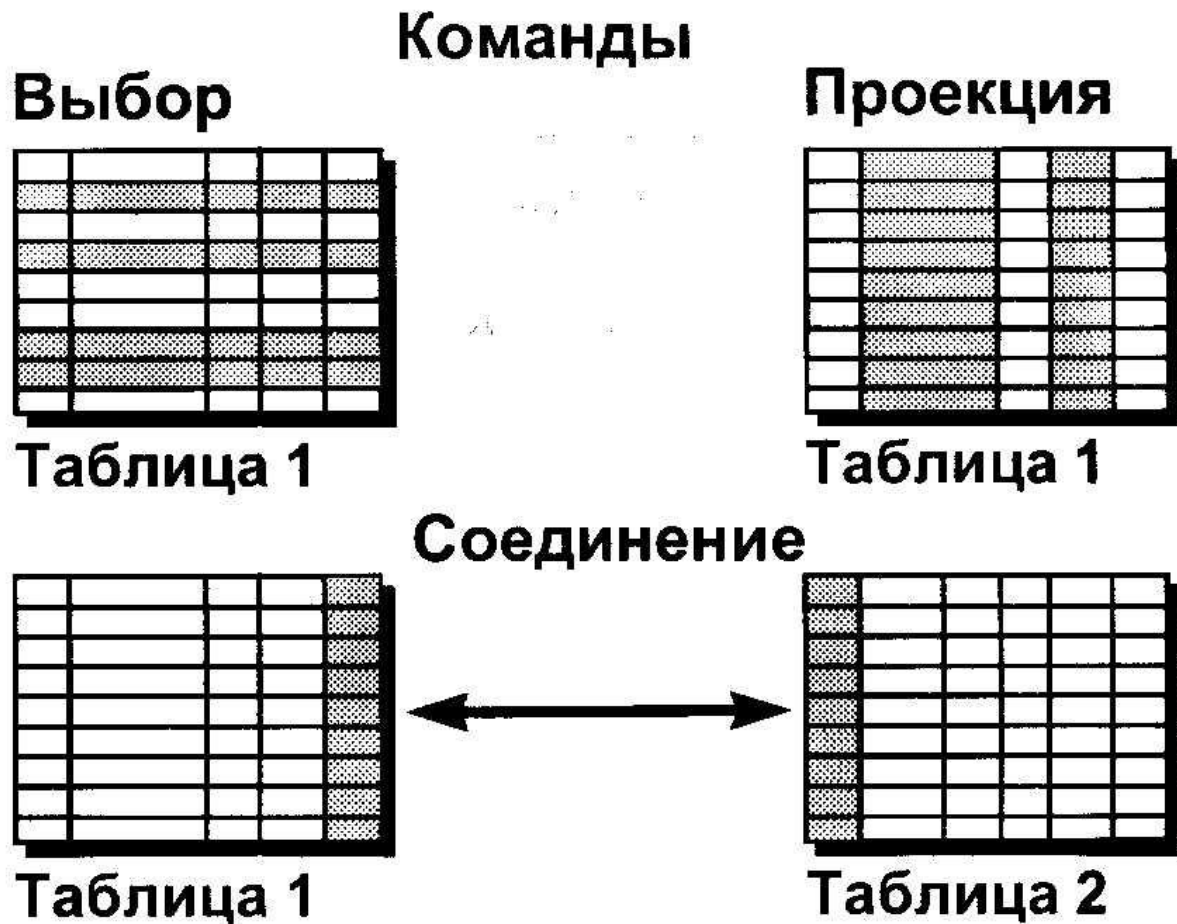


Язык SQL

Реализация Oracle 10g

Возможности команды SELECT языка SQL



Основная команда SELECT

- SELECT [DISTINCT] {*, column [*alias*],...)
- FROM *table*;
- SELECT указывает, *какие* столбцы
- FROM указывает, из *какой* таблицы

Написание команд SQL

- Команды SQL не различают регистры символов
- Команды SQL могут занимать одну или несколько строк
- Ключевые слова нельзя сокращать и размещать на двух строках
- Предложения обычно пишутся на отдельных строках
- Для облегчения чтения используются табуляция и отступы

Выбор конкретных столбцов

```
SQL> SELECT deptno, loc  
2 FROM dept;
```

DEPTNO	LOC
10	NEW YORK
20	DALLAS
30	CHICAGO
40	BOSTON

- **Арифметические выражения**
- **Создаются из данных типа NUMBER и DATE с помощью арифметических операторов**

Использование арифметических операторов

```
SQL> SELECT ename, sal, sal+300  
2 FROM emp;
```

ENAME	SAL	SAL+300
KING	5000	5300
BLAKE	2850	3150
CLARK	2450	2750
JONES	2975	3275
MARTIN	1250	1550
ALLEN	1600	1900

...

14 rows selected.

Неопределенное значение (NULL)

- Неопределенное значение (NULL) - это когда значение недоступно, не присвоено, неизвестно или неприменимо.
- Это не ноль и не пробел

```
SQL> select ename NAME, 12*sal+comm  
2  from    emp  
3  WHERE   ename='KING';
```

NAME	12*SAL+COMM
-----	-----
KING	

Псевдоним (алиас) столбца

- Альтернативный заголовок столбца
- Удобен при вычислениях
- Следует сразу за именем столбца;
ключевое
слово **AS** между именем столбца с
псевдонимом необязательно
- Заключается в двойные кавычки, если
содержит пробелы, специальные
СИМВОЛЫ
или различает регистры символов

Использование псевдонимов столбцов

```
SQL> SELECT ename AS name, sal salary
2 FROM emp;
```

NAME	SALARY
-----	-----
...	

```
SQL> SELECT ename "Name",
2 sal*12 "Annual Salary"
3 FROM emp;
```

Name	Annual Salary
-----	-----
...	

