



UNIVERSIDADE FEDERAL DE PELOTAS  
CENTRO DE DESENVOLVIMENTO TECNOLÓGICO  
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

Disciplina de Projeto de Banco de Dados

Guia das Aulas Práticas – (comandos para o SGBD mySQL)

Passo inicial – conexão com servidor

```
# mysql -h host -u user -p senha
```

1. Criação do esquema para definição do banco de dados “Companhia”.

```
> CREATE DATABASE Companhia;
```

No caso do mySQL, para escolher o banco a ser usado:

```
> USE Companhia;
```

2. Esquema conceitual completo (com domínios e restrições de integridade) para criação do banco. Este esquema deve ser usado em apoio à criação do banco de dados, a partir das operações em 3 e 4 - **NÃO COPIE E COLE OS COMANDOS DESTES ESQUEMA, POIS AS RESTRIÇÕES DEVEM SER ESTABELECIDAS EM ORDEM:**

```
CREATE TABLE EMPREGADO (  
    PNome      VARCHAR(15)          NOT NULL,  
    MNome      CHAR,                NOT NULL,  
    SNome      VARCHAR(15)          NOT NULL,  
    NSS        CHAR(9)              NOT NULL,  
    DataNasc   DATE,                NOT NULL,  
    Endereco   VARCHAR(30),         NOT NULL,  
    Sexo       CHAR,                NOT NULL,  
    Salario    DECIMAL(10, 2),      NOT NULL,  
    NssSuper   CHAR(9),             NOT NULL,  
    NDep       INT                  NOT NULL,  
    PRIMARY KEY (NSS),  
    FOREIGN KEY (NssSuper) REFERENCES EMPREGADO (NSS)  
        ON DELETE SET NULL  
        ON UPDATE CASCADE,  
    FOREIGN KEY (NDep) REFERENCES DEPARTAMENTO (DNumero) );  
  
CREATE TABLE DEPARTAMENTO (  
    DNome      VARCHAR(15)          NOT NULL,  
    DNumero    INT                  NOT NULL,  
    NssGer     CHAR(9)              NOT NULL,  
    DataInicGer DATE,                NOT NULL,  
    PRIMARY KEY (DNumero),  
    FOREIGN KEY (NssGer) REFERENCES EMPREGADO (NSS)  
        ON DELETE CASCADE );
```

```

CREATE TABLE LOCAIS_DEPTO (
    DNUMERO          INT          NOT NULL,
    DLOCALIZACAO     VARCHAR(15)  NOT NULL,
    PRIMARY KEY (DNUMERO, DLOCALIZACAO),
    FOREIGN KEY (DNUMERO) REFERENCES DEPARTAMENTO (DNUMERO)
    ON DELETE CASCADE);

```

```

CREATE TABLE PROJETO (
    PNOOME            VARCHAR(15)   NOT NULL,
    PNUMERO           INT           NOT NULL,
    PLOCALIZACAO      VARCHAR(15),
    DNUM              INT           NOT NULL,
    PRIMARY KEY (PNUMERO),
    FOREIGN KEY (DNUM) REFERENCES DEPARTAMENTO (DNUMERO));

```

```

CREATE TABLE TRABALHA_EM (
    NSSEMP            CHAR(9)       NOT NULL,
    PNRO              INT           NOT NULL,
    HORAS             DECIMAL(3,1)  NOT NULL,
    PRIMARY KEY (NSSEMP, PNRO),
    FOREIGN KEY (NSSEMP) REFERENCES EMPREGADO (NSS)
    ON DELETE CASCADE,
    FOREIGN KEY (PNRO) REFERENCES PROJETO (PNUMERO)
    ON DELETE CASCADE);

```

```

CREATE TABLE DEPENDENTE (
    NSSEMP            CHAR(9)       NOT NULL,
    NOMEDEPENDENTE    VARCHAR(15)   NOT NULL,
    SEXO              CHAR,
    DATANIV           DATE,
    RELACAO           VARCHAR(8),
    PRIMARY KEY (NSSEMP, NOMEDEPENDENTE),
    FOREIGN KEY (NSSEMP) REFERENCES EMPREGADO (NSS));

```

### 3. Comandos para criação do Banco de Dados Companhia

3.a) Criação da tabela EMPREGADO (conforme esquema em 2, sem restrições de integridade):

```

CREATE TABLE EMPREGADO (
    PNOOME            VARCHAR(15)   NOT NULL,
    MNOOME            CHAR,
    SNOOME            VARCHAR(15)   NOT NULL,
    NSS               CHAR(9)       NOT NULL,
    DATANASC          DATE,
    ENDERECO          VARCHAR(30),
    SEXO              CHAR,
    SALARIO           DECIMAL(10, 2),
    NSSSUPER          CHAR(9),
    NDEP              INT           NOT NULL);

