

Лекция № 14.

Язык реляционных баз данных SQL

Язык реляционных баз данных SQL

Язык SQL (Structured Query Language) появился в середине 70-х годов и был разработан в рамках экспериментальной реляционной СУБД System R.

В основу SQL положена комбинация реляционного исчисления кортежей и реляционной алгебры. При этом возможности SQL шире, чем у этих средств.

SQL в том или ином варианте присутствует практически во всех коммерческих СУБД.

Существует несколько стандартов языка, но все они во многом сводятся в основном к аккуратной технической обработке идей SQL, впервые появившегося в системе System R.

Язык реляционных баз данных SQL

Первоначально SQL был ориентирован главным образом на удобную и понятную пользователю формулировку запросов. Позже в него помимо операторов формулирования запросов были включены и другие средства, делающие его полным языком БД, такие как:

- определение схемы БД и манипулирование ею;
- определение ограничений целостности и триггеров (процедур, поддерживающих ограничение целостности);
- определение структур физического уровня (индексов);
- определение представлений БД;
- определение авторизации доступа к отношениям и их полям;
- определение точек сохранения транзакций и откатов;
- поддержка встраивания SQL в языки программирования и динамического исполнения операторов языка SQL.

Формулирование запросов к РБД

Запрос к РБД формулируется оператором выборки данных *SELECT*. Средствами SQL можно формулировать простые запросы к соединениям нескольких отношений и вложенных подзапросов в предикатах (условиях) выборки. В предикатах с вложенными подзапросами можно употреблять теоретико-множественные операции сравнения, использовать кванторы и т.п.

Результатом выполнения оператора *SELECT* будет некоторое отношение, в общем случае являющееся не множеством, а мультимножеством кортежей (в нем могут присутствовать кортежи-дубликаты).

В результирующем отношении могут выполняться различные группирования данных по полям в соответствии с заданными условиями.

Оператор выборки данных SELECT

В простейшем случае оператор SELECT выглядит следующим образом:

SELECT <имена столбцов> FROM <имена таблиц>

В полной форме оператора могут присутствовать дополнительные разделы:

SELECT <имена столбцов>

FROM <имена таблиц>

WHERE <условие соединения> AND <условие выборки записей>

GROUP BY <имена столбцов>

HAVING <условие выборки групп>

ORDER BY <имена столбцов>

Оператор выборки данных SELECT

Результатом выполнения оператора SELECT в простейшей форме

SELECT <имена столбцов> FROM <имена таблиц>

будет таблица, составленная из заданных столбцов указанных таблиц (или одной таблицы).

Если в выборке участвует несколько таблиц, то для однозначной идентификации их столбцов указывается полное имя столбца:

<имя таблицы>.<имя столбца>

Если вместо списка имен столбцов указать символ *, то результирующая таблица будет состоять из всех столбцов всех указанных таблиц.

Оператор выборки данных **SELECT**

Вместо имени столбца в операторе может быть указано любое выражение, в том числе и константа. В этом случае в качестве значений этого столбца будут выступать результаты вычисления указанного выражения для каждой записи результирующей таблицы.

Использование в запросе выражений позволяет вычислять комбинацию данных из нескольких столбцов, а использование констант позволяет вставлять столбцы с комментариями.

В большинстве случаев имена столбцов в результирующей таблице совпадают с именами столбцов из исходных таблиц, однако они могут быть изменены для сохранения уникальности:

```
SELECT <имя столбца> AS <новое имя>, ...  
FROM <имена таблиц>.
```

Для столбцов-выражений имена порождаются автоматически.

Оператор выборки данных **SELECT**

Для имен таблиц, указанных в запросе, можно задавать синонимы:

SELECT <имена столбцов>

FROM <имя таблицы> <синоним>, ...

В этом случае во всех остальных частях запроса вместо имен этих таблиц следует использовать их синонимы.

Такой прием в частности может быть использован, когда в запросе указано более одного вхождения одной и той же таблицы.

При выполнении запроса в таблице (например, в результате проекции) могут оказаться одинаковые записи. Чтобы исключить дублирующиеся записи, перед именами полей в команде нужно поместить ключевое слово *DISTINCT*:

SELECT DISTINCT <имена столбцов> FROM <имена таблиц>.

