

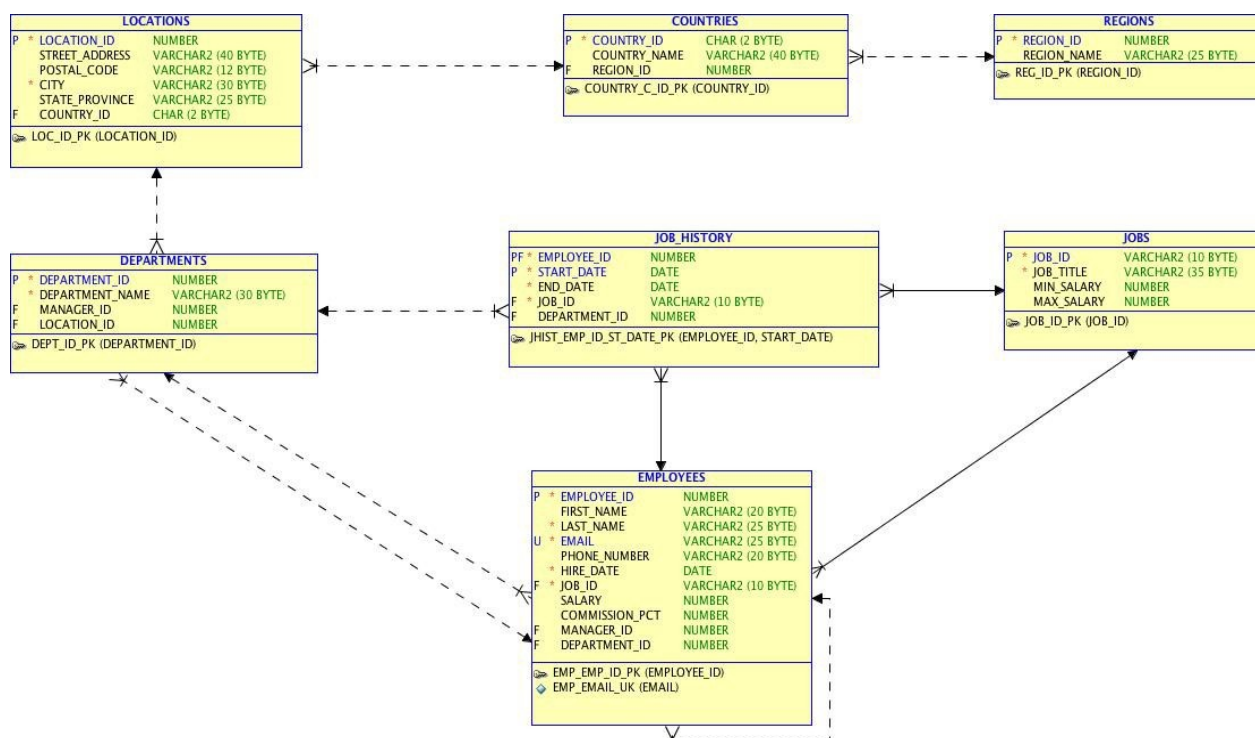
Engenharia de Computação  
BDI e Lab. BD I – 1º Semestre de 2024  
Prof. Evandrino G. Barros (evandrino@cefetmg.br)

Exercício 01 de BDI e LBDI - Funções SQL – 1 ponto para cada disciplina, em duplas.  
Divulgação: 22/03/2024 – Entrega: 22/03/2024 (até 23:59) pelo AVA de BD I  
somente. Enviar somente uma resposta por dupla.

Este exercício tem uma parte prática primeiro, na qual há uma série de exemplos de funções SQL. Na última página, encontra-se o exercício propriamente dito.

Na primeira parte, são apresentadas as principais funções SQL que rodam na maioria dos SGBDs. As interfaces das funções são do SGBD Oracle, mas podem ter outras interfaces em outros bancos e até mesmo nomes diferentes. Essas funções devem ser usadas no exercício.