Дублирование строк

- По умолчанию выдаются все строки, включая дубликаты.

```
SQL> SELECT deptno  
2 FROM emp;
```

```
DEPTNO  
-----  
10  
30  
10  
20  
...  
14 rows selected.
```

Устранение строк-дубликатов

- Дубликаты устраняются с помощью ключевого слова **DISTINCT** в команде **SELECT**.

```
SQL> SELECT DISTINCT deptno  
2 FROM emp;
```

DEPTNO
10
20
30

Ограничение количества выбираемых строк путем отбора

EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7698	BLAKE	MANAGER		30
7782	CLARK	MANAGER		10
7566	JONES	MANAGER		20
...				

"...выбрать всех
служащих отдела
10"



EMP

EMPNO	ENAME	JOB	...	DEPTNO
7839	KING	PRESIDENT		10
7782	CLARK	MANAGER		10
7934	MILLER	CLERK		10

Ограничение количества выбираемых строк

- Количество возвращаемых строк можно ограничить с помощью предложения **WHERE**.

```
SELECT          [DISTINCT] {*, column [alias], ...}  
FROM           table  
[WHERE        condition(s)];
```

- Предложение **WHERE** следует за предложением **FROM**.

Использование предложения WHERE

```
SQL> SELECT ename, job, deptno  
2 FROM emp  
3 WHERE job='CLERK';
```

ENAME	JOB	DEPTNO
JAMES	CLERK	30
SMITH	CLERK	20
ADAMS	CLERK	20
MILLER	CLERK	10

Прочие операторы сравнения

Оператор	Значение
BETWEEN ...AND...	Находится в диапазоне от одного значения до другого (включительно)
IN(list)	Совпадает с каким-либо значением списка
LIKE	Соответствует символьному шаблону
IS NULL	Является неопределенным значением

Использование оператора BETWEEN

- Оператор BETWEEN используется для вывода строк по диапазону значений.

```
SQL> SELECT      ename, sal
      2 FROM        emp
      3 WHERE      sal BETWEEN 1000 AND 1500;
```

ENAME	SAL
MARTIN	1250
TURNER	1500
WARD	1250
ADAMS	1100
MILLER	1300

↑
Нижняя
граница

↑
Верхняя
граница

Использование оператора IN

- Оператор IN используется для проверки на входжение значений в список.

```
SQL> SELECT empno, ename, sal, mgr
2 FROM emp
3 WHERE mgr IN (7902, 7566, 7788);
```

EMPNO	ENAME	SAL	MGR
7902	FORD	3000	7566
7369	SMITH	800	7902
7788	SCOTT	3000	7566
7876	ADAMS	1100	7788

Использование оператора LIKE

- Оператор LIKE используется для поиска символьных значений по шаблону с метасимволами.
- Условия поиска могут включать алфавитные и цифровые символы.
- % обозначает ноль или много символов
- _ обозначает один символ

```
SQL> SELECT   ename
2  FROM      emp
3  WHERE     ename LIKE 'S%';
```

Логические операторы

Оператор	Значение
AND	Возвращает результат ИСТИННО, если выполняются оба условия.
OR	Возвращает результат ИСТИННО, если выполняется любое из условий.
NOT	Возвращает результат ИСТИННО, если следующее условие не выполняется.

Предложение ORDER BY

- Предложение ORDER BY используется для сортировки строк
 - ASC: сортировка по возрастанию (используется по умолчанию)
 - DESC: сортировка по убыванию
- В команде SELECT предложение ORDER BY указывается последним.

```
SQL> SELECT      ename, job, deptno, hiredate
2 FROM          emp
3 ORDER BY hiredate;
```

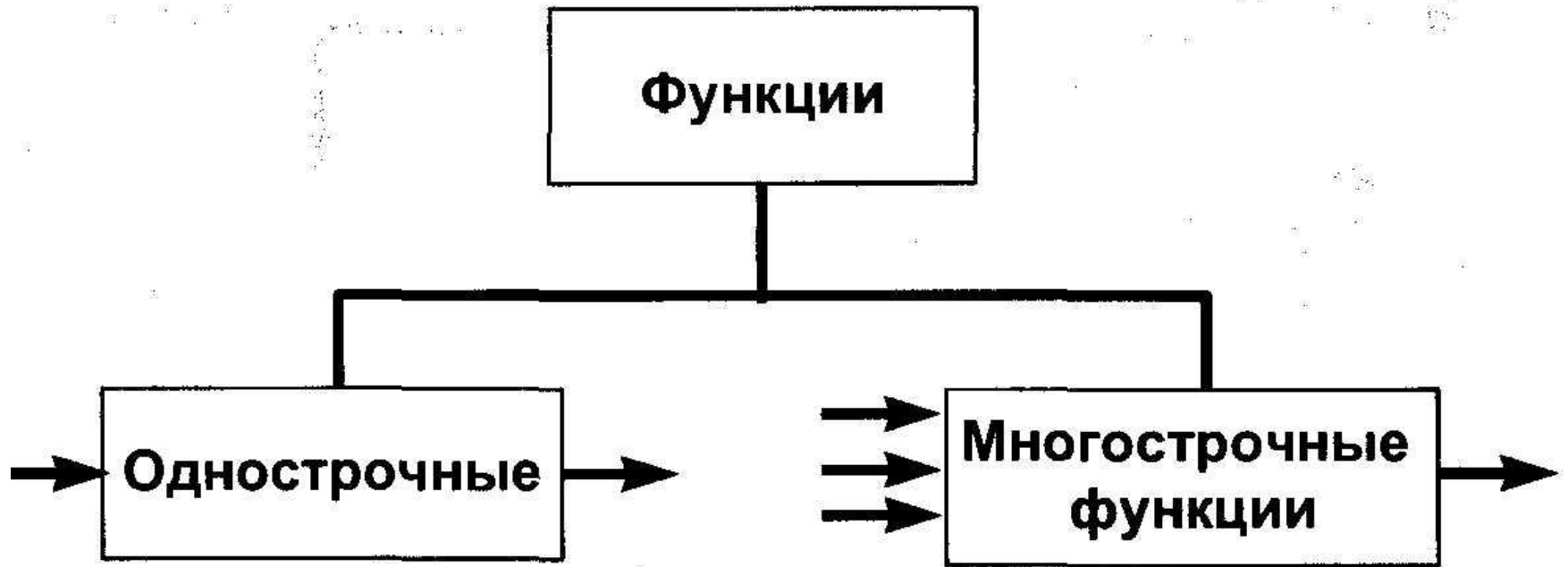
ENAME	JOB	DEPTNO	HIREDATE
SMITH	CLERK	20	17-DEC-80
ALLEN	SALESMAN	30	20-FEB-81

```
...
14 rows selected.
```

