

Язык SQL

СУБД System R - экспериментальная исследовательская система с языком **SEQUEL** (позже **SQL**), созданная IBM:

- Полный реляционный язык БД
- Операторы манипулирования БД
- Средства определения и манипулирования схемой БД
- Определение ограничений целостности
- Определение представлений
- Определение индексов
- Авторизация доступа к отношениям и их полям
- Точки сохранения транзакций и откаты

Стандарты SQL1, SQL2, SQL3 (1992)

Содержит только набор стандартных операторов доступа к данным, хранящимся в базе данных. Операторы SQL встраиваются в базовый язык программирования

Достоинства:

- Повсеместная распространенность
- Быстрое обучение в простых случаях
- Связывание с различными языками программирования
- Поддержка ODBC и JDBC
- Фактор времени: научились хорошо реализовывать.

Недостатки:

- Несоответствие реляционной модели данных (наличие дубликатов, необязательность первичного ключа, возможность упорядочения результатов)
- Недостаточно продуманный механизм неопределенных значений
- Сложность формулировок и громоздкость.

Литература

Базы данных: модели реализации / Т. С. Карпова. - СПб.: Питер, 2001. - 304 с.: ил.

Хомоненко А. Д., Цыганков В. М., Мальцев М. Г. Базы данных: Учебник для высших учебных заведений /Под. ред. проф. А. Д. Хомоненко. - СПб.: КОРОНА принт, 2000. - 416 с.

Оператор выбора SELECT

Язык запросов (Data Query Language) в SQL состоит из единственного оператора SELECT Этот единственный оператор поиска реализует все операции реляционной алгебры. Синтаксис оператора SELECT имеет следующий вид:

SELECT [ALL | DISTINCT] < список полей >|*)

FROM < Список таблиц >

[WHERE < Предикат - условие выборки или соединения >]

[GROUP BY < Список полей результата >]

[HAVING < Предикат - условие для группы >]

[ORDER BY < Список полей , по которым упорядочить вывод >]

Ключевое слово ALL означает, что в результирующий набор строк включаются все строки, удовлетворяющие условиям запроса.

Значит, в результирующий набор могут попасть одинаковые строки. И это нарушение принципов теории отношений.

DISTINCT означает, что в результирующий набор включаются только различные строки, то есть дубликаты строк результата не включаются в набор. Символ *. означает, что в результирующий набор включаются все столбцы из исходных таблиц запроса. В разделе FROM задается перечень исходных отношений (таблиц) запроса. В разделе WHERE задаются условия отбора строк результата или условия соединения кортежей исходных таблиц

В разделе GROUP BY задается список полей группировки.

В разделе HAVING задаются предикаты-условия, накладываемые на каждую группу.

- В части ORDER BY задается список полей упорядочения результата, то есть список полей, который определяет порядок сортировки в результирующем отношении.

- Предикат вхождения в множество IN (множество) истинен тогда, когда сравниваемое значение входит в множество заданных значений. При этом множество значений может быть задано простым перечислением или встроенным подзапросом. Одновременно существует противоположный предикат NOT IN (множество), который истинен тогда, когда сравниваемое значение не входит в заданное множество.

- Предикаты сравнения с образцом LIKE и NOT LIKE . Предикат LIKE требует задания шаблона, с которым сравнивается заданное значение, предикат истинен, если сравниваемое значение соответствует шаблону, и ложен в противном случае. Предикат NOT LIKE имеет противоположный смысл.

По стандарту в шаблон могут быть включены специальные символы:

- символ подчеркивания (`_`) — для обозначения любого одиночного символа;
- символ процента (`%`) — для обозначения любой произвольной последовательности символов;
- остальные символы, заданные в шаблоне, обозначают самих себя.

Предикат сравнения с неопределенным значением `IS NULL`. Это значение при появлении дополнительной информации в любой момент времени может быть заменено на некоторое конкретное значение. Для выявления равенства значения некоторого атрибута неопределенному применяют специальные стандартные предикаты: **< имя атрибута >IS NULL** и **< имя атрибута > IS NOT NULL**. Если в данном кортеже (в данной строке) указанный атрибут имеет неопределенное значение, то предикат `IS NULL` принимает значение «Истина» (`TRUE`), а предикат `IS NOT NULL` — «Ложь» (`FALSE`), в противном случае предикат `IS NULL` принимает значение «Ложь», а предикат `IS NOT NULL` принимает значение «Истина».

