

SQL – Structured Query Language (Структурированный Язык Запросов).
Обеспечивает описание
структур данных,
ограничений целостности и
требований к обновляемым/выбираемым данным.



8.7. Оператор SELECT

8.7.1. Общие сведения

Обеспечивает выборку данных.

Является «гибридом» РА и РИ с переменными-кортежами.

Может использоваться

- как самостоятельная часть приложения;
- как основа определения представления (объекта SQL-системы);
- как часть другого оператора SQL (подзапрос);
- как часть табличного выражения,
(т.е., операнд операций JOIN, UNION, INTERSECT и EXCEPT).

Результат действия – **НЕИМЕНОВАННАЯ ТАБЛИЦА.**

Запрос **любой сложности** можно сформулировать в виде **одного** оператора **SELECT**, содержащего несколько подзапросов.

8.7.2. Синтаксическая диаграмма

Только смысл конструкций.
Пара [] обозначает
необязательную конструкцию.

SELECT [**ALL** | **DISTINCT**] целевой_список
FROM источник_данных
[**WHERE** условное_выражение]
[**GROUP BY** список_столбцов_группирования]
[**HAVING** условное_выражение]
[**ORDER BY** список_столбцов_упорядочения];

СИМВОЛ ‘;’ ЕСТЬ ПРИЗНАК ЗАВЕРШЕНИЯ ОПЕРАТОРА.
ОБЯЗАТЕЛЕН!!!

Предложения **SELECT** и **FROM** – *обязательные*.

SELECT определяет состав столбцов производной таблицы (S-таблицы).

FROM определяет источник данных запроса (F-таблицу).

Остальные четыре: **WHERE**, **GROUP BY**, **HAVING**, **ORDER BY**
задают ограничения на содержание и вид выходных данных.

Порядок следования предложений в записи оператора **SELECT**
должен соответствовать указанному в диаграмме.

8.7.3. Простейший пример

Получить имена и статусы всех поставщиков.

```
SELECT ALL S.Snam, S.St  
FROM S;
```

```
SELECT DISTINCT S.Snam, S.St  
FROM S;
```

Базовая таблица S

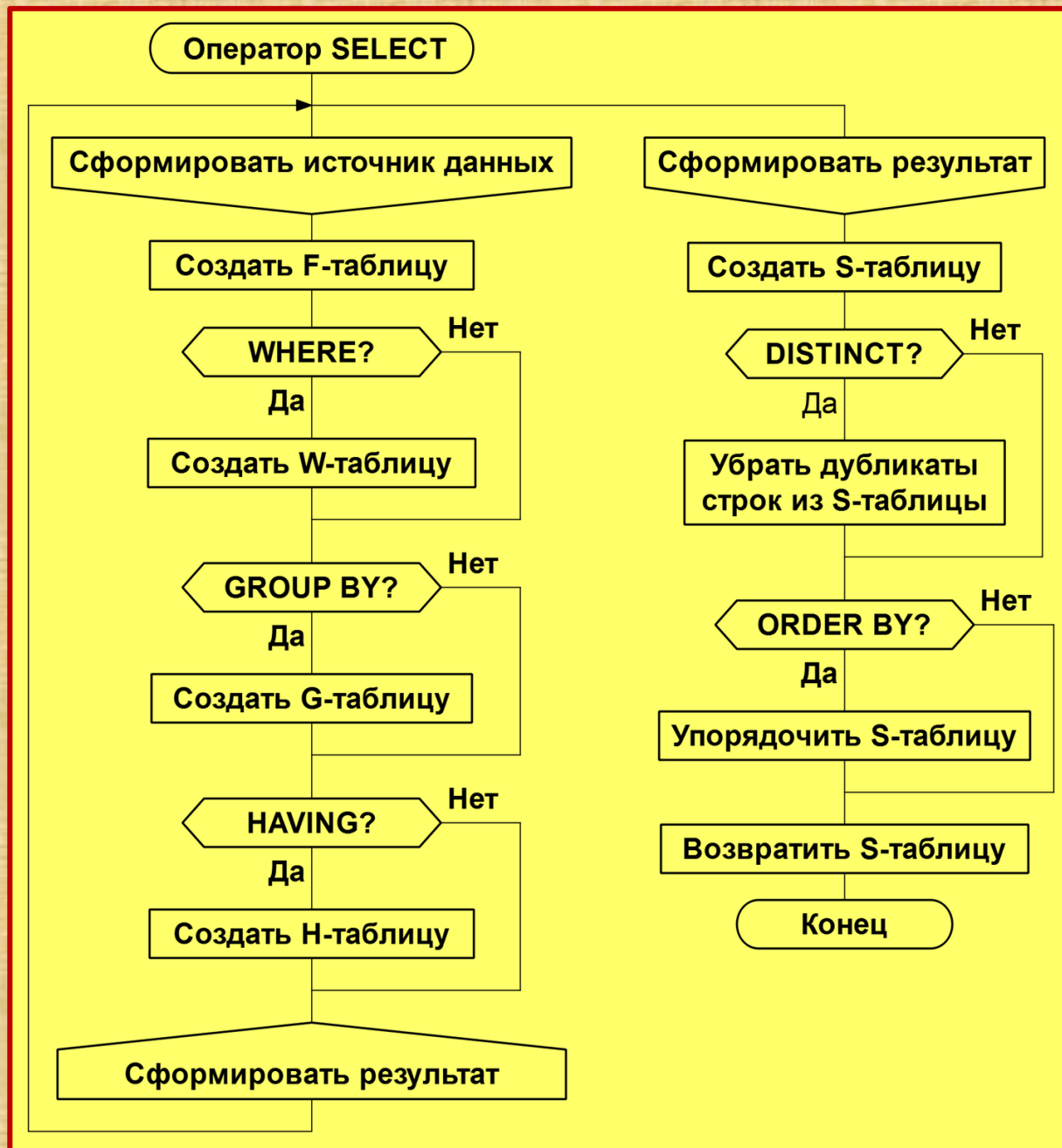
Snum	Snam	St	Ci
S019	Иван	100	Томск
S151	Николай	50	Асино
S002	Григорий	80	Яя
S423	Петр	20	Каргасок
S156	Иван	100	Асино
S105	Петр	90	Каштак
S009	Владимир	30	Синюки
S094	Егор	100	Токио