Оператор выборки данных *SELECT*

1. Соединение таблиц в запросе

С точки зрения реляционной алгебры результат выполнения оператора *SELECT* представляет собой проекцию прямого произведения отношений. На практике же обычно требуется получить не прямое произведение отношений, а их соединение.

Для этого в операторе *SELECT* нужно указать *условие соединения*:

```
SELECT <имена столбцов> FROM <имена таблиц>  
WHERE <условие соединения>
```

В качестве условия соединения может выступать сравнение двух атрибутов таблиц, а также конъюнкция таких условий (с помощью логической связки *AND*).

Оператор выборки данных **SELECT**

2. Задание условий выборки записей

Кроме условия соединения, в запросе может быть указано также и *условие выборки*. В этом случае оно обязательно должно следовать после условия соединения (если последнее задано):

```
SELECT <имена столбцов> FROM <имена таблиц>  
      WHERE <условие соединения> AND <условие выборки>
```

Задание условия выборки позволяет включать в результирующую таблицу не все записи, а только те из них, которые удовлетворяют этому условию.

Условие выборки может быть простым или содержать подзапрос, а также может состоять из нескольких частей, соединенных логическими связками *AND*, *OR* и *NOT*.

Оператор выборки данных SELECT

К простым условиям выборки относятся:

- сравнение значения атрибута со значением другого атрибута или любого выражения:

<атрибут> <сравнение> <атрибут>

<атрибут> <сравнение> <выражение> ,

где в качестве операции сравнения могут использоваться символы равенства (=), неравенства (< >, !=, #) и меньше/больше (<, <=, >, >=);

- проверка на принадлежность (не принадлежность) значения атрибута заданному интервалу:

<атрибут> BETWEEN <начало> AND <конец>

<атрибут> NOT BETWEEN <начало> AND <конец>;

Оператор выборки данных SELECT

- проверка на принадлежность (не принадлежность) значения атрибута заданному множеству значений:
<атрибут> IN (<набор значений>)
<атрибут> NOT IN (<набор значений>);
- проверка на соответствие (не соответствие) значения символьного атрибута заданному образцу:
<атрибут> LIKE <образец>
<атрибут> NOT LIKE <образец> ,

где в качестве образца может выступать любая строка, содержащая специальные символы:

- “ ” — сопоставляющийся с любым одним символом,
- “_” — сопоставляющийся с любым одним символом,
- “%” — сопоставляющийся с любым количеством любых символов.

Оператор выборки данных SELECT

Условия с подзапросом содержат внутри себя вложенный запрос к тем же или другим таблицам. Этот подзапрос должен формировать таблицу, состоящую из одного столбца, который интерпретируется как множество значений для последующей проверки истинности условия.

К таким условиям относятся:

- проверка на непустоту результата подзапроса:

EXISTS (<подзапрос>)

т.е. существует ли хотя бы одна запись во множестве, образованном результатом подзапроса;

- сравнение значения атрибута со всеми значениями результата подзапроса:

<атрибут> <сравнение> ALL (<подзапрос>)

т.е. сравнимо ли значение атрибута со всеми значениями из множества, образованного результатом подзапроса;

Оператор выборки данных SELECT

- сравнение значения атрибута с хотя бы одним значением результата подзапроса:

<атрибут> <сравнение> SOME (<подзапрос>)

т.е. сравнимо ли значение атрибута хотя бы с одним значением из множества, образованного результатом подзапроса;

- проверка на принадлежность (не принадлежность) значения атрибута множеству, образованному результатом подзапроса:

<атрибут> IN (<подзапрос>)

<атрибут> NOT IN (<подзапрос>)

т.е. принадлежит ли (не принадлежит ли) значение атрибута множеству, образованному результатом подзапроса.

Оператор выборки данных SELECT

3. Группировка записей и использование функций подсчета

В результирующую таблицу можно помещать не только существующие значения столбцов или результат вычисления выражения для каждой записи, но также и некоторую статистику (количество, сумму, среднее арифметическое и т.п.) по всем значениям столбца. Для этого используются средства группировки записей и набор специальных функций подсчета.

Группировка записей по одному или нескольким атрибутам задается с помощью раздела *GROUP BY*:

```
SELECT <имена столбцов> FROM <имена таблиц>  
      GROUP BY <имена столбцов>
```

Оператор выборки данных SELECT

Группировка записей по заданному атрибуту заключается в том, что все записи с одинаковыми значениями атрибута объединяются в одну группу и в результирующую таблицу попадает только один представитель от каждой группы.