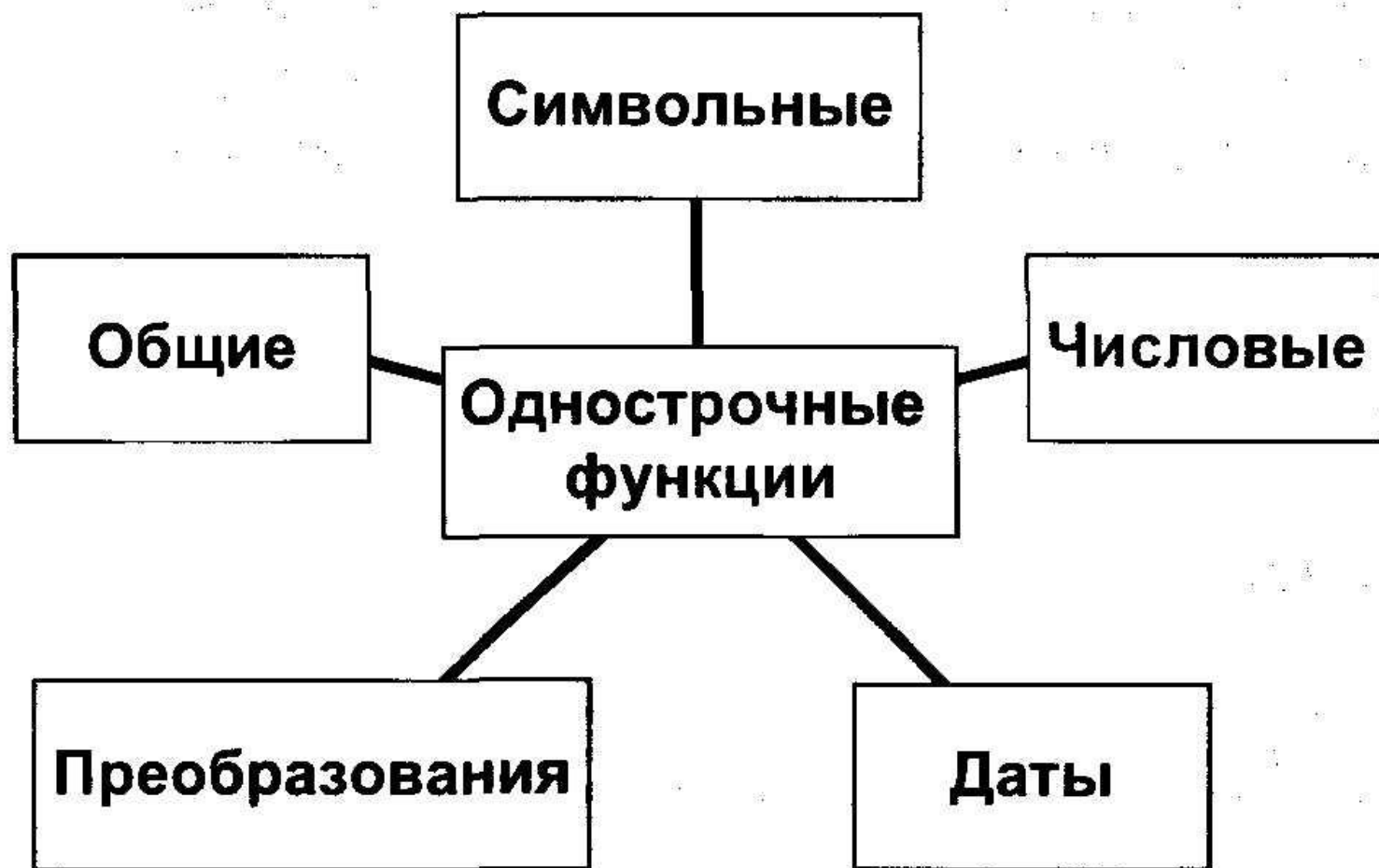
Однострочные функции

- **Различные типы функций в SQL.**
- **Включение в команды SELECT функций различных типов — символьных, числовых и типа "дата".**
- **Функции преобразования данных и их использование.**

Два типа функций SQL



Однострочные функции



Символьные функции

Символьные
функции

Функции преобразования
регистра символов

LOWER

UPPER

INITCAP

Функции манипулирования
символами

CONCAT

SUBSTR

LENGTH

INSTR

LPAD

Использование функций преобразования регистра

Вывод номера служащего, фамилии и номера отдела для служащего по фамилии Blake.

```
SQL> SELECT empno, ename, deptno
 2 FROM emp
 3 WHERE ename = 'blake';
no rows selected
```

```
SQL> SELECT empno, ename, deptno
 2 FROM emp
 3 WHERE LOWER(ename) = 'blake';
```

EMPNO	ENAME	DEPTNO
7698	BLAKE	30

Использование функций манипулирования символами

```
SQL> SELECT ename, CONCAT (ename, job), LENGTH(ename),  
2          INSTR(ename, 'A')  
3 FROM      emp  
4 WHERE     SUBSTR(job,1,5) = 'SALES';
```

ENAME	CONCAT (ENAME, JOB)	LENGTH (ENAME)	INSTR (ENAME, 'A')
MARTIN	MARTINSALESMAN	6	2
ALLEN	ALLENSALESMAN	5	1
TURNER	TURNERSALESMAN	6	0
WARD	WARDSALESMAN	4	2

Работа с датами

- • Oracle хранит данные во внутреннем цифровом формате.
- - Век, год, месяц, число, часы, минуты, секунды
- По умолчанию дата выдается в формате DD-MON-YY (число- месяц-год)
- Функция SYSDATE возвращает текущие дату и время
- DUAL - это фиктивная таблица, используемая для просмотра SYSDATE

