SQL (диалект Oracle). Выборки с помощью SELECT

лекция-семинар

1. Фильтрация (where) и сортировка (order by)
2. Многотабличные запросы, inner join
3. Single row functions. Псевдонимы (alias)
4. Group functions, группировка (group by, having)
5. Внешние соединения (left, right, full outer join)
6. Операции над выборками (union, minus, intersect)
7. Подзапросы, в т.ч. многострочные (in, any, all)
8. Иерархические запросы в Oracle (connect by)

1. Простейшие запросы

select table_name from user_tables

TABLE_NAME
EMPLOYEE
DEPARTMENT

select sysdate from dual

SYSDATE
11/16/2007 23:00:11

общий вид простых запросов:

```
SELECT [DISTINCT] {*, column [alias],...}
[WHERE … [AND …] [OR …]]
FROM   table;
```

DISTINCT

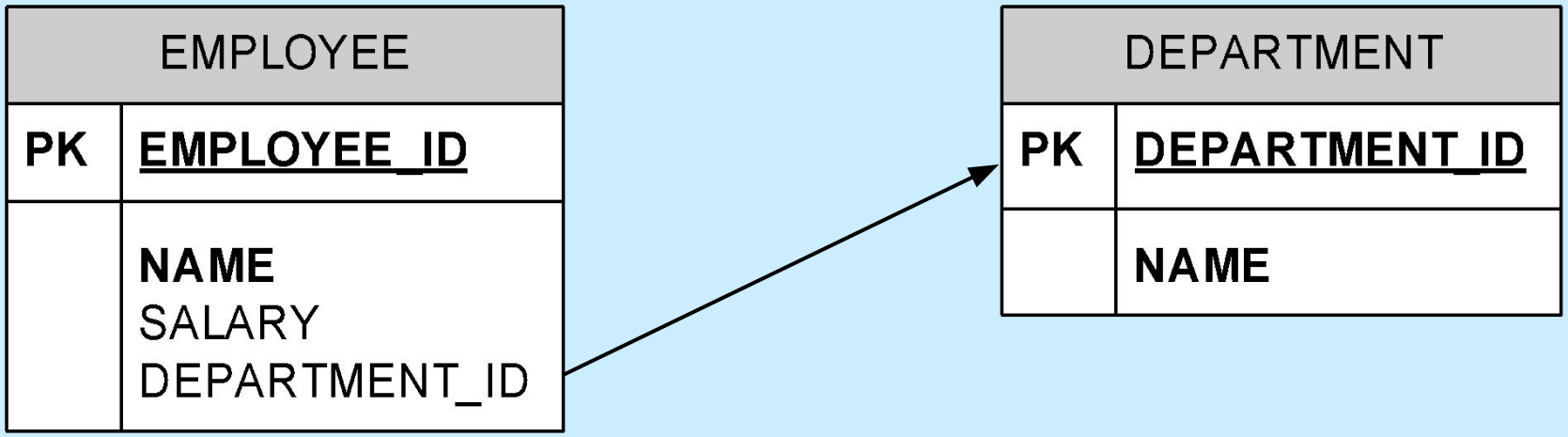
- Выражение вида SELECT DISTINCT позволяет выбрать только уникальные записи

(Использовать только тогда, когда это действительно необходимо: сильно падает скорость)

select distinct salary from employee

SALARY
1500
2000
2500
3000
3500
5000

Структура используемых таблиц



Наполнение используемых таблиц

EMPLOYEE_ID	NAME	SALARY	DEPARTMENT_ID
1	Ivan Ivanov	2000	2
2	Petr Sidorov	3000	1
3	Andrey Petrov	1500	1
4	Sergey Vasiliev	1500	
5	Oleg Fedorov	2500	1
6	Mihail Hohlov	2000	3
7	Alexey Alexeev	3500	2
8	Sergey Frolov	5000	1
9	Ivan Stulov	2000	3
10	Maxim Hlebnikov	2500	

