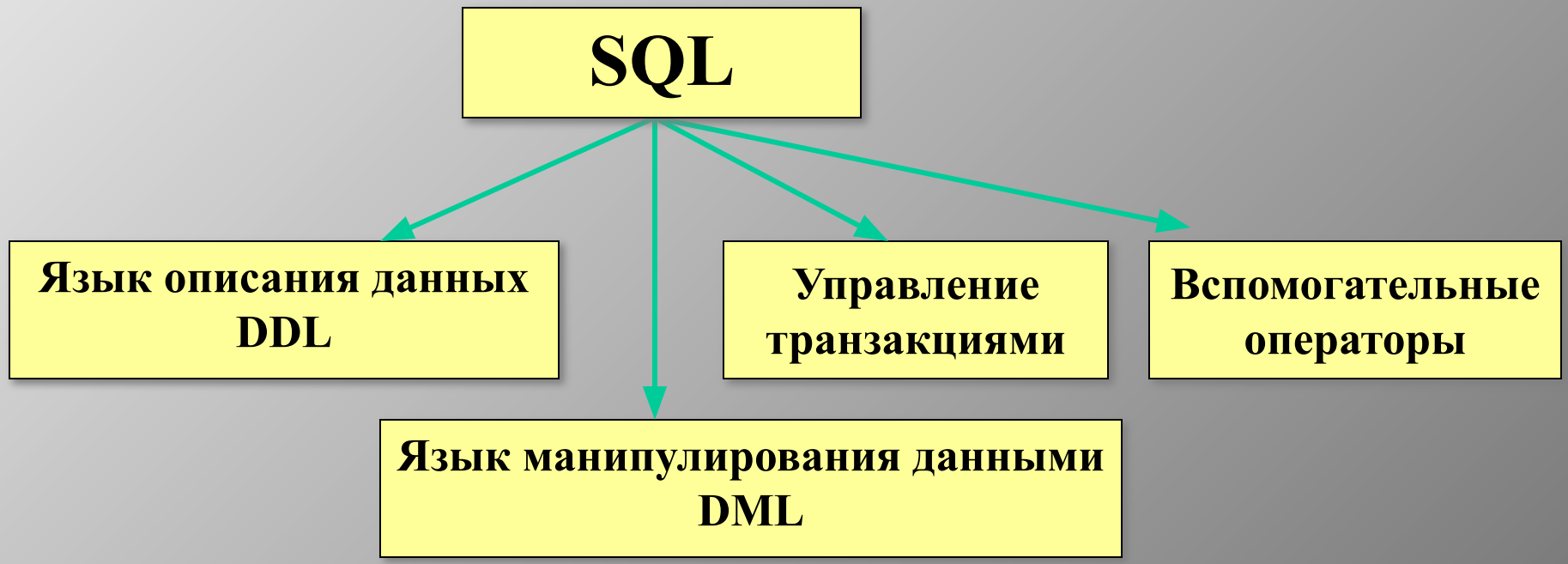
Тема 5.
Язык запросов SQL

Язык запросов SQL

Язык запросов SQL (Structured Query Language) был разработан корпорацией IBM в рамках исследовательского проекта по созданию реляционной СУБД System R (сначала язык назывался SEQUEL). После того как SQL поддержали в своих продуктах фирмы Oracle Corporation, Informix Corp. и Ingres Corp., он стал доминирующим языком для СУБД больших и средних вычислительных систем. Этот статус был подтвержден серией стандартов Американского национального института стандартов ANSI и Международной организации по стандартизации ISO (SQL-86, SQL-89 и SQL-92). Последний действующий стандарт опубликован как ANSI X3.135-1992 и ISO/IEC 9075:1992



Язык запросов SQL

create table *name* **(***column*, ...**)**

name – имя создаваемой таблицы; *column* – спецификация атрибута таблицы.

Спецификация каждого атрибута имеет следующий вид:

name type [*nullability*] [*default*] [*constraints*],

где *name* – имя атрибута; *type* – тип данных; *nullability* – допустимость неопределенных значений; *default* – значение по умолчанию; *constraints* – ограничения целостности.

Допустимость задания неопределенных значений атрибута таблицы задается ключевыми словами ***with null*** или ***not null***:

- ***with null*** – атрибут может иметь неопределенное значение. Это является умолчанием для всех типов данных;
- ***not null*** – атрибут не может иметь неопределенное значение ***null***.

Значение по умолчанию задается ключевыми словами ***with default*** или ***not default***:

- ***not default*** – значение атрибута при вводе обязательно;
- ***with default*** – если значение не задано, то для числовых типов данных и денежной единицы значение устанавливается равным нулю, а для символьных типов и даты – равным пустой строке;
- ***default*** *value* – если значение атрибута при вводе не задано, то оно устанавливается равным *value*.