Если задано разбиение по нескольким столбцам, то оно осуществляется последовательно, т.е. сначала все записи разбиваются на группы по первому указанному столбцу, потом внутри каждой группы по второму столбцу и т.д. Вместо имен столбцов можно указывать их порядковые номера в результирующей таблице.

Оператор выборки данных *SELECT*

Разбиение на группы обычно производится для подсчета статистики по столбцам. Для этого в операторе *SELECT* вместо имени столбца нужно указать одну из стандартных функций от значений этого столбца:

COUNT(*<имя столбца>*) – количество значений в столбце;

SUM(*<имя столбца>*) – сумма значений в столбце;

AVG(*<имя столбца>*) – среднее арифметическое в столбце;

MIN(*<имя столбца>*) – минимальное значение в столбце;

MAX(*<имя столбца>*) – максимальное значение в столбце.

Эти функции действуют над всеми значениями столбца внутри каждой группы. В этом случае представитель каждой группы в результирующей таблице будет содержать результат вычисления функции в соответствующем столбце.

Оператор выборки данных SELECT

Если группировка с помощью GROUP BY не задана, то вся исходная таблица считается одной группой и функции подсчета применяются ко всем значениям заданного столбца (или столбцов). Результирующая таблица в этом случае всегда будет состоять лишь из одной записи.

При использовании функций подсчета перед именем столбца можно указать ключевое слово *DISTINCT*, например:

COUNT(DISTINCT <имя столбца>).

В этом случае в подсчете будут участвовать только различные значения в столбце.

В функции *COUNT()* также вместо имени конкретного столбца можно указать символ *.

Оператор выборки данных SELECT

Вместо имени столбца в вызове функций может быть указано любое выражение. В этом случае будет осуществляться подсчет не значений столбца, а результатов вычисления заданного выражения для всех записей группы.

Таким образом, можно получать статистику не только по данным из столбца, но и по некоторой комбинации данных из одного или нескольких столбцов.

Оператор выборки данных **SELECT**

4. Задание условий выборки групп

Для того чтобы в результирующую таблицу попадали представители не всех групп, а только некоторых из них, удовлетворяющих заданному *условию выборки групп*, необходимо указать это условие в разделе *HAVING*:

```
SELECT <имена столбцов> FROM <имена таблиц>  
GROUP BY <имена столбцов >  
HAVING <условие выборки групп>
```

В этом условии, так же, как и в условии выборки записей, можно использовать операции сравнения, но их аргументами уже могут быть не только значения атрибутов, но и вызовы функций подсчета для значений столбцов. Подзапросы в этом условии использовать не разрешается. Условие может состоять из нескольких частей, соединенных логическими связками *AND*, *OR* и *NOT*.

Оператор выборки данных SELECT

5. Упорядочение записей

Для упорядочения записей в результирующей таблице запроса необходимо в разделе `ORDER BY` указать имена одного или нескольких столбцов, по которым последовательно будет производиться упорядочение записей:

```
SELECT <имена столбцов> FROM <имена таблиц>  
ORDER BY <имена столбцов>
```

Сначала записи упорядочиваются по первому столбцу, затем для записей с одинаковым значением в этом столбце – по второму столбцу и т.д. Если после имени столбца стоит признак `ASC`, то для этого столбца упорядочение производится по возрастанию значений, если `DESC`, то по их убыванию. По умолчанию происходит упорядочение по возрастанию (`ASC`).

Вместо имен столбцов можно также указывать их порядковые номера в результирующей таблице.

Оператор выборки данных SELECT

6. Объединение результатов запросов

Результаты двух или нескольких запросов можно объединить в одну таблицу, используя операцию объединения запросов *UNION*:

<запрос> UNION <запрос>

В этом случае результирующие таблицы запросов будут объединены в одну и должны иметь одинаковый формат. Таким образом можно объединить результаты нескольких запросов последовательно, при необходимости используя скобки.

По умолчанию дублирующиеся записи из объединенной таблицы исключаются. Для того чтобы они оставались в таблице, нужно указать ключевое слово *ALL*:

<запрос> UNION ALL <запрос>

Объединение результатов нескольких запросов может потребоваться при создании сложного запроса, учитывающего различные альтернативные условия.