

СРСП 5. Выборка данных (предложение SELECT)

Предложение SELECT (выбрать) имеет следующий формат:

```
SELECT [[ALL] | DISTINCT]{ * | элемент_select [,элемент_select] ...}  
FROM      базовая_таблица | представление [псевдоним]  
          [,базовая_таблица | представление [псевдоним]] ...  
[WHERE     фраза]  
[GROUP BY  фраза [HAVING фраза]];
```

где элемент_select - это одна из следующих конструкций:

```
[таблица.]* | [таблица.]столбец | SQL_функция | переменная  
              | (выражение) | системная_переменная
```

В этом предложении нижеследующие конструкции означают

SELECT - (выбрать) данные из указанных столбцов и (если необходимо) выполнить перед выводом их преобразование в соответствии с указанными выражениями и (или) функциями

FROM - (из) перечисленных таблиц, в которых расположены эти столбцы

WHERE - (где) строки из указанных таблиц должны удовлетворять указанному перечню условий отбора строк

GROUP BY - (группируя по) указанному перечню столбцов с тем, чтобы получить для каждой группы единственное агрегированное значение, используя во фразе SELECT SQL-функции SUM (сумма), COUNT (количество), MIN (минимальное значение), MAX (максимальное значение) или AVG (среднее значение)

ORDER BY - а затем упорядочить результаты выбора данных. При этом упорядочение можно производить в порядке возрастания - ASC (ASCending) или убывания DESC (DESCending), а по умолчанию принимается ASC.

HAVING - (имея) в результате лишь те группы, которые удовлетворяют указанному перечню условий отбора групп

Кроме традиционных операторов сравнения (= | <> | < | <= | > | >=) в WHERE фразе используются условия BETWEEN (между), LIKE (похоже на), IN (принадлежит), IS NULL (не определено) и EXISTS (существует), которые могут предваряться оператором NOT (не). Критерий отбора строк формируется из одного или нескольких условий, соединенных логическими операторами:

AND - когда должны удовлетворяться оба разделяемых с помощью AND условия;

OR - когда должно удовлетворяться одно из разделяемых с помощью OR условий;

AND NOT - когда должно удовлетворяться первое условие и не должно второе;

OR NOT - когда или должно удовлетворяться первое условие или не должно удовлетворяться второе, причем существует приоритет AND над OR (сначала выполняются все операции AND и только после этого операции OR). Для получения желаемого результата WHERE условия должны быть введены в правильном порядке, который можно организовать введением скобок.

При обработке условия числа сравниваются алгебраически - отрицательные числа считаются меньшими, чем положительные, независимо от их абсолютной величины. Строки символов сравниваются в соответствии с их представлением в коде, используемом в конкретной СУБД, например, в коде ASCII. Если сравниваются две строки символов, имеющих разные длины, более короткая строка дополняется справа пробелами для того, чтобы они имели одинаковую длину перед осуществлением сравнения.

Пусть в базе данных имеются три взаимосвязанных таблицы:

Товары

Код товара	Наименование	Ед_изм	Категория_товара
1	Товар_1	шт	электроника
2	Товар_2	шт	мебель
3	Товар_3	литр	напитки
4	Товар_4	кг	мясные изделия
5	Товар_5	шт	книги

Поставщики

Код поставщика	Наименование	Номер счета	Банк
1	Поставщик_1	1111	Народный
2	Поставщик_2	2222	Альянс
3	Поставщик_3	3333	Народный
4	Поставщик_4	4444	Альянс
5	Поставщик_5	5555	Народный

Движение товара

Дата	№ накладной	Код поставщика	Код товара	Количество	Цена
01.01.06	1	1	2	100	10
10.01.06	2	5	3	50	20
20.01.06	3	1	2	100	10
30.01.06	4	2	1	50	15
01.03.06	5	3	4	100	25
10.03.06	6	4	3	200	20
20.03.06	7	1	2	10	12
30.03.06	8	2	1	50	15
01.10.06	9	3	4	10	25
10.10.06	10	5	3	20	20
20.10.06	11	1	2	100	12
30.10.06	12	2	1	5	15

Простая выборка

Запрос: выдать название и категории товаров

```
SELECT    Наименование, Категория_товара
FROM      Товары;
```

дает результат

Наименование	Категория_товара
Товар_1	электроника
Товар_2	мебель
Товар_3	напитки
Товар_4	мясные изделия
Товар_5	книги

При необходимости получения полной информации о поставщиках, можно было бы дать запрос

```
SELECT    Код поставщика, Наименование, Номер счета, Банк
FROM      Поставщики;
```

или использовать его более короткую нотацию:

```
SELECT    *
FROM      Поставщики;
```

Здесь "звездочка" (*) служит кратким обозначением всех имен полей в таблице, указанной во фразе FROM. При этом порядок вывода полей соответствует порядку, в котором эти поля определялись при создании таблицы.