Пример создания таблицы

```
* create table product (
*  number  integer2  primary key not null,
*  name    char(20)  unique not null,
*  status  integer1  check (status in(1, 2, 3)) not null default 1,
*  price   float4    check (price > 0) with null default null,
*  updated date      not null default 'today',
*  comment char(460) not null with default)\g
continue
*
```

Получение информации о таблице

```
* help table product\g

Name:                  product
Owner:                 ingres
Created:               25-feb-2010 11:48:28
Location:              ii_database
Type:                  user table
Version:               OI2.0
Page size:             2048
Cache priority:        0
Alter table version:   0
Alter table totwidth:  500
Row width:             500
Number of rows:        0
Storage structure:     heap
Compression:           none
Duplicate Rows:        allowed
Number of pages:       3
Overflow data pages:   0
Journaling:            enabled after the next checkpoint
Base table for view:   no
Permissions:           none
Integrities:           yes
Optimizer statistics:  none
```

Получение информации о таблице (продолжение)

```
Column Information:

                                                                  Key
Column Name                    Type       Length Nulls Defaults  Seq
number                         integer         2   no      no
name                           char           20   no      no
status                         integer         1   no   value
price                          float           4  yes    null
updated                        date                no   value
comment                        char          460   no     yes

Secondary indexes:
Index Name                     Structure  Keyed On
"$produ_u000000ee00000000"     btree      number
"$produ_u000000f000000000"     btree      name

continue
*
```

Ввод данных

insert into *table* [**(***column, ...* **)**] [***values(****expr*, ... **)**] | [*subselect*]

В данном операторе можно задавать либо список значений атрибутов *expr*, либо запрос *subselect*. Используя список значений атрибутов, можно вставить только одну запись за одно выполнение оператора ***insert.*** Если задан запрос *subselect*, то оператор ***insert*** вставляет все записи, полученные в результате выполнения запроса.

Пример ввода данных

```
* insert into product(number, name) values(10, 'Стойка 600x200');
* insert into product(number, name) values(60, 'Подставка');
* insert into product(number, name) values(30, 'Короб световой');
* insert into product(number, name) values(80, 'Стойка 1200x800');
* insert into product(number, name) values(140, 'Держатель малый');
* insert into product(number, name) values(50, 'Держатель средний');
* insert into product(number, name) values(90, 'Держатель большой');
* insert into product(number, name) values(130, 'Стойка 3 полки');
* insert into product(number, name) values(70, 'Полка с подсветкой')
* \g
(1 row)
(1 row)
(1 row)
(1 row)
(1 row)
(1 row)
(1 row)
(1 row)
(1 row)
continue
*
```

Результат ввода данных

```
* select number, name, status, price, updated from product\g
+------+--------------------+------+----------+--------------------+
|number|name                |status|price     |created             |
+------+--------------------+------+----------+--------------------+
|    10|Стойка 600x200      |     1|          |27.02.2010 10:41:25 |
|    60|Подставка           |     1|          |25.02.2010 10:41:25 |
|    30|Короб световой      |     1|          |25.02.2010 10:41:25 |
|    80|Стойка 1200x800     |     1|          |25.02.2010 10:41:25 |
|   140|Держатель малый     |     1|          |25.02.2010 10:41:25 |
|    50|Держатель средний   |     1|          |25.02.2010 10:41:25 |
|    90|Держатель большой   |     1|          |25.02.2010 10:41:25 |
|   130|Стеллаж с подсветкой|     1|          |25.02.2010 10:41:25 |
|    70|Полка с подсветкой  |     1|          |25.02.2010 10:41:25 |
+------+--------------------+------+----------+--------------------+
(9 rows)
continue
*
```

Контроль целостности данных

```
* insert into product (number, name) values (70, 'Стеллаж малый')\g
E_US1194 Duplicate key on INSERT detected.
    (Fri Feb 25 15:59:44 2010)
continue
* insert into product (number, name) values (157, 'Подставка')\g
E_US1194 Duplicate key on INSERT detected.
    (Fri Feb 25 16:00:18 2010)
continue
* insert into product values (40, 'Столик малый', 4, 700, '', '')\g
E_US1905 Check integrity constraint '"$produ_c0000010400000000"'
    on table '"product"' was violated.
    (Fri Feb 25 16:10:54 2010)
continue
* insert into product values (40, 'Столик малый', 1, 0, '', '')\g
E_US1905 Check integrity constraint '"$produ_c0000010600000000"'
    on table '"product"' was violated.
    (Fri Feb 25 16:11:03 2010)
continue
*
```

Управление транзакциями

Под транзакцией понимается последовательность операторов SQL, которая заканчивается операторами ***commit*** или ***rollback***, используемыми для управления транзакциями.

Оператор ***commit*** подтверждает выполнение транзакции. После выполнения этого оператора действия запросов транзакции нельзя отменить.

Оператор ***rollback*** отменяет все изменения в базе данных, произведенные с момента начала транзакции. Это позволяет управлять *транзакционной целостностью* базы данных, когда неполное выполнение последовательности операторов может привести к возникновению противоречивости данных. При выполнении операторов SQL можно проверить наличие ошибки и в этом случае выполнить “откат” транзакции с помощью оператора ***rollback***, как показано в приведенном ниже примере.

Оператор ***savepoint*** позволяет отметить промежуточные точки транзакции. В этом случае откат транзакции возможен до любой из промежуточных точек.

