Banco de Dados ARA0040

Turma 3002

Tema 04

SQL básico

Leitura específica:

ELMASRI, R.; NAVATHE, S. Sistemas de banco de dados. 7. São Paulo: Pearson, 2018.
Capítulo 06 – SQL básica

Introdução

- Linguagem SQL

- SQL

SQL é uma linguagem de banco de dados abrangente: tem instruções para definição de dados, consultas e atualizações. Logo, ela é uma DDL *e* uma DML. Além disso, ela tem habilidades para definir visões sobre o banco de dados, para especificar segurança e autorização, para definir restrições de integridade e para especificar controles de transação. Ela também possui regras para embutir instruções SQL em uma linguagem de programação de uso geral, como Java ou C/C++.[1]

Definição de tipos de Dados

- Tabela, linha e coluna

- Instrução ‘create ‘

Definição de tipos de Dados

- ESQUEMA –

- instrução CREATE SCHEMA

```
CREATE SCHEMA EMPRESA AUTHORIZATION 'Jsilva';
```

Definição de tipos de Dados

- Catálogo

Definição de tipos de Dados
Comando CREATE TABLE

- Especificar uma nova relação dando NOME e especificando atributos e restrições

https://www.programiz.com/sql/online-compiler/

Definição de tipos de Dados
Comando CREATE TABLE

Exemplo....

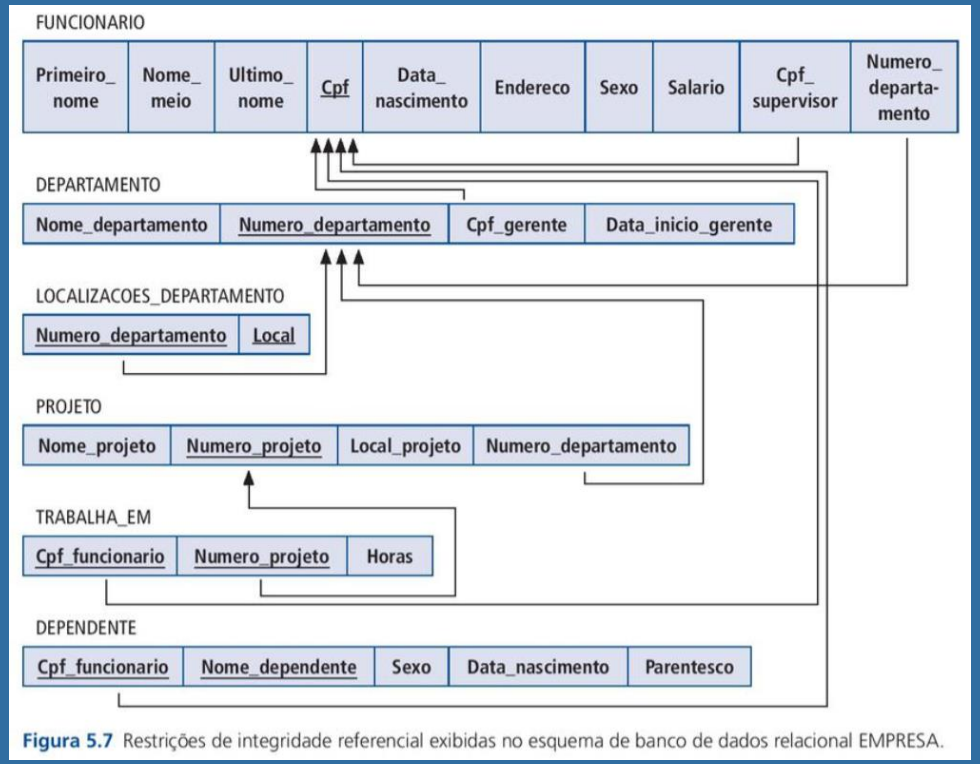

Figura 5.7 Restrições de integridade referencial exibidas no esquema de banco de dados relacional EMPRESA.

