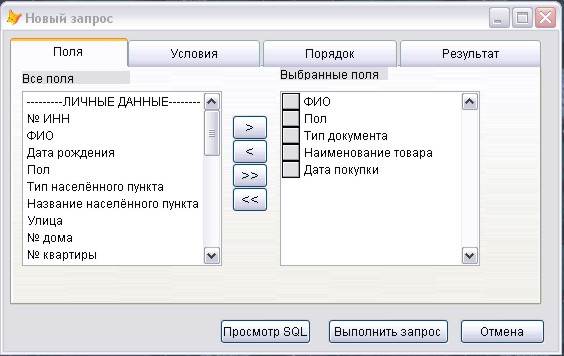
## Задание №9. Нестандартный запрос

### Как должен работать нестандартный запрос

Нестандартный запрос предназначен для генерации «на лету» заранее не описанных запросов к БД на основе прямого использования метаданных о схеме БД либо из сформированных вручную вспомогательных таблиц, описывающих всю или часть схемы БД. Интерфейс нестандартного запроса должен позволять конструировать новые запросы по принципу «не набирать, а выбирать».

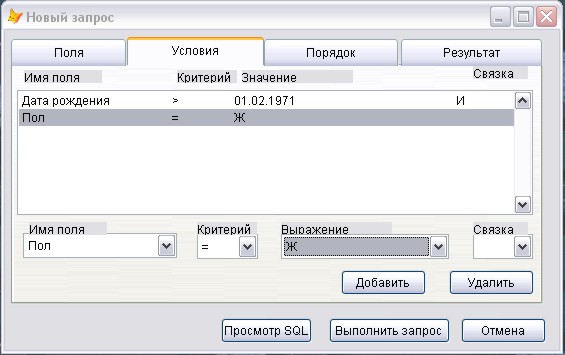
Примерный вид графического пользовательского интерфейса:



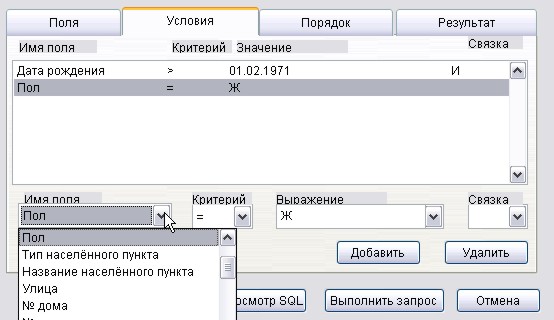
На вкладке «Поля» ­– все поля из всех таблиц из последнего задания предыдущей контрольной точки (Задание №8. «Автоматизированное рабочее место»). Названия полей можно сгруппировать по смыслу, отделив друг от друга специальными символами, как показано на рисунке. На данной вкладке пользователь выбирает поля для вывода результатов запроса (заголовки полей в результирующей таблице, т. е. в тексте сгенерированного SQL запроса данный список полей должен непосредственно следовать за ключевым словом SELECT). Например, для выбранных полей на рисунке выше:

SELECT people\_info.fio, people\_info.pol, sogl\_info.doc\_type, goods.name, pep\_good.sale\_date

Далее, пользователь должен иметь возможность наложить определённые условия на результат запроса:



В поле «Имя поля» должны попадать **все поля из всех таблиц**, а не только те, которые были выбраны на предыдущей вкладке.



В поле «Критерий» - операции сравнения, доступные для типа данных выбранного поля.