```

**3.b)** Adicione a chave primária da tabela EMPREGADO separadamente:

```
> ALTER TABLE EMPREGADO ADD PRIMARY KEY (NSS);
```

**3.c)** Antes de adicionar as chaves estrangeiras em EMPREGADO, insira um primeiro registro:

```
> INSERT INTO EMPREGADO (PNAME, MNAME, SNAME, NSS, DATANASC, ENDereco, SEXO, SALARIO, NDEP) VALUES ('James', 'E', 'Borg', '888665555', '1927-11-10', 'Av. H. 8', 'M', 5500, 1);
```

**3.d)** Adicione a chave estrangeira (NSSUPER, que referencia EMPREGADO) na tabela EMPREGADO:

```
> ALTER TABLE EMPREGADO ADD FOREIGN KEY (NSSUPER) REFERENCES EMPREGADO (NSS)
    ON DELETE SET NULL
    ON UPDATE CASCADE;
```

**3.e)** Crie a tabela DEPARTAMENTO:

```
CREATE TABLE DEPARTAMENTO (
    DNAME          VARCHAR(15)      NOT NULL,
    DNUMERO        INT              NOT NULL,
    NSSGER         CHAR(9)          NOT NULL,
    DATINICGER     DATE,
    PRIMARY KEY (DNUMERO));
```

**3.f)** Insira um primeiro registro na tabela DEPARTAMENTO:

```
> INSERT INTO DEPARTAMENTO VALUES ('Gerencial', 1, '888665555', '1971-06-19');
```

**3.g)** Adicione a chave estrangeira NDEP em EMPREGADO que referencia a tabela DEPARTAMENTO (DNUMERO):

```
> ALTER TABLE EMPREGADO ADD FOREIGN KEY (NDEP) REFERENCES DEPARTAMENTO (DNUMERO);
```

**3.h)** Adicione a chave estrangeira NSSGER em DEPARTAMENTO que referencia a tabela EMPREGADO (NSS):

→ para fazerem

**3.i)** Altere a tabela DEPARTAMENTO para aceitar o valor DEFAULT '888665555' na coluna NSSGER:

→ para fazerem

**3.j)** Crie as demais tabelas – NÃO ESQUEÇA DAS RESTRIÇÕES CONFORME ESQUEMA:

→ (criar na ordem definida em 2)

#### 4. Faça algumas alterações nas tabelas

4.a) Altere a tabela DEPARTAMENTO para receber valores únicos na coluna DNOME:

```
> ALTER TABLE DEPARTAMENTO ADD CONSTRAINT DEPARTAMENTO_NOME UNIQUE (DNOME);
```

4.b) Altere a coluna HORAS da tabela TRABALHA\_EM para aceitar valores nulos:

```
> ALTER TABLE TRABALHA_EM MODIFY HORAS DECIMAL(3,1) NULL;
```

4.c) Defina atributos únicos:

→ para fazerem

4.d) Crie o atributo FUNCAO em EMPREGADO:

→ para fazerem

#### 5. Alguns comandos extras, para visualização das tabelas criadas:

5.a) Mostrar as tabelas do schema criado:

```
> SHOW TABLES;
```

```
> SHOW TABLES FROM COMPANHIA;
```

5.b) Mostrar a estrutura de determinada tabela:

```
> DESCRIBE EMPREGADO;
```

```
> SHOW COLUMNS FROM EMPREGADO;
```

5.c) Mostra toda a estrutura de criação da tabela:

```
> SHOW CREATE TABLE EMPREGADO;
```

#### 6. Com as tabelas criadas, insira dados nas tabelas:

(Sendo que 1 campo da tabela EMPREGADO e 1 campo da tabela DEPARTAMENTO já estão criados). Cuidado com a ordem de inserção nas tabelas!

Na próxima página estão as tabelas completas, com instâncias

6.a) Inserção de valores em DEPARTAMENTO:

```
> INSERT INTO DEPARTAMENTO (DNOME, DNUMERO, DATINICGER) VALUES  
    ('Pesquisa', 5, '1978-05-22'),  
    ('Administrativo', 4, '1985-01-19');
```

## Relações do BD Companhia:

### EMPREGADO

PNOME	MNOME	SNOME	NSS	DATANASC	ENDEREÇO	SEXO	SALARIO	NSSSUPER	NDEP
John	B	Smith	123456789	09-JAN-55	R. A, 1	M	3000	333445555	5
Franklin	T	Wong	333445555	08-DEZ-45	R. B, 2	M	4000	888665555	5
Alicia	J	Zelaya	999887777	19-JUL-58	Av. C, 3	F	2500	987654321	4
Jennifer	S	Wallace	987654321	20-JUN-31	Trav. D, 4	F	4300	888665555	4
Ramesh	K	Narayan	666884444	15-SET-52	R. E, 5	M	3800	333445555	5
Joyce	A	English	453453453	31-JUL-62	R. F, 6	F	2500	333445555	5
Ahmad	V	Jabbar	987987987	29-MAR-59	Av G, 7	M	2500	987654321	4
James	E	Borg	888665555	10-NOV-27	Av H, 8	M	5500	null	1

