



ПОСТРОЕНИЕ ЗАПРОСОВ К БД. ОПЕРАТОР SELECT

Алексей Кузьмин

Руководитель Учебного центра филиала в г.
Саратове, компания «Неофлекс»

ВАШ СОЮЗНИК В РАЗВИТИИ

Научиться строить как простые, так и сложные запросы к БД с помощью оператора SELECT

select <список_знач> from <источник>

В oracle запрещены запросы без источника данных

выход – использование в качестве источника данных таблицу-заглушку dual

Select <выражение> from dual

Примеры:



select (123+45)/56 from dual - получение значения вычисляемого выражения

Основные операторы



Операция	Оператор
Конкатенация строк:	'строка 1' 'строка 2'
логические операторы:	and, or, not
операторы сравнения:	>, <, =, >=, <=, <>
арифметические операторы:	+, -, *, /
проверка вхождение в диапазон:	between n and m
проверка вхождение в множество:	in (val1, val2...)
проверка на пустоту/непустоту:	is null / is not null
...всех:	all (например, x>all(4,5,y,z))
...одного из:	any (например, x> any(4,5,y,z))
замена в случае пустого значения:	nvl (выражение, замена)

Использование регулярных выражений

● **regex_like** (строка, регулярное_выражение)

любой символ	. (точка)
включение определённого символа	'<символ>'
один символ из набора символов	[abc] или (a b c)
один символ не из набора символов	[^abc]
один символ из диапазона символов	[a-f]
один символ не из диапазона символов	[^a-f]
буква ноль или один раз	<буква>?

Простые запросы select

<code>select * from имя_т</code>	Выборка всех строк со всеми столбцами из таблицы
<code>select <список_ст> from имя_т</code>	Выборка всех строк с некоторыми столбцами из таблицы
<code>select <список_ст> from имя_т where <условие_отбора_строк></code>	Выборка некоторых строк с некоторыми столбцами из таблицы

Использование псевдонимов столбцов:

```
select <имя_ст> as <псевдоним> from <источник>
```

Удаление/оставление повторяющихся записей в итоговом наборе:

```
select distinct <список_ст> from <источник>
```

```
select all <список_ст> from <источник> (по умолч.)
```

Сортировка по столбцу (ASC – по возрастанию(умолч), DESC – по убыванию)

```
select <список ст> from <источник> where <условие> order by <имя_ст> ASC
```

Сортировка по нескольким столбцам

```
select <список ст> from <источник> where <условие> order by <имя_ст> ASC, <имя_ст2> DESC
```

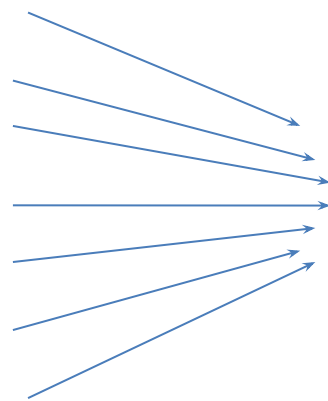
Агрегатная функция – функция, принимающая неопределённое

Описание	Агрегатная функция
К количество значений в столбце	count (<ст>)
количество строк в наборе	count (*)
сумма по столбцу	sum (<ст>)
минимум по столбцу	min(<ст>)
максимум по столбцу	max(<ст>)
среднее значение по столбцу	avg (<ст>)
стандартное отклонение	stdev(<ст>)
дисперсия	variance (<ст>)

Использование агрегатных функций

Фамилия	Отдел	Зарплата	
Иванов	IT	25000	
Петров	QA	34000	
Сидоров	IT	40000	
Захаров	QA	25000	
Мышин	QA	30000	
Орлов	IT	34000	
Соколов	IT	45000	

средняя зп
33286



```
select avg (Зарплата) as 'средняя зп' from  
Сотрудники
```

Использование группировки



Группировка – объединение исходного набора записей в группы

Фамилия

Отдел

Зарплата

записей по указанному критерию и вывода по одной итоговой строке

для каждой группы

Сидоров

IT

40000

Захаров

QA

25000

Мышин

QA

30000

Орлов

IT

34000

Соколов

IT

45000

Отдел

средняя зп

IT

36000

QA

29667

```
select Отдел, avg (Зарплата) as "средняя зп" from Сотрудники group by
```