Результат ALL

Snam	St
Иван	100
Николай	50
Григорий	80
Петр	20
Иван	100
Петр	90
Владимир	30
Егор	100

Результат DISTINCT

Snam	St
Николай	50
Григорий	80
Петр	20
Иван	100
Петр	90
Владимир	30
Егор	100




8.7.5. Синтаксические единицы оператора SELECT

Целевой список

целевой_список ::= {элемент_целевого_списка,...}
элемент_целевого_списка ::= выражение [[AS] псевдоним] | [спецификатор.*]
выражение ::= имя столбца F-таблицы | вызов агрегатной функции |
 выражение для вычисления значения | литерал
спецификатор ::= имя таблицы источника данных

целевой список

SELECT DISTINCT  S.*, 'поставил', SPJ.Pnum AS Деталь, sum(SPJ.Qt)

S.Snum	S.Nam	S.St	S.Ci		Деталь	
S012	Иван	100	Яя	поставил	P123	12500
S038	Иван	75	Томск	поставил	P095	500

8.7.6. Источник данных

источник_данных ::= {элемент_источника,...}

элемент_источника ::= имя_таблицы [[AS] псевдоним [(имя_столбца,...)]] |
подзапрос [AS] псевдоним [(имя_столбца,...)] |
соединение

ПРИМЕР:

FROM SPJ, S INNER JOIN J AS SJ

соединение ::= перекрёстное_соединение |
естественное_соединение |
соединение_по_именам_столбцов |
соединение_по_условию |
UNION_соединение

8.7.7. Примеры запросов

Сравнить с выражениями РА и РИ

1) Получить полные сведения обо всех производимых изделиях.

РА

J;

РИ

RANGE OF JX IS J
JX;

SQL

SELECT J.Jnum, J.Jnam, J.Ci FROM J;

SELECT Jnum, Jnam, Ci FROM J;

SELECT J.* FROM J;

SELECT * FROM J;

SELECT ALL * FROM J;

SELECT DISTINCT * FROM J;

TABLE J;

2) Получить номера и названия изделий, производимых в Томске.

ПИ: RANGE OF JX IS J (JX.Jnum, JX.Jnam) **WHERE** JX.Ci = 'Томск';

SQL: SELECT J.Jnum, J.Jnam **FROM** J **WHERE** J.Ci = 'Томск';

3) Получить значения номеров поставщиков,
выполняющих поставки для изделия **J1**.

ПИ: SPJX.Snum **WHERE** SPJX.Jnum = 'J1';

SQL: SELECT Snum **FROM** SPJ **WHERE** Jnum = 'J1';

4) Получить значения номеров поставщиков,
поставляющих деталь **P1** для изделия **J1**.

PA: (SPJ **WHERE** Jnum = 'J1' **AND** Pnum = 'P1')[Snum];

SQL: SELECT Snum **FROM** SPJ **WHERE** Jnum = 'J1' **AND** Pnum = 'P1';

5) Получить значения наименований изделий,
для которых выполняет поставки поставщик **S1**.

JX.Jn **WHERE EXISTS** SPJX (SPJX.Snum = 'S1' **AND** SPJX.Jnum = J X.Jnum);

SELECT Jnam **FROM** J, SPJ **WHERE** SPJX.Snum = 'S1' **AND** SPJ.Jnum = J.Jnum;

6) Получить значения цветов деталей, поставляемых поставщиком S1.