Пример управления транзакциями

```
* insert into product(number, name) values(99, 'Экран навесной')\g
(1 row)
continue
* select number, name from product where number = 99\g
+------+--------------------+
|number|name                |
+------+--------------------+
|    99|Экран навесной      |
+------+--------------------+
(1 row)
continue
* rollback\g
continue
* select number, name from product where number = 99\g
+------+--------------------+
|number|name                |
+------+--------------------+
+------+--------------------+
(0 rows)
continue
*
```

Пример управления транзакциями (продолжение)

```
* insert into product(number, name) values(99, 'Экран навесной')\g
(1 row)
continue
* savepoint p1\g
continue
* insert into product(number, name) values(100, 'Экран напольный')\g
(1 row)
* rollback to p1\g
continue
* select number, name from product where number in (99, 100)\g
+------+--------------------+
|number|name                |
+------+--------------------+
|    99|Экран навесной      |
+------+--------------------+
(1 rows)
continue
*
```

Изменение структуры хранения

modify *name* ***to*** *type* [***unique***] ***on*** *column*, ... [*with*]

name – имя таблицы, *type* – тип структуры хранения (***heap***, ***hash***, ***isam*** или ***btree***), *with* состоит из ключевого слова ***with*** и следующих за ним параметров: ***fillfactor = n*** и др.

Опция ***on*** задает первичный ключ таблицы, который может состоять из одного или нескольких атрибутов *column*. Если значения ключа должны быть уникальными, то необходимо указать опцию ***unique***.

Параметр ***fillfactor*** задает коэффициент заполнения страницы в процентах.

По умолчанию параметр ***fillfactor*** равен для ***hash*** 50 %, для ***isam*** – 80 %, для ***btree*** – 80 %.

Пример таблицы в структуре HEAP

```
* select number, name, tid, tid/512 as page from product\g
+------+--------------------+-------------+-------------+
|number|name                |tid          |page         |
+------+--------------------+-------------+-------------+
|    10|Стойка 600x200      |            0|            0|
|    60|Подставка           |            1|            0|
|    30|Короб световой      |            2|            0|
|    80|Стойка 1200x800     |            3|            0|
|   140|Держатель малый     |          512|            1|
|    50|Держатель средний   |          513|            1|
|    90|Держатель большой   |          514|            1|
|   130|Стеллаж с подсветкой|          515|            1|
|    70|Полка с подсветкой  |         1024|            2|
+------+--------------------+-------------+-------------+
(9 rows)
continue
*
```

Пример таблицы в структуре ISAM

```
* modify product to isam unique on number with fillfactor = 100\g
(9 rows)
continue
* select number, name, tid, tid/512 as page from product\g
+------+--------------------+-------------+-------------+
|number|name                |tid          |page         |
+------+--------------------+-------------+-------------+
|    10|Стойка 600x200      |            0|            0|
|    30|Короб световой      |            1|            0|
|    50|Держатель средний   |            2|            0|
|    60|Подставка           |            3|            0|
|    70|Полка с подсветкой  |          512|            1|
|    80|Стойка 1200x800     |          513|            1|
|    90|Держатель большой   |          514|            1|
|   130|Стеллаж с подсветкой|          515|            1|
|   140|Держатель малый     |         1024|            2|
+------+--------------------+-------------+-------------+
(9 rows)
continue
*
```

Пример таблицы в структуре ISAM, fillfactor 50%

```
* modify product to isam unique on number with fillfactor = 50\g
(9 rows)
continue
* select number, name, tid, tid/512 as page from product\g
+------+--------------------+-------------+-------------+
|number|name                |tid          |page         |
+------+--------------------+-------------+-------------+
|    10|Стойка 600x200      |            0|            0|
|    30|Короб световой      |            1|            0|
|    50|Держатель средний   |          512|            1|
|    60|Подставка           |          513|            1|
|    70|Полка с подсветкой  |         1024|            2|
|    80|Стойка 1200x800     |         1025|            2|
|    90|Держатель большой   |         1536|            3|
|   130|Стеллаж с подсветкой|         1537|            3|
|   140|Держатель малый     |         2048|            4|
+------+--------------------+-------------+-------------+
(9 rows)
continue
*
```

Пример таблицы в структуре HASH

```
* modify product to hash unique on number with fillfactor = 100\g
(9 rows)
continue
* select number, name, tid, tid/512 as page from product\g
+------+--------------------+-------------+-------------+
|number|name                |tid          |page         |
+------+--------------------+-------------+-------------+
|    50|Держатель средний   |          512|            1|
|    70|Полка с подсветкой  |         1536|            3|
|    80|Стойка 1200x800     |         2048|            4|
|   140|Держатель малый     |         2560|            5|
|    10|Стойка 600x200      |         5120|           10|
|   130|Стеллаж с подсветкой|         5632|           11|
|    90|Держатель большой   |         7168|           14|
|    30|Короб световой      |         7680|           15|
|    60|Подставка           |         7681|           15|
+------+--------------------+-------------+-------------+
(9 rows)
continue
*
```