Definição de tipos de Dados
Comando CREATE TABLE

```
CREATE TABLE FUNCIONARIO
    (Primeiro_nome          VARCHAR(15)         NOT NULL,
    Nome_meio               CHAR,
    Ultimo_nome             VARCHAR(15)         NOT NULL,
    Cpf                     CHAR(11),           NOT NULL,
    Data_nascimento         DATE,
    Endereco                VARCHAR(30),
    Sexo                    CHAR,
    Salario                 DECIMAL(10,2),
    Cpf_supervisor          CHAR(11),
    Numero_departamento     INT                 NOT NULL,
    PRIMARY KEY (Cpf) );
CREATE TABLE DEPARTAMENTO
    (Nome_departamento      VARCHAR(15)         NOT NULL,
    Numero_departamento     INT                 NOT NULL,
    Cpf_gerente             CHAR(11),           NOT NULL,
    Data_inicio_gerente     DATE,
    PRIMARY KEY (Numero_departamento),
    UNIQUE (Nome_departamento),
```

Figura 6.1 Instruções de definição de dados CREATE TABLE da SQL para definição do esquema EMPRESA da Figura 5.7. (*continua*)

```
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf) );
CREATE TABLE LOCALIZACOES_DEPARTAMENTO
    (Numero_departamento    INT                 NOT NULL,
    Local                   VARCHAR(15)         NOT NULL,
    PRIMARY KEY (Numero_departamento, Local),
    FOREIGN KEY (Numero_departamento) REFERENCES
    DEPARTAMENTO(Numero_departamento) );
CREATE TABLE PROJETO
    (Nome_projeto           VARCHAR(15)         NOT NULL,
    Numero_projeto          INT                 NOT NULL,
    Local_projeto           VARCHAR(15),
    Numero_departamento     INT                 NOT NULL,
    PRIMARY KEY (Numero_projeto),
    UNIQUE (Nome_projeto),
    FOREIGN KEY (Numero_departamento) REFERENCES DEPARTAMENTO(Numero_
    departamento) );
```

Definição de tipos de Dados
Comando CREATE TABLE

```
CREATE TABLE TRABALHA_EM
    (Cpf_funcionario        CHAR(11)        NOT NULL,
    Numero_projeto          INT             NOT NULL,
    Horas                   DECIMAL(3,1)    NOT NULL,
    PRIMARY KEY (Cpf_funcionario, Numero_projeto),
    FOREIGN KEY (Cpf_funcionario) REFERENCES FUNCIONARIO(Cpf),
    FOREIGN KEY (Numero_projeto) REFERENCES PROJETO(Numero_projeto) );
CREATE TABLE DEPENDENTE
    (Cpf_funcionario        CHAR(11),       NOT NULL,
    Nome_dependente         VARCHAR(15)     NOT NULL,
    Sexo                    CHAR,
    Data_nascimento         DATE,
    Parentesco                              VARCHAR(8),
    PRIMARY KEY (Cpf_funcionario, Nome_dependente),
    FOREIGN KEY (Cpf_funcionario) REFERENCES FUNCIONARIO(Cpf) );
```

Figura 6.1 Instruções de definição de dados CREATE TABLE da SQL para definição do esquema EMPRESA da Figura 5.7. (*continuação*)

Definição de tipos de Dados
Comando CREATE TABLE

- Explicitando o esquema

CREATE TABLE EMPRESA.FUNCIONARIO

em vez de

CREATE TABLE FUNCIONARIO

https://www.programiz.com/sql/online-compiler/

Definição de tipos de Dados

Tipo de dados

- Os tipos de dados **numéricos** incluem números inteiros de vários tamanhos (INTEGER ou INT e SMALLINT) e números de ponto flutuante (reais) de várias precisões (FLOAT ou REAL e DOUBLE PRECISION). Os números formatados podem ser declarados usando DECIMAL(i, j) — ou DEC(i, j) ou NUMERIC(i, j) — em que i, a *precisão*, é o número total de dígitos decimais e j, a *escala*, é o número de dígitos após o ponto decimal. O valor padrão para a escala é zero e, para a precisão, é definido pela implementação.