Отдел

Использование группировки

Группировка возможна по нескольким столбцам

Фамилия	Отдел	Зарплата	Отдел	Фамилия	кол-во
Иванов	IT	25000	IT	Иванов	1
Петров	QA	34000	IT	Сидоров	2
Сидоров	IT	40000	IT	Захаров	1
Захаров	QA	25000	QA	Петров	1
Иванов	QA	30000	QA	Захаров	1
Сидоров	IT	34000	QA	Иванов	1
Захаров	IT	45000	QA		

```
select Фамилия, Отдел, count (*) as кол-во from Сотрудники group by Отдел, Фамилия
```

Фильтрация групп

оператор **having** – задаёт условие отбора групп

Фамилия	Отдел	Зарплата
Иванов	IT	25000
Петров	QA	34000
Сидоров	IT	40000
Захаров	QA	25000
Иванов	QA	30000
Сидоров	IT	34000
Захаров	IT	45000

Отдел	Фамилия	кол-во
IT	Сидоров	2

```
select Фамилия, Отдел, count (*) as кол-во from Сотрудники group by Отдел,  
Фамилия having кол-во >1
```



Предложение **rollup**

Включает в итоговый набор итоговое значение по каждому из уровню группировки и по всем уровням группировки



Предложение **cube**

Включает в итоговый набор итоговое значение по каждому из уровню группировки и по каждой комбинации уровней группировки

Использование множественных операторов

Оператор	Описание
union	объединение итоговых наборов (исключая повторения)
union all	объединение итоговых наборов (не исключая повторения)
intersect	пересечение итоговых наборов
minus	вычитание итоговых наборов

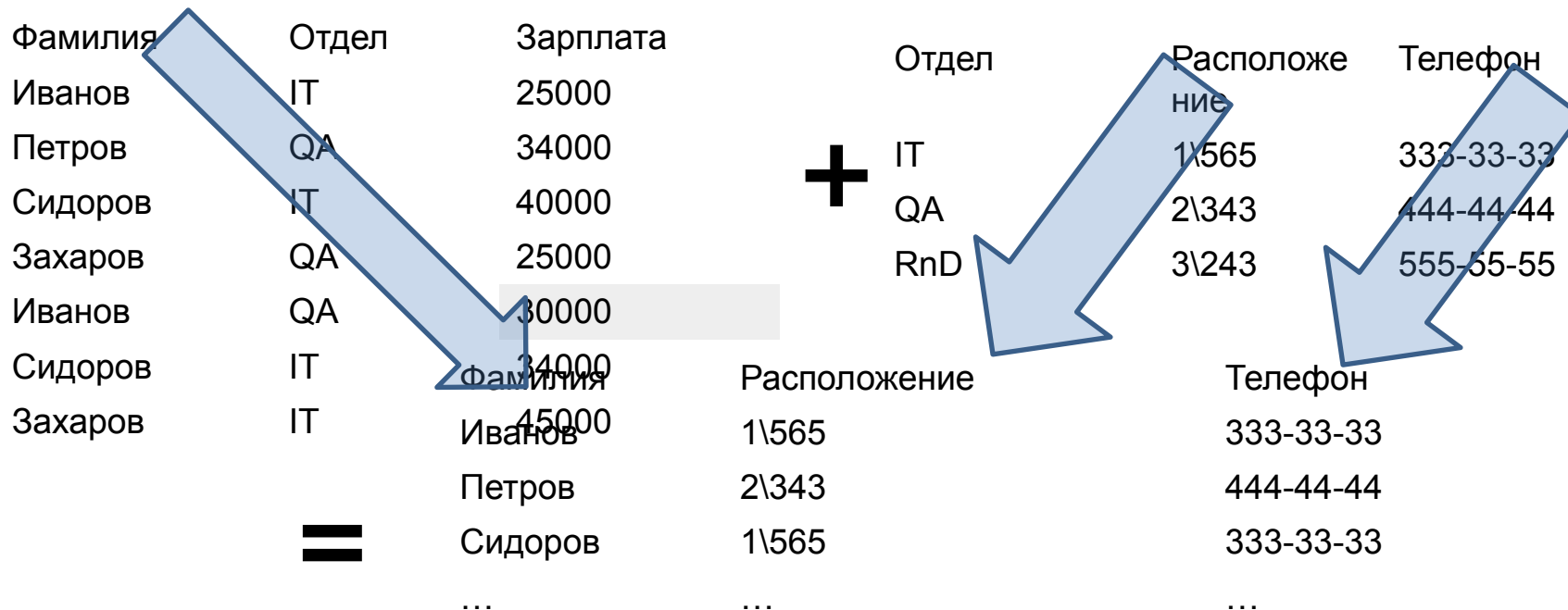
select.....