Пример таблицы в структуре BTREE

```
* modify product to btree unique on number with fillfactor = 100\g
(9 rows)
continue
* select number, name, tid, tid/512 as page from product\g
+------+--------------------+-------------+-------------+
|number|name                |tid          |page         |
+------+--------------------+-------------+-------------+
|    10|Стойка 600x200      |         1024|            2|
|    30|Короб световой      |         1025|            2|
|    50|Держатель средний   |         1026|            2|
|    60|Подставка           |         1027|            2|
|    70|Полка с подсветкой  |         1536|            3|
|    80|Стойка 1200x800     |         1537|            3|
|    90|Держатель большой   |         1538|            3|
|   130|Стеллаж с подсветкой|         1539|            3|
|   140|Держатель малый     |         2048|            4|
+------+--------------------+-------------+-------------+
(9 rows)
continue
*
```

Создание вторичных индексов

create index *name* ***on*** *table* **(***column, ...***)** [***with structure*** **=** *type*]

name – имя индекса; *table* – имя таблицы, для которой создается индекс;
column – один или несколько атрибутов, по которым строится индекс;
type – тип структуры хранения для индекса (**isam**, **hash** или **btree**,
по умолчанию – **isam**).

Пример создания вторичного индекса

```
* create index product_name on product(name)\g
(9 rows)
continue
* select * from product_name\g
+--------------------+-------------+
|name                |tidp         |
+--------------------+-------------+
|Держатель большой   |         1538|
|Держатель малый     |         2048|
|Держатель средний   |         1026|
|Короб световой      |         1025|
|Подставка           |         1027|
|Полка с подсветкой  |         1536|
|Стеллаж с подсветкой|         1539|
|Стойка 1200x800     |         1537|
|Стойка 600x200      |         1024|
+--------------------+-------------+
(9 rows)
continue
*
```

Изменение данных

update *table* [***from*** *table, ...*] ***set*** *column = expr, ...* [***where*** *condition*]

Оператор ***update*** изменяет значения заданных в ***set*** атрибутов таблицы *table* для всех записей, которые соответствуют условию поиска *condition*. Выражение *expr* в опции ***set*** может быть сконструировано с использованием констант и значений атрибутов из обновляемой таблицы или таблиц, указанных в опции ***from***.

Пример изменения данных

```
* update product set status = 2, price = 120, updated = 'now'
* where number = 60\g
(1 row)
continue
* select number, name, status, price, updated from product\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     1|           |25.02.2010 10:41:25 |
|    30|Короб световой      |     1|           |25.02.2010 10:41:25 |
|    50|Держатель средний   |     1|           |25.02.2010 10:41:25 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|    70|Полка с подсветкой  |     1|           |25.02.2010 10:41:25 |
|    80|Стойка 1200x800     |     1|           |25.02.2010 10:41:25 |
|    90|Держатель большой   |     1|           |25.02.2010 10:41:25 |
|   130|Стеллаж с подсветкой|     1|           |25.02.2010 10:41:25 |
|   140|Держатель малый     |     1|           |25.02.2010 10:41:25 |
+------+--------------------+------+-----------+--------------------+
(9 rows)
continue
*
```

Удаление данных

delete from *table* [***where*** *condition*]

Оператор ***delete*** удаляет из таблицы *table* все записи, которые соответствуют заданному условию поиска *condition*.

Пример удаления данных

```
* delete from product where number = 90\g
(1 row)
continue
* select number, name, status, price, updated from product\g
+------+--------------------+------+----------+-------------------+
|number|name                |status|price     |updated            |
+------+--------------------+------+----------+-------------------+
|    10|Стойка 600x200      |     2|  1500.000|27.02.2010 12:08:26|
|    30|Короб световой      |     1|          |25.02.2010 10:41:25|
|    50|Держатель средний   |     1|          |25.02.2010 10:41:25|
|    60|Подставка           |     2|   120.000|27.02.2010 11:36:47|
|    70|Полка с подсветкой  |     1|          |25.02.2010 10:41:25|
|    80|Стойка 1200x800     |     1|          |25.02.2010 10:41:25|
|   130|Стойка 3 полки      |     2|  2300.000|27.02.2010 12:08:26|
|   140|Держатель малый     |     3|   200.000|27.02.2010 12:08:26|
+------+--------------------+------+----------+-------------------+
(8 rows)
continue
*
```

Процедуры

create procedure *name* [**(***param* [**=**] *type* [*nullability*] [*default*]**, …)**] ***as*** [***declare*** *declare_section*] ***begin*** *statement*; … ***end***.

Процедура идентифицируется уникальным именем *name* и может получать набор входных параметров *param*. Для каждого параметра задается тип данных *type*, а также могут быть заданы допустимость неопределенных значений *nullability* и значение по умолчанию *default*. В секции ***declare*** могут описываться локальные переменные. Тело процедуры состоит из набора операций *statement*, который может содержать операторы ***commit***, ***delete***, ***endloop***, ***execute procedure***, ***if then***, ***else***, ***insert***, ***return***, ***rollback***, ***select***, ***update***, ***while***, а также выражения присваивания ‘**=**’ или ‘**:=**’. В процедуре оператор ***select*** может возвращать только одну запись и присваивать возвращаемые значения локальным переменным.
Оператор вызыва процедуры имеет синтаксис

execute procedure *name* [**(***param* **=** *value*, …**)**] [***into*** *status*]

name – имя процедуры, *param* – имя параметра, *value* – его значение.
Опция **into** задает целочисленную переменную, которой присваивается значение, возвращаемое оператором **return** из процедуры.