Definição de tipos de Dados
Tipo de dados

- Tipos de dados de **cadeia de caracteres** (também chamados de *string* de caracteres) são de tamanho fixo — CHAR(*n*) ou CHARACTER(*n*), em que *n* é o número de caracteres — ou de tamanho variável — VARCHAR(*n*) ou CHAR VARYING(*n*) ou CHARACTER VARYING(*n*), em que *n* é o número máximo de caracteres. Ao especificar um valor literal de cadeia de caracteres, ele é colocado entre aspas simples (apóstrofos), e é *case sensitive* (diferencia maiúsculas de minúsculas).[3] Para cadeias de caracteres de tamanho fixo, uma cadeia mais curta é preenchida com caracteres em branco à direita. Por exemplo, se o valor 'Silva' for para um atributo do tipo CHAR(10), ele é preenchido com cinco caracteres em branco para se tornar 'Silva ', se necessário. Os espaços preenchidos geralmente são ignorados quando as cadeias são comparadas. Para fins de comparação, as cadeias de caracteres são consideradas ordenadas em ordem alfabética (ou lexicográfica); se uma cadeia *str1* aparecer antes de outra cadeia *str2* em ordem alfabética, então *str1* é considerada menor que *str2*.[4] Também há um operador de concatenação indicado por || (barra vertical dupla), que pode concatenar duas cadeias de caracteres em SQL. Por exemplo, 'abc' || 'XYZ' resulta em uma única cadeia 'abcXYZ'. Outro tipo de dado de cadeia de caracteres de tamanho variável, chamado CHARACTER LARGE OBJECT ou CLOB, também está disponível para especificar colunas que possuem grandes valores de texto, como documentos. O tamanho máximo de CLOB pode ser especificado em kilobytes (K), megabytes (M) ou gigabytes (G). Por exemplo, CLOB(20M) especifica um tamanho máximo de 20 megabytes.

Definição de tipos de Dados

Tipo de dados

Tipos de dados de **sequência de bits** podem ser de tamanho fixo n — BIT(n) — ou de tamanho variável — BIT VARYING(n), em que n é o número máximo de bits. O valor padrão para n, o tamanho de uma cadeia de caracteres ou sequência de bits, é 1. Os literais de sequência de bits são colocados entre apóstrofos, mas precedidos por um B para distingui-los das cadeias de caracteres; por exemplo, B'10101'.[5] Outro tipo de dados de sequência de bits de tamanho variável, chamado BINARY LARGE OBJECT ou BLOB, também está disponível para especificar colunas que possuem grandes valores binários, como imagens. Assim como para CLOB, o tamanho máximo de um BLOB pode ser especificado em kilobits (K), megabits (M) ou gigabits (G). Por exemplo, BLOB(30G) especifica um tamanho máximo de 30 gigabits.

Um tipo de dado **booleano** tem os valores tradicionais TRUE (verdadeiro) ou FALSE (falso). Em SQL, em razão da presença de valores NULL (nulos), uma lógica de três valores é utilizada, de modo que um terceiro valor possível para um tipo de dado booleano é UNKNOWN (indefinido). Discutiremos a necessidade de UNKNOWN e a lógica de três valores no Capítulo 7.

