

Тема 9. Оператор SELECT

Оператор `SELECT` – центральная конструкция `SQL`, является специальным видом селекции одного или нескольких отношений с последующей проекцией на указанные атрибуты. Оператор `SELECT` заменяет операции декартова произведения, соединения, селекции и проекции реляционной алгебры.

Синтаксис оператора:

```
SELECT [ALL | DISTINCT] {список_данных | *}  
FROM список_таблиц  
[WHERE условие_выборки]  
[GROUP BY список_столбцов]  
[HAVING условие_выборки_групп]  
[ORDER BY ключ_сортировки]
```

Оператор производит выборку и вычисления над данными одной или нескольких таблиц. Результатом является таблица, в которой могут быть (`ALL`) или не могут быть (`DISTINCT`) повторяющиеся строки.

Список данных определяет столбцы результирующей таблицы. Он состоит из имен столбцов исходных таблиц и выражений над столбцами. Символ `*` означает, что результирующая таблица содержит имена всех столбцов исходных отношений.

Предложение `FROM` перечисляет исходные таблицы запроса.

Предложение `WHERE` указывает условия отбора кортежей.

Предложение `GROUP BY` выделяет в результирующем множестве группы кортежей с совпадающими значениями в полях группировки. Группы используются для вычисления групповых операций.

Предложение `HAVING` действует совместно с `GROUP BY` и используется для селекции групп.

Предложение `ORDER BY` задает порядок сортировки результирующей таблицы.

В условиях предложения `WHERE` используются следующие предикаты:

- Сравнение;
- BETWEEN A AND B;
- IN (множество) (NOT IN (множество)) – проверка принадлежности элемента множеству. Множество – перечисление или вложенный подзапрос.
- LIKE (NOT LIKE) – сравнение с шаблоном (_ представляет любой одиночный символ, % - произвольную последовательность символов).
- IS NULL (IS NOT NULL) – сравнение с неопределенным значением.
- EXISTS (NOT EXISTS) – предикат существования для вложенных запросов.

В условиях используются логические операции AND, OR, NOT.

Примеры запросов

Пример.

Выдать фамилии покупателей с положительным балансом.

```
SELECT Фамилия
FROM Покупатели
WHERE Баланс > 0
```

В отношении Покупатели селектируются кортежи с положительным балансом, и результат селекции проецируется на атрибут Фамилия. При этом создается неявная переменная-кортеж, ассоциированная с именем отношения, пробегающая по всем кортежам отношения.

Пример.

Напечатать названия и адреса поставщиков, поставляющих шурупы или гвозди.

```
SELECT DISTINCT Название, Адрес
FROM Поставщики
WHERE Товар=шурупы OR Товар=гвозди
```

В отношении Поставщики ищутся кортежи, столбец Товар которых содержит слова «шуропы» или «гвозди», и проецируются на атрибуты Название, Адрес. Так как указано DISTINCT, результат будет содержать в точности один кортеж для каждого поставщика шурупов или гвоздей.

Пример «Петров».

Выдать атрибуты Название, Товар, Цена поставщиков, имеющих товары, заказанные покупателем Петровым.

```
SELECT Название, Товар, Цена
FROM Поставщики
WHERE Товар IN
      ( SELECT Товар
        FROM Заказы
        WHERE Фамилия=Петров)
```

В данном случае имеет место композиция двух запросов. Внутренний запрос селектирует в отношении Заказы кортежи с фамилией покупателя Петров и проецирует их на атрибут Товар. Внешний оператор выбирает в отношении Поставщики кортежи, компонент Товар которых принадлежит найденному ранее множеству. Выбранные кортежи проецируются на атрибуты Название, Товар, Цена.

SQL позволяет использовать псевдонимы отношений. Псевдоним задается предложением FROM R T, где T – псевдоним отношения R. Внутри предложения WHERE можно ссылаться на компонент A кортежа T посредством T.A. Псевдонимы требуются, если в запросе одновременно используются несколько разных ссылок (разных переменных-кортежей) на одно отношение.

Пример.

Напечатать фамилии всех покупателей, проживающих по одному адресу с какими-нибудь другими покупателями.

```
SELECT Фамилия
FROM Покупатели T
```

```
WHERE 0<
      ( SELECT COUNT (*)
        FROM Покупатели
        WHERE Фамилия<>Т.Фамилия AND Адрес=Т.Адрес)
```

Переменная-кортеж Т внешнего запроса пробегает по всем кортежам отношения Покупатели. Во внутреннем запросе для каждого значения Т в этом же отношении селективируются строки с адресом Т.Адрес и фамилией, отличной от Т.Фамилия.

При этом используется переменная-кортеж вложенного запроса, связанная с отношением Покупатели. Определяется число найденных строк (функция count (*)). Проверяется условие внешнего запроса. Если условие выполняется, то значение Т поступает в результат.

В предложении FROM можно указать более чем одно отношение. В этом случае вычисляется декартово произведение отношений. В предложении WHERE указывается условие селекции декартова произведения, в предложении SELECT – компоненты декартова произведения. Если отношения имеют одинаковые атрибуты, то к ним можно обращаться имя_отношения. имя_атрибута.

Пример «Петров».

```
SELECT Название, Поставщики.Товар, Цена
FROM Поставщики, Заказы
WHERE Фамилия=Петров AND Поставщики.Товар=Заказы.Товар
```

Как видим, использование декартова произведения позволяет заменить композицию запросов одним простым запросом. Декартово произведение таблиц R и S можно записать так:

```
SELECT DISTINCT *
FROM R, S
```