

Вложенные запросы (подзапросы) в SQL Oracle

1. Цели лабораторной работы

- Изучить возможности SQL Oracle по формулировке и обработке подзапросов.
- Приобрести практический опыт по формулировке и обработке подзапросов с использованием SQL*Plus.

2. Теоретические основы

Запрос – это операция, которая позволяет отыскивать данные из одной или несколько таблиц. При наличии вложенных запросов запрос верхнего уровня называется предложением SELECT, а запрос, вложенный в предложение SELECT, называется подзапросом. Таким образом, **подзапрос** (вложенный запрос) – это запрос, результат которого передается в качестве аргумента в другой запрос. Подзапросы позволяют связывать в единое целое несколько запросов.

Подзапросы используются для:

- определения множества строк, который должны быть вставлены в целевую таблицу в предложениях INSERT или CREATE TABLE;
- определения одного или более значений, присваиваемых существующим строка в предложении UPDATE;
- получения значений для фраз WHERE, HAVING или START WITH в предложениях SELECT, UPDATE, и DELETE;
- определения значений указанного столбца в списке INSERT ... VALUES;
- определения таблицы, которая используются соответствующим запросом.

Это производится путем размещения подзапроса во фразе FROM соответствующего запроса, как если бы это было именем таблицы. Вы можете также использовать таким образом подзапросы вместо таблиц в предложениях INSERT, UPDATE и DELETE.

Используемые таким образом подзапросы могут использовать переменные связывания (correlation variables), однако только такие, которые определены только в

самом подзапросе, ссылки на внешние переменные не допустимы. Внешние ссылки (подзапросы с левой корреляцией - left-correlated subqueries) допустимы только во фразе FROM предложения SELECT.

Подзапрос дает ответ на содержательные запросы, имеющие сложную структуру. Например, для определения, кто работает на кафедре Иванова, вы сначала используете подзапрос для определения кафедры, на которой работает Иванов, а затем отвечаете на основной запрос путем формулировки предложения SELECT.

Подзапрос может содержать другие подзапросы. Oracle не ограничивает глубину вложенности подзапросов.

Если таблица в подзапросе имеет такое же имя, что и таблица внешнего запроса, то для ссылки на столбцы внешнего запроса их необходимо уточнять именем таблицы или аliasом таблицы. Чтобы ваши запросы было легче воспринимать, всегда квалифицируйте столбцы в подзапросе именем или аliasом таблицы.

Oracle выполняет **корреляционные (связанный) подзапрос**, когда подзапрос ссылается на столбец таблицы внешнего запроса. Связанный подзапрос вычисляется для каждой строки, обрабатываемой внешним предложением. Внешним предложением может быть SELECT, UPDATE или DELETE.

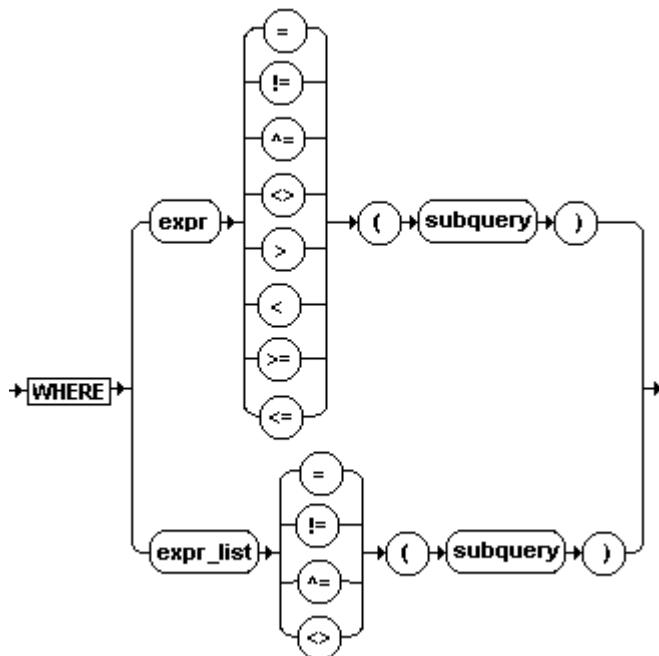
Связанный подзапрос дает ответы на такие содержательные запросы, ответы которых требуют вычисления подзапросов для каждой строки внешнего запроса. Например, связанный подзапрос используется для определения преподавателей, которые зарабатывают больше, чем средняя зарплата по кафедре. В этом случае связанный подзапрос для каждого преподавателя вычисляет среднюю зарплату на его кафедре.