Execute todos os exemplos da primeira parte. Somente o que se pede no exercício deve ser entregue. Ou seja, não inclua na solução os comandos de exemplos abaixo e nem os seus resultados.



## 1. Funções de valores simples:

- **ABS(n)**= Devolve o valor absoluto de (n).
- **CEIL(n)**=Obtém o valor inteiro imediatamente superior ou igual a "n".
- **FLOOR(n)** = Devolve o valor inteiro imediatamente inferior ou igual a "n".
- **MOD (m, n)**= Devolve o resto resultante de dividir "m" entre "n".
- **NVL (valor, expressão)**= Substitui um valor nulo por outro valor.
- **POWER (m, expoente)**= Calcula a potência de um número.
- **ROUND (numero [, m])**= Arredonda números com o número de dígitos de precisão indicados.
- **SIGN (valor)**= Indica o signo do "valor".
- **SQRT(n)**= Devolve a raiz quadrada de "n".
- **TRUNC (numero, [m])**= Trunca números para que tenham uma certa quantidade de dígitos de precisão.
- **VAIRANCE (valor)**= Devolve a média de um conjunto de valores.

Exemplos:

-- calcula o salário diário de cada empregado

```
select first_name nome, salary, salary/30 salario
from employees;
```

-- calcula o salário diário de cada empregado

-- com arredondamento para inteiros

```
select first_name nome, salary, round(salary/30) salario
from employees;
```

-- calcula o salario diario de cada empregado

-- com arredondamento para inteiros e com filtro

-- de arredondamento

```
select first_name nome, salary, round(salary/30) salario
from employees
where round(salary/30)>130;
```

## 2. Funções que devolvem valores de caracteres:

- **CHR(n)** = Devolve o caractere cujo valor em binário é equivalente a "n".
- **CONCAT (cad1, cad2)**= Devolve "cad1" concatenada com "cad2".
- **LOWER (cad)**= Devolve a cadeia "cad" em minúsculas.
- **UPPER (cad)**= Devolve a cadeia "cad" em maiúsculas.
- **INITCAP (cad)**= Converte a cadeia "cad" a tipo título.
- **LPAD (cad1, n[,cad2])**= Adiciona caracteres à esquerda da cadeia até que tenha uma certa longitude.
- **RPAD (cad1, n[,cad2])**= Adiciona caracteres à direita até que tenha uma certa longitude.
- **LTRIM (cad [,set])**= Suprime um conjunto de caracteres à esquerda da cadeia.
- **RTRIM (cad [,set])**= Suprime um conjunto de caracteres à direita da cadeia.
- **REPLACE (cad, cadeia\_busca [, cadeia\_substitucao])**= Substitui um caractere ou caracteres de uma cadeia com 0 ou mais caracteres.
- **SUBSTR (cad, m [,n])**= Obtém parte de uma cadeia.
- **TRANSLATE (cad1, cad2, cad3)**= Converte caracteres de uma cadeia em caracteres diferentes, segundo um plano de substituição marcado pelo usuário.

Exemplos:

```
-- retorna o primeiro e último nome e a data de  
-- admissão dos empregados que começam com R no  
-- ultimo nome - observe o uso do operador like
```

```
select first_name, last_name, hire_date  
from employees  
where last_name like 'R%';
```

```
-- retorna o primeiro e último nome e a data de  
-- admissão dos empregados que começam com R no  
-- último nome - observe o uso da função substr
```

```
select first_name, last_name, hire_date  
from employees  
where substr(last_name,1,1)='R';
```

```
-- retorna o primeiro e último nome e a data de  
-- admissão dos empregados começam com R no
```

```
-- último nome. Observe o uso da substr e da função  
-- upper para garantir que todos os last_name's estão  
-- em maiúsculas
```

```
select first_name, last_name from employees  
where substr(upper(last_name),1,1)='R';
```

```
-- Concatenação do primeiro nome com o último nome  
-- dos empregados que começam com R no último nome
```

```
select concat(first_name,last_name)  
from employees  
where substr(upper(last_name),1,1)='R';
```

```
-- Concatenação do primeiro nome com o último nome  
-- (separados por espaço) dos empregados que  
-- começam com R no último nome
```

```
select concat(concat(first_name,' '),last_name)  
from employees  
where substr(upper(last_name),1,1)='R';
```

```
-- Apresentando o mesmo resultado da sentença  
-- anterior, mas utilizando o operador de concatenação ||
```

```
select first_name || ' ' || last_name "Nome Completo"  
from employees  
where substr(upper(last_name),1,1)='R';
```