Результат запроса:

Код поставщика	Наименование	Номер счета	Банк
1	Поставщик 1	1111	Народный
2	Поставщик 2	2222	Альянс
3	Поставщик 3	3333	Народный
4	Поставщик 4	4444	Альянс
5	Поставщик 5	5555	Народный

Выдать список банков, в которых обслуживаются поставщики товаров:

```
SELECT    Банк
FROM      Поставщики;
```

Результат запроса:

Банк
Народный
Альянс
Народный
Альянс
Народный

В предыдущем примере был выдан правильный, но не совсем удачный перечень банков: из него не были исключены дубликаты. Для исключения дубликатов и одновременного упорядочения перечня необходимо дополнить запрос ключевым словом DISTINCT (различный, различные), как показано в следующем примере:

```
SELECT DISTINCT Банк
FROM      Поставщики;
```

Результат запроса:

Банк
Альянс
Народный

Выборка вычисляемых значений

Из синтаксиса фразы SELECT видно, что в ней может содержаться не только перечень столбцов таблицы или символ *, но и выражения.

Например, если нужно получить стоимость поставленных товаров по накладной, то нужно сформулировать запрос:

```
SELECT    № накладной, (Количество*Цена)
FROM      Движение товара;
```

Результат выборки:

№ накладной	Количество*Цена
1	1000

2	100
3	1000
4	750
5	2500
6	4000
7	120
8	750
9	250
10	400
11	1200
12	75

Фраза SELECT может включать не только выражения, но и отдельные числовые или текстовые константы. Следует отметить, что текстовые константы должны заключаться в апострофы (').

SELECT № накладной, 'Стоимость =' , (Количество*Цена)
FROM Движение товара;

№ накладной		Количество*Цена
1	Стоимость =	1000
2	Стоимость =	100
3	Стоимость =	1000
4	Стоимость =	750
5	Стоимость =	2500
6	Стоимость =	4000
7	Стоимость =	120
8	Стоимость =	750
9	Стоимость =	250
10	Стоимость =	400
11	Стоимость =	1200
12	Стоимость =	75

Имеется также возможность дать название вычисляемому полю. При этом используется следующий синтаксис

SELECT № накладной, (Количество*Цена) AS Стоимость
FROM Движение товара;
Результат выборки:

№ накладной	Стоимость
1	1000
2	100
3	1000
4	750
5	2500
6	4000
7	120
8	750
9	250
10	400
11	1200
12	75

Выборка с использованием фразы *WHERE*

Во фразе WHERE для отбора нужных строк таблицы можно использовать операторы сравнения = (равно), <> (не равно), < (меньше), <= (меньше или равно), > (больше), >= (больше или равно), которые могут предваряться оператором NOT, создавая, например, отношения "не меньше" и "не больше".

Так, для получения перечня накладных, выписанных организаций с кодом 1 можно сформировать запрос

```
SELECT    № накладной, Дата
FROM      Движение товара
WHERE     Код поставщика = 1;
```

и получить в результате выборки:

№ накладной	Дата
1	01.01.06
3	20.01.06
7	20.03.06
11	20.10.06

Возможность использования нескольких условий, соединенных логическими операторами AND, OR, AND NOT и OR NOT, позволяет осуществить более детальный отбор строк. Так, для получения перечня накладных, содержащих товары стоимостью менее 20 тенге и количеством >=50 :

```
SELECT    № накладной, Дата, Количество, Цена
FROM      Движение товара
WHERE     Цена<20 AND Количество>=50;
```

Результат запроса имеет вид

№ накладной	Дата	Количество	Цена
1	01.01.06	100	10
3	20.01.06	100	10
4	30.01.06	50	15
8	30.03.06	50	15
11	20.10.06	100	12

Использование *BETWEEN*

С помощью BETWEEN ... AND ... (находится в интервале от ... до ...) можно отобрать строки, в которых значение какого-либо столбца находится в заданном

диапазоне. BETWEEN чувствителен к порядку, и первое значение в предложении должно быть первым по алфавитному или числовому порядку.

Например, выдать перечень накладных, в которых значение цены товаров находится в диапазоне от 10 до 15:

```
SELECT    № накладной, Дата. Количество, Цена
FROM      Движение товара
WHERE     Цена BETWEEN 10 AND 15;
```

Результат запроса имеет вид

№ накладной	Дата	Количество	Цена
1	01.01.06	100	10
3	20.01.06	100	10
4	30.01.06	50	15
7	20.03.06	10	12
8	30.03.06	50	15
11	20.10.06	100	12
12	30.10.06	5	15

Можно задать и NOT BETWEEN (не принадлежит диапазону между), например:

```
SELECT    № накладной, Дата. Количество, Цена
FROM      Движение товара
WHERE     Цена NOT BETWEEN 10 AND 15;
```

Результат запроса имеет вид

№ накладной	Дата	Количество	Цена
2	10.01.06	50	20
5	01.03.06	100	25
6	10.03.06	200	20
9	01.10.06	10	25
10	10.10.06	20	20

Использование IN

Оператор IN определяет набор значений в которое данное значение может или не может быть включено.