Definição de tipos de Dados

Tipo de dados

O tipo de dados **DATE** possui dez posições, e seus componentes são DAY (dia), MONTH (mês) e YEAR (ano), na forma DD-MM-YYYY. O tipo de dado TIME (tempo) tem pelo menos oito posições, com os componentes HOUR (hora), MINUTE (minuto) e SECOND (segundo) na forma HH:MM:SS. Somente datas e horas válidas devem ser permitidas pela implementação SQL. Isso implica que os meses devem estar entre 1 e 12 e os dias devem estar entre 1 e 31; além disso, um dia deve ser uma data válida para o mês correspondente. A comparação < (menor que) pode ser usada com datas ou horas — uma data *anterior* é considerada menor que uma data posterior, e da mesma forma com a hora. Os valores literais são representados por cadeias com apóstrofos precedidos pela palavra-chave DATE ou TIME; por exemplo, DATE '27-09-2018' ou TIME '09:12:47'. Além disso, um tipo de dado TIME(*i*), em que *i* é chamado de *precisão em segundos fracionários de tempo*, especifica *i* + 1 posições adicionais para TIME — uma posição para um caractere separador de período adicional (.), e *i* posições para especificar as frações de um segundo. Um tipo de dados TIME WITH TIME ZONE inclui seis posições adicionais para especificar o *deslocamento* com base no fuso horário universal padrão, que está na faixa de +13:00 a –12:59 em unidades de HOURS:MINUTES. Se WITH TIME ZONE não for incluído, o valor padrão é o fuso horário local para a sessão SQL.

ESPECIFICANDO RESTRIÇÕES

Restrições de atributos e defaults de atributo

- NULL como valor de atributo

- Valor DEFAULT

ESPECIFICANDO RESTRIÇÕES

Restrições de atributos e defaults de atributo

```
CREATE TABLE FUNCIONARIO
 ( . . . ,
    Numero_departamento INT        NOT NULL        DEFAULT 1,
 CONSTRAINT CHAVEPRIMFUNCIONARIO
     PRIMARY KEY (Cpf),
 CONSTRAINT CHAVEESTRFUNC_SUPERVISOR
     FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)
            ON DELETE SET NULL                ON UPDATE CASCADE,
 CONSTRAINT CHAVEESTRFUNC_DEPARTAMENTO
     FOREIGN KEY(Numero_departamento) REFERENCES DEPARTAMENTO(Numero_departamento)
            ON DELETE SET DEFAULT             ON UPDATE CASCADE);
CREATE TABLE DEPARTAMENTO
 ( . . . ,
    Cpf_gerente CHAR(11)        NOT NULL        DEFAULT '88866555576',
    . . . ,
 CONSTRAINT CHAVEPRIMDEPARTAMENTO
     PRIMARY KEY(Numero_departamento),
 CONSTRAINT CHAVEUNICADEPARTAMENTO
     UNIQUE (Nome_departamento),
 CONSTRAINT CHAVEESTRDEPARTAMENTO_FUNC
     FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)
            ON DELETE SET DEFAULT             ON UPDATE CASCADE);
CREATE TABLE LOCALIZACOES_DEPARTAMENTO
 ( . . . ,
 PRIMARY KEY (Numero_departamento, Local),
 FOREIGN KEY (Numero_departamento) REFERENCES
                                   DEPARTAMENTO(Numero_departamento)
            ON DELETE CASCADE                 ON UPDATE CASCADE);
```

ESPECIFICANDO RESTRIÇÕES

Restrições de atributos e defaults de atributo

- Usando uma cláusula CHECK após uma definição

```
Numero_departamento INT NOT NULL CHECK (Numero_departamento > 0 AND
Numero_departamento < 21);
```

- Também utilizada com CREATE

```
CREATE DOMAIN D_NUM AS INTEGER
CHECK (D_NUM > 0 AND D_NUM < 21);
```

https://www.programiz.com/sql/online-compiler/

ESPECIFICANDO RESTRIÇÕES

Restrições de chave e integridade referencial

- PRIMARY KEY

```
CREATE TABLE FUNCIONARIO
    (Primeiro_nome      VARCHAR(15)      NOT NULL,
    Nome_meio           CHAR,
    Ultimo_nome         VARCHAR(15)      NOT NULL,
    Cpf                 CHAR(11),        NOT NULL,
    Data_nascimento     DATE,
    Endereco            VARCHAR(30),
    Sexo                CHAR,
    Salario             DECIMAL(10,2),
    Cpf_supervisor      CHAR(11),
    Numero_departamento INT              NOT NULL,
    PRIMARY KEY (Cpf) );
```

