



Disciplina: Banco de Dados

ADS

Condições Lógicas

Operadores	Descrição
AND	Retorna TRUE se as duas condições forem atendidas
OR	Retornará TRUE se uma das condições for verdadeira
NOT	Retorna TRUE se a condição for falsa



And

Nesse exemplo, o operador AND, força a consulta para ser obrigada a atender as duas condições estabelecidas. Para retornar algum resultado, o salário terá que ser maior que 10000 e o job_id conter a palavra %MAN%.

```
select employee_id, last_name, job_id, salary from employees  
where salary >=10000  
and job_id like '%MAN%';
```

Or

O comando OR, diferentemente do AND, necessita que somente uma das condições seja atendida.

```
select employee_id, last_name, job_id, salary from employees  
where salary >=10000  
or job_id like '%MAN%';
```

Nesse exemplo, se qualquer uma das condições forem atendidas, os resultados serão exibidos.



Regras de Precedência

Podemos também, combinar as condições AND, WHERE e OR para filtrarmos os resultados das consultas:

```
select employee_id, last_name, job_id, salary from employees  
where job_id = 'SA_REP'  
or job_id= 'AD_PRES'  
and salary > 15000;
```

Primeiramente o job_id tem que ser igual a 'AS_REP' ou 'AD_PRES', e o salário maior que 15000.



Ordenação das linhas – Order by

É possível com a função order by, ordenar os dados de maneira crescente ou decrescente:

- ASC: ordem crescente, default
- DESC: ordem decrescente

```
select last_name, job_id, department_id, hire_date  
from employees  
order by hire_date;
```



Ordenação das linhas – Order by

A cláusula order by é inserida por último nas instruções de SELECT e pode ser utilizada, combinando mais de uma coluna ou pelos seus respectivos apelidos.

```
select last_name, job_id, department_id, hire_date  
from employees  
order by hire_date, job_id desc;
```

```
select last_name, job_id, department_id "DEPARTAMENTO", hire_date  
from employees  
order by departamento desc;
```



Concatenação

É possível concatenar strings junto as colunas de uma tabela no retorno dos dados de uma consulta sql. O símbolo para utilização da concatenação é ||.

```
select '0 funcionario'||'id'||employee_id||' '||last_name||' tem o salario de'||s.salary "Concatenado" from hr.employees s
```

Annotations from left to right:

- string
- concatenação
- string
- simulação de espaço
- coluna
- coluna
- string
- coluna
- apelido da coluna

Concatenação

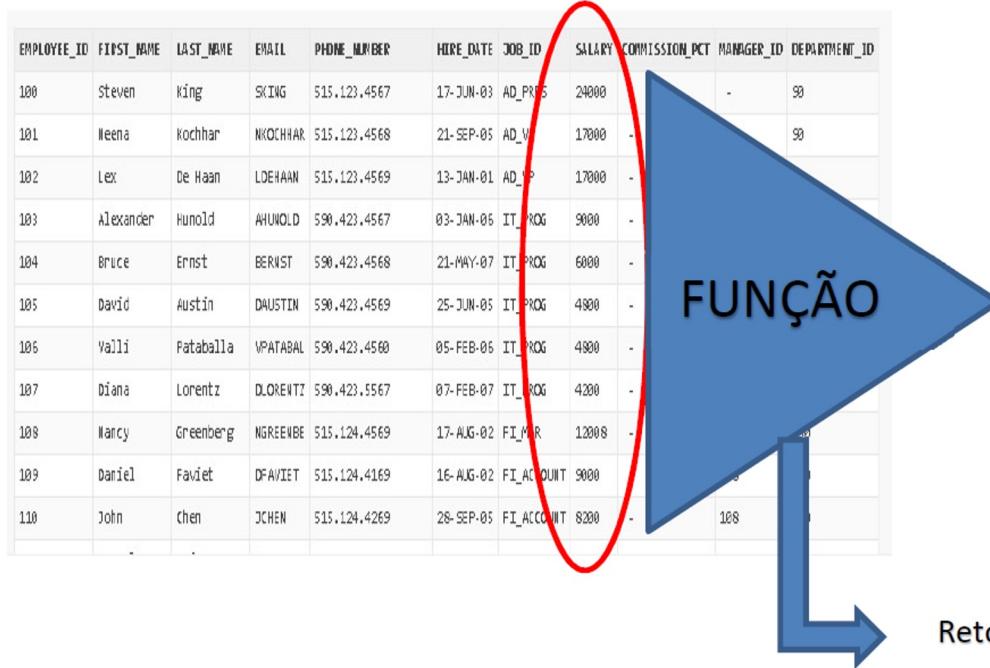
Nosso resultado sera:

	Concatenado	
1	O funcionario id 100 King tem o salario de 24000	...
2	O funcionario id 101 Kochhar tem o salario de 17000	...
3	O funcionario id 102 De Haan tem o salario de 17000	...
4	O funcionario id 103 Hunold tem o salario de 9000	...
5	O funcionario id 104 Ernst tem o salario de 6000	...
6	O funcionario id 105 Austin tem o salario de 4800	...
7	O funcionario id 106 Pataballa tem o salario de 4800	...
8	O funcionario id 107 Lorentz tem o salario de 4200	...
9	O funcionario id 108 Greenberg tem o salario de 12000	...
10	O funcionario id 109 Faviet tem o salario de 9000	...
11	O funcionario id 110 Chen tem o salario de 8200	...
12	O funcionario id 111 Sciarra tem o salario de 7700	...
13	O funcionario id 112 Urman tem o salario de 7800	...
14	O funcionario id 113 Popp tem o salario de 6900	...
15	O funcionario id 114 Raphaely tem o salario de 11000	...
16	O funcionario id 115 Khoo tem o salario de 3100	...
17	O funcionario id 116 Baida tem o salario de 2900	...
18	O funcionario id 117 Tobias tem o salario de 2800	...
19	O funcionario id 118 Wigmann tem o salario de 2600	...



Funções de Grupo

As funções de grupo são utilizadas para retornar um determinado resultado de um grupo de registros.



select max(salary) from employees

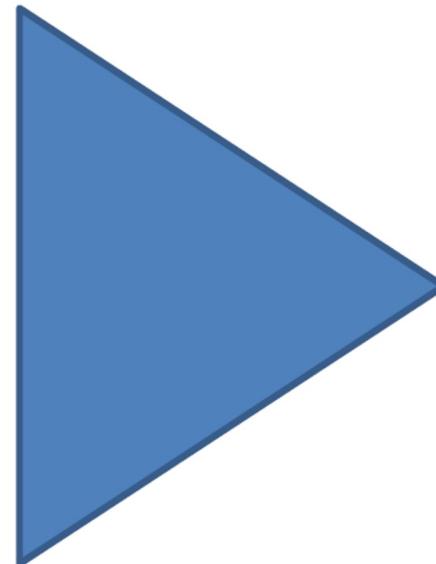
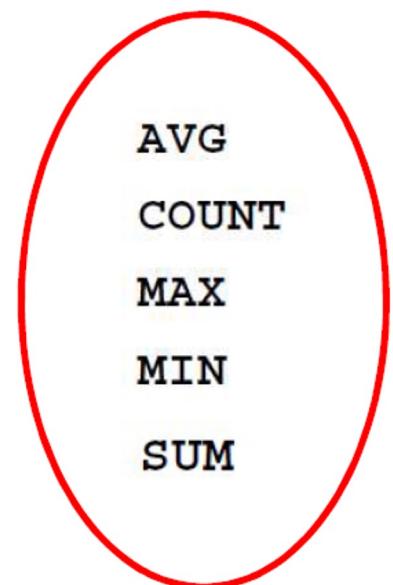
Retorna resultado de um grupo de registros



Funções de Grupo

No exemplo acima, foi utilizada a função MAX para buscar o maior salário da coluna “salary”.

As funções de grupo mais utilizadas são:



Média
Contar
Máxima
Mínima
Soma



AVG e SUM

As funções AVG, SUM são utilizadas para os dados do tipo number, e podem ser combinadas com os filtros WHERE e outras lógicas já demonstradas da linguagem SQL.

AVG e SUM

As funções MIN e MAX, além de poderem ser utilizadas nas mesmas condições das outras funções, funcionam também, com registros do tipo caractere e data. No exemplo abaixo, estamos buscando a somatória de todos os salários, o menor salário registrado, o maior salário, assim como a média salarial de todos os funcionários que sejam do departamento número 100.

```
1 select sum(salary), min(salary), max(salary), avg(salary) from hr.employees  
2 where department_id=100;
```

SUM(SALARY)	MIN(SALARY)	MAX(SALARY)	AVG(SALARY)
51608	6900	12008	8601.33

Count

A função COUNT utilizada junto ao “*” é utilizada para contar quantos registros possuem uma determinada tabela.

```
1 select count(*) from hr.employees;
```

COUNT(*)
107

Com a função count, podemos também utilizar condições lógicas para retornarmos as respectivas quantidades de registro.