Пример процедуры

```
* create procedure set_updating_date (number integer2 not null) as
* begin
*    update product set updated = 'now' where number = :number;
* end\g
continue
*
```

Правила

create rule *name* ***after insert*** | ***update***[**(***column*, …**)**] | ***delete***
[***on*** | ***of*** | ***from*** | ***into***] *table*
[***referencing*** [***old as*** *old_name*] [***new as*** *new_name*]]
[***where*** *condition*]
execute procedure *procedure* [**(***param* = *value*, …**)**],

name – имя правила; *column* – атрибут таблицы, изменение которого приводит к срабатыванию правила; *table* – таблица, для которой создается правило.

Список параметров *param* позволяет передавать значения *value* в вызываемую процедуру *procedure*. Опция ***referencing*** позволяет переназначать ключевые слова, задающие, какое – новое или старое – значение записи таблицы передается через параметры в процедуру. По умолчанию этими ключевыми словами являются ***new*** и ***old***. В опции ***where*** задается условие *condition* вызова процедуры.

Пример правила

```
* create rule on_updating_product after
* update(number, name, status, price, comment) on product
* execute procedure set_updating_date(number = new.number)\g
continue
*
```

Результат совместной работы правила и процедуры

```
* update product set status = 2, price = 1500 where number = 10;
* update product set status = 2, price = 2300
* where name = 'Стойка 3 полки';
* update product set status = 3, price = 200 where number = 140\g
continue
* select number, name, status, price, updated from product\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     2|   1500.000|27.02.2012 12:08:26 |
|    30|Короб световой      |     1|           |25.02.2010 10:41:25 |
|    50|Держатель средний   |     1|           |25.02.2010 10:41:25 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|    70|Полка с подсветкой  |     1|           |25.02.2010 10:41:25 |
|    80|Стойка 1200x800     |     1|           |25.02.2010 10:41:25 |
|    90|Держатель большой   |     1|           |25.02.2010 10:41:25 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2012 12:08:26 |
|   140|Держатель малый     |     3|    200.000|27.02.2012 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(8 rows)
continue
*
```

Получение данных

select [***all*** |***distinct***] [***first*** *n*] * | *expression* [*as column*], …
[***from*** *table* [*alias*], *…*
[***where*** *condition*]
[***group by*** *column*, *…*]
[***having*** *condition*]
[***union*** [***all***] (*select*)]
[***order by*** *column* [***asc*** | ***desc***], *…*]

В целевом списке *expression* указываются атрибуты, которые должны быть возвращены оператором ***select***. Если в целевом списке указывается звездочка, то возвращаются все атрибуты соответствующей таблицы. Возвращаемые столбцы могут содержать значения, считываемые из столбцов таблиц базы данных или вычисляемые во время запроса. Опция ***first*** задает максимальное число записей, возвращаемых по итогам запроса. Опция ***distinct*** позволяет удалить повторяющиеся строки из результата запроса.

Получение данных (продолжение)

В опции ***from*** указывается список таблиц, участвующих в запросе. Каждой таблице в этом списке можно назначить короткое имя *alias*, которое используется для более компактного написания текста запроса.

Опция ***where*** задает, что в результаты запроса следует включать только строки, удовлетворяющие условию поиска *condition*.

Опция ***group by*** позволяет создавать итоговый запрос. Обычный запрос включает в результаты запроса по одной строке для каждой строки из базы данных. Итоговый запрос вначале группирует строки базы данных таким образом, чтобы в каждой группе они имели одинаковые значения во всех столбцах, по которым осуществляется группирование, а затем включает в результаты запроса одну итоговую строку для каждой группы.

Опция ***having*** означает, что в результат запроса следует включать только группы, удовлетворяющие условию поиска *condition*.

Опция ***union*** позволяет объединить результаты нескольких запросов, при этом опция указывает, что из результатов запроса не должны удаляться дублирующие записи ***all***.

Получение данных (продолжение)

Опция ***order by*** задает сортировку результатов запроса на основании данных, содержащихся в столбцах *column*. Опции ***asc*** и ***desc*** задают сортировку по возрастанию или по убыванию соответственно.

В целевом списке запроса могут задаваться агрегативные функции:
count(*expr*) – количество значений в столбце;
sum(*expr*) – сумма всех значений столбца;
avg(*expr*) – среднее всех значений, содержащихся в столбце;
max(*expr*) – максимальное значение столбца;
min(*expr*) – минимальное значение столбца.

Агрегативные функции вычисляются для всех записей таблицы или для каждой группы, если в запросе задано группирование данных с помощью опции ***group by***.

Для работы с неопределенными значениями поддерживается функция ***ifnull***(*expr1, expr2*), вычисляющая выражение *expr1*, и если оно равно ***null***, возвращающая *expr2*.