https://www.programiz.com/sql/online-compiler/

ESPECIFICANDO RESTRIÇÕES

Restrições de chave e integridade referencial

- UNIQUE

```
CREATE TABLE DEPARTAMENTO
    (Nome_departamento      VARCHAR(15)     NOT NULL,
    Numero_departamento     INT             NOT NULL,
    Cpf_gerente             CHAR(11),       NOT NULL,
    Data_inicio_gerente     DATE,

    PRIMARY KEY (Numero_departamento),
    UNIQUE (Nome_departamento),
```

https://www.programiz.com/sql/online-compiler/

ESPECIFICANDO RESTRIÇÕES

Restrições de chave e integridade referencial

- INTEGRIDADE REFERENCIAL

```
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf) );
CREATE TABLE LOCALIZACOES_DEPARTAMENTO
    (Numero_departamento    INT             NOT NULL,
    Local                   VARCHAR(15)     NOT NULL,
    PRIMARY KEY (Numero_departamento, Local),
    FOREIGN KEY (Numero_departamento) REFERENCES
    DEPARTAMENTO(Numero_departamento) );
CREATE TABLE PROJETO
    (Nome_projeto           VARCHAR(15)     NOT NULL,
    Numero_projeto          INT             NOT NULL,
    Local_projeto           VARCHAR(15),
    Numero_departamento     INT             NOT NULL,
    PRIMARY KEY (Numero_projeto),
    UNIQUE (Nome_projeto),
    FOREIGN KEY (Numero_departamento) REFERENCES DEPARTAMENTO(Numero_
    departamento) );
```

https://www.programiz.com/sql/online-compiler/

ESPECIFICANDO RESTRIÇÕES

Restrições de chave e integridade referencial

- INTEGRIDADE REFERENCIAL
  - ON DELETE
  - ON UPDATE

```
CREATE TABLE DEPARTAMENTO
 ( . . . ,
    Cpf_gerente CHAR(11)          NOT NULL          DEFAULT '88866555576',
    . . . ,
 CONSTRAINT CHAVEPRIMDEPARTAMENTO
    PRIMARY KEY(Numero_departamento),
 CONSTRAINT CHAVEUNICADEPARTAMENTO
    UNIQUE (Nome_departamento),
 CONSTRAINT CHAVEESTRDEPARTAMENTO_FUNC
    FOREIGN KEY (Cpf_gerente) REFERENCES FUNCIONARIO(Cpf)
          ON DELETE SET DEFAULT                    ON UPDATE CASCADE);
CREATE TABLE LOCALIZACOES_DEPARTAMENTO
 ( . . . ,
 PRIMARY KEY (Numero_departamento, Local),
 FOREIGN KEY (Numero_departamento) REFERENCES
                                                  DEPARTAMENTO(Numero_departamento)
          ON DELETE CASCADE                       ON UPDATE CASCADE);
```

https://www.programiz.com/sql/online-compiler/

DANDO NOMES AS RESTRIÇÕES

- CONSTRAIN

```
CREATE TABLE FUNCIONARIO
 ( . . . ,
    Numero_departamento INT        NOT NULL          DEFAULT 1,
 CONSTRAINT CHAVEPRIMFUNCIONARIO
    PRIMARY KEY (Cpf),
 CONSTRAINT CHAVEESTRFUNC_SUPERVISOR
    FOREIGN KEY (Cpf_supervisor) REFERENCES FUNCIONARIO(Cpf)
          ON DELETE SET NULL                      ON UPDATE CASCADE,
 CONSTRAINT CHAVEESTRFUNC_DEPARTAMENTO
    FOREIGN KEY(Numero_departamento) REFERENCES DEPARTAMENTO(Numero_departamento)
          ON DELETE SET DEFAULT                   ON UPDATE CASCADE);
```

https://www.programiz.com/sql/online-compiler/

ESPECIFICANDO RESTRICÕES SOBRE TUPLAS USANDO CHECK

```
CHECK (Data_criacao <= Data_inicio_gerente)
```

https://www.programiz.com/sql/online-compiler/

Exercícios!