Арифметические операции с датами

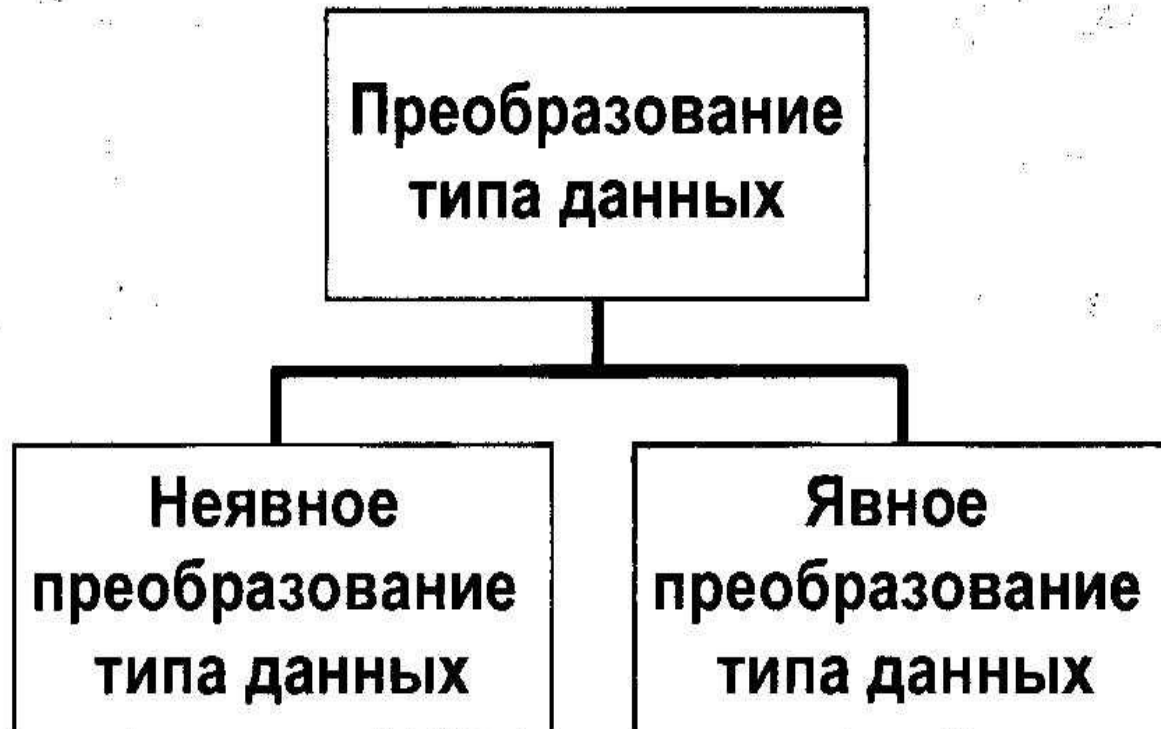
- Результатом прибавления числа к дате и вычитания числа из даты является дата.
- Результатом вычитания одной даты из другой является количество дней, разделяющих эти даты.
- Прибавление часов к дате производится путем деления количества часов на 24.

Использование арифметических операторов с датами

```
SQL> SELECT ename, (SYSDATE-hiredate)/7 WEEKS  
2 FROM emp  
3 WHERE deptno = 10;
```