Получение данных (продолжение)

В условии поиска допустимы следующие операторы:

- операторы сравнения =, <>, >, >=, <, <=;
- ***like*** – оператор сравнения по образцу, который имеет синтаксис *column* [***not***] ***like*** *pattern*, где *pattern* – строка символов, в которой могут быть подстановочные знаки “%” и “_”. Знак “%” совпадает с любой последовательностью из нуля и более символов. Знак “_” совпадает с любым отдельным символом;
- ***between*** – оператор проверки на принадлежность диапазону значений, который имеет синтаксис *y* [***not***] ***between*** *x* ***and*** *y*;
- ***in*** – оператор проверки на членство в множестве, который имеет синтаксис *y* [***not***] ***in*** **(***x, …, z***)** или *expression* [***not***] ***in*** **(***subquery***)**;
- ***exists*** – оператор проверки на принадлежность запросу, который имеет синтаксис [***not***] ***exists*****(***subquery***)**;
- ***is null*** – оператор проверки на неопределенное значение, который имеет синтаксис ***is*** [***not***] ***null***;
- ***any*** – оператор проверки, который имеет синтаксис *operator any*(*subselect*), где *operator* – один из операторов сравнения. Проверяемое значение поочередно сравнивается с каждым значением, возвращаемым вложенным запросом. Если любое из этих сравнений дает результат *true*, то проверка ***any*** возвращает результат *true*;

Получение данных (продолжение)

- ***all*** – оператор проверки, который имеет синтаксис *operator* ***all*****(***subselect***)**, где *operator* – один из операторов сравнения. Проверяемое значение поочередно сравнивается с каждым значением, возвращаемым вложенным запросом. Если все сравнения дают результат *true*, то проверка ***all*** возвращает результат *true*.

Условия поиска можно объединять с помощью логических операторов ***and***, ***or***, ***not***. Приоритет логических операторов можно изменять с помощью круглых скобок. По умолчани наивысший приоритет имеет оператор ***not***, затем следует ***and***, низший приоритет у оператора ***or***.

Запрос с проверкой неопределенного значения

```
* select number, name, status, price, updated from product
* where price is not null\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
|   140|Держатель малый     |     3|    200.000|27.02.2010 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(3 rows)
continue
*
```

Запрос с использованием оператора *like*

```
* select number, name, status, price, updated from product
* where name like 'Стойка%'\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
|    80|Стойка 1200x800     |     1|           |25.02.2010 10:41:25 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(3 rows)
continue
*
```

Запрос с использованием оператора *between*

```
* select number, name, status, price, updated from product
* where price between 1000 and 10000\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(2 rows)
continue
*
```

Запрос с использованием оператора *in*

```
* select number, name, status, price, updated from product
* where status in (2, 3)\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
|   140|Держатель малый     |     3|    200.000|27.02.2010 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(4 rows)
continue
*
```

Запрос с сортировкой по возрастанию

```
* select number, name, status, price, updated from product
* order by name\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|   140|Держатель малый     |     3|    200.000|27.02.2010 12:08:26 |
|    50|Держатель средний   |     1|           |25.02.2010 10:41:25 |
|    30|Короб световой      |     1|           |25.02.2010 10:41:25 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|    70|Полка с подсветкой  |     1|           |25.02.2010 10:41:25 |
|    80|Стойка 1200x800     |     1|           |25.02.2010 10:41:25 |
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
+------+--------------------+------+-----------+--------------------+
(8 rows)
continue
*
```

Запрос с сортировкой по убыванию

```
* select number, name, status, price, updated from product
* order by updated desc\g
+------+--------------------+------+-----------+--------------------+
|number|name                |status|price      |updated             |
+------+--------------------+------+-----------+--------------------+
|   130|Стойка 3 полки      |     2|   2300.000|27.02.2010 12:08:26 |
|    10|Стойка 600x200      |     2|   1500.000|27.02.2010 12:08:26 |
|   140|Держатель малый     |     3|    200.000|27.02.2010 12:08:26 |
|    60|Подставка           |     2|    120.000|27.02.2010 11:36:47 |
|    50|Держатель средний   |     1|           |25.02.2010 10:41:25 |
|    80|Стойка 1200x800     |     1|           |25.02.2010 10:41:25 |
|    30|Короб световой      |     1|           |25.02.2010 10:41:25 |
|    70|Полка с подсветкой  |     1|           |25.02.2010 10:41:25 |
+------+--------------------+------+-----------+--------------------+
(8 rows)
continue
*
```

Запрос с агрегативной функцией *count*

```
* select count(price) as num_rows from product\g
+-------------+
|num_rows     |
+-------------+
|            4|
+-------------+
(1 row)
continue
* select count(*) as num_rows from product\g
+-------------+
|num_rows     |
+-------------+
|            8|
+-------------+
(1 row)
continue
*
```

Запрос с группированием

```
* select status, count(*) as num_rows from product group by status\g
+------+-------------+
|status|num_rows     |
+------+-------------+
|     1|            4|
|     2|            3|
|     3|            1|
+------+-------------+
(3 rows)
continue
*
```

Запрос с группированием

```
* select status, max(price) as num_rows from product
* group by status\g
+------+-----------+
|status|num_rows   |
+------+-----------+
|     1|           |
|     2|   2300.000|
|     3|    200.000|
+------+-----------+
(3 rows)
continue
*
```