1. Implemente o Exemplo e teste o código
2. Faça modificações para teste
3. Comece a implementar o seu modelo ER

https://www.programiz.com/sql/online-compiler/

INSTRUÇÃO INSERT

```
U1: INSERT INTO     FUNCIONARIO
    VALUES          ('Ricardo', 'K', 'Marini', '65329865388', '30-12-1962',
                    'Rua Itapira, 44, Santos, SP', 'M', 37000, '65329865388', 4 );
```

```
U1A: INSERT INTO    FUNCIONARIO (Primeiro_nome, Ultimo_nome, Numero_
                    departamento, Cpf)
     VALUES         ('Ricardo', 'Marini', 4, '65329865388');
```

https://www.programiz.com/sql/online-compiler/

INSTRUÇÃO DELETE

```
U4A:  DELETE FROM   FUNCIONARIO
      WHERE         Ultimo_nome = 'Braga';
U4B:  DELETE FROM   FUNCIONARIO
      WHERE         Cpf = '12345678966';
U4C:  DELETE FROM   FUNCIONARIO
      WHERE         Numero_departamento = 5;
U4D:  DELETE FROM   FUNCIONARIO;
```

https://www.programiz.com/sql/online-compiler/

INSTRUÇÃO UPDATE

```
U5:  UPDATE  PROJETO
     SET     Local_projeto = 'Santo André', Numero_departamento = 5
     WHERE   Numero_projeto = 10;
```

```
U6:  UPDATE  FUNCIONARIO
     SET     Salario = Salario * 1.1
     WHERE   Numero_departamento = 5;
```

https://www.programiz.com/sql/online-compiler/

FUNCIONARIO

Primeiro_nome	Nome_meio	Ultimo_nome	Cpf	Data_nascimento	Endereco	Sexo	Salario	Cpf_supervisor	Numero_departamento
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Alice	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

DEPARTAMENTO

Nome_departamento	Numero_departamento	Cpf_gerente	Data_inicio_gerente
Pesquisa	5	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

LOCALIZACOES_DEPARTAMENTO

Numero_departamento	Local
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo

TRABALHA_EM

Cpf_funcionario	Numero_projeto	Horas
12345678966	1	32,5
12345678966	2	7,5
66688444476	3	40,0
45345345376	1	20,0
45345345376	2	20,0
33344555587	2	10,0
33344555587	3	10,0
33344555587	10	10,0
33344555587	20	10,0
99988777767	30	30,0
99988777767	10	10,0
98798798733	10	35,0
98798798733	30	5,0
98765432168	30	20,0
98765432168	20	15,0
88866555576	20	NULL

PROJETO

Nome_projeto	Numero_projeto	Local_projeto	Numero_departamento
ProdutoX	1	Santo André	5
ProdutoY	2	Itu	5
ProdutoZ	3	São Paulo	5
Informatização	10	Mauá	4
Reorganização	20	São Paulo	1
Novosbenefícios	30	Mauá	4

DEPENDENTE

Cpf_funcionario	Nome_dependente	Sexo	Data_nascimento	Parentesco
33344555587	Alicia	F	05-04-1986	Filha
33344555587	Tiago	M	25-10-1983	Filho
33344555587	Janaína	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha
12345678966	Elizabeth	F	05-05-1967	Esposa

CONSULTA DE RECUPERAÇÃO BASICA

```
SELECT    <lista atributos>
FROM      <lista tabelas>
WHERE     <condição>;
```

em que

- <lista atributos> é uma lista de nomes de atributo cujos valores devem ser recuperados pela consulta.
- <lista tabelas> é uma lista dos nomes de relação exigidos para processar a consulta.
- <condição> é uma expressão condicional (booleana) que identifica as tuplas a serem recuperadas pela consulta.

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 0. Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.

```
C0:  SELECT Data_nascimento, Endereco
     FROM   FUNCIONARIO
     WHERE  Primeiro_nome='João' AND Nome_meio='B' AND
            Ultimo_nome='Silva';
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 0. Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.