ENAME	WEEKS
KING	830.93709
CLARK	853.93709
MILLER	821.36566

Функции преобразования

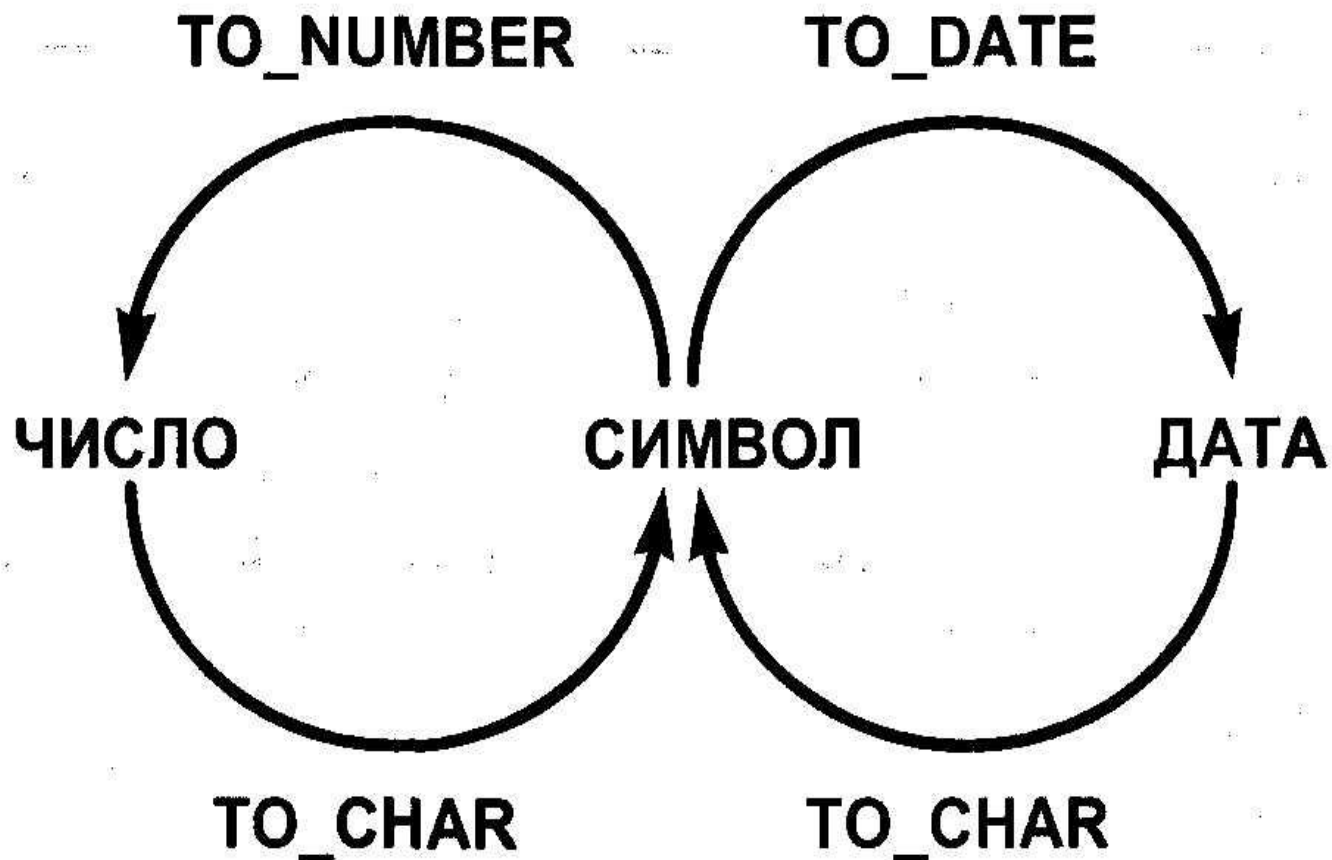


Неявное преобразование ТИПОВ ДАННЫХ

- Для операций присваивания Oracle может автоматически выполнять следующие преобразования:

Исходный формат	Новый формат
VARCHAR2 или CHAR	NUMBER
VARCHAR2 или CHAR	DATE
NUMBER	VARCHAR2
DATE	VARCHAR2

Явное преобразование типов данных



Функция TO_CHAR с датами

```
TO_CHAR(date, 'fmt')
```

Модель формата:

- Должна быть заключена в апострофы. Различает символы верхнего и нижнего регистров. • Может включать любые разрешенные элементы формата даты. • Использует элемент *fm* для удаления конечных пробелов и ведущих нулей.
- Отделяется от значения даты запятой.

Элементы формата даты

YYYY	Полный год цифрами
YEAR	Год прописью
MM	Двузначное цифровое обозначение месяца
MONTH	Полное название месяца
DY	Трехзначное алфавитное сокращенное название дня недели
DAY	Полное название дня недели

Элементы модели формата даты

- Элементы, которые задают формат части даты, обозначающей время.

HH24:MI:SS AM

15:45:32 PM

- Символьные строки добавляются в кавычках.

DD "of" MONTH

12 of OCTOBER

- Числовые суффиксы используются для вывода числительных прописью.

ddspth

fourteenth

Функция TO_CHAR с числами

```
TO_CHAR(number, 'fmt')
```

Форматы, используемые с функцией TO_CHAR для вывода символьного значения в виде числа

9	Цифра
0	Вывод нуля
\$	Плавающий знак доллара
L	Плавающий символ местной валюты
.	Вывод десятичной точки
,	Вывод разделителя троек цифр

Функция TO_CHAR с числами

```
SQL> SELECT TO_CHAR(sal, '$99,999') SALARY  
2 FROM emp  
3 WHERE ename = 'SCOTT';
```

SALARY

\$3,000

Функции TO_NUMBER и TO_DATE

- Преобразование строки символов в числовой формат с помощью функции TO_NUMBER

```
TO_NUMBER(char)
```

- Преобразование строки символов в формат даты с помощью функции TO_DATE

```
TO_DATE(char[, 'fmt'])
```

Темы

- Команды **SELECT** для выборки данных из более, чем одной таблицы с помощью эквисоединений и прочих видов соединений.
- Использование внешних соединений для просмотра данных, не удовлетворяющих обычным условиям соединения
- Соединение таблицы с собой

Выборка данных из нескольких таблиц

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
...
7934	MILLER	10

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

↓ ↓

EMPNO	DEPTNO	LOC
7839	10	NEW YORK
7698	30	CHICAGO
7782	10	NEW YORK
7566	20	DALLAS
7654	30	CHICAGO
7499	30	CHICAGO
...

14 rows selected.

Что такое соединение?

- Соединение используется для выборки данных из более, чем одной таблицы.
- Условие соединения указывается в предложении **WHERE**.
- Если одно и то же имя столбца присутствует более, чем в одной таблице, к имени столбца добавляется имя таблицы в виде префикса .

Декартово произведение

- Декартово произведения образуется , если:
- Опущено условие соединения.
- Условие соединения недействительно.
- Все строки первой таблицы соединяются со всеми строками второй таблицы.
- Во избежание получения декартова произведения предложение **WHERE** всегда должно включать допустимое условие соединения.

Получение декартова произведения

EMP (14 rows)

EMPNO	ENAME	...	DEPTNO
7839	KING	...	10
7698	BLAKE	...	30
...			
7934	MILLER	...	10

DEPT (4 rows)

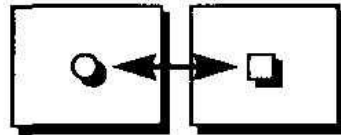
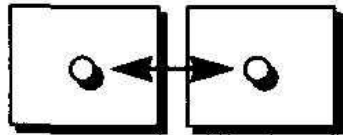
DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

“Декартово
произведение:
14*4=56 строк”

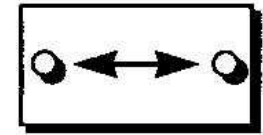
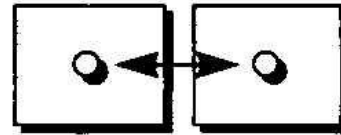
ENAME	DNAME
KING	ACCOUNTING
BLAKE	ACCOUNTING
...	
KING	RESEARCH
BLAKE	RESEARCH
...	
56 rows selected.	

Виды соединений

Эквисоединение



Внешнее
соединение



Не-эквисоединение



Соединение
таблицы с
собой



Что такое эквисоединение?

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20
...		
14 rows selected.		

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS
...		
14 rows selected.		