Введение Null -значений вызвало необходимость модификации классической двузначной логики и превращения ее в трехзначную. Все логические операции, производимые с неопределенными значениями, подчиняются этой логике в соответствии с заданной таблицей истинности:

A	B	Not A	AAB	A V B
TRUE	TRUE	FALSE	TRUE	TRUE
TRUE	FALSE	FALSE	FALSE	TRUE
TRUE	Null	FALSE	Null	TRUE
FALSE	TRUE	TRUE	FALSE	TRUE
FALSE	FALSE	TRUE	FALSE	FALSE
FALSE	Null	TRUE	FALSE	Null
Null	TRUE	Null	Null	TRUE
Null	FALSE	Null	FALSE	Null
Null	Null	Null	Null	Null

Предикаты существования `EXIST` и несуществования `NOT EXIST` .

Эти предикаты относятся к встроенным подзапросам.

`SELECT` — ключевое слово, которое сообщает СУБД, что эта команда — запрос. Все запросы начинаются этим словом с доследующим пробелом. За ним может следовать способ выборки — с удалением дубликатов (`DISTINCT`) или без удаления (`ALL` , подразумевается по умолчанию). Затем следует список перечисленных через запятую столбцов, которые выбираются запросом из таблиц, или символ '*' (звездочка) для выбора всей строки.

`FROM` — ключевое слово, подобно `SELECT` , которое должно быть представлено в каждом запросе. Оно сопровождается пробелом и затем именами таблиц, используемых в качестве источника информации. Таблицам можно присвоить имена-псевдонимы, что бывает полезно для осуществления операции соединения таблицы с самой собою или для доступа из вложенного подзапроса к текущей записи внешнего запроса

Все последующие разделы оператора SELECT являются необязательными. Самый простой запрос SELECT без необязательных частей соответствует просто декартову произведению.

Например, выражение `SELECT * FROM R1, R2` соответствует декартову произведению таблиц R 1 и R 2

Выражение

`SELECT R1.A, R2.B FROM R1, R2`

соответствует проекции декартова произведения двух таблиц на два столбца A из таблицы R 1 и B из таблицы R 2, при этом дубликаты всех строк сохранены, в отличие от операции проектирования в реляционной алгебре, где при проектировании по умолчанию все дубликаты кортежей уничтожаются.

WHERE — ключевое слово, за которым следует предикат — условие, налагаемое на запись в таблице, которому она должна удовлетворять, чтобы попасть в выборку, аналогично операции селекции в реляционной алгебре.

Примеры

Вывести список всех групп (без повторений), где должны пройти экзамены.

R1= (ФИО, Дисциплина, Оценка); R 2 = (ФИО, Группа);

R3=(Группы, Дисциплина)

```
SELECT DISTINCT Группы FROM R3
```

Результат:

Группа

4906

4807

Вывести список студентов, которые сдали экзамен по дисциплине «Базы данных» на «отлично».

```
SELECT ФИО
```

```
FROM R 1
```

```
WHERE Дисциплина = " Базы данных " AND Оценка = 5
```

Вывести список всех студентов, которым надо сдавать экзамены с указанием названий дисциплин, по которым должны проводиться эти экзамены.

```
SELECT ФИО , Дисциплина
```

```
FROM R2.R3
```

```
WHERE R2.Группа =R 2.Группа
```

Вывести список лентяев, имеющих несколько двоек.

```
SELECT DISTINCT R1.ФИО
```

```
FROM R1 a, R2 b :
```

```
WHERE a. ФИО = b.ФИО AND
```

```
a. Дисциплина <> b. Дисциплина AND
```

```
a.Оценка <= 2 AND b.Оценка <- 2;
```

Здесь мы использовали псевдонимы для именованя отношения R 1 а и b, так как для записи условий поиска нам необходимо работать сразу с двумя экземплярами данного отношения.

Логика работы оператора выбора (декартово произведение— селекция—проекция) не совпадает с порядком описания в нем данных.

SQL изначально разрабатывался для применения конечными пользователями, и его стремились сделать возможно ближе к языку естественному, а не к языку алгоритмическому.

«Найти студентов, пришедших на экзамен, но не сдававших его с указанием названия дисциплины». Оператор SELECT будет выглядеть следующим образом:

```
SELECT ФИО . Дисциплина
```

```
FROM R1
```

```
WHERE Оценка IS NULL
```