

Nome - João Gabriel Wolochen 1751310  
Nome - Gabriel Rodrigues Shiota 1999672  
Nome - Caio Fernando Peres 2000784

### LISTA DE EXERCÍCIOS - SQL

Dada às relações abaixo:

#### SISTEMA DE ALOCAÇÃO DE SALAS

**DEPARTAMENTO** {IDDepartamento, Nome, Sigla}

**CURSO** {IDCurso, Nome, Sigla, CargaHoraria, IDDepartamento}

**SALA** {IDSala, Sigla, TipoSala, Capacidade, IDDepartamento}

*TipoSala = "Teórica", "Laboratório", "Auditório", "Gabinete Professor"*

**FUNCIONARIO** {IDFuncionario, Nome, CPF, Email, DataAdmissao, DataDemissao, Salario, IDEndereco, IDDepartamento}

**ENDERECO** {IDEndereco, UF, Cidade, Bairro, Rua, CEP}

**EVENTO** {IDEvento, Nome, Descricao, Sigla}

**ALOCACAO** {IDAlocacao, IDEvento, IDSala, DataAlocacao, DataInicio, DataTermino, HoraInicio, HoraTermino, DiaSemana, IDFuncionario}

*DiaSemana = "Segunda", "Terça", "Quarta", "Quinta", "Sexta", "Sábado"*

**Elaborar os comandos SQL, conforme se pede:**

1. Selecione as Cidades do estado do Paraná ordenando-as por ordem crescente do Nome da cidade;

```
select endereco.UF,endereco.cidade  
from endereco  
where endereco.UF like 'PR'  
order by cidade;
```

2. Selecione o Nome do Funcionário o seu endereço. Ordene de forma decrescente de nome do Funcionário e por ordem crescente de UF;

```
select funcionario.nome,endereco.uf  
from funcionario,endereco
```

```
where funcionario.IDEndereco = endereco.IDEndereco  
order by nome desc,uf;
```

3. Selecione o Nome e CPF dos Funcionários do Departamento Acadêmico de Informática;

```
select funcionario.nome,funcionario.cpf  
from funcionario, departamento  
where Departamento.Nome like 'Cien%' and #####  
funcionario.IDDepartamento = departamento.IDDepartamento;
```

4. Selecione o nome e sigla dos cursos do Departamento Acadêmico de Informática;

```
select curso.nome,curso.sigla  
from curso,departamento  
where Departamento.Nome like 'Cien%' and #####  
curso.IDDepartamento = Departamento.IDDepartamento;
```

5. Seleciona todos os Funcionários do Departamento Acadêmico de Produção de moram em Curitiba;

```
select endereco.cidade,departamento.nome,funcionario.nome  
from endereco,departamento,funcionario  
where departamento.nome like 'Fina%' and #####  
endereco.cidade like 'Curitiba' and  
funcionario.IDDepartamento = Departamento.IDDepartamento and  
funcionario.IDEndereco = Endereco.IDEndereco;
```

6. Selecione o Nome e Email dos Funcionários que foram admitidos a partir do ano de 2015. Ordene de forma crescente da data de Admissão;

```
select funcionario.nome,funcionario.email,funcionario.dataadmissao  
from funcionario  
where funcionario.dataadmissao > '2014-12-31'  
order by dataadmissao;
```

**7.** Selecione os Funcionários que estão ativos na data de hoje;

```
select funcionario.nome  
from funcionario  
where funcionario.datademissao IS NULL;
```

**8.** Selecione a quantidade de Funcionário que residem no Paraná, Santa Catarina e no Rio Grande do Sul, produzindo informações por estado;

```
SELECT count(*) as Pessoas  
from funcionario, endereco  
where (funcionario.idendereco = endereco.IDEndereco) and ((endereco.UF = 'PR')  
or (endereco.UF = 'SC') or (endereco.UF = 'RS'));
```

**9.** Selecione os Funcionários com menos de 30 anos de serviço na data de hoje;

```
select nome,dataadmissao from funcionario  
where (CURDATE() - 00300000) < dataadmissao ;
```

**10.** Selecione a quantidade de Funcionários por Departamento. Ordene da menor para a maior quantidade;

```
select departamento.IDDepartamento, COUNT(funcionario.IDDepartamento) AS  
QuantidadeDeFuncionarios FROM departamento RIGHT JOIN funcionario  
ON funcionario.IDDepartamento=departamento.IDDepartamento GROUP BY  
departamento.IDDepartamento ORDER BY QuantidadeDeFuncionarios;
```