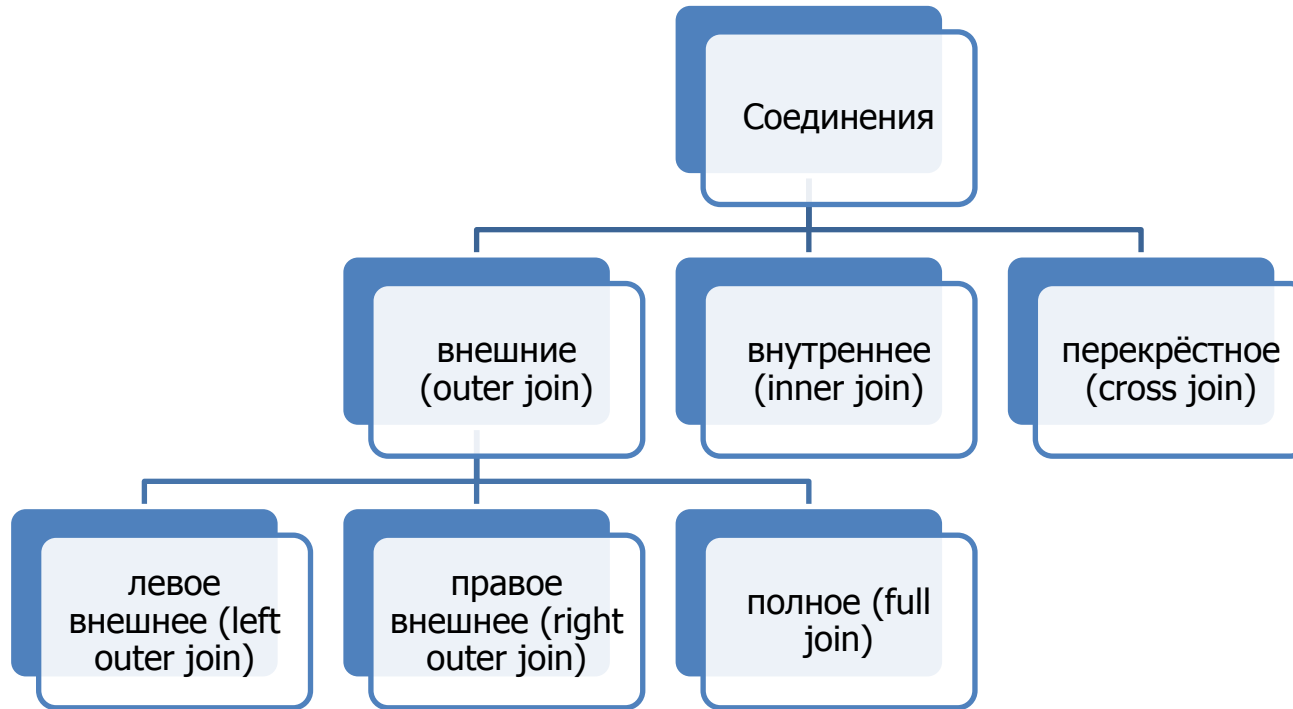
union | union all | | intersect | | minus

select.....

...