Главный ключ Внешний ключ



Выборка записей с помощью ЭКВИСОЕДИНЕНИЙ

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2          dept.deptno, dept.loc  
3 FROM emp, dept  
4 WHERE emp.deptno=dept.deptno;
```

EMPNO	ENAME	DEPTNO	DEPTNO	LOC
7839	KING	10	10	NEW YORK
7698	BLAKE	30	30	CHICAGO
7782	CLARK	10	10	NEW YORK
7566	JONES	20	20	DALLAS

...

14 rows selected.

Различение столбцов с одинаковыми именами

- Для различения одноименных столбцов из разных таблиц используются префиксы в виде имен таблиц.
- Использование префиксов в виде имен таблиц увеличивает производительность.
- Одноименные столбцы из разных таблиц можно различать по их псевдонимам.

Дополнительные условия поиска с оператором AND

EMP

EMPNO	ENAME	DEPTNO
7839	KING	10
7698	BLAKE	30
7782	CLARK	10
7566	JONES	20
7654	MARTIN	30
7499	ALLEN	30
7844	TURNER	30
7900	JAMES	30
7521	WARD	30
7902	FORD	20
7369	SMITH	20

...

14 rows selected.

DEPT

DEPTNO	DNAME	LOC
10	ACCOUNTING	NEW YORK
30	SALES	CHICAGO
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
30	SALES	CHICAGO
20	RESEARCH	DALLAS
20	RESEARCH	DALLAS

...

14 rows selected.

Псевдонимы таблиц

Использования псевдонимов таблиц упрощает запросы.

```
SQL> SELECT emp.empno, emp.ename, emp.deptno,  
2         dept.deptno, dept.loc  
3 FROM    emp, dept  
4 WHERE   emp.deptno=dept.deptno;
```

```
SQL> SELECT e.empno, e.ename, e.deptno,  
2         d.deptno, d.loc  
3 FROM    emp e, dept d  
4 WHERE   e.deptno=d.deptno;
```

Соединение более, чем двух таблиц

CUSTOMER

NAME	CUSTID
-----	-----
JOCKSPORTS	100
TKB SPORT SHOP	101
VOLLYRITE	102
JUST TENNIS	103
K+T SPORTS	105
SHAPE UP	106
WOMENS SPORTS	107
...	
9 rows selected.	

ORD

CUSTID	ORDID
-----	-----
101	610
102	611
104	612
106	601
102	602
106	
106	
21 rows	

ITEM

ORDID	ITEMID
-----	-----
610	3
611	1
612	1
601	1
602	1
64 rows selected.	

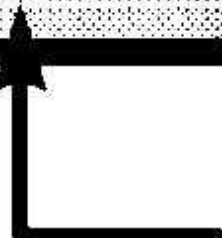
Внешние соединения

EMP

ENAME	DEPTNO
-----	-----
KING	10
BLAKE	30
CLARK	10
JONES	20
...	

DEPT

DEPTNO	DNAME
-----	-----
10	ACCOUNTING
30	SALES
10	ACCOUNTING
20	RESEARCH
...	
40	OPERATIONS



**В отделе OPERATIONS
служащих нет**

Внешние соединения

- Внешнее соединение используется для выборки строк, не удовлетворяющих обычным условиям соединения.
- Оператором внешнего соединения является знак плюс (+).

```
SELECT table.column, table.column  
FROM table1, table2  
WHERE table1.column(+) = table2.column;
```

```
SELECT table.column, table.column  
FROM table1, table2  
WHERE table1.column = table2.column(+);
```

Использование внешних соединений

```
SQL> SELECT   e.ename, d.deptno, d.dname
  2  FROM      emp e, dept d
  3  WHERE     e.deptno(+) = d.deptno
  4  ORDER BY e.deptno;
```

ENAME	DEPTNO	DNAME
-----	-----	-----
KING	10	ACCOUNTING
CLARK	10	ACCOUNTING
...		
	40	OPERATIONS

15 rows selected.

Темы

- **Общие сведения об имеющихся групповых функциях**
- **Использование групповых функций**
- **Вывод данных по группам с помощью предложения GROUP BY**
- **Включение и исключение групп с помощью предложения HAVING**

Что такое групповые функции?

Групповые функции работают с множеством строк и возвращают один результат на группу.

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

“максимальный
оклад в
таблице EMP”

MAX (SAL)

5000

Типы групповых функций

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **SUM**

Использование групповых функций

```
SELECT      column, group_function(column)  
FROM        table  
[WHERE      condition]  
[ORDER BY  column];
```

Использование функций AVG и SUM

Функции AVG и SUM применяются к столбцам с числовыми данными

```
SQL> SELECT  AVG(sal), MAX(sal),  
2           MIN(sal), SUM(sal)  
3 FROM      emp  
4 WHERE     job LIKE 'SALES%';
```

AVG (SAL)	MAX (SAL)	MIN (SAL)	SUM (SAL)
1400	1600	1250	5600

Использование функций MIN и MAX

Функции MAX и MIN применяются к данным любого типа

```
SQL> SELECT MIN(hiredate), MAX(hiredate)
2 FROM emp;
```

MIN(HIRED	MAX(HIRED
17-DEC-80	12-JAN-83

Использование функции COUNT

COUNT(*) возвращает количество строк в таблице

```
SQL> SELECT COUNT (*)  
2 FROM emp  
3 WHERE deptno = 30;
```

```
COUNT (*)  
-----
```

```
6
```

Использование функции COUNT

COUNT(*expr*) возвращает количество строк с определенными значениями (не NULL)

```
SQL> SELECT COUNT(comm)
      2 FROM emp
      3 WHERE deptno = 30;
```

```
COUNT (COMM)
-----
4
```

Создание групп данных

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

2916.6667

2175

1566.6667

“средний
оклад
в таблице
EMP
по каждому
отделу”

DEPTNO	AVG (SAL)
10	2916.6667
20	2175
30	1566.6667

Создание групп данных: предложение GROUP BY

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[ORDER BY  column];
```

Предложение GROUP BY разбивает строки
таблицы на группы

Использование предложения GROUP BY

Все столбцы, которые входят в список SELECT и к которым не применяются групповые функции, должны быть указаны в предложении GROUP BY.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667

Использование предложения GROUP BY

Столбец, указанный в GROUP BY, может отсутствовать в списке SELECT.

```
SQL> SELECT      AVG(sal)
  2  FROM          emp
  3  GROUP BY deptno;
```

```
AVG (SAL)
-----
2916.6667
      2175
1566.6667
```