```
C0:  SELECT Data_nascimento, Endereco
     FROM   FUNCIONARIO
     WHERE  Primeiro_nome='João' AND Nome_meio='B' AND
            Ultimo_nome='Silva';
```

(a)

Data_nascimento	Endereco
09-01-1965	Rua das Flores, 751, São Paulo, SP

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 1. Recuperar o nome e o endereço de todos os funcionários que trabalham para o departamento 'Pesquisa'.

```
C1: SELECT Primeiro_nome, Ultimo_nome, Endereco
    FROM   FUNCIONARIO, DEPARTAMENTO
    WHERE  Nome_departamento ='Pesquisa' AND DEPARTAMENTO.Numero_
           departamento = FUNCIONARIO.Numero_departamento;
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 1. Recuperar o nome e o endereço de todos os funcionários que trabalham para o departamento 'Pesquisa'.

```
C1: SELECT Primeiro_nome, Ultimo_nome, Endereco
    FROM   FUNCIONARIO, DEPARTAMENTO
    WHERE  Nome_departamento ='Pesquisa' AND DEPARTAMENTO.Numero_
           departamento = FUNCIONARIO.Numero_departamento;
```

Primeiro_nome	Ultimo_nome	Endereco
João	Silva	Rua das Flores, 751, São Paulo, SP
Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
Joice	Leite	Av. Lucas Obes, 74, São Paulo, SP

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 2. Para cada projeto localizado em 'Mauá', liste o número do projeto, o número do departamento que o controla e sobrenome, endereço e data de nascimento do gerente do departamento.

```
C2: SELECT Numero_projeto, DEPARTAMENTO.Numero_departamento, Ultimo_
        nome, Endereco, Data_nascimento
    FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
    WHERE PROJETO.Numero_departamento = DEPARTAMENTO.Numero_depar-
        tamento AND Cpf_gerente = Cpf AND Local_projeto = 'Mauá';
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 2. Para cada projeto localizado em 'Mauá', liste o número do projeto, o número do departamento que o controla e sobrenome, endereço e data de nascimento do gerente do departamento.

```
C2: SELECT Numero_projeto, DEPARTAMENTO.Numero_departamento, Ultimo_
        nome, Endereco, Data_nascimento
    FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
    WHERE PROJETO.Numero_departamento = DEPARTAMENTO.Numero_depar-
        tamento AND Cpf_gerente = Cpf AND Local_projeto = 'Mauá';
```

Numero_projeto	DEPARTAMENTO. Numero_departa-mento	Ultimo_nome	Endereco	Data_nascimento
10	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941
30	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 2. Para cada projeto localizado em 'Mauá', liste o número do projeto, o número do departamento que o controla e sobrenome, endereço e data de nascimento do gerente do departamento.

```
C2: SELECT Numero_projeto, DEPARTAMENTO.Numero_departamento, Ultimo_
        nome, Endereco, Data_nascimento
    FROM PROJETO, DEPARTAMENTO, FUNCIONARIO
    WHERE PROJETO.Numero_departamento = DEPARTAMENTO.Numero_depar-
        tamento AND Cpf_gerente = Cpf AND Local_projeto = 'Mauá';
```

Numero_projeto	DEPARTAMENTO. Numero_departa-mento	Ultimo_nome	Endereco	Data_nascimento
10	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941
30	4	Souza	Av. Artur de Lima, 54, Santo André, SP	20-06-1941

CONSULTA DE RECUPERAÇÃO BASICA

- Nomes de atributos ambíguos, apelido, renomeação e variáveis de tulplas

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 8. Para cada funcionário, recupere o primeiro e o último nome do funcionário e o primeiro e o último nome de seu supervisor imediato.

