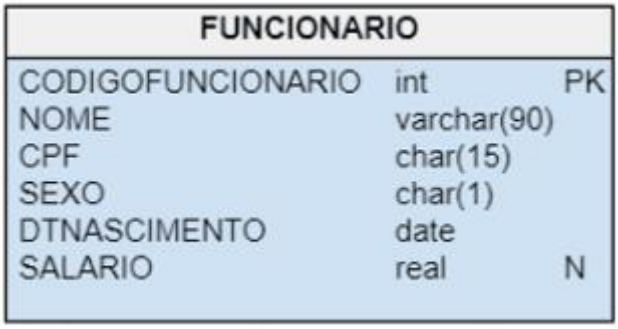
Operar consultas envolvendo agrupamento de dados

CONSULTAS COM GROUP BY E HAVING



FUNCIONARIO		
CODIGOFUNCIONARIO	int	PK
NOME	varchar(90)	
CPF	char(15)	
SEXO	char(1)	
DTNASCIMENTO	date	
SALARIO	real	N

```
CREATE TABLE FUNCIONARIO (
            CODIGOFUNCIONARIO int NOT NULL,
            NOME varchar(90) NOT NULL,
            CPF CHAR(15) NULL,
            SEXO CHAR(1) NOT NULL,
            DTNASCIMENTO date NOT NULL,
            SALARIO real NULL,
            CONSTRAINT FUNCIONARIO_PK PRIMARY KEY (CODIGOFUNCIONARIO)
);
```

```
INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,
DTNASCIMENTO, SALARIO) VALUES (1,'ROBERTA SILVA
BRASIL',NULL,'F','20/02/1980',7000);

INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,
DTNASCIMENTO, SALARIO) VALUES (2,'MARIA SILVA
BRASIL',NULL,'F','20/09/1988',9500);

INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,
DTNASCIMENTO, SALARIO) VALUES (3,'GABRIELLA PEREIRA
LIMA',NULL,'F','20/02/1990',6000);

INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,
DTNASCIMENTO, SALARIO) VALUES (4,'MARCOS PEREIRA
BRASIL',NULL,'M','20/02/1999',6000);

INSERT INTO FUNCIONARIO (CODIGOFUNCIONARIO, NOME, CPF, SEXO,
DTNASCIMENTO, SALARIO) VALUES (5,'HEMERSON SILVA
BRASIL',NULL,'M','20/12/1992',4000);
```

Exibir todo o conteúdo após a criação da tabela e a inserção dos registros

SELECT * FROM FUNCIONARIO;

GRUPO DE DADOS

A maior parte dessas consultas está atrelada ao uso de alguma função de resumo, por exemplo, SUM, AVG, MIN e MAX, as quais representam, respectivamente, soma, média, mínimo e máximo.

GRUPO DE DADOS COM GROUP BY

A cláusula GROUP BY serve para exibir resultados de consulta de acordo com um grupo especificado. Ela é declarada após a cláusula FROM, ou após a cláusula WHERE, caso exista na consulta.

SELECT SEXO

FROM FUNCIONARIO

GROUP BY SEXO;

No entanto, vamos perceber que o uso mais conhecido da cláusula GROUP BY ocorre quando associada a funções de agregação, tais como COUNT, MIN, MAX e AVG.

VEJAMOS ALGUNS EXEMPLOS:

Consulta 01:

Retornar o número de funcionários por sexo.

SELECT SEXO, COUNT(*) AS QUANTIDADE

FROM FUNCIONARIO

GROUP BY SEXO;

Consulta 02:

Retornar a média salarial por sexo.

SELECT SEXO,

AVG(SALARIO) AS MEDIASALARIAL

FROM FUNCIONARIO

GROUP BY SEXO;

Consulta 03:

Retornar, por mês de aniversário, a quantidade de colaboradores, o menor salário, o maior salário e o salário médio. Ordene os resultados por mês de aniversário.

```
SELECT EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,
        COUNT(*) AS QUANTIDADE,
        MIN(SALARIO) AS MENORSALARIO,
        ROUND(AVG(SALARIO)::NUMERIC,0) AS MEDIASALARIAL,
        MAX(SALARIO) AS MAIORSALARIO
FROM FUNCIONARIO
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO)
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO);
```

O SGBD realiza o agrupamento de dados de acordo com o mês de nascimento dos funcionários. Depois, para cada grupo encontrado, as funções de agregação são executadas e, em seguida, exibidos os resultados. Perceba também que, na linha 4, utilizamos a função ROUND com objetivo de mostrar ao usuário final somente a parte inteira dos valores resultantes da média salarial.

Consulta 04:

Retornar, por mês de aniversário, o mês, o sexo e a quantidade de colaboradores. Apresentar os resultados ordenados pelo mês.

```
SELECT EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,
        SEXO,
        COUNT(*) AS QUANTIDADE
FROM FUNCIONARIO
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO),SEXO
ORDER BY EXTRACT(MONTH FROM DTNASCIMENTO);
```

O SGBD realiza o agrupamento de dados de acordo com os valores do mês de aniversário. Em seguida, no contexto de cada mês encontrado, mais um grupo é construído por sexo. Finalmente, para cada ocorrência mês/sexo, o número de colaboradores é calculado.

GRUPO DE DADOS COM GROUP BY E HAVING

Você vai vivenciar situações onde será necessário estabelecer algum tipo de filtro, tendo como base um cálculo originado a partir de uma função de agregação, não sendo possível usar a cláusula WHERE. Nesses casos, utilizamos a cláusula HAVING, que serve justamente para esse propósito.

Consulta 05:

Suponha que o departamento de recursos humanos esteja estudando a viabilidade de oferecer bônus de 5% aos funcionários por mês de nascimento, mas limitado somente aos casos onde há mais de um colaborador aniversariando. Assim, para cada mês em questão, deseja-se listar o mês, o número de colaboradores e o valor do bônus.

```
SELECT EXTRACT(MONTH FROM DTNASCIMENTO) AS MES,
        COUNT(*) AS QUANTIDADE,
        SUM(SALARIO*0.5) AS TOTALBONUS
FROM FUNCIONARIO
GROUP BY EXTRACT(MONTH FROM DTNASCIMENTO)
HAVING COUNT(*) > 1
ORDER BY EXTRACT (MONTH FROM DTNASCIMENTO);
```

Note que estamos diante de uma estrutura de consulta muito similar ao código da consulta 03. Porém, estamos interessados em retornar somente o(s) registro(s) cujo valor da coluna quantidade seja maior que a unidade. Acontece que quantidade é uma coluna calculada com auxílio de uma função de agregação, não sendo possível programar um filtro na cláusula WHERE (WHERE QUANTIDADE>1). Assim, declaramos o filtro de interesse fazendo uso da cláusula HAVING, conforme linha 6 da consulta.