### 3. Funções que devolvem valores numéricos:

- **ASCII(cad)**= Devolve o valor ASCII da primeira letra da cadeia "cad".
- **INSTR (cad1, cad2 [, comeco [,m]])**= Permite uma busca de um conjunto de caracteres em uma cadeia, mas não suprime nenhum caractere depois.
- **LENGTH (cad)**= Devolve o número de caracteres de cad.

Exemplo:

```
-- Apresentando o tamanho dos nomes com a função lenght
```

```
select first_name || ' ' || last_name "Nome Completo",  
       length(first_name || ' ' || last_name) Tamanho  
from employees  
where substr(upper(last_name),1,1)='R'
```

#### 4. Funções para a manipulação de datas:

- **SYSDATE**= Devolve a data do sistema.
- **ADD\_MONTHS (data, n)**= Devolve a data "data" incrementada em "n" meses.
- **LASTDAY (data)**= Devolve a data do último dia do mês que contém "data".
- **MONTHS\_BETWEEN (data1, data2)**= Devolve a diferença em meses entre as datas "data1" e "data2".
- **NEXT\_DAY (data, cad)**= Devolve a data do primeiro dia da semana indicado por "cad" depois da data indicada por "data".

Exemplos:

-- Retorna o número de dias inteiros trabalhados  
-- por cada empregado

```
select round(sysdate - hire_date) DiasDeTrab  
from employees;
```

-- Retorna o número de meses inteiros trabalhados  
-- por cada empregado

```
select round(months_between(sysdate,hire_date))  
       MesesDeTrab,  
       hire_date DataAdmissao  
from employees;
```

#### 5. Funções de conversão:

- **TO\_CHAR**= Transforma um tipo DATE ou NUMBER em uma cadeia de caracteres.
- **TO\_DATE**= Transforma um tipo NUMBER ou CHAR em DATE.
- **TO\_NUMBER**= Transforma uma cadeia de caracteres em NUMBER

Exemplo:

```
-- Retorna a data de admissão de cada empregado
-- tanto no formato DEFAULT (dd/mm/yy) quanto
-- no formato 'DD/Month/YYYY', obtido com
-- a conversão da data com a função to_char
```

```
select hire_date DataAdmissao,
       to_char(hire_date,'DD/Month/YYYY') DiaExtenso
from employees;
```

## 6. Funções de grupos de valores:

- **AVG(n)**= Calcula o valor médio de "n" ignorando os valores nulos.
- **COUNT (\* | Expressão)**= Conta o número de vezes que a expressão avalia algum dado com valor não nulo. A opção "\*" conta todas as linhas selecionadas.
- **MAX (expressão)**= Calcula o máximo.
- **MIN (expressão)**= Calcula o mínimo.
- **SUM (expressão)**= Obtém a soma dos valores da expressão.
- **GREATEST (valor1, valor2...)**= Obtém o maior valor da lista.
- **LEAST (valor1, valor2...)**= Obtém o menor valor da lista.