В поле «Выражение» должны подчитываться значения выбранного поля «Имя поля» из соответствующей таблицы БД. Условия могут быть связаны через И, ИЛИ. Созданные условия добавляются в раздел запроса WHERE. Например, здесь это будет так:

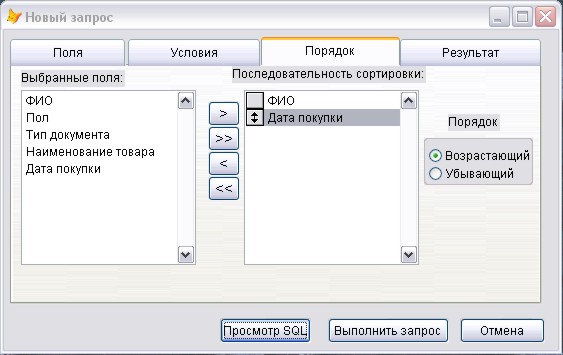
SELECT people\_info.fio, people\_info.pol, sogl\_info.doc\_type, goods.name, pep\_good.sale\_date

FROM people\_info,solg\_info, goods, pep\_good

WHERE peoples\_info.birth\_date = {01.02.1971} AND peoples\_info.pol = ‘Ж’

Фигурные скобки {} или апострофы ‘’ ставятся в зависимости от типа поля (строковое, дата и т. п.)

Далее, на вкладке «порядок» - возможность задать сортировку по выбранным полям. Данную вкладку реализовывать необязательно.



Результат выбора попадает в раздел запроса ORDER BY. Например,

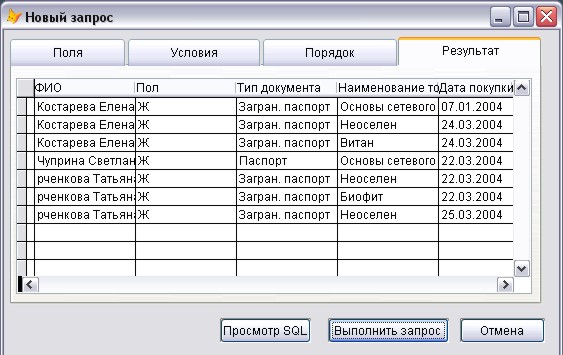
SELECT people\_info.fio, people\_info.pol, sogl\_info.doc\_type, goods.name, pep\_good.sale\_date

FROM people\_info,solg\_info, goods, pep\_good

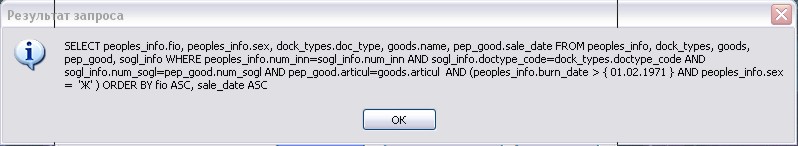
WHERE peoples\_info.birth\_date = {01.02.1971} AND peoples\_info.pol = ‘Ж’

ORDER BY peoples\_info.fio ASC, pep\_good.sale\_date ASC

На вкладке «Результат» - результат выполнения запроса

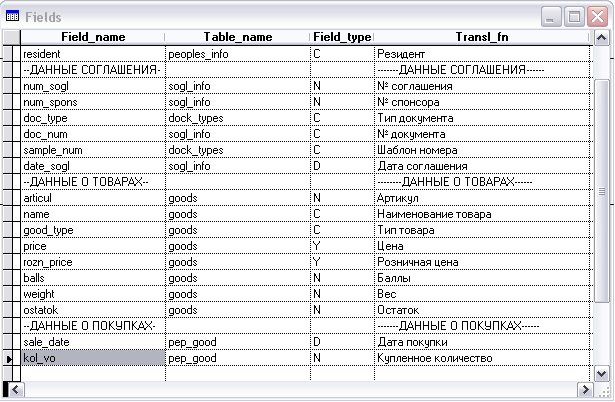


Кроме того, должна быть возможность посмотреть текст созданного SQL-запроса по нажатию на кнопку «Просмотр SQL»