### DEPARTAMENTO

DNOME	DNÚMERO	NSSGER	DATINICGER
Pesquisa	5	333445555	22-MAI-78
Administrativo	4	987654321	01-JAN-85
Gerencial	1	888665555	19-JUN-71

### LOCAIS\_DEPTO

DNÚMERO	DLOCALIZAÇÃO
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

### PROJETO

PNOME	PNÚMERO	PLOCALIZAÇÃO	DNUM
ProdutoX	1	Bellaire	5
ProdutoY	2	Sugarland	5
ProdutoZ	3	Houston	5
Automação	10	Stafford	4
Reorganização	20	Houston	1
Beneficiamento	30	Stafford	4

### TRABALHA EM

NSSEMP	PNRO	HORAS
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	Null

### DEPENDENTE

NSSEMP	NOMEDEPENDENTE	SEXO	DATANIV	RELAÇÃO
333445555	Alice	F	05-ABR-76	FILHA
333445555	Theodore	M	25-OUT-73	FILHO
333445555	Joy	F	03-MAI-48	ESPOSA
987654321	Abner	M	29-FEV-78	MARIDO
123456789	Michael	M	01-JAN-78	FILHO
123456789	Alice	F	31-DEZ-78	FILHA
123456789	Elizabeth	F	05-MAI-57	ESPOSA

### 6.b) Inserção de valores em EMPREGADO:

```
> INSERT INTO EMPREGADO VALUES ('Franklin', 'T', 'Wong', '333445555',  
'1945-12-08', 'R. B. 2', 'M', 4000, '888665555', 5, NULL);  
  
> INSERT INTO EMPREGADO VALUES ('john', 'B', 'Smith', '123456789', '1955-  
01-09', 'R. A. 1', 'M', 3000, '333445555', 5, NULL);
```

### 6.c) Atualização dos valores previamente inseridos em DEPARTAMENTO:

```
> UPDATE DEPARTAMENTO SET NSSGER='987654321' WHERE DNUMERO=4;
```

→ para fazerem - atualizar o Departamento de número 5, que passa a ter como gerente o funcionário com NSS '333445555'

### 6.d) Inserção de valores nas demais tabelas:

```
> INSERT INTO LOCAIS_DEPTO VALUES  
    (1, 'Houston'),  
    (4, 'Stafford'),  
    (5, 'Bellaire');  
  
> INSERT INTO PROJETO VALUES  
    ('ProdutoX', 1, 'Bellaire', 5),  
    ('ProdutoY', 2, 'Sugarland', 5),  
    ('ProdutoZ', 3, 'Houston', 5);  
  
> INSERT INTO TRABALHA_EM VALUES  
    ('123456789', 1, 32.5),  
    ('123456789', 2, 7.5),  
    ('666884444', 3, 40.0),  
    ('453453453', 1, 20.0);  
  
> INSERT INTO TRABALHA_EM (NSSEMP, PNRO) VALUES (987654321, 20);  
  
> INSERT INTO DEPENDENTE VALUES  
    ('333445555', 'Alice', 'F', '1976-04-05', 'FILHA'),  
    ('333445555', 'Theodore', 'M', '1973-10-25', 'FILHO'),  
    ('333445555', 'Joy', 'F', '1948-05-03', 'ESPOSA');
```

### 6.e) Mostrar os dados inseridos em cada uma das tabelas:

```
> SELECT * FROM EMPREGADO;  
  
> SELECT * FROM DEPARTAMENTO;  
  
> SELECT * FROM LOCAIS_DEPTO;  
  
> SELECT * FROM PROJETO;  
  
> SELECT * FROM TRABALHA_EM;  
  
> SELECT * FROM DEPENDENTE;
```

**6.f) Remover algum valor inserido na tabela:**

**> DELETE FROM EMPREGADO WHERE SNOME='Wong';**

**7. Com o banco de dados criado e instanciado, iniciar as consultas:**

a) **Consulta 1** – Recuperar a data de aniversário e o endereço do empregado chamado 'John B. Smith'.

**SELECT DATANASC, ENDERECO  
FROM EMPREGADO  
WHERE PNAME='John' AND MNAME='B' AND SNOME='Smith';**

b) **Consulta 2** – Obter o nome e o endereço dos empregados que trabalham para o departamento de 'Pesquisa'.

**→ para fazerem**

c) **Consulta 3** - Para todo projeto localizado em 'Stafford', listar o número do projeto, o número do departamento responsável, o sobrenome, endereço e data de nascimento do gerente responsável pelo departamento.

