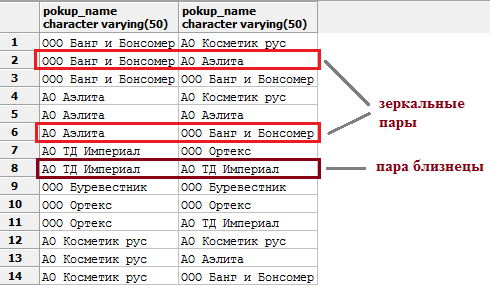
Так как надо образовать пары, то необходимо соединить таблицу *Покупатель* с ее копией. Далее из образованных пар покупателей выбрать те, где совпадает город.

SELECT p1.pokup\_name, p2.pokup\_name

FROM spokup p1, spokup p2

WHERE p1.city=p2.city

Результат запроса



Виден недостаток данного запроса: в результат попали пары близнецы (А,А) и зеркальные пары (А,В) (В,А). Эти пары необходимо исключить. Для этого используем еще один предикат, чтобы код покупателя из первой копии был больше кода покупателя из второй (можно выбрать обратное условие).

SELECT p1.city, p1.pokup\_name, p2.pokup\_name

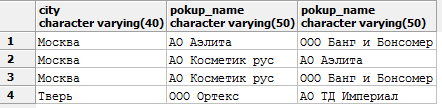
FROM spokup p1, spokup p2

WHERE p1.city=p2.city

and p1.pokup\_kod>p2.pokup\_kod

ORDER BY p1.city

Результат запроса:



При этом необходимо использовать ***алиасы*** (***псевдонимы***), которые позволяют различать соединяемые копии таблиц

**3) Внешнее соединение.**

Естественное или внутреннее соединение оставляло строки, совпадающие по соединяемым колонкам. В примере 1 покупатели, которые не получали продукцию не попали в результат запроса. Покупатель с кодом 5 не попали в результат выборки. Также в результат не попадут продажи, в которых по каким-то причинам не задан код покупателя (продажа с идентификатором 6).

Более полную картину дают внешние соединения. Внешнее соединение таблиц А и В – может быть:

* левое
* правое
* полное

*Левое* соединение

А LEFT OUTER JOIN В ON А.PK=B.FK

включает все строки левой таблицы А независимо от того, есть ли связанные с ней строки в таблице В.

*Правое* соединение

А RIGHT OUTER JOIN В ON А.PK=B.FK

включает все строки правой таблицы В независимо от того, есть ли связанные с ней строки в таблице А.

*Полное* соединение

А FULL OUTER JOIN В ON А.PK=B.FK

объединяет левое и правое соединения.

Все виды соединений можно представить в виде наглядных интерпретаций (рис. 6).

|  |  |  |  |
| --- | --- | --- | --- |
| ***Внутреннее*** | ***Левое*** | ***Правое*** | ***Полное*** |
|  |  |  |  |

**Рис.6 Наглядные интерпретации всех видов соединений**

***Пример 4.*** Вывести для всех покупателей вывести полную информацию об их покупках

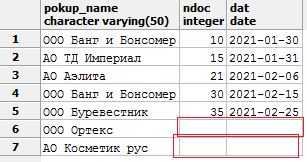
SELECT sp.pokup\_name, ndoc, dat

FROM spokup sp LEFT OUTER JOIN book b

ON b.pokup\_kod=sp. pokup\_kod

Будут отобраны все покупатели, независимо от того приобретали они продукцию или нет. Если покупатель не покупал продукцию, то колонки ndoc и dat будут содержать null; а если покупал, то число его вхождений в результат выборки равно числу его покупок.

Результат запроса:



***Пример 5*.** Вывести информацию о всех продажах с указанием наименования покупателя.

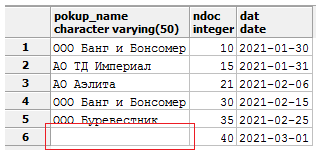
SELECT sp.pokup\_name, ndoc, dat

FROM spokup sp RIGHT OUTER JOIN book b

ON b.pokup\_kod=sp. pokup\_kod

Будут отобраны все строки книги продаж, независимо от того указан ли в продаже покупатель. Если покупатель не указан, то колонка pokup\_name будет содержать null.

Результат запроса:



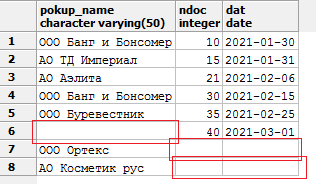
***Пример 6*.** Вывести информацию о всех продажах и покупателях

SELECT sp.pokup\_name, ndoc, dat

FROM spokup sp FULL OUTER JOIN book b

ON b.pokup\_kod=sp. pokup\_kod

Результат объединяет данные предыдущих двух запросов:



***Упражнение 2.***

В БД имеются две таблицы

Продукция (SPROD) Книга продаж (BOOK)

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Код** | **Наименов.** | **Группа** |  | **Ид-р книги** | **Покупатель** | **Код прод.(FK)** | **Кол-во** |
| 010 | Samsung 32″ | Телевизоры |  | 1 | АО Смена | 010 | 10 |
| 011 | LG DF4 | СВЧ печь |  | 2 | ООО Темп | 011 | 12 |
| 012 | Bork A3 | СВЧ печь |  | 3 | АО Смена | 011 | 25 |
| 013 | LG 41″ | Телевизоры |  | 4 | АО Смена | null | 16 |
| 014 | Panasonic 26″ | Телевизоры |  | 5 | Null | 012 | 8 |
|  |  |  |  | 6 | ПАО Крок | 013 | 18 |
| **Prod\_kod** | **Prod\_name** | **Grup** |  | **Book\_Id** | **Pokup** | **Prod\_kod** | **Kol** |

Cоставить операторы запроса для выполнения различных видов соединений (JOIN):

- внутреннего;

- внешнего (левого);

- внешнего (правого);

- внешнего (полного);

и привести результаты их выполнения.

Для выполнения задания создайте временные таблицы (временные таблицы существуют только в текущем сеансе запроса)

WITH sprod(prod\_kod,prod\_name,grup) as

(select '010', 'Samsung 32', 'Телевизоры'

union

select '011', 'LG DF4', 'СВЧ печь'

union

select '012', 'Bork A3', 'СВЧ печь'

union

select '013', 'LG 41', 'Телевизоры'

union

select '014', 'Panasonic 26', 'Телевизоры'),

book (book\_id,pokup,prod\_kod,kol) as

(select 1, 'АО Смена', '010',10

union

select 2, 'ООО Темп', '011',12

union

select 3, 'АО Смена', '011',25

union

select 4, 'АО Смена', null,16

union

select 5, null, '012',8

union

select 6, 'ПАО Крок', '013',18)

SELECT … текст оператора запроса

**Задание**

1. Выполните упражнения
2. Составьте похожие запросы для вашей БД