Запрос с группированием и постусловием

```
* select status, max(price) as num_rows from product
* group by status having max(price) is not null\g
+------+-----------+
|status|num_rows   |
+------+-----------+
|     2|   2300.000|
|     3|    200.000|
+------+-----------+
(2 rows)
continue
*
```

Создание двух таблиц для многотабличных запросов

```
* create table client (
*   id      integer2 primary key not null,
*   name    char(10) unique not null,
*   address char(20) not null with default)\g
continue
* create table orders (
*   number   integer2 primary key not null,
*   client   integer2 not null,
*   date     date     not null,
*   product  integer2 not null,
*   quantity integer2 not null)\g
continue
*
```

Включение данных в таблицы

```
* create table client (
* insert into client values (1, 'ООО Круиз', 'СПб., ул. Герцена 12');
* insert into client values (2, 'ТОО Сигма', 'Москва, ул. Мира 8');
* insert into client values (3, 'ЗАО Вега', 'СПб., пр. Медиков 7');
* insert into client values (4, 'ЗАО Рубин', 'СПб., пр. Науки 17')\g
continue
* insert into orders values (1, 2, '20.03.2010', 60, 200);
* insert into orders values (2, 1, '22.03.2010', 130, 40);
* insert into orders values (3, 1, '22.03.2010', 10, 60);
* insert into orders values (4, 2, '25.03.2010', 140, 132);
* insert into orders values (5, 1, '28.03.2010', 60, 80);
* insert into orders values (6, 3, '30.03.2010', 130, 15);
* insert into orders values (7, 2, '30.03.2010', 10, 24);
* insert into orders values (8, 3, '01.04.2010', 60, 50)\g
continue
* commit\g
continue
*
```

Запрос на получение данных с вложенным запросом

```
* select * from client where id in (select client from orders)\g
+------+-----------+--------------------+
|id    |name       |address             |
+------+-----------+--------------------+
|     1|ООО "Круиз"|СПб., ул. Герцена 12|
|     2|ТОО "Сигма"|Москва, ул. Мира 8  |
|     3|ЗАО "Вега" |СПб., пр. Медиков 7 |
+------+-----------+--------------------+
(3 rows)
continue
*
```

Запрос с соединением таблиц

```
* select o.number, o.date, c.name as client, p.name as product,
* o.quantity, o.quantity*p.price as total
* from orders o join client c on c.id = o.client
* join product p on p.number = o.product
* where date_trunc('month', o.date) = '01.03.2010' order by o.date\g
Executing . . .
+------+----------+----------+----------------+------+-----------+
|number|date      |client    |product         |quanti|total      |
+------+----------+----------+----------------+------+-----------+
|     1|20.03.2010|ТОО "Сигма"|Подставка       |   200|  24000.000|
|     2|22.03.2010|ООО "Круиз"|Стойка 3 полки  |    40|  92000.000|
|     3|22.03.2010|ООО "Круиз"|Стойка 600x200  |    60|  90000.000|
|     4|25.03.2010|ТОО "Сигма"|Держатель малый |   132|  26400.000|
|     5|28.03.2010|ООО "Круиз"|Подставка       |    80|   9600.000|
|     7|30.03.2010|ТОО "Сигма"|Стойка 600x200  |    24|  36000.000|
|     6|30.03.2010|ЗАО "Вега" |Стойка 3 полки  |    15|  34500.000|
+------+----------+----------+----------------+------+-----------+
(7 rows)
continue
*
```

Запрос с внешним соединением

```
* select c.id, c.name as client, sum(o.quantity*p.price) as total
* from orders o right join client c on c.id = o.client
* left join product p on p.number = o.product
* group by c.id, c.name order by total desc\g
+------+----------+----------+
|id    |client    |total     |
+------+----------+----------+
|     4|ЗАО "Рубин"|          |
|     1|ООО "Круиз"| 191600.000|
|     2|ТОО "Сигма"|  86400.000|
|     3|ЗАО "Вега" |  40500.000|
+------+----------+----------+
(4 rows)
continue
*
```

Представления

create view *name* **as** *select* [***with check*** *option*]

name – имя представления.
В опции ***as*** задается оператор ***select***, определяющий содержание “виртуальной таблицы”. Обновление представления разрешено, если представление имеет одну исходную таблицу, его колонки не ссылаются на агрегативные функции и в операторе ***select*** не используются опции ***distinct***, ***group by***, ***union*** и ***having***. Опция ***with check*** устанавливает режим контроля, при котором СУБД автоматически проверяет каждую операцию ***insert*** или ***update***, выполняемую над представлением, чтобы удостовериться, что полученные в результате строки удовлетворяют условиям поиска в определении представления. Если добавляемая или обновляемая строка не удовлетворяет этим условиям, то выполнение оператора ***insert*** или ***update*** завершается ошибкой.