```
C8: SELECT F.Primeiro_nome, F.Ultimo_nome, S.Primeiro_nome, S.Ultimo_nome
    FROM   FUNCIONARIO AS F, FUNCIONARIO AS S
    WHERE  F.Cpf_supervisor=S.Cpf;
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Consulta 8. Para cada funcionário, recupere o primeiro e o último nome do funcionário e o primeiro e o último nome de seu supervisor imediato.

```
C8: SELECT F.Primeiro_nome, F.Ultimo_nome, S.Primeiro_nome, S.Ultimo_nome
    FROM   FUNCIONARIO AS F, FUNCIONARIO AS S
    WHERE  F.Cpf_supervisor=S.Cpf;
```

F.Primeiro_nome	F.Ultimo_nome	S.Primeiro_nome	S.Ultimo_nome
João	Silva	Fernando	Wong
Fernando	Wong	Jorge	Brito
Alice	Zelaya	Jennifer	Souza
Jennifer	Souza	Jorge	Brito
Ronaldo	Lima	Fernando	Wong
Joice	Leite	Fernando	Wong
André	Pereira	Jennifer	Souza

CONSULTA DE RECUPERAÇÃO BASICA

- Nomes de atributos ambíguos, apelido, renomeação e variáveis de tulplas

FUNCIONARIO **AS** F(Pn, Nm, Un, Cpf, Dn, End, Sexo, Sal, Cpfs, Nd)

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

- Pode usar o mecanismo de nomeação em qualquer consulta

```
C1B: SELECT F.Primeiro_nome, F.Ultimo_nome, F.Endereco
     FROM   FUNCIONARIO AS F, DEPARTAMENTO AS D
     WHERE  D.Nome_departamento = 'Pesquisa' AND
            D.Numero_departamento=F.Numero_departamento;
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

- Cláusula WHERE não especificada e uso do asterisco

CONSULTA DE RECUPERAÇÃO BASICA

Consultas 9 e 10. Selecionar todos os Cpfs de FUNCIONARIO (C9) e todas as combinações de Cpf de FUNCIONARIO e Nome_departamento de DEPARTAMENTO (C10) no banco de dados.

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Consultas 9 e 10. Selecionar todos os Cpfs de FUNCIONARIO (C9) e todas as combinações de Cpf de FUNCIONARIO e Nome_departamento de DEPARTAMENTO (C10) no banco de dados.

```
C9:  SELECT Cpf
     FROM   FUNCIONARIO;

C10: SELECT Cpf, Nome_departamento
     FROM   FUNCIONARIO, DEPARTAMENTO;
```

CONSULTA DE RECUPERAÇÃO BASICA

Cpf
12345678966
33344555587
99988777767
98765432168
66688444476
45345345376
98798798733
88866555576

Cpf	Nome_departamento
12345678966	Pesquisa
33344555587	Pesquisa
99988777767	Pesquisa
98765432168	Pesquisa
66688444476	Pesquisa
45345345376	Pesquisa
98798798733	Pesquisa
88866555576	Pesquisa
12345678966	Administração
33344555587	Administração
99988777767	Administração
98765432168	Administração
66688444476	Administração

45345345376	Administração
98798798733	Administração
88866555576	Administração
12345678966	Matriz
33344555587	Matriz
99988777767	Matriz
98765432168	Matriz
66688444476	Matriz
45345345376	Matriz
98798798733	Matriz
88866555576	Matriz

CONSULTA DE RECUPERAÇÃO BASICA

```
C1C: SELECT *
     FROM   FUNCIONARIO
     WHERE  Numero_departamento = 5;

C1D: SELECT *
     FROM   FUNCIONARIO, DEPARTAMENTO
     WHERE  Nome_departamento = 'Pesquisa' AND DEPARTAMENTO.Numero_
            departamento = FUNCIONARIO.Numero_departamento;

C10A: SELECT *
     FROM   FUNCIONARIO, DEPARTAMENTO;
```

https://www.programiz.com/sql/online-compiler/

CONSULTA DE RECUPERAÇÃO BASICA

Primeiro_nome	Nome_meio	Ultimo_nome	Cpf	Data_nascimento	Endereco	Sexo	Salario	Cpf_supervisor	Numero_departamento
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40000	88866555576	5
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25000	33344555587	5