Далее мы обсудим использование подзапросов в предложении SELECT.

2.1. Подзапрос во фразе WHERE

2.1.1. Подзапрос в простом условии сравнения

Синтаксис:



Описание:

При использовании простых условий сравнения с подзапросом во фразе WHERE применяются следующие правила:

- Подзапрос должен возвращать единственную строку;
- Если левая часть равна *expr*, то подзапрос должен возвращать единственную строку с единственным значением с типом, совместимым с типом *expr*;
- Если левая часть является списком выражений (*expr_list*), то подзапрос должен возвращать единственную строку со списком значений, который соответствует по количеству и типу значениям из *expr_list*. В этом случае оператор сравнения дает TRUE, если каждое значение в *expr_list* равно (в случае *=*) или не равно (в случае *!=*, *&=*, *<>*) каждому значению, возвращаемому подзапросом;

Примеры:

1. Выбрать кафедры, которые располагаются в том же корпусе, что факультет информатики:

```
SELECT Name  
FROM DEPARTMENT  
WHERE Building = (SELECT Building  
                   FROM FACULTY  
                   WHERE UPPER(Name) = 'INFORMATICS');
```

2. Выбрать факультеты, чьи фонды меньше фонда кафедры CAD:

```
SELECT Name  
FROM FACULTY  
WHERE Fund < (SELECT Fund  
               FROM DEPARTMENT  
               WHERE UPPER(Name) = 'CAD');
```

3. Выбрать преподавателей, у которых salary + commission превышает более чем на 100 половину salary + commission преподавателя Bill:

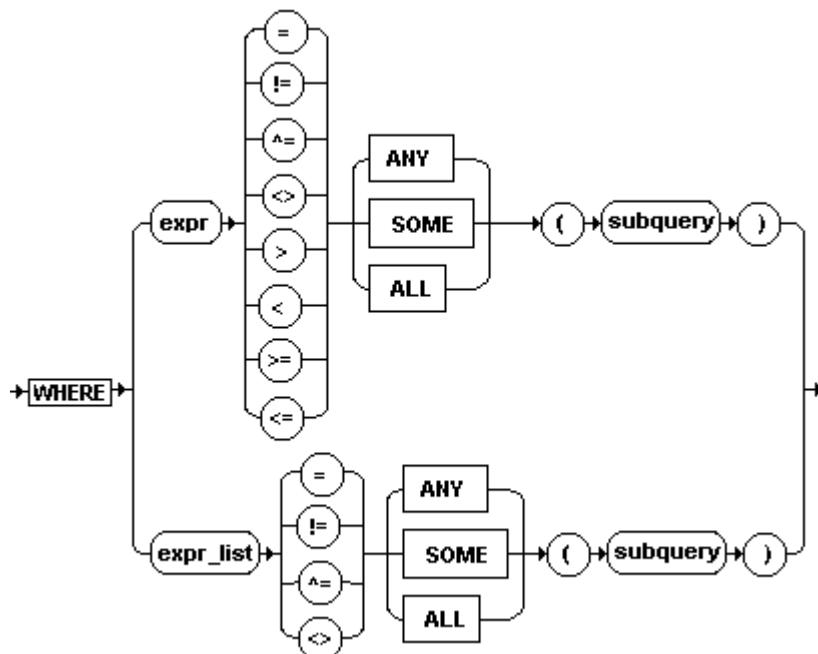
```
SELECT Name  
FROM TEACHER  
WHERE Salary + Commission + 100 > (SELECT (Salary + Commission) / 2  
                                       FROM TEACHER  
                                       WHERE UPPER(Name) = 'BILL');
```

