

Лабораторная работа №4

Многотабличные запросы и подзапросы в среде PostgreSQL

Цель работы: Изучить порядок формирования многотабличных запросов и подзапросов в среде PostgreSQL. Приобрести навыки работы с многотабличными запросами и подзапросами в среде PostgreSQL.

Задание. Общая часть: Ознакомиться с теоретическими сведениями о возможностях создания многотабличных запросов и подзапросов в среде PostgreSQL. Выполнить выборку из созданных таблиц согласно варианту. Если в базе нет данных, удовлетворяющих условиям запроса, то добавить подходящие данные. Продемонстрировать результаты работы.

Вариант 1: Найти всех нападающих с правым хватом клюшки и в возрасте больше, чем средний возраст нападающих с левым хватом клюшки. Найти всех защитников, набравших более 5 очков и с ростом больше, чем средний рост защитников в возрасте до 25 лет. Найти всех полузащитников в возрасте от 20 до 30 лет и с окладом больше, чем средний оклад нападающих из России.

Вариант 2: Найти все бакалейные товары и с ценой больше, чем средняя цена бакалейных товаров, поставляемых из Москвы. Найти все молочные продукты, поставляемых из Омска и Томска и с ценой больше, чем средняя цена товаров, поставленных за последние три месяца. Найти все товары, поставленные за последние полгода, и с ценой больше, чем средняя цена товаров, поставляемых из России и Индии.

Вариант 3: Найти всех водителей в возрасте до 30 лет и имеющих заработок больше, чем средний заработок водителей, получивших права от 18 до 20 лет. Найти всех водителей, выполнившие заказы за последние три месяца на сумму более 20000 рублей и в возрасте больше, чем средний

возраст водителей, не состоящих в браке. Найти всех водителей в возрасте от 30 до 50 лет, родившихся в Новосибирске, и имеющих заработок больше, чем средний заработок водителей, приехавших из Омска и Томска.

Вариант 4: Найти все повести и романы, в которых от 250 до 500 страниц, и с ценой больше, чем средняя цена книг заданного издательства. Найти всю литературу автора Иванова, выпущенную в Москве и Новосибирске, и с ценой выше, чем средняя цена произведений жанра «детектив», выпущенных за последние полгода, автором Ивановым. Найти все поэтические произведения заданного издательства и с ценой больше, чем средняя цена книг, выпущенных в Чехии и Словакии.

Вариант 5: Найти все автомобили с правым рулем и автоматической коробкой передач, чья стоимость больше, чем средняя стоимость машин малолитражек из Японии. Найти все седаны и минивены, выпущенные в Германии и США, и с ценой выше, чем средняя стоимость машин Японского производства, выпущенных за последние три года. Найти все машины с левым рулем, выпущенные в Японии и Франции, и с ценой больше, чем средняя цена машин с механической коробкой передач, выпущенных в России.

Вариант 6: Найти все фрукты с датой поставки за последние полгода и с ценой больше, чем средняя цена фруктов поставщика Иванова. Найти все овощи, вес которых на складе от 150 до 350 кг, и чья стоимость больше, чем средняя стоимость овощей, поступивших за последние две недели. Найти все арбузы из Узбекистана и Таджикистана и с ценой больше, чем средняя цена овощей из Омска и Томска.

Вариант 7: Найти все журналы и еженедельники, заказ которых составляет в пределах от 200 до 500 экземпляров, и с ценой больше, чем средняя цена продукции, проданной за наличный расчет. Найти всю продукцию, произведенную для клиентов из Омска и Томска, оплаченную безналичным путем и стоимость которой больше, чем средняя стоимость продукции, произведенной за последние полгода. Найти все газеты и

бюллетени заданного издательства и имеющих цену больше, чем средняя цена журналов, выпущенных в Омске и Томске.

Вариант 8: Найти всех жильцов, которые только частично оплатили ремонт, и у которых были выполнены ремонтные работы на сумму больше, чем средняя стоимость всех заказов на ремонтные работы, выполненные за последний месяц. Найти всех мастеров с опытом работы от 5 до 10 лет, и которые выполнили заказы на ремонтные работы на сумму больше, чем средняя стоимость всех заказов, выполненных мастерами из Москвы и Санкт-Петербурга. Найти всех мастеров, которые занимаются ремонтными работами вида «ремонт крыши» и «сантехнические работы» и выполнили заказов на сумму больше, чем средняя стоимость заказов, выполненных за последний месяц мастерами со стажем работы от 3 до 5 лет.