Count

```
1 select count(*) from hr.employees  
2 where department_id=10
```

COUNT(*)
1

Criação de Grupo de Dados

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13000
202	Pat	Fay	PFAY	603.123.6666	17-AUG-05	MK_REP	6000
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000
115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-03	PU_CLERK	3100
116	Shelli	Baida	SBAIDA	515.127.4563	24-DEC-05	PU_CLERK	2900
117	Sigal	Tobias	STOBIAS	515.127.4564	24-JUL-05	PU_CLERK	2800
118	Guy	Himuro	GHIMURO	515.127.4565	15-NOV-06	PU_CLERK	2600
119	Karen	Colmenares	KCOLMENA	515.127.4566	10-AUG-07	PU_CLERK	2500
203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-02	HR REP	6500



DEPARTMENT_ID	TRUNC(AVG(SALARY))
10	4400
20	9500
20	4150
30	6500
30	3475
30	5760
30	10000
30	8955
30	19333
40	8601

Criação de Grupo de Dados

Salário médio por departamento

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY
200	Jennifer	Whalen	JWHALEN	515.123.4444	17-SEP-03	AD_ASST	4400
201	Michael	Hartstein	MHARTSTE	515.123.5555	17-FEB-04	MK_MAN	13800
202	Pat	Fay	PFAY	603.423.6666	17-AUG-05	MK_REP	6000
114	Den	Raphaely	DRAPHEAL	515.127.4561	07-DEC-02	PU_MAN	11000
115	Alexander	Khoo	AKHOO	515.127.4562	18-MAY-03	PU_CLERK	3100
116	Shelli	Baida	SBaida	515.127.4563	24-DEC-05	PU_CLERK	2900
117	Sigal	Tobias	STOBIA	515.127.4564	24-JUL-05	PU_CLERK	2800
118	Guy	Himuro	GHIMURO	515.127.4565	15-NOV-06	PU_CLERK	2600
119	Karen	Colmenares	KCOLMENA	515.127.4566	18-AUG-07	PU_CLERK	2500
203	Susan	Mavris	SMAVRIS	515.123.7777	07-JUN-02	HR REP	6500

DEPARTMENT_ID	TRUCK(AVG(SALARY))
10	4400
20	9500
20	4150
30	6500
30	3475
30	5760
30	18000
30	8955
30	19333
40	8601

```
1 select department_id, avg(salary) from hr.employees  
2 group by department_id;
```

* Todas as colunas que não estão utilizando funções de grupo devem estar no "group by".

```
1 select department_id, avg(salary) from hr.employees  
2 group by department_id;
```



Criação de Grupo de Dados

Agora que conhecemos as funções de grupo e sua utilização, a quantidade de combinações ficam quase ilimitadas e precisamos nos atentar quanto a ordem de utilização do GROUP BY, ORDER BY, AND, WHERE e outros.

```
1 select department_id, avg(salary) from hr.employees  
2 where department_id in (10, 20)  
3 group by department_id  
4 order by department_id desc;
```

CUIDADO!!!!

Não utilizar cláusulas WHERE para restringir funções de grupo.



Insert

O comando insert é utilizado para adicionar uma nova linha em uma tabela.

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
10	Desocupado	108	1000



Inserção de novo registo na tabela



```
insert into hr.departments (department_id, department_name, manager_id, location_id) values (10, 'Desocupado', 108, 1000)
```



Insert

Método Implícito: omite a coluna da lista de colunas

```
insert into hr.departments values (10, 'Desocupado', 108, 1000)
```

A inserção no método implícito, é utilizada quando conhecemos as colunas das tabelas e realizaremos uma inserção em todos os registros das tabelas



Insert

Método Explícito: especifica as colunas da tabela que realizaremos a inserção dos dados.

```
insert into hr.departments (department_id, department_name, manager_id, location_id) values (10, 'Desocupado', 108, 1000)|
```

O método explícito, é utilizado quando
desejamos inserir um registro em somente
uma coluna ou quando
gostaríamos de especificar as colunas que
serão inseridos os registros.

Insert

**CUIDADO!!!
Lembrem-se das características
das chaves primárias!!!!**



Exercícios

- 1) O departamento de Recursos Humanos exige alguns relatórios de alguns determinados departamentos.

Seguir os requisitos abaixo para buscar os resultados solicitados (tabela employees).

- a) Exibir o menor salário da empresa.
- b) Exibir o maior salário da empresa
- c) Exibir a média salarial da empresa.
- d) Exibir a somatória do salário agrupada por cargo
- e) Exibir a média salarial de cada departamento (department_id)
- f) Exibir a média salarial dos funcionários com ID de departamento maior que 80



Exercícios

- 2) Crie um relatório para exibir o número do gerente e o salário do funcionário com menor remuneração desse gerente. Exclua todas as pessoas cujo gerente seja desconhecido. Exclua todos os grupos em que o salário seja menor ou igual a 6000. Classifique a saída em ordem decrescente.
- 3) Inserir 10 novos funcionários na tabela Clientes criada na aula, sendo 5 no modo implícito e 5 no modo explícito. Atentar quanto a obrigatoriedade de registros.



Universidade
Cruzeiro do Sul

www.cruzeirodosul.edu.br