4. Выбрать преподавателей, которые работают на той же кафедре, что и Bill и занимают ту же должность, что и Bill:

```
SELECT Name  
FROM TEACHER  
WHERE (DepNo, Post) = (SELECT DepNo, Post  
                        FROM TEACHER  
                        WHERE UPPER(Name) = 'BILL');
```

2.1.2. Подзапрос в условии сравнения групп

Синтаксис:



Описание:

При использовании условий сравнения групп с подзапросом во фразе WHERE применяются следующие правила:

- Подзапрос может возвращать ноль или более строк;
- Если левая часть равна *expr*, то подзапрос должен возвращать строки с единственным значением, которые совместимы по типу с *expr*;
- Если левая часть равна *expr_list*, то подзапрос должен возвращать строки со списком значений, который соответствует по количеству и типу с *expr_list*.

ANY и SOME эквивалентны и сравнивают значение слева с каждым значением списка справа, возвращаемого подзапросом. Подзапрос может вернуть ноль или более строк. Условие равно TRUE, если по крайней мере одна строка подзапроса удовлетворяет условию (соответствует оператору сравнения) по отношению к значению (списку значений), определенному левым операндом, в противном получаем FALSE. Если подзапрос не возвращает строк, то получаем FALSE.

ALL сравнивают значение слева с каждым значением списка справа, возвращаемого подзапросом. Дает TRUE, если ВСЕ строки, возвращаемые подзапросом, удовлетворяют условию (соответствуют оператору сравнения) по отношению к значению (списку значений), определенному левым операндом, в противном получаем FALSE. Если подзапрос не возвращает строк, то получаем TRUE

Примеры:

1. Выдать кафедры, фонд которых больше фонда по крайней мере одного из факультетов:

```
SELECT Name  
FROM DEPARTMENT  
WHERE Fund > ANY (SELECT Fund FROM FACULTY);
```

ANY, ALL и агрегатные функции. Обратите внимание, что оператор “ $<$ ANY” эквивалентно утверждению: “левое значение меньше, чем максимальное значение из множества, задаваемого правым operandом”, а оператор “ $>$ ANY” эквивалентен следующему утверждению “левое значение больше, чем минимальное значение из множества, задаваемого правым operandом”. Поэтому операторы ANY могут быть выражены через функции MAX и MIN в подзапросе.

В свою очередь, “ $<$ ALL” (“ $>$ ALL”) эквивалентно утверждению: «левое значение меньше (больше), чем минимальное (максимальное) значение из множества, задаваемого правым operandом».

2. Выдать кафедры, фонд которых больше фонда по крайней мере одного из факультетов:

```
SELECT Name  
FROM DEPARTMENT  
WHERE Fund > ANY (SELECT Fund FROM FACULTY);
```

```
SELECT Name  
FROM DEPARTMENT
```

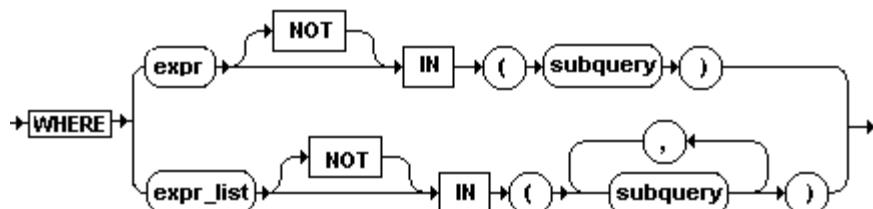
```
WHERE Fund > (SELECT MIN(Fund) FROM FACULTY);
```

3. Выдать группы, которые имеют рейтинг больше, чем рейтинг всех групп пятого курса кафедры “DBMS”:

```
SELECT Num  
FROM SGROUP  
WHERE Rating > ALL (SELECT Rating  
                      FROM SGROUP, DEPARTMENT  
                     WHERE SGROUP.DepNo = DEPARTMENT.DepNo AND  
                           UPPER(DEPARTMENT.Name) = 'DBMS' AND  
                           SGROUP.Course = 5);
```

2.1.3. Подзапрос в условии проверки вхождения элемента во множество

Синтаксис:



Описание:

Это условие в таком синтаксисе проверят вхождение элемента (списка элементов) во множество (множество списков), создаваемое подзапросом.