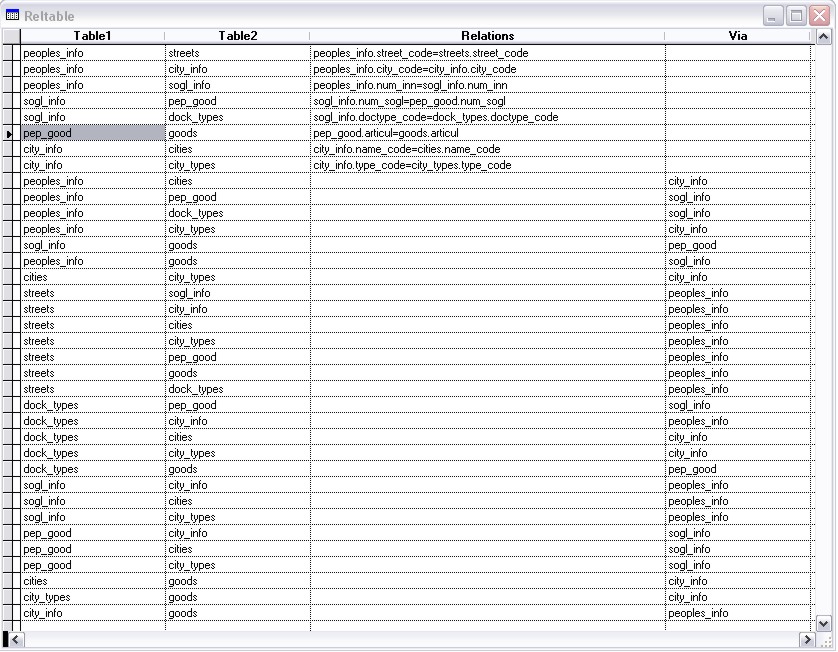
Как видно из текста запроса, в разделе WHERE кроме условий, заданных пользователем, содержатся ещё и условия соединения таблиц по ключевым полям. Это необходимо для корректного выполнения запроса. Рассмотрим, что необходимо для того, чтобы правильно построить эти условия соединения.

# Вспомогательная таблица с метаданными о таблицах БД



Используется на первой вкладке для вывода всех полей всех таблиц БД, а также для учёта типа поля при формировании условия на вкладке «Условия». Таблица содержит два поля с названиями перечисленных полей. Одно используется при формировании части SELECT, а другое, переведённое на русский, для вывода на форму для наглядного описания имён полей.

# Вспомогательная таблица с метаданными о связях между таблицами БД



Данная таблица используется для формирования строки соединения между любыми двумя таблицами БД. Как видно на рисунке, сначала здесь прописываются соединения между соседними таблицами (по каким полям соединены соседние таблицы).

А затем, прописываем соединения каждой таблицы с каждой, которые непосредственно не соединены, но между которыми существует путь. Для этого в поле “Via” записываем имя таблицы, через которую они соединены. Например, пусть у нас есть таблицы 1, 2, 3, 4.

Соединены они так: 1<-2<-3<-4

Тогда сначала записываем непосредственные связи таблиц:

1 2 1.code = 2.code\_1

2 3 2.code = 3.code\_2

3 4 3.code = 4.code\_3

Затем прописываем остальные связи:

1 3 соединены через 2

1 4 соединены через 2

2 4 соединены через 3

Допустим, пользователь выбрал в запрос поля из таблиц 1 и 4. Нам нужно соединить эти две таблицы. Для этого сначала ищем в нашей таблице строчку 1 4. Они соединены через 2. Поэтому ищем, как соединены 1 и 2, 2 и 4. 1 и 2 соединены через 1.code = 2.code\_1. Получили первое условие соединения.

Далее, ищем 2 и 4. Они соединены через 3. Ищем 2 и 3, 3 и 4. Получаем ещё условия

2.code = 3.code\_2 и 3.code = 4.code\_3

Соединяем все условия через AND:

WHERE 1.code = 2.code\_1 AND

2.code = 3.code\_2 AND

3.code = 4.code\_3

***Замечание:*** важно уметь демонстрировать способность программы нестандартного запроса автоматически учитывать изменения во вспомогательных таблицах с метаданными.