DEPARTMENT_ID	NAME
1	R&D
2	Sales
3	QA
4	IT

Выражения, ограничивающие набор строк (WHERE)

```
select name, salary
from employee
where department_id is null
```

NAME	SALARY
Sergey Vasiliev	1500
Maxim Hlebnikov	2500

```
select name, salary
from employee
where salary between 2500 and 3000
```

NAME	SALARY
Petr Sidorov	3000
Oleg Fedorov	2500
Maxim Hlebnikov	2500

```
select name, salary
from employee
where salary not in (2000, 2500, 3000, 3500)
```

NAME	SALARY
Andrey Petrov	1500
Sergey Vasiliev	1500
Sergey Frolov	5000

Сортировка (ORDER BY)

- DESC – сортировка в направлении убывания

```
select salary, name
from employee
order by salary, name
```

SALARY	NAME
1500	Andrey Petrov
1500	Sergey Vasiliev
2000	Ivan Ivanov
2000	Ivan Stulov
2000	Mihail Hohlov
2500	Maxim Hlebnikov
2500	Oleg Fedorov
3000	Petr Sidorov
3500	Alexey Alexeev
5000	Sergey Frolov

```
select e.salary, e.name
from employee e    //e - alias
order by e.salary desc, e.name
```

SALARY	NAME
5000	Sergey Frolov
3500	Alexey Alexeev
3000	Petr Sidorov
2500	Maxim Hlebnikov
2500	Oleg Fedorov
2000	Ivan Ivanov
2000	Ivan Stulov
2000	Mihail Hohlov
1500	Andrey Petrov
1500	Sergey Vasiliev

- Можно использовать функцию вместо атрибута сортировки после order by
  (а также вместо выбираемого атрибута после select и в условии отбора после where)

```
select salary, name
from employee
order by dbms_random.random //это функция в пакете
```

SALARY	NAME
1500	Andrey Petrov
1500	Sergey Vasiliev
2000	Ivan Stulov
3500	Alexey Alexeev
2500	Maxim Hlebnikov
3000	Petr Sidorov
5000	Sergey Frolov
2000	Ivan Ivanov
2000	Mihail Hohlov
2500	Oleg Fedorov

SALARY	NAME
2500	Oleg Fedorov
2000	Mihail Hohlov
3000	Petr Sidorov
2000	Ivan Stulov
2000	Ivan Ivanov
2500	Maxim Hlebnikov
1500	Andrey Petrov
1500	Sergey Vasiliev
5000	Sergey Frolov
3500	Alexey Alexeev

2. Многотабличные запросы. Соединение таблиц (Inner Join)

```
select e.name, d.name department //department – alias столбца
from employee e, department d //e, d – alias таблиц
where e.department_id = d.department_id   order by department
```

NAME	DEPARTMENT
Mihail Hohlov	QA
Ivan Stulov	QA
Petr Sidorov	RD
Andrey Petrov	RD
Sergey Frolov	RD
Oleg Fedorov	RD
Ivan Ivanov	Sales
Alexey Alexeev	Sales

- *Упражнение 1: Выполнить аналогичный запрос с выдачей зарплат, причем зарплаты <> 5000 (AND)*

Функции

- Есть два типа функций (имена – из Oracle):
  - Single Row Functions – функции, применяемые к конкретной строке выборки (SIN, ROUND, DECODE, NVL, SUBSTR, ..)
  - Group Functions – функции, применяющиеся к некоторому подмножеству выборки (COUNT, AVG, STDDEV, MAX, MIN)

3. Single Row Functions

- Функции, применяемые к одной записи (точнее, к одной ячейке) из выборки
- Типы Single Row Functions:
  - Математические функции: SIN, EXP, MOD, …
  - Функции для работы со строками, датами и др.: SUBSTR, LOWER, LPAD, NEXT_DATE, …
  - Функции преобразования типов: TO_DATE, TO_CHAR, TO_NUMBER, …
  - Функции, переопределяющие значения: DECODE, NVL, …