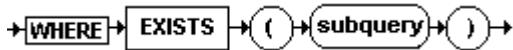
Пример:

1. Выбрать преподавателей, которые имеют лекции по крайней мере одному такому предмету, по которым читает лекции преподаватель Bill:

```
SELECT Name  
FROM TEACHER T, LECTURE L  
WHERE T.TchNo = L.TchNo AND  
      SbjNo IN (SELECT SbjNo  
                  FROM TEACHER TCH, LECTURE LEC  
                 WHERE TCH.TchNo = LEC.TchNo AND UPPER(TCH.Name) =  
                       'BILL');
```

2.1.4. Подзапрос в условии EXISTS

Синтаксис:



Описание:

Дает TRUE, если подзапрос возвращает по крайней мере одну строку.

Так как EXISTS обычно используется в связанных подзапросах, мы его обсудим подробнее позже.

2.2. Связанные подзапросы

Для того, чтобы связать подзапрос с внешним запросом (предложением), необходимо в подзапросе была ссылка на столбец внешнего запроса. Подзапрос вычисляется для каждой строки, обрабатываемой внешним запросом (предложением). В качестве внешнего предложения могут выступать SELECT, UPDATE или DELETE.

Следующие примеры дают общий синтаксис использования связанных подзапросов:

```
SELECT select_list
FROM table1 t_alias1
WHERE expr operator
(SELECT column_list
FROM table2 t_alias2
WHERE t_alias1.column operator t_alias2.column);
UPDATE table1 t_alias1
SET column =
(SELECT expr
FROM table2 t_alias2
WHERE t_alias1.column = t_alias2.column);
DELETE FROM table1 t_alias1
WHERE column operator
(SELECT expr
FROM table2 t_alias2
WHERE t_alias1.column = t_alias2.column);
```

Далее мы обсудим использование связанных подзапросов во фразе WHERE предложения SELECT.

2.2.1. Связанные подзапросы во фразе WHERE

Примеры:

1. Выдать преподавателей, которые имеют по крайней мере одну лекцию:

```
SELECT Name  
FROM TEACHER  
WHERE EXISTS (SELECT *  
               FROM LECTURE  
              WHERE LECTURE.TchNo = TEACHER.TchNo);
```

Здесь в условии LECTURE.TchNo = TEACHER.TchNo подзапроса мы ссылаемся на внешний запрос. Поэтому подзапрос является связанным.

2. Выдать преподавателей, которые не имеют ни одной лекции:

```
SELECT Name  
FROM TEACHER  
WHERE NOT EXISTS (SELECT*  
                   FROM LECTURE  
                  WHERE LECTURE.TchNo = TEACHER.TchNo);
```

2.3. Простые и связанные подзапросы во фразе HAVING

Вы можете использовать простые и связанные подзапросы во фразе HAVING.

Если вы используете связанный подзапрос в фразе HAVING, то в подзапросе можно ссылаться на те столбцы внешнего запроса, которые могут использоваться в фразе HAVING (обычно это столбцы, по которым производится группирование).

Примеры:

1. Перечислить факультеты, у которых сумма фондов финансирования всех их кафедр превышает более чем на 20000 фонд финансирования той кафедры факультета, которая имеет максимальный фонд.

```
SELECT F1.Name  
FROM   FACULTY F1, DEPARTMENT D1  
WHERE  F1.FacNo = D1.FacNo  
GROUP BY F1.Name  
HAVING SUM(D1.Fund) > (SELECT 200000 + MAX(D2.Fund)  
                           FROM   FACULTY F2, DEPARTMENT D2  
                          WHERE  F2.FacNo = D2.FacNo AND F1.Name =  
                                F2.Name);
```

2.4. Простые подзапросы во фразе FROM

Фраза FROM может содержать не только список имен таблиц, но и подзапросы. Для ссылки на такие таблицы-подзапросы следует приписать подзапросу алиас.