Соединения необходимы для вывода в одном итоговом наборе согласованных данных из нескольких связанных таблиц





Внутреннее соединение

Выводит в итоговый набор только те записи, у которых есть соответствующая запись в связанной таблице

Фамилия	Отдел	Зарплата	Отдел	Расположение	Телефон
Иванов	IT	25000			
Петров	IT	34000	IT	1\565	333-33-33
Сидоров	IT	40000	QA	2\343	444-44-44
Сидоров	Adm	34000	RnD	3\243	555-55-55
Захаров	Adm	45000	Расположение		Телефон
		Иванов	1\565		333-33-33
		Петров	1/565		33333333
		Сидоров	1\565		333-33-33

```
select Фамилия, Расположение, Телефон  
from Сотрудники inner join Отделы  
on Отделы.Отдел = Сотрудники.Отдел
```

Внешнее левое соединение

Выводит в итоговый набор все записи из левой (подчинённой) таблицы и только соответствующие записи из правой (главной) таблицы

Фамилия	Отдел	Зарплата	+	Отдел	Расположение	Телефон	=
Иванов	IT	25000					
Петров	IT	34000		IT	1\565	333-33-33	
Сидоров	IT	40000		<u>QA</u>	<u>2\343</u>	<u>444-44-44</u>	
Сидоров	null	34000		RnD	3\243	555-55-55	
Захаров	null	45000					
		Фамилия		Расположение		Телефон	
		Иванов		1\565		333-33-33	
		Петров		2\565		33333333	
		Сидоров		1\565		333-33-33	
		Сидоров		null		null	
		Захаров		null		null	

```
select Отделы.Отдел, Расположение, Телефон  
from Сотрудники left outer join Отделы  
on Отделы.Отдел = Сотрудники.Отдел
```

Внешнее правое соединение

Выводит в итоговый набор все записи из правой (главной) таблицы и только соответствующие из левой (подчинённой) записи из таблицы

Фамилия	Отдел	Зарплата		Отдел	Расположение	Телефон	
Иванов	IT	25000					
Петров	QA	34000		IT	1\565	333-33-33	=
Сидоров	IT	40000	+	QA	2\343	444-44-44	
Сидоров	Adm	34000		RnD	3\243	555-55-55	
Сахаров	Adm	45000					
		Фамилия		Расположение		Телефон	
		Иванов		1\565		333-33-33	
		Петров		2\343		444-44-44	
		Сидоров		1\565		333-33-33	
		null		3\243		555-55-55	

```
select Фамилия, Расположение, Телефон  
from Сотрудники right outer join Отделы  
on Отделы.Отдел = Сотрудники.Отдел
```

Внешнее полное соединение

Комбинация внешнего правого и внешнего левого соединения

Имя	Отдел	Зарплата	Отдел	Расположение	Телефон
Иванов	IT	25000			
Петров	QA	34000	IT	1\565	333-33-33
Сидоров	IT	40000	QA	2\343	444-44-44
Сидоров	Adm	34000	RnD	3\243	555-55-55
Захаров	Adm	45000			

+

Фамилия	Расположение	Телефон
Иванов	1\565	333-33-33
Петров	2\343	444-44-44
Сидоров	1\565	333-33-33
Сидоров	null	null
Захаров	null	null
null	3\243	555-55-55

=

```
select Отделы.Отдел, Расположение, Телефон
from Сотрудники full outer join Отделы
on Отделы.Отдел = Сотрудники.Отдел
```

Перекрёстное соединение (декартово произведение)