Математические функции

```
select name, mod(salary, 1000) mod
from employee
where name like '%Petrov' //like – сравнение строк по маске
```

NAME	MOD
Andrey Petrov	500

```
select sin(1) from dual
```

SIN(1)
0.84

Функции работы со строками

- *Упражнение 2: в запросе упражнения 1 выдавать имена с большой буквы (остальные - строчные), независимо от того, как они хранятся в БД. Применить SUBSTR, UPPER, LOWER, CONCAT (или оператор ||)*

Функции переопределения значения

```
select name, salary, nvl(department_id, 0) department_id
from employee
```

NAME	SALARY	DEPARTMENT_ID
Ivan Ivanov	2000	2
Petr Sidorov	3000	1
Andrey Petrov	1500	1
Sergey Vasiliev	1500	0
Oleg Fedorov	2500	1
Mihail Hohlov	2000	3
Alexey Alexeev	3500	2
Sergey Frolov	5000	1
Ivan Stulov	2000	3
Maxim Hlebnikov	2500	0

```
select name, decode (salary, 1500, 'Good', 2000, 'Very Good', 'Cool!')
    status
from employee
```

NAME	STATUS
Ivan Ivanov	Very good
Petr Sidorov	Cool!
Andrey Petrov	Good
Sergey Vasiliev	Good
Oleg Fedorov	Cool!
Mihail Hohlov	Very good
Alexey Alexeev	Cool!
Sergey Frolov	Cool!
Ivan Stulov	Very good
Maxim Hlebnikov	Cool!

4. Group Functions. Выражение GROUP BY

- Выражение GROUP BY используется для разбиения выборки на группы с равными значениями в заданном(ых) столбце(ах)
- Групповая функция (COUNT, AVG, …) – для подсчета одного числа по каждой группе (или по всей выборке, если group by не задано)

```
select salary, count(*) number
from employee
group by salary

select avg(salary) average_salary
from employee
```

SALARY	NUMBER
1500	2
2000	3
2500	2
3000	1
3500	1
5000	1

```
select department_id, max(salary) max_salary,
  min(salary) min_salary
from employee
group by department_id
```

DEPARTMENT_ID	MAX_SALARY	MIN_SALARY
1	5000	1500
2	3500	2000
3	2000	2000
	2500	1500

- *Упражнение 3: на основе запроса упражнения 1 подсчитать среднюю зарплату по каждому отделу (выдавать название отдела, а не его id).*
- *Упражнение 4 (с outer join, см. далее): –//–, но также выдать ср. зарплату сотрудников без отдела (применить функцию NVL для названия несуществующего отдела)*

Условие HAVING

```
select department_id, avg(salary)
from employee
group by department_id
having max(salary) > 2000
```

DEPARTMENT_ID	AVG(SALARY)
1	3000
2	2750
	2000

- Если требуется отфильтровать строки до группировки – where, если после группировки – having

5. Внутр. и внешние соединения

Две формы записи Inner Join

Используется обычно:

```
select e.name, d.name department
from employee e, department d
where e.department_id = d.department_id
```

Стандарт ANSI:

```
select e.name, d.name department
from employee e
inner join department d
on e.department_id = d.department_id
```

NAME	DEP…
Sergey Frolov	RD
Oleg Fedorov	RD
Andrey Petrov	RD
Petr Sidorov	RD
Alexey Alexeev	Sales
Ivan Ivanov	Sales
Ivan Stulov	QA
Mihail Hohlov	QA

Left Outer Join (внешнее соединение)

Стандарт ANSI:

```
select e.name, d.name department
from employee e
left outer join department d
on e.department_id = d.department_id
```

NAME	DEP…
Sergey Frolov	RD
Oleg Fedorov	RD
Andrey Petrov	RD
Petr Sidorov	RD
Alexey Alexeev	Sales
Ivan Ivanov	Sales
Ivan Stulov	QA
Mihail Hohlov	QA
Maxim Hlebnikov	
Sergey Vasiliev	