**SELECT PNUMERO, DNUM, SNOME, DATANASC, ENDERECO  
FROM PROJETO, DEPARTAMENTO, EMPREGADO  
WHERE DNUM=DNUMERO AND NSSGER=NSS AND PLOCALIZACAO='Stafford';**

d) **Consulta 4** – a mesma Consulta 3, mas usando qualificadores:

**→ para fazerem**

e) **Consulta 5** – Uso de Aliases – Para cada empregado, recupere o nome do empregado e o nome de seu supervisor imediato.

**→ para fazerem**

f) **Consulta 6** – sem a cláusula WHERE especificada.

d.1) Recupere o NSS de todos os empregados.

**SELECT NSS FROM EMPREGADO;**

d.2) Recupere o NSS e o nome do departamento de todos os empregados.

**→ para fazerem**

g) **Consulta 7** – Recuperar todos os atributos dos empregados que trabalham no departamento 'Pesquisa'.

**→ para fazerem**

h) **Consulta 8** – uso do qualificador DISTINCT.

d.1) Recupere o salário de todos os empregados.

→ para fazerem

d.2) Recupere salários não duplicados dos empregados.

→ para fazerem

i) **Consulta 9** - Listar os números de projetos em que o empregado de sobrenome 'Wong' trabalhe ou que sejam controlados por algum departamento gerenciado pelo empregado de sobrenome 'Wong':

```
SELECT PNUMERO
FROM PROJETO, DEPARTAMENTO, EMPREGADO
WHERE DNUM=DNUMERO AND NSSGER=NSS AND SNOME='Wong'
UNION
(SELECT PNUMERO
FROM PROJETO, TRABALHA_EM, EMPREGADO
WHERE PNUMERO=PNRO AND NSSEMP=NSS AND SNOME='Wong');
```

i.1) Esta mesma consulta pode ser feita de forma aninhada:

→ para fazerem

j) **Consulta 10** – Consultas Aninhadas Correlacionadas – Recupere o nome de cada empregado que tenha um dependente com o mesmo nome do empregado.

→ para fazerem

k) **Consulta 11** – Alteração da Consulta 2 usando operador 'JOIN' – Obter o nome e o endereço dos empregados que trabalham para o departamento de 'Pesquisa'.

```
SELECT PNAME, SNOME, ENDERECO
FROM EMPREGADO JOIN DEPARTAMENTO ON DNUMERO=NDEP
WHERE DNAME='Pesquisa';
```

l) **Consulta 12** – Buscar o nome e as localizações do departamento 'Pesquisa' – com e sem cláusula JOIN.

→ para fazerem

m) **Consulta 13** – Buscar os nomes dos empregados e seus supervisores, mantendo os empregados que não possuem supervisores.

```
SELECT E.PNAME, E.SNOME, S.PNAME, S.SNOME
FROM (EMPREGADO AS E LEFT OUTER JOIN EMPREGADO AS S ON
E.NSSUPER=S.NSS);
```



n) **Consulta 14** - Encontrar o maior salário, o menor salário, e a média salarial de todos os empregados.

```
SELECT MAX(SALARIO), MIN(SALARIO), AVG(SALARIO) FROM EMPREGADO;
```

o) **Consulta 15** - Recuperar o total de empregados da companhia (Consulta A) e o número de empregados do departamento Pesquisa (Consulta B).

```
SELECT COUNT (*) FROM EMPREGADO;
```

p) **Consulta 16** – Clausula ‘GROUP BY’ - Para cada departamento, recuperar o seu número, a quantidade de empregados que possui e a sua média salarial.

```
SELECT NDEP, COUNT (*), AVG (SALARIO)  
FROM EMPREGADO  
GROUP BY NDEP;
```

q) **Consulta 17** - Para cada projeto, recuperar o número do projeto, seu nome e o número de empregados que trabalham no projeto.

→ para fazerem

r) **Consulta 18** – uso clausula ‘HAVING’ - Para cada projeto em que trabalhem mais de dois empregados, recupere o número do projeto e o nome do projeto e o número de empregados que trabalham no projeto.

```
SELECT PNUMERO, PNOME, COUNT (*)  
FROM PROJETO, TRABALHA_EM  
WHERE PNUMERO=PNRO  
GROUP BY PNUMERO, PNOME  
HAVING COUNT(*)>2;
```

s) **Consulta 19** – uso de operadores aritméticos - Recupere todos os empregados (nome e sobrenome) e seus respectivos salários que trabalham no projeto ‘ProdutoX’ com um aumento de 10%.

→ para fazerem

t) **Consulta 20** – cláusula ORDER BY - Recuperar a lista de empregados e dos projetos em que eles trabalhem, ordenados pelo departamento do empregado e cada departamento ordenado alfabeticamente pelo sobrenome do empregado.

→ para fazerem