Всевозможные комбинации записей правой и

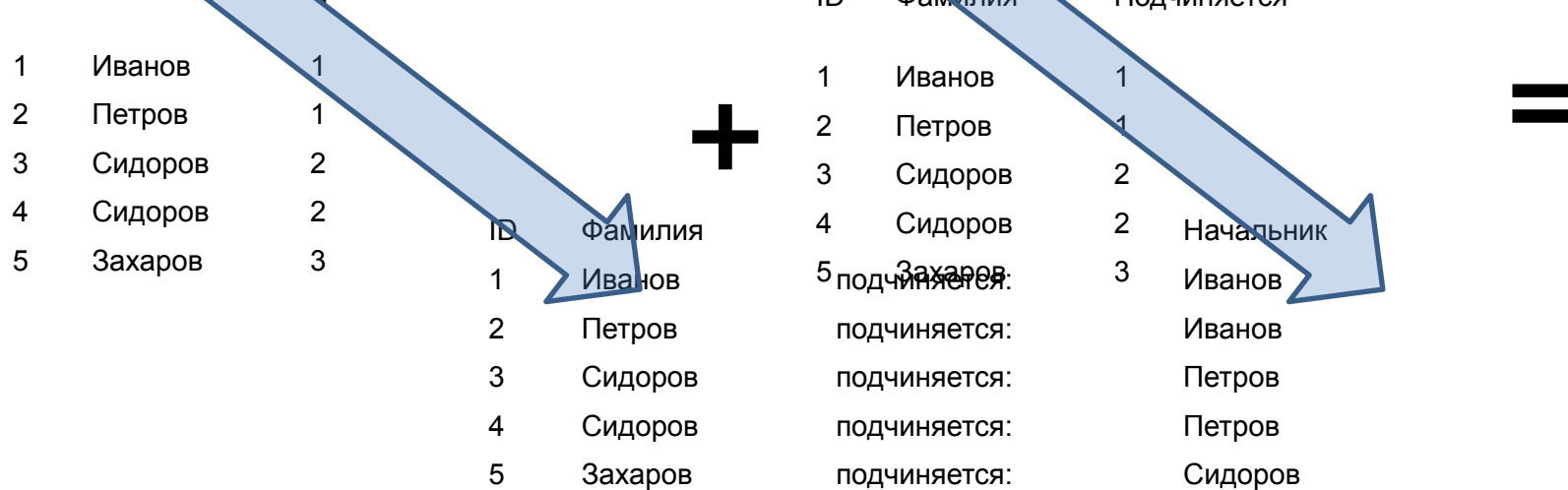
левой таблицы

Имя	Отдел	Зарплата	Отдел	Расположение	Телефон	
Иванов	IT	25000				
Петров	QA	34000				
Сидоров	IT	40000				
Сидоров	Adm	34000				
Захаров	Adm	45000				
			+			
			IT	1\565	333-33-33	
			QA	2\343	444-44-44	
			RnD	3\243	555-55-55	
			Расположение		Телефон	
				Иванов	1\565	333-33-33
				Иванов	2\343	444-44-44
				Иванов	3\243	555-55-55
				Петров	1\565	333-33-33
				Петров	2\343	444-44-44
				Петров	3\243	555-55-55
			

```
select Отделы.Отдел, Расположение, Телефон  
from Сотрудники cross join Отделы
```

Самосоединение (соединение таблицы с копией себя)

Всевозможные комбинации записей правой и левой таблицы



```
select сотрудники_подчин.Фамилия, 'подчиняется: ', сотрудники_нач.  
Фамилия
```

```
from Сотрудники as сотрудники_подчин inner join Сотрудники as
```

```
сотрудники_нач
```

Иерархические (рекурсивные)

запросы

Используются для вывода записей, состоящих в иерархических отношениях

Операторы, используемые при построении иерархических запросов:

level	возвращает уровень записи в иерархии
connect by	устанавливает условие подчинения записи-потомка другой записи-родителя
start with	устанавливает условие, позволяющее найти первую в иерархии запись
prior	ссылается на родительскую запись иерархии

```
select level, ID_сотрудника, Подчиняется from Сотрудники  
start with Подчиняется is null  
connect by prior ID_сотрудника = Подчиняется
```

- **в условии отбора записи фигурирует значение, вычисляемое подзапросом**

```
select <список_ст> from <имя_т> where ст <оператор><подзапрос>
```

```
select Название_должности from Должности  
where Оклад > (select avg (Оклад) from Должности)
```

- **в качестве источника данных служит итоговый набор подзапроса**

```
select <список_ст> from (<подзапрос> )
```


Вложенные коррелированные запросы

- Внутренний подзапрос обращается к данным внешнего запроса;
 - Внутренний подзапрос выполняется заново для каждой записи внешнего запроса
- Например**, отбираем только те отделы, в которых существуют сотрудники:

```
select * from отделы as внешняя_табл
where exists(select * from Сотрудники as внутренняя_табл
where внутренняя_табл.ID_Отдела=внешняя_табл.ID_Отдела);
```

- http://download.oracle.com/docs/cd/B19306_01/server.102/b14200/queries.htm#i2068094
- <http://sql-language.ru/>

БЛАГОДАРЮ ЗА ВНИМАНИЕ

Задавайте, пожалуйста, вопросы

Алексей Кузьмин

Руководитель Учебного центра филиала в г. Саратове,
компания «Неофлекс»

www.neoflex.ru

Телефон: 8 -962-62-11-431

E-mail: alkuzmin@neoflex.ru

Neoflex
ИТ-РЕШЕНИЯ ДЛЯ БИЗНЕСА 