Пример создания представления

```
* create view sales as select o.number, o.date, c.name as client,
* p.name as product, o.quantity, o.quantity*p.price as total
* from orders o join client c on c.id = o.client
* join product p on p.number = o.product\g
continue
*
```

Запрос с использованием представления

```
* select * from sales\g
+------+----------+----------+----------------+------+----------+
|number|date      |client    |product         |quanti|total     |
+------+----------+----------+----------------+------+----------+
|     3|22.03.2010|ООО "Круиз"|Стойка 600x200  |    60| 90000.000|
|     7|30.03.2010|ТОО "Сигма"|Стойка 600x200  |    24| 36000.000|
|     5|28.03.2010|ООО "Круиз"|Подставка       |    80|  9600.000|
|     1|20.03.2010|ТОО "Сигма"|Подставка       |   200| 24000.000|
|     8|01.04.2010|ЗАО "Вега" |Подставка       |    50|  6000.000|
|     6|30.03.2010|ЗАО "Вега" |Стойка 3 полки  |    15| 34500.000|
|     2|22.03.2010|ООО "Круиз"|Стойка 3 полки  |    40| 92000.000|
|     4|25.03.2010|ТОО "Сигма"|Держатель малый |   132| 26400.000|
+------+----------+----------+----------------+------+----------+
(8 rows)
continue
*
```

Запрос с соединением представления и таблицы

```
* select s.date, s.client, c.address, s.product, s.quantity
* from sales s join client c on c.name = s.client
* where s.date between '28.03.2010' and '01.04.2010'\g
+-----------+-----------+--------------------+---------------+------+
|date       |client     |address             |product        |quanti|
+-----------+-----------+--------------------+---------------+------+
|30.03.2010 |ЗАО "Вега" |СПб., пр. Медиков 7 |Стойка 3 полки |    15|
|01.04.2010 |ЗАО "Вега" |СПб., пр. Медиков 7 |Подставка      |    50|
|28.03.2010 |ООО "Круиз"|СПб., ул. Герцена 12|Подставка      |    80|
|30.03.2010 |ТОО "Сигма"|Москва, ул. Мира 8  |Стойка 600x200 |    24|
+-----------+-----------+--------------------+---------------+------+
(4 rows)
continue
*
```

Запрос с представлением и агрегативными функциями

```
* select count(number) as orders, max(quantity) as max_lot,
* sum(total) as total from sales\g
+-------------+------+-----------+
|orders       |max_lo|total      |
+-------------+------+-----------+
|            8|   200| 318500.000|
+-------------+------+-----------+
(1 row)
continue
*
```

Ограничения целостности

create integrity on *table* ***is*** *condition*

Параметр *table* определяет таблицу, для которой задается контроль целостности. Параметр *condition* определяет условия контроля целостности. Заданные условия вычисляются для всех обновляемых записей таблицы. Если добавляемая или обновляемая в запросе строка не удовлетворяет этим условиям, то выполнение оператора ***insert*** или ***update*** завершается ошибкой.

Пример ограничения целостности

```
* create integrity on orders is quantity > 0\g
continue
*
*
*
*
* insert into orders values (7, 2, '20.03.2010', 80, 0)\g
Executing . . .
 (0 rows)
continue
* update orders set quantity = -10 where number = 1\g
Executing . . .
 (0 rows)
continue
*
```

Последовательности

create sequence *name* [*options*]

Параметр *name* задает уникальное имя последовательности, а *options* – набор опций, управляющих последовательностью. Среди опций применяются:

- ***start with*** *number* – задает начальное значение для последовательности;
- ***increment by*** *number* – определяет инкремент (декремент);
- ***maxvalue*** *number* – задает максимальное значение;
- ***minvalue*** *number* – задает минимальное значение;
- ***cycle*** – разрешение цикличности;
- ***nocycle*** – запрещение цикличности.

Для доступа к значениям последовательности применяется оператор ***nextval***, который возвращает следующее значение и инкрементирует последовательность в соответствии с заданными параметрами.

Пример использования последовательности

```
* create sequence numbers start with 1 increment by 1\g
continue
*
*
*
* select numbers.nextval as number, date, client, product, quantity
* from sales where date between '28.03.2010' and '01.04.2010'\g
+------+----------+----------+--------------+------+
|number|date      |client    |product       |quanti|
+------+----------+----------+--------------+------+
|     1|30.03.2010|ЗАО "Вега"|Стойка 3 полки|    15|
|     2|01.04.2010|ЗАО "Вега"|Подставка     |    50|
|     3|28.03.2010|ООО "Круиз"|Подставка    |    80|
|     4|30.03.2010|ТОО "Сигма"|Стойка 600x200|   24|
+------+----------+----------+--------------+------+
(4 rows)
continue
*
```

Использование последовательности при вводе данных

```
* create sequence num_ord start with 1 increment by 1
* maxvalue 32767 nocycle\g
continue
*
*
* insert into orders values (num_ord.nextval, 2, 'today', 20, 100);
* insert into orders values (num_ord.nextval, 1, 'today', 80, 200);
* insert into orders values (num_ord.nextval, 4, 'today', 60, 120);
* insert into orders values (num_ord.nextval, 5, 'today', 10, 400)\g
(1 row)
(1 row)
(1 row)
(1 row)
continue
*
```

Удаление элементов логической схемы

Для удаления созданных элементов логической схемы данных в языке SQL применяется оператор ***drop***:

drop table *name*
drop index *name*
drop procedure *name*
drop rule *name*
drop view *name*
drop sequence *name*
drop integrity on *name* ***all*** | *number1, number2, ...*

name – имя удаляемого объекта.