Группировка по нескольким столбцам

EMP

DEPTNO	JOB	SAL
10	MANAGER	2450
10	PRESIDENT	5000
10	CLERK	1300
20	CLERK	800
20	CLERK	1100
20	ANALYST	3000
20	ANALYST	3000
20	MANAGER	2975
30	SALESMAN	1600
30	MANAGER	2850
30	SALESMAN	1250
30	CLERK	950
30	SALESMAN	1500
30	SALESMAN	1250

“просуммировать
оклады в
таблице EMP
по каждой
должности
внутри каждого
отдела”

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
20	MANAGER	2975
30	CLERK	950
30	MANAGER	2850
30	SALESMAN	5600

Использование предложения GROUP BY

```
SQL> SELECT deptno, job, sum(sal)
2 FROM emp
3 GROUP BY deptno, job;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900

9 rows selected.

Недействительные запросы с групповыми функциями

Все столбцы и выражения из списка SELECT, не являющиеся групповой функцией, должны быть включены в предложение GROUP BY.

```
SQL> SELECT deptno, COUNT(ename)
2 FROM emp;
```

```
SELECT deptno, COUNT(ename)
```

```
*
```

```
ERROR at line 1:
```

```
ORA-00937: not a single-group group function
```

В предложении GROUP BY недостает столбца

Недействительные запросы с групповыми функциями

- Предложение WHERE для исключения групп не используется.
- Для исключения некоторых групп следует пользоваться предложением HAVING

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 WHERE AVG(sal) > 2000
4 GROUP BY deptno;
```

```
WHERE AVG(sal) > 2000
*
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Исключение групп

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

5000

3000

2850

“максимальный
оклад
в отделе
превышает
\$2900”

DEPTNO	MAX(SAL)
10	5000
20	3000

Исключение групп: предложение HAVING

Для исключения групп пользуйтесь предложением
HAVING

- Строки группируются.
- Применяется групповая функция.
- Выводятся группы, удовлетворяющие условию в предложении HAVING.

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY  group_by_expression]
[HAVING    group_condition]
[ORDER BY  column];
```


Использование предложения HAVING

```
SQL> SELECT deptno, max(sal)
2 FROM emp
3 GROUP BY deptno
4 HAVING max(sal) > 2900;
```

DEPTNO	MAX (SAL)
10	5000
20	3000

Использование предложения HAVING

```
SQL> SELECT      job, SUM(sal) PAYROLL
  2 FROM          emp
  3 WHERE         job NOT LIKE 'SALES%'
  4 GROUP BY     job
  5 HAVING        SUM(sal)>5000
  6 ORDER BY     SUM(sal);
```

JOB	PAYROLL
ANALYST	6000
MANAGER	8275

Вложенные групповые функции

Вывод максимального среднего оклада.

```
SQL> SELECT max(avg(sal))  
2 FROM emp  
3 GROUP BY deptno;
```

```
MAX(AVG(SAL))  
-----  
2916.6667
```

Заключение

```
SELECT      column, group_function (column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

Последовательность оценки предложений:

- **WHERE**
- **GROUP BY**
- **HAVING**

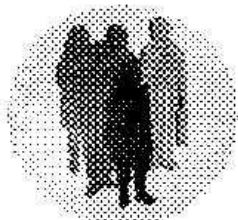
Подзапросы

- **Типы проблем, решаемых с помощью подзапросов**
- **Определение подзапросов**
- **Типы подзапросов**
- **Написание однострочных и многострочных подзапросов**

Использование подзапроса для решения проблемы

“У кого оклад больше, чем у Джонса?”

Главный запрос



“У кого из служащих оклад больше, чем у Джонса?”

Подзапрос



“Каков оклад Джонса?”



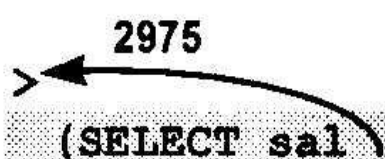
Подзапросы

```
SELECT  select_list
FROM    table
WHERE   expr operator
        (SELECT  select_list
         FROM    table);
```

- Подзапрос (внутренний запрос) выполняется один раз до главного запроса.
- Результат подзапроса используется главным запросом (внешним запросом).

Использование подзапроса

```
SQL> SELECT  ename
2 FROM      emp
3 WHERE     sal >
4           (SELECT sal
5            FROM   emp
6            WHERE  empno=7566) ;
```

A diagram consisting of a curved arrow pointing from the value '2975' to the '>' operator in the WHERE clause of the main query. The subquery text '(SELECT sal FROM emp WHERE empno=7566)' is highlighted with a grey background.

ENAME

KING

FORD

SCOTT

Указания по использованию подзапросов

- **Подзапрос должен быть заключен в скобки.**
- **Подзапрос должен находиться справа от оператора сравнения.**
- **Подзапрос не может содержать предложение ORDER BY.**
- **В однострочных подзапросах используются однострочные операторы.**
- **В многострочных подзапросах используются многострочные операторы.**

Типы подзапросов

- Однострочный подзапрос



- Многострочный подзапрос



- Многостолбцовый подзапрос



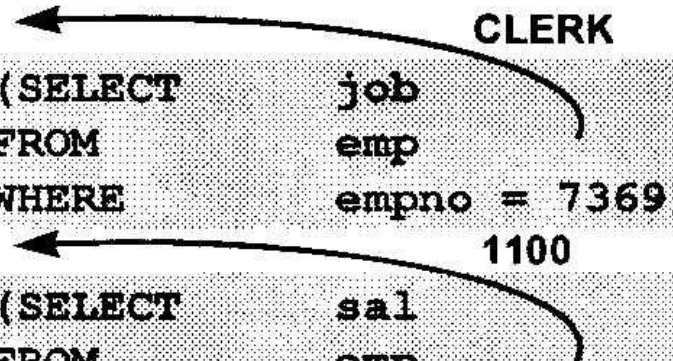
Однострочные подзапросы

- Возвращают только одну строку
- Используют однострочные операторы сравнения

Оператор	Значение
=	Равно
>	Больше, чем
>=	Больше или равно
<	Меньше, чем
<=	Меньше или равно
<>	Не равно

Выполнение однострочных подзапросов