Выдать сведения о накладных, по которым были поставлены товары с кодами 1 и 2:

```
SELECT    *
FROM      Движение товара
WHERE     Код товара IN (1, 2);
```

Результат выборки:

Дата	№ накладной	Код поставщика	Код товара	Количество	Цена
01.01.06	1	1	2	100	10
20.01.06	3	1	2	100	10
30.01.06	4	2	1	50	15
20.03.06	7	1	2	10	12
30.03.06	8	2	1	50	15
20.10.06	11	1	2	100	12
30.10.06	12	2	1	5	15

Рассмотренная форма IN является в действительности просто краткой записью последовательности отдельных сравнений, соединенных операторами OR. Предыдущее предложение эквивалентно такому:

```
SELECT      *
FROM        Движение товара
WHERE       Код товара=1 OR Код товара=2;
Можно задать и NOT IN (не принадлежит).
```

Использование LIKE

LIKE применим только к полям типа CHAR или VARCHAR, с которыми он используется чтобы находить подстроки. Т.е. он ищет поле символа чтобы видеть, совпадает ли с условием часть его строки. В качестве условия он использует групповые символы - специальные символы которые могут соответствовать чему-нибудь. Имеются два типа групповых символов используемых с LIKE:

- символ _ (подчеркивание) – заменяет любой одиночный символ,
- символ % (процент) – заменяет любую последовательность из N символов (где N может быть нулем),

Обычная форма "имя_столбца LIKE текстовая_константа" для столбца текстового типа позволяет отыскать все значения указанного столбца, соответствующие образцу, заданному "текстовой_константой".

Выдать список организаций, в наименовании которых присутствует сочетание символов «оставщик»

```
SELECT      Наименование
FROM        Поставщики
WHERE       Наименование LIKE '%оставщик%';
Результат выборки:
```

Наименование
Поставщик 1
Поставщик 2
Поставщик 3
Поставщик 4
Поставщик 5

Вовлечение неопределенного значения (NULL-значения)

Если при загрузке данных не введено значение в какое-либо поле таблицы, то СУБД поместит в него NULL-значение. Аналогичное значение можно ввести в поле таблицы, выполняя операцию изменения данных. Так, при отсутствии сведений о цене товара в момент поставки в таблице Движение товара в столбце Цена будет введено NULL и там будет храниться код NULL-значения, а не 0, 0. или пробел.

В этом случае для выявления неправильно оформленных накладных (в которых отсутствует цена на товар) можно дать запрос

```
SELECT      № накладной
FROM        Движение товара
WHERE       Цена IS NULL;
```

Естественно, что для выявления правильно оформленных накладных продуктов, существующих в кладовой, следует дать запрос

```
SELECT      № накладной
FROM        Движение товара
```

WHERE Цена IS NOT NULL;

Использование условий

столбец IS NULL и столбец IS NOT NULL

вместо, например,

столбец = NULL и столбец <> NULL

связано с тем, что ничто - и даже само NULL-значение - не считается равным другому NULL-значению. (Несмотря на это, два неопределенных значения рассматриваются, однако, как дубликаты друг друга при исключении дубликатов, и предложение SELECT DISTINCT даст в результате не более одного NULL-значения.)

Задание к СРСП 5:

Исходные данные: Пусть имеются таблицы БД со следующей структурой

Продавцы (Таб.номер, ФИО, город)

Товары (Номер товара, Наименование)

Движение товаров (дата, таб.номер продавца, код товара, количество товара, цена, ФИО покупателя)

1. Напишите команду SELECT которая бы вывела код товара, сумму (как цена*количество товара) и дату для всех строк из таблицы Движение товаров.
2. Напишите запрос который вывел бы все строки из таблицы Движение товаров для которых номер продавца = 1001.
3. Напишите запрос который вывел бы таблицу Продавцы в следующем порядке: город, ФИО, таб.номер
4. Напишите запрос который вывел бы значения таб.номер всех продавцов в текущем порядке из таблицы Движение товаров без каких бы то ни было повторений.
5. Напишите запрос который вывел бы ФИО продавцов, проживающих в г.Караганде
6. Выведите список кодов товаров, цена которых находится в диапазоне от 200 до 1000 тенге
7. Выведите ФИО покупателей, ФИО которых начинаются с буквы «З»
8. Выдать сведения о накладных, по которым были поставлены товары с кодами 10, 100, 1000