Имеется класс запросов, которые не могут быть выражены без подзапроса во фразе FROM. К ним, в частности, относятся такие запросы, которые требуют независимого вычисления двух или более запросов, и затем совместного использования результатов такого запроса.

Пример:

Выдать средний фонд финансирования факультетов и среднюю зарплату преподавателей:

```
SELECT Fac.AvgFund, Tch.AvgSalary  
FROM (SELECT AVG(Fund) AS AvgFund FROM FACULTY) Fac,  
      (SELECT AVG(Salary) AS AvgSalary      FROM TEACHER) Tch;
```

2.5. Подзапросы во фразе SELECT

Во фразе SELECT можно использовать простые (независимые, несвязанные) и связанные (коррелированные) запросы. **В обоих случаях подзапрос должен возвращать одно значение.**

Подзапрос является простым, если в нем не используются атрибуты таблиц, определенных в основном (внешнем запросе). При использовании простого подзапроса он вычисляется однократно и возвращенное им значение вставляется в соответствующее место во все строки, формируемые внешним запросом.

Пример. Для каждого факультета вывести его название, фонд финансирование, а также максимальный и минимальный фонды финансирования среди всех кафедр ВУЗа

```
SELECT Name AS "Факультет",  
       Fund AS "Фонд факультета",  
       (SELECT MAX(Fund) FROM DEPARTMENT) AS "МАКС фонд кафедр",  
       (SELECT MIN(Fund) FROM DEPARTMENT) AS "МИН фонд кафедр"  
FROM FACULTY
```

Подзапрос является связанным (коррелированным), если в нем используются атрибуты таблиц, определенных во внешней запросе. В этом случае подзапрос вычисляется для каждой строки, формируемой для фразы SELECT.

Пример. По каждому факультету, расположенному в корпусе 6, вывести:

- название факультета
- количество групп этого факультета с рейтингом, более 20
- количество преподавателей-профессоров

```
SELECT Name AS "Факультет",
       (SELECT COUNT (DISTINCT GrpPK)
        FROM DEPARTMENT d, SGROUP g
        WHERE f.FacPK=d.FacFK AND d.DepPK=g.DepFK AND g.Rating > 20)
    AS "К-во групп",
       (SELECT COUNT (DISTINCT TchPK)
        FROM DEPARTMENT d, TEACHER t
        WHERE f.FacPK=d.FacFK AND d.DepPK=t.DepFK AND
              UPPER(t.Post)= 'профессор')           AS "К-во профессоров"
   FROM FACULTY f
  WHERE Building= '6';
```