**11.** Selecione o valor total dos salários dos Funcionários do Departamento Acadêmico de Informática, Eletrônica e Mecânica. Apresente o total por departamento

```
select sum(funcionario.salario), departamento.nome  
from funcionario  
inner join departamento on  
funcionario.IDDepartamento=departamento.IDDepartamento
```

```
where departamento.sigla = 'DAINF' OR departamento.sigla = 'DEMEC' OR
departamento.sigla = 'DEELE'
group by departamento.sigla;
```

- 12.** Selecione o nome e sigla do Departamento que possui o maior número de cursos;

```
select departamento.Sigla
from departamento, curso
where departamento.IDDepartamento = curso.IDDepartamento
having count(*) = (select max(quant)
from (select count(*) as QUANT
from curso group by IDDepartamento) as T);
```

- 13.** Selecione o nome do Departamento que possui o número de Funcionários acima da média de todos os demais Departamentos;

```
select departamento.Nome FROM departamento, FUNCIONARIO
WHERE departamento.IDDepartamento=funcionario.IDDepartamento
GROUP BY funcionario.IDDepartamento
HAVING COUNT(*) > (
SELECT AVG(Contar) AS Media FROM (
SELECT COUNT(*) AS Contar FROM funcionario JOIN departamento ON
funcionario.IDDepartamento = departamento.IDDepartamento GROUP BY
departamento.IDDepartamento)AS T );
```

- 14.** Altere a data de admissão para o dia de hoje do Funcionário Lucas Alves Pedroso;

```
update funcionario
set DataAdmissao = now()
where nome like 'Lucas Alves Pedroso';
```

- 15.** Apresente o comando DDL para criação da tabela ALOCAÇÃO, com sua PK e FK's;

```
CREATE TABLE alocao (IDAlocacao INT NOT NULL, IDEvento INT NOT NULL,
IDSala INT NOT NULL, DataAlocacao DATE NOT NULL, DataInicio DATE NOT NULL,
```

```

DataTermino DATE NOT NULL, HoralInicio TIME NOT NULL, HoraTermino TIME NOT
NULL, DiaSemana VARCHAR(20) NOT NULL, IDFuncionario INT NOT NULL, PRIMARY
KEY (IDAlocacao ), FOREIGN KEY (IDEvento) REFERENCES EVENTO(IDEvento)
ON DELETE RESTRICT, FOREIGN KEY (IDSala) REFERENCES SALA(IDSala)
ON DELETE RESTRICT, FOREIGN KEY (IDFuncionario) REFERENCES
FUNCIONARIO(IDFuncionario) ON DELETE RESTRICT);
ALTER TABLE alocacao
DROP COLUMN DataAlocacao;
ALTER TABLE alocacao
ADD DataAlocacao DATE;
DROP TABLE alocacao;

```

**16.** Excluir a Sala que nunca foi alocada;

```

DELETE FROM sala WHERE NOT EXISTS(SELECT * FROM alocacao WHERE
alocacao.IDSala=SALA.IDSala);

```

**17.** Incluir a coluna Data de Nascimento na tabela Funcionário;

```

alter table funcionario add column Datadenascimento;

```

**18.** Excluir a coluna Email da tabela Funcionário;

```

alter table funcionario drop column email;

```

**19.** Apresente o comando DDL que permite criar um índice na tabela Departamento na coluna Sigla;

```

CREATE TABLE departamento (IDDepartamento INT NOT NULL, Nome VARCHAR(200)
NOT NULL, Sigla VARCHAR(10) NOT NULL, PRIMARY KEY (IDDepartamento));
ALTER TABLE departamento
ADD Indice INT;
ALTER TABLE departamento
DROP COLUMN Indice;

```

**20.** Crie uma store procedure para atualizar a tabela alocação.

```
CREATE PROCEDURE AtualizarAlocacao(IDAloc INT, IDEvento INT, IDSala  
INT, DataAlocacao DATE, DataInicio DATE, DataTermino DATE, HoraInicio TIME,  
HoraTermino TIME, DiaSemana VARCHAR(20), IDFuncionario INT)  
BEGIN  
UPDATE alocacao WHERE IDAlocacao = IDAloc;  
END//
```

```
CALL  
AtualizarAlocacao(2,3,2,'2018-10-03','2018-11-04','2018-11-05','22:59:59','23:30:00','Sábado',1);
```