Exemplo:

```
-- Exemplo de função para grupo de valores.
-- Neste caso, a soma de salários de toda a
-- tabela é retornada.
```

```
select sum(salary)
from employees;
```

```
-- Neste caso, a soma de salários é retornada por departamento.
-- Para isso, é necessário agrupar as linhas por departamento para que a
-- função seja aplicada por cada grupo obtido. O agrupamento das linhas é
--- obtido por meio da cláusula GROUP BY. Para que as somas
-- retornadas sejam associadas aos devidos
-- departamentos, deve-se projetar também a coluna usada no GROUP BY.
--
```

```
select department_id, sum(salary)
```

```
from employees  
GROUP BY department_id;
```

```
-- No SGBD Oracle, quando uma sentença tiver função de grupo, somente  
-- as colunas no group by podem ser projetadas. Mais de uma função de  
-- grupo pode ser usada no projeto, assim como mais de uma coluna  
-- ou expressão pode ser no GROUP BY.
```

```
-- para retornar a soma, o máximo e média de salário por departamento:  
select department_id, sum(salary), max(salary), avg(salary)
```

```
from employees  
GROUP BY department_id;
```

```
-- para retornar a soma e média salarial  
-- por cargo em cada departamento
```

```
select department_id, job_id, sum(salary), max(salary), min(salary)  
from employees  
GROUP BY department_id, job_id;
```

## Exercício 2 de BD I e 3 de LBD I

Resolva, as seguintes questões, apresentando o comando SQL apropriado. Também apresente os resultados obtidos. Faça isso a partir do esquema de banco de dados relacional já apresentado acima. Use o banco de dados na nuvem que já tem as tabelas disponíveis para consulta por todos os usuários do banco.

1) Apresente o primeiro e último nomes e o mês, por extenso, de admissão (coluna Hire\_Date de EMPLOYEES) de todos os empregados que começam com a letra 'R' no último nome. Não utilize LIKE, mas funções SQL. O formato de exibição por extenso das datas deve ser 'DD-Month-YYYY'. Exemplo: '01-Janeiro-2021'.

- 
- 2) Apresente o nome completo (primeiro nome concatenado com o último nome, separados por branco (use a função **concat**)) de todos os empregados que trabalham no departamento que tem “executive” (em minúsculas) no nome. O nome do departamento deve aparecer no resultado em maiúsculas. Utilize funções SQL e junções. Não utilize o operador LIKE.
  - 3) Apresente o nome completo e a data de admissão (coluna Hire\_Date de Employees) de todos os empregados que foram admitidos no ano de 1999. Use a coluna HIRE\_DATE com funções de data. O formato de exibição das datas deve ser ‘DD-Month-YYYY’. Exemplo: ‘01-janeiro-2021’.
  - 4) Apresente o nome completo, o salário diário, arredondado, e o anual de todos os empregados que ganham comissão (coluna Comission\_Pct). O salário (Salary) em EMPLOYEES é mensal para todos os empregados.
  - 5) Apresente o nome completo de todos empregados, com salário mensal e o salário mensal mais a comissão (que é um percentual mensal de 0 a 1). Caso não tenha comissão, o empregado tem a coluna Comission\_Pct igual a NULL. Empregados sem comissão devem ser incluídos no resultado, naturalmente a comissão deles é zero. Dê nomes adequados às colunas de resultado.
  - 6) Apresente somente a inicial de cada nome (primeiro e último nomes) de empregados que são gerentes de departamento. Incluir o nome do departamento.
  - 7) Apresente o nome completo de todos os empregados, a quantidade de anos e meses trabalhando na empresa. Os resultados devem ser inteiros. A data atual do SGBD é fornecida pela função SYSDATE.
  - 8) Apresente a média, arredondada, o máximo, o mínimo salário por departamento, incluindo o nome do departamento. Considere toda a tabela.
  - 9) Apresente a quantidade de empregados por departamento. Faça somente isso para empregados que têm ‘clerk’ na coluna JOB\_ID (código do cargo). O departamento deve aparecer. Utilize o operador LIKE ou a função SUBSTR, com o devido tratamento para as linhas existentes.