Вариант 9: Найти всех мастеров, родившихся в Омске и Томске, которые занимаются ремонтом техники вида «микроволновка» и «кофеварка» и выполнившие заказы на сумму больше, чем средняя стоимость заказов, выполненных мастерами в возрасте старше 45 лет. Найти мастеров со стажем работы от 10 до 20 лет, и выполнивших заказы на сумму больше, чем средняя стоимость заказов, выполненных за последние три месяца мастерами из Москвы и Санкт-Петербурга. Найти всех мастеров, которые занимаются ремонтом техники вида «фен» и «мультиварка» и выполнившие заказы на сумму больше, чем средняя стоимость заказов, выполненных мастерами со стажем работы от 2 до 5 лет.

Вариант 10: Найти всех руководителей туров, которые обслуживают только автобусные туры и выполнившие заказы на туры со стоимостью больше, чем средняя стоимость заказов на все туры, выполненные за последние три месяца. Найти все туры, выполненные за последний месяц, и со стоимостью больше, чем средняя стоимость железнодорожных туров, выполненных за последние две недели. Найти все авиатуры в страны Турция и Египет, предлагаемые для семейного отдыха и чья стоимость больше, чем средняя стоимость туров, позиционируемых как туры-шопинги.

Содержание отчета

Отчет должен содержать титульный лист, цель работы, задание, коды команд на каждом этапе выполнения работы, результаты выполнения команд (скриншоты), выводы и анализ результатов работы.

Выборка данных с использованием языка запросов SQL

Центральное место в SQL занимает команда *SELECT*, предназначенная для построения запросов и выборки данных из таблиц и представлений. Данные, возвращаемые в результате запроса, называются итоговым набором; как и таблицы, они состоят из записей и полей.

Данные итогового набора не хранятся на диске в какой-либо постоянной форме. Итоговый набор является лишь временным представлением данных, полученных в результате запроса. Структура полей итогового набора может соответствовать структуре исходной таблицы, но может и радикально отличаться от нее. Итоговые наборы даже могут содержать поля, выбранные из других таблиц.

Из-за своей особой роли в PostgreSQL команда *SELECT* также является самой сложной командой, обладающей многочисленными секциями и параметрами. Ниже приведено общее определение синтаксиса *SELECT*.

```
SELECT [ALL / DISTINCT [ON (выражение [...])]]  
цель [AS имя] [...] [FROM источник [...]]  
[[NATURAL] тип_объединения источник  
[ON условие | USING (список_полей)]] [...]  
[WHERE условие] [GROUP BY критерий [...]]  
[HAVING условие [...]]  
[{UNION | INTERSECT | EXCEPT} [ALL] подзапрос]  
[ORDER BY выражение [ASC / DESC / USING оператор] [...]]  
[FOR UPDATE [OF таблица [...]]]  
[LIMIT {число / ALL} [OFFSET начало]]
```

В этом описании *источник* представляет собой имя таблицы или подзапрос. Эти общие формы имеют следующий синтаксис:

FROM {[*ONLY*] *таблица*
[[*AS*] синоним [(синоним_поля[...])]]
(запрос) [*AS*] синоним [(синоним_поля [...])]]}

ALL. Необязательное ключевое слово *ALL* указывает на то, что в выборку включаются все найденные записи.

DISTINCT [*ON* (выражение [...])]. Секция *DISTINCT* определяет поле или выражение, значения которого должны входить в итоговый набор не более одного раза.

Цель [*AS* имя] [...]. В качестве *цели* обычно указывается имя поля, хотя *цель* также может быть константой, идентификатором, функцией или общим выражением. Перечисляемые *цели* разделяются запятыми, существует возможность динамического назначения имен *целей* в секции *AS*. Звездочка (*) является сокращенным обозначением всех несистемных полей, вместе с ней в списке могут присутствовать и другие *цели*.

FROM *источник* [...]. В секции *FROM* указывается *источник*, в котором PostgreSQL ищет заданные *цели*. В данном случае *источник* является именем таблицы или подзапроса. Допускается перечисление нескольких *источников*, разделенных запятыми (примерный аналог перекрестного запроса). Синтаксис секции *FROM* подробно описан ниже.

[*NATURAL*] *тип_объединения* *источник* [*ON* *условие* | *USING* (*список_полей*)]. Источники *FROM* могут группироваться в секции *JOIN* с указанием типа объединения (*INNER*, *FULL*, *OUTER*, *CROSS*). В зависимости от типа объединения также может потребоваться уточняющее условие или список полей.

WHERE *условие*. Секция *WHERE* ограничивает итоговый набор заданными критериями. Условие должно возвращать простое логическое значение (true или false), но оно может состоять из нескольких внутренних условий, объединенных логическими операторами (например, AND или OR).

GROUP BY критерий [...]. Секция *GROUP BY* обеспечивает группировку записей по заданному критерию. Причем критерий может быть простым именем поля или произвольным выражением, примененным к значениям итогового набора.