Используется в Oracle:

```
select e.name, d.name department
from employee e, department d
where e.department_id = d.department_id(+)
```

Right Outer Join

Стандарт ANSI:

```
select e.name, d.name department
from employee e
right outer join department d
on e.department_id = d.department_id
```

NAME	DEP…
Sergey Frolov	RD
Oleg Fedorov	RD
Andrey Petrov	RD
Petr Sidorov	RD
Alexey Alexeev	Sales
Ivan Ivanov	Sales
Ivan Stulov	QA
Mihail Hohlov	QA
	IT

Используется в Oracle:

```
select e.name, d.name department
from employee e, department d
where e.department_id(+) = d.department_id
```

Full Outer Join

Стандарт ANSI:

```
select e.name, d.name department
from employee e
full outer join department d
on e.department_id = d.department_id
```

NAME	DEP…
Sergey Frolov	RD
Oleg Fedorov	RD
Andrey Petrov	RD
Petr Sidorov	RD
Alexey Alexeev	Sales
Ivan Ivanov	Sales
Ivan Stulov	QA
Mihail Hohlov	QA
Maxim Hlebnikov	
Sergey Vasiliev	
	IT

Так неправильно!:

```
select e.name, d.name department
from employee e, department d
where e.department_id(+) = d.department_id(+)
```

6. Теоретико-множественные операции над выборками

Объединение множеств:

```
select name, salary from employee
where department_id=2      union [all]
select name, salary from employee
where department_id is null
```

Вычитание множеств:

```
select department_id from department    minus
select department_id from employee
```

- *Упражнение 5: придумать осмысленный запрос с пересечением множеств* - intersect

7. Подзапросы

Подзапрос с единственным результатом:

```
select name, salary from employee
where salary > (select salary from
employee where name='Oleg Fedorov')
```

2500

NAME	SALARY
Petr Sidorov	3000
Alexey Alexeev	3500
Sergey Stulov	5000

Многострочный подзапрос и сравнение с его результатами (in, any, all):

```
select name, salary from employee
where salary < all (select salary from
employee where department_id=2)
```

2000, 3500

NAME	SALARY
Andrey Petrov	1500
Sergey Vasiliev	1500

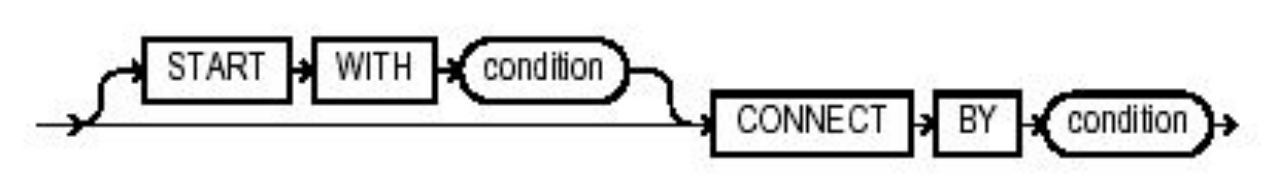
<ANY – меньше максимума; >ANY – больше минимума;

<ALL – меньше минимума; >ALL – больше максимума;

=ANY – эквивалентно IN; <>ALL – эквивалентно NOT IN…

Упражнение 6: Выбрать сотрудников, чья зарплата превышает среднюю з/п по какому-либо отделу

8. Иерархические запросы (Oracle)



- [start with условие] определяет корень(ни) дерева
- connect by и prior задает отношение parent-child
- можно использовать псевдостолбец level

Пусть в таблице employee есть столбец manager references employee(employee_id). Тогда перечисление всех сотрудников с их подчиненными (если они есть):

```
select lpad(‘ ’,3*(level-1))||name “name” from employee
connect by prior employee_id = manager_id //prior – перед PK
```

- *Упражнение 7: построить иерархию начальников-подчиненных, в которой корни – только топ-менеджеры*