ПИ: PX.Co **WHERE EXISTS** SPJX (SPJX.Snum = 'S1' **AND** SPJX.Pnum = PX.Pnum);

SQL: **SELECT DISTINCT** Co **FROM** SPJ, P **WHERE** Snum = 'S1' **AND** SPJ.Pnum = P.Pnum;

7) Получить номера поставщиков, поставляющих детали для изделий J1 и J2.

ПИ: SPJX.Snum **WHERE** SPJX.Jnum = 'J1' **AND**
EXISTS SPJY (SPJY.Snum = SPJX.Snum **AND** Jnum = 'J2');

SQL в стиле ПИ

SELECT Snum **FROM** SPJ **AS** SPJX **WHERE** SPJX.Jnum = 'J1' **AND**
EXISTS (**SELECT** * **FROM** SPJ **AS** SPJY
WHERE SPJY.Snum = SPJX.Snum **AND** SPJY.Jnum = 'J2');

SQL в стиле PA

SELECT Snum **FROM** SPJ **WHERE** Jnum = 'J1'
INTERSECT
SELECT Snum **FROM** SPJ **WHERE** Jnum = 'J2';

SQL в стиле SQL

SELECT Snum **FROM** SPJ
WHERE Jnum = 'J1' **AND**
Snum **IN** (**SELECT** Snum **FROM** SPJ **WHERE** Jnum = 'J2');

8) Получить значения номеров поставщиков, поставляющих для изделия J1 красную деталь.

```
SELECT Snum FROM SPJ, P
      WHERE Jnum = 'J1' AND P.Pnum = SPJ.Pnum AND Co = 'красный';
```

9) Получить значения номеров деталей,
поставляемых для каждого изделия, производимого в Томске.

```
ПИ: SPJX.Pnum WHERE FORALL JX
      ( IF JX.Ci='Томск' THEN EXISTS SPJY ( SPJY.Jnum = JX.Jnum AND
                                             SPJY.Pnum = SPJX.Pnum ) );
```

```
SQL: SELECT X.Pnum FROM SPJ AS X
      WHERE NOT EXISTS ( SELECT *
                        FROM J
                        WHERE Ci='Томск' AND
                        NOT EXISTS ( SELECT *
                                    FROM SPJ AS Y
                                    WHERE Y.Jnum=J.Jnum AND
                                    Y.Pnum = X.Pnum
                                    )
                        );
```

10) Получить значения номеров поставщиков, поставляющих
красные детали для изделий, производимых в Томске или Яе.

```
SPJX.Snum WHERE EXISTS PX EXISTS JX  
  ( PX.Co = 'красный' AND SPJX.Pnum = PX.Pnum AND  
    JX.Jnum = SPJX.Jnum AND JX.Ci = 'Томск' OR JX.Ci = 'Яя' );
```

```
SELECT DISTINCT Snum FROM SPJ, P, J  
  WHERE Co = 'красный' AND SPJ.Pnum = P.Pnum AND  
    J.Jnum = SPJ.Jnum AND J.Ci='Томск' OR J.Ci='Яя';
```

11) Получить значения номеров изделий, снабжаемых по крайней
мере одним поставщиком, расположенным не в том же самом городе.

```
SELECT DISTINCT Jnum FROM SPJ, S, J  
  WHERE J.Jnum = SPJ.Jnum AND S.Snum = SPJ.Snum AND NOT J.Ci = S.Ci;
```

12) Получить значения номеров изделий,
для которых не поставляется ни одной красной детали из Томска.

```
SELECT Jnum FROM J  
WHERE NOT EXISTS ( SELECT * FROM SPJ, P WHERE  
                        ( SPJ.Jnum = J.Jnum AND SPJ.Pnum = P.Pnum AND  
                        P.Ci = 'Томск' AND Co = 'красный'  
                        )  
);
```

13) Получить имена поставщиков, поставляющих деталь P2.

```
SELECT Sname FROM S, SPJ WHERE SPJ.Pnum = 'P2' AND S.Snum = SPJ.Snum;
```

14) Получить имена поставщиков, поставляющих **все** детали.

```
SELECT Snam  
FROM S WHERE NOT EXISTS  
    ( SELECT * FROM P WHERE NOT EXISTS  
        ( SELECT * FROM SPJ  
            WHERE ( P.Pnum = SPJ.Pnum AND SPJ.Snum = S.Snum )  
        )  
);
```

Домашнее задание

Перевести следующий запрос на SQL

15) Получить имена поставщиков, поставляющих **все поставляемые** детали.

SX.Snam **WHERE FOR ALL PX**

(IF EXISTS SPJX SPJX.Pnum = PX.Pnum

THEN EXISTS SPJY (SPJY.Pnum = PX.Pnum AND SPJY.Snum = SX.Snum)

);

Написать следующие запросы на SQL

16) Получить номера деталей, поставляемых для какого-либо изделия,
производимого в Яе поставщиком из Яи.

17) Получить номера поставщиков, поставляющих хотя бы одну деталь которая
поставляется хотя бы одним поставщиком, поставляющим красную деталь.