HAVING условие [...]. Секция *HAVING* похожа на секцию *WHERE*, но условие проверяется на уровне целых групп, а не отдельных записей.

{*UNION* | *INTERSECT* | *EXCEPT*} [*ALL*] подзапрос. Выполнение одной из трех операций, в которых участвуют два запроса (исходный и дополнительный); итоговые данные возвращаются в виде набора с обобщенной структурой, из которого удаляются дубликаты записей (если не было задано ключевое слово *ALL*). *UNION* - объединение (записи, присутствующие в любом из двух наборов). *INTERSECT* - пересечение (записи, присутствующие одновременно в двух наборах). *EXCEPT* - исключение (записи, присутствующие в основном наборе *SELECT*, но не входящие в подзапрос).

ORDER BY выражение. Сортировка результатов команды *SELECT* по заданному выражению.

[*ASC* | *DESC* | *USING оператор*]. Порядок сортировки, определяемой секцией *ORDER BY* выражение: по возрастанию (*ASC*) или по убыванию (*DESC*). С ключевым словом *USING* может задаваться оператор, определяющий порядок сортировки (например, < или >).

FOR UPDATE [*OF* таблица [...]]. Возможность монопольной блокировки возвращаемых записей. В транзакционных блоках *FOR UPDATE* блокирует записи указанной таблицы до завершения транзакции. Заблокированные записи не могут обновляться другими транзакциями.

LIMIT {число | *ALL*}. Ограничение максимального количества возвращаемых записей или возвращение всей выборки (*ALL*).

OFFSET начало. Точка отсчета записей для секции *LIMIT*. Например, если в секции *LIMIT* установлено ограничение в 100 записей, а в секции

OFFSET -50, запрос вернет записи с номерами 50-150 (если в итоговом наборе найдется столько записей).

Ниже описаны компоненты секции *FROM*.

[ONLY] *таблица*. Имя таблицы, используемой в качестве источника для команды *SELECT*. Ключевое слово *ONLY* исключает из запроса записи всех таблиц-потомков.

[AS] *синоним*. Источникам *FROM* могут назначаться необязательные псевдонимы, упрощающие запрос. Ключевое слово *AS* является необязательным.

(*запрос*) *[AS]* *синоним*. В круглых скобках находится любая синтаксически правильная команда *SELECT*. Итоговый набор, созданный запросом, используется в качестве источника *FROM* так, словно выборка производится из статической таблицы. При выборке из подзапроса обязательно должен назначаться синоним.

(*синоним_поля* [...]). Синонимы могут назначаться не только всему источнику, но и его отдельным полям. Перечисляемые синонимы полей разделяются запятыми и группируются в круглых скобках за синонимом источника *FROM*. Синонимы перечисляются в порядке следования полей в таблице, к которой они относятся.

Целями команды *SELECT* могут быть не только простые поля, но и произвольные выражения (включающие вызовы функций или различные операции с идентификаторами) и константы. Синтаксис команды практически не изменяется, появляется лишь одно дополнительное требование - все самостоятельные выражения, идентификаторы и константы должны разделяться запятыми. В списке разрешены произвольные комбинации разнотипных целей.

Команда *SELECT* также может использоваться для простого вычисления и вывода результатов выражений и констант. В этом случае она не содержит секции *FROM* или имен столбцов.

Для каждой цели в списке может задаваться необязательная секция *AS*, которая назначает синоним (новое произвольное имя) для каждого поля в итоговом наборе. Имена синонимов подчиняются тем же правилам, что и имена обычных идентификаторов (в частности, они могут содержать внутренние пробелы или совпадать с ключевыми словами при условии заключения их в апострофы и т. д.)

Назначение синонима не влияет на исходное поле и действует лишь в контексте итогового набора, возвращаемого запросом. Секция *AS* особенно удобна при «выборке» выражений и констант, поскольку синонимы позволяют уточнить смысл неочевидных выражений или констант.

В секции *FROM* указывается источник данных - таблица или итоговый набор. Секция может содержать несколько источников, разделенных запятыми.

Использование нескольких источников данных в PostgreSQL требует осторожности. В результате выполнения команды *SELECT* для нескольких источников без секций *WHERE* и *JOIN*, уточняющих связи между источниками, возвращается полное декартово произведение источников. Иначе говоря, итоговый набор содержит все возможные комбинации записей из всех источников. Обычно для уточнения связей между источниками, перечисленными через запятую в секции *FROM*, используется секция *WHERE*.

Для предотвращения неоднозначности в «полные» имена столбцов включается имя таблицы. При этом используется специальный синтаксис, называемый точечной записью (название связано с тем, что имя таблицы отделяется от имени поля точкой). Точечная запись обязательна только при наличии неоднозначности между наборами данных.