```
SQL> SELECT  ename, job
2  FROM      emp
3  WHERE     job =
4             (SELECT  job
5             FROM      emp
6             WHERE     empno = 7369)
7  AND       sal >
8             (SELECT  sal
9             FROM      emp
10            WHERE     empno = 7876);
```



ENAME	JOB
-----	-----
MILLER	CLERK

Использование групповых функций в подзапросах

```
SQL> SELECT  ename, job, sal
2 FROM      emp
3 WHERE     sal =
4           (SELECT  MIN(sal)
5            FROM      emp);
```

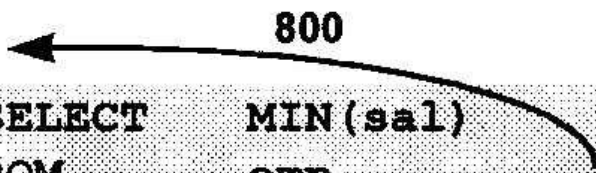


ENAME	JOB	SAL
-----	-----	-----
SMITH	CLERK	800

Предложение HAVING с подзапросами

- Сервер Oracle сначала выполняет подзапрос.
- Сервер Oracle возвращает результаты в предложение HAVING главного запроса.

```
SQL> SELECT      deptno, MIN(sal)
  2  FROM        emp
  3  GROUP BY    deptno
  4  HAVING      MIN(sal) >
  5              (SELECT  MIN(sal)
  6                  FROM    emp
  7                  WHERE   deptno = 20);
```



The diagram illustrates the execution of the HAVING clause. A curved arrow points from the value '800' to the comparison operator '>' in the HAVING clause, indicating that the result of the subquery is used to evaluate the condition.

Что неправильно в этой команде?

```
SQL> SELECT empno, ename
2 FROM emp
3 WHERE sal =
4         (SELECT MIN(sal)
5         FROM emp
6         GROUP BY deptno);
```

```
ERROR:
ORA-01427: single-row subquery returns more than
one row

no rows selected
```

Будет ли выполнена эта команда?

```
SQL> SELECT ename, job
2 FROM emp
3 WHERE job =
4           (SELECT job
5            FROM emp
6            WHERE ename='SMYTHE');
```

```
no rows selected
```

Подзапрос не возвращает никаких значений

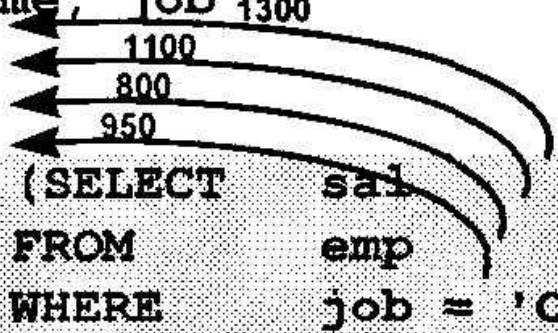
Многострочные запросы

- Возвращают более одной строки
- Используют многострочные операторы сравнения

Оператор	Значение
IN	Равно любому члену списка
ANY	Сравнение значения с любым значением, возвращаемым подзапросом
ALL	Сравнение значения с каждым значением, возвращаемым подзапросом

Использование оператора ANY в многострочных подзапросах

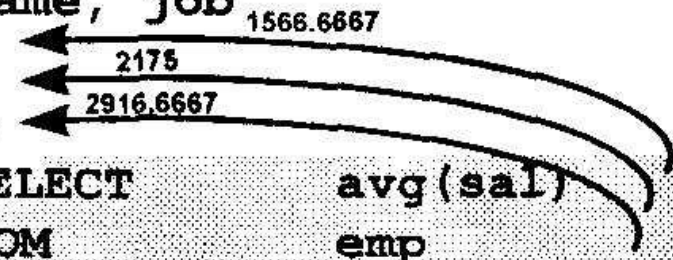
```
SQL> SELECT empno, ename, job 1300
2 FROM emp 1100
3 WHERE sal < ANY 800
4 (SELECT sal 950
5 FROM emp
6 WHERE job = 'CLERK')
7 AND job <> 'CLERK';
```



EMPNO	ENAME	JOB
7654	MARTIN	SALESMAN
7521	WARD	SALESMAN

Использование оператора ALL в многострочных подзапросах

```
SQL> SELECT empno, ename, job
2 FROM emp
3 WHERE sal > ALL
4 (SELECT avg(sal)
5 FROM emp
6 GROUP BY deptno);
```



EMPNO	ENAME	JOB
7839	KING	PRESIDENT
7566	JONES	MANAGER
7902	FORD	ANALYST
7788	SCOTT	ANALYST

Заключение

Подзапросы полезны, когда запрос основан на неизвестных значениях.

```
SELECT  select_list
FROM    table
WHERE   expr operator
        (SELECT select_list
         FROM   table);
```

Неопределенные значения в подзапросе

```
SQL> SELECT  employee.ename
  2  FROM    emp employee
  3  WHERE   employee.empno NOT IN
  4          (SELECT manager.mgr
  5           FROM   emp manager);
no rows selected.
```

Использование подзапроса в предложении FROM

```
SQL> SELECT a.ename, a.sal, a.deptno, b.salavg
 2 FROM emp a, (SELECT deptno, avg(sal) salavg
 3 FROM emp
 4 GROUP BY deptno) b
 5 WHERE a.deptno = b.deptno
 6 AND a.sal > b.salavg;
```

ENAME	SAL	DEPTNO	SALAVG
KING	5000	10	2916.6667
JONES	2975	20	2175
SCOTT	3000	20	2175

...

6 rows selected.

Заключение

- **Многостолбцовый подзапрос возвращает значения нескольких столбцов.**
- **Сравнение столбцов в многостолбцовых подзапросах может быть парным и непарным.**
- **Многостолбцовый подзапрос может также использоваться в предложении FROM команды SELECT.**