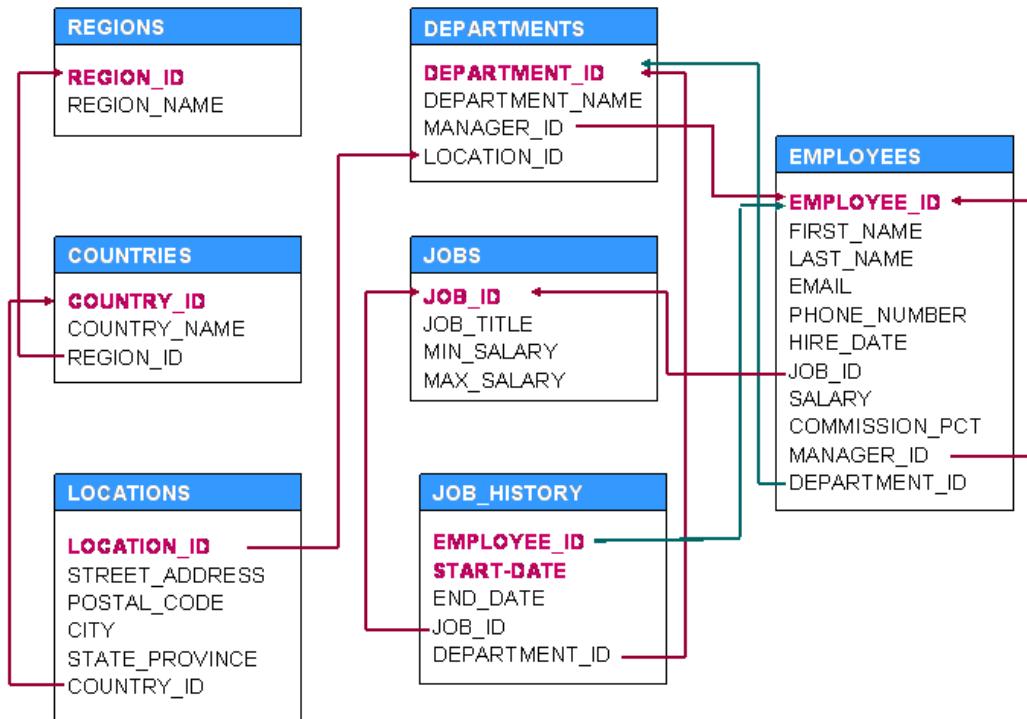
Примечание: для получения дополнительной информации по данной теме также полезно почитать данную статью:

[https://docs.microsoft.com/ru-ru/sql/relational-databases/performance/subqueries?
view=aps-pdw-2016](https://docs.microsoft.com/ru-ru/sql/relational-databases/performance/subqueries?view=aps-pdw-2016)

В рамках данной статьи разбирается некоторая модификация стандартного языка запросов SQL, но синтаксис, используемый в окружения ключевых слов GO, соответствует стандарту, используемому в СУБД ORACLE.

3. Задание для аудиторной и самостоятельной работы:

Схема таблиц HR



Далее приводятся вариант с 7 запросами, которые относятся к следующим категориям (в порядке их следования):

- 1) Некоррелируемые подзапросы;
- 2) Коррелируемые (зависимые, связанные) подзапросы;
- 3) Коррелируемые подзапросы и предикат EXISTS;
- 4) Коррелируемые подзапросы и предикат ANY, SOME, ALL;
- 5) Подзапросы во фразе HAVING;
- 6) Подзапросы во фразе FROM;
- 7) Подзапросы во фразе SELECT.

3.1. Варианты запросов

- 1) По каждому департаменту, расположенному в том же городе, что и департамент, руководителем которого является King, вывести следующую информацию в столбцах с соответствующими именами:
 - название департамента (department_name);
 - имя руководителя (first_name, last_name);
 - город департамента (city_department)
- 2) Вывести названия департаментов, которые располагаются в США (department_name и country).
- 3) Вывести названия департаментов, располагающихся в США, и фамилии их руководителей, в которых работает хотя бы один программист и имя первого программиста (если их несколько) (department_name и country, director_last_name, programmer_first_name job_title = programmer).
- 4) Вывести всех работников (id, first_name, last_name, salary + те поля что написаны в каждом пункте), у которых:
 - 4.1) Зарплата больше средней зарплаты (при выводе среднего значение допускается не больше 2 знаков после запятой) всех работников (all_average_salary);
 - 4.2) Зарплата которых выше, чем зарплата одного любого сотрудника в отделе “IT” (нужно выбрать одного любого сотрудника, который есть в этом отделе) (worker_first_name, worker_last_name, worker_salary этого выбранного сотрудника);
 - 4.3) Зарплаты с премией больше, чем любая из сумм зарплаты с премией в стране “Россия”;
 - 4.4) Зарплата больше 100 000 и нет премии (где prize = 0).

- 5) Придумать самостоятельно;
- 6) Придумать самостоятельно;
- 7) Придумать самостоятельно;