Если в качестве источника используется итоговый набор, созданный подзапросом, то весь подзапрос заключается в круглые скобки. По наличию круглых скобок PostgreSQL узнает о том, что команда *SELECT*

интерпретируется как подзапрос, и выполняет ее перед выполнением внешней команды *SELECT*.

Источникам данных в секции *FROM* - таблицам, подзапросам и т. д. - можно назначать синонимы в секции *AS* (по аналогии с отдельными полями). Синонимы часто используются для упрощения точечной записи. Наличие синонима для набора данных позволяет обращаться к нему при помощи точечной записи, что делает команды SQL более компактными и наглядными.

Синонимы можно назначать не только для источников данных в секции *FROM*, но и для отдельных полей внутри этих источников. Для этого за синонимом источника данных приводится список синонимов полей, разделенный запятыми и заключенный в круглые скобки. Список синонимов полей представляет собой последовательность идентификаторов, перечисленных в порядке следования полей в структуре таблицы (слева направо).

Список синонимов полей не обязан содержать данные обо всех полях. К тем полям, для которых не были заданы синонимы, можно обращаться по обычным именам. Если единственное поле, которому требуется назначить синоним, находится после других полей, вам придется включить в список все предшествующие поля (синоним поля может совпадать с его именем). В противном случае PostgreSQL не сможет определить, какому полю назначается синоним, и решит, что речь идет о первом поле таблицы.

Необязательное ключевое слово *DISTINCT* исключает дубликаты из итогового набора. Если ключевое слово *ON* отсутствует, из результатов запроса с ключевым словом *DISTINCT* исключаются записи с повторяющимися значениями целевых полей.

Проверяются только поля, входящие в целевой список *SELECT*.

В общем случае PostgreSQL выбирает записи, исключаемые из итогового набора при наличии секции *ON*, по своему усмотрению. Если в запрос вместе с *DISTINCT* входит секция *ORDER BY*, вы можете

самостоятельно задать порядок выборки полей так, чтобы нужные записи оказались в начале. Если вместо исключения всех дубликатов достаточно сгруппировать записи с повторяющимися значениями некоторого критерия, воспользуйтесь секцией *GROUP BY*.

В секции *WHERE* задается логическое условие, которому должны удовлетворять записи итогового набора. На практике команда *SELECT* почти всегда содержит как минимум одну уточняющую секцию *WHERE*. Секция *WHERE* может содержать несколько условий, объединенных логическими операторами (например, *AND* или *OR*) и возвращающими одно логическое значение.

Количество условий, объединяемых в секции *WHERE*, не ограничено, хотя при наличии двух и более условий обычно выполняется группировка при помощи круглых скобок, наглядно демонстрирующая логическую связь между условиями.

Контрольные вопросы

1. Каким образом можно выполнить связывание таблиц?
2. Какие типы связывания используются наиболее часто?
3. С какой целью используется связывание таблиц?
4. Как формируется подзапрос?
5. Как задается параметр для отбора данных в подзапросе?
6. Порядок формирования многотабличных запросов.
7. В каких случаях используются многотабличные запросы?
8. Как задается критерий отбора данных в многотабличных запросах?

Список источников

1. Журналы онлайн (русский) [Электронный ресурс]: Журнал Linux Format за ноябрь 2006 года. – Режим доступа: <http://av5.com/journals-magazines-online/> . - загл. с экрана.
2. Дж Уорсли Дж., Дрейк Дж. PostgreSQL. Для профессионалов - СПб.: Питер, 2003. -496 с.
3. PostgreSQL (русский) [Электронный ресурс]: Документация по PostgreSQL. – Режим доступа: <http://postgresql.ru.net> . - загл. с экрана.
4. Википедия свободная энциклопедия (русский) [Электронный ресурс]: PostgreSQL. – Режим доступа: <http://ru.wikipedia.org>. - загл. с экрана.
5. Журналы онлайн (русский) [Электронный ресурс]: Журнал Системный Администратор за октябрь 2004 года. – Режим доступа: <http://av5.com/journals-magazines-online/> . - загл. с экрана.
6. Компьютерная документация (русский) [Электронный ресурс]: Функции для работы с датой и временем. – Режим доступа: <http://www.hardline.ru/> . - загл. с экрана.
7. Linux и Windows: помощь админам и пользователям (русский) [Электронный ресурс]: 15 команд для управления PostgreSQL. – Режим доступа: <http://www.guruadmin.ru> . - загл. с экрана.
8. Учебники по программированию (русский) [Электронный ресурс]: SQL-сервер в Linux. – Режим доступа: <http://www.book-perlphpsql.ru/> . - загл. с экрана.
9. Мини-портал Linux (русский) [Электронный ресурс]: Оптимизация Apache + PHP + PostgreSQL. – Режим доступа: <http://linux.opennet.ru> . - загл. с экрана.