

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

PASSO 1: Preparação do Ambiente

Abra a janela de comandos SQL e execute o trecho a seguir.

```
-- -----  
-- Sequence structure for seq_cidade  
-- -----  
DROP SEQUENCE IF EXISTS "public"."seq_cidade";  
CREATE SEQUENCE "public"."seq_cidade"  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 20  
CACHE 1;  
  
-- -----  
-- Sequence structure for seq_contratacao  
-- -----  
DROP SEQUENCE IF EXISTS "public"."seq_contratacao";  
CREATE SEQUENCE "public"."seq_contratacao"  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1000  
CACHE 1;  
  
-- -----  
-- Sequence structure for seq_empregado  
-- -----  
DROP SEQUENCE IF EXISTS "public"."seq_empregado";  
CREATE SEQUENCE "public"."seq_empregado"  
INCREMENT 1  
MINVALUE 1  
MAXVALUE 9223372036854775807  
START 1000  
CACHE 1;  
  
-- -----  
-- Sequence structure for seq_empresa  
-- -----  
DROP SEQUENCE IF EXISTS "public"."seq_empresa";  
CREATE SEQUENCE "public"."seq_empresa"  
INCREMENT 1  
MINVALUE 1
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

MAXVALUE 9223372036854775807

START 1000

CACHE 1;

-- -----

-- Sequence structure for seq_estado

-- -----

DROP SEQUENCE IF EXISTS "public"."seq_estado";

CREATE SEQUENCE "public"."seq_estado"

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 1000

CACHE 1;

-- -----

-- Sequence structure for seq_profissao

-- -----

DROP SEQUENCE IF EXISTS "public"."seq_profissao";

CREATE SEQUENCE "public"."seq_profissao"

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 1000

CACHE 1;

-- -----

-- Sequence structure for seq_projeto

-- -----

DROP SEQUENCE IF EXISTS "public"."seq_projeto";

CREATE SEQUENCE "public"."seq_projeto"

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 1000

CACHE 1;

-- -----

-- Sequence structure for seq_salario

-- -----

DROP SEQUENCE IF EXISTS "public"."seq_salario";

CREATE SEQUENCE "public"."seq_salario"

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

INCREMENT 1

MINVALUE 1

MAXVALUE 9223372036854775807

START 1000

CACHE 1;

-- -----

-- Table structure for alocacao

-- -----

DROP TABLE IF EXISTS "public"."alocacao";

CREATE TABLE "public"."alocacao" (

 "codemp" int4 NOT NULL,

 "codproj" char(2) COLLATE "pg_catalog"."default" NOT NULL,

 "qtdhoras" int4 NOT NULL

)

;

-- -----

-- Records of alocacao

-- -----

INSERT INTO "public"."alocacao" VALUES (101, 'PD', 100);

INSERT INTO "public"."alocacao" VALUES (508, 'TK', 120);

INSERT INTO "public"."alocacao" VALUES (832, 'PD', 200);

INSERT INTO "public"."alocacao" VALUES (270, 'FS', 80);

INSERT INTO "public"."alocacao" VALUES (101, 'MX', 120);

INSERT INTO "public"."alocacao" VALUES (245, 'MX', 80);

INSERT INTO "public"."alocacao" VALUES (330, 'TK', 160);

INSERT INTO "public"."alocacao" VALUES (619, 'MX', 100);

-- -----

-- Table structure for cidade

-- -----

DROP TABLE IF EXISTS "public"."cidade";

CREATE TABLE "public"."cidade" (

 "codcid" int4 NOT NULL DEFAULT nextval('seq_cidade'::regclass),

 "nome" varchar(60) COLLATE "pg_catalog"."default" NOT NULL,

 "codest" int4 NOT NULL

)

;

-- -----

-- Records of cidade

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
-- -----  
INSERT INTO "public"."cidade" VALUES (1, 'Sorocaba', 1);  
INSERT INTO "public"."cidade" VALUES (2, 'S o Caetano', 1);  
INSERT INTO "public"."cidade" VALUES (3, 'Resende', 2);  
INSERT INTO "public"."cidade" VALUES (4, 'Blumenau', 3);  
INSERT INTO "public"."cidade" VALUES (5, 'Salvador', 4);  
INSERT INTO "public"."cidade" VALUES (6, 'Lauro de Freitas', 4);  
INSERT INTO "public"."cidade" VALUES (7, 'Cama ari', 4);  
INSERT INTO "public"."cidade" VALUES (8, 'Feira de Santana', 4);  
INSERT INTO "public"."cidade" VALUES (9, 'Florian polis', 3);  
INSERT INTO "public"."cidade" VALUES (10, 'Rio de Janeiro', 2);  
INSERT INTO "public"."cidade" VALUES (20, 'Santo Amaro', 4);  
INSERT INTO "public"."cidade" VALUES (21, 'Camamu', 4);  
INSERT INTO "public"."cidade" VALUES (22, 'Mutuipe', 4);
```

```
-- -----  
-- Table structure for contratacao  
-- -----
```

```
DROP TABLE IF EXISTS "public"."contratacao";  
CREATE TABLE "public"."contratacao" (  
    "codcont" int4 NOT NULL DEFAULT nextval('seq_contratacao '::regclass),  
    "dtadm" date NOT NULL,  
    "dtdem" date,  
    "codemp" int4  
)  
;
```

```
-- -----  
-- Records of contratacao  
-- -----
```

```
INSERT INTO "public"."contratacao" VALUES (1, '2000-01-01', NULL, 101);  
INSERT INTO "public"."contratacao" VALUES (2, '2001-11-01', NULL, 204);  
INSERT INTO "public"."contratacao" VALUES (3, '2002-10-01', '2015-02-11', 330);  
INSERT INTO "public"."contratacao" VALUES (4, '2002-08-05', NULL, 134);  
INSERT INTO "public"."contratacao" VALUES (5, '2008-09-09', NULL, 245);  
INSERT INTO "public"."contratacao" VALUES (6, '2009-07-01', '2012-02-20', 508);  
INSERT INTO "public"."contratacao" VALUES (7, '2010-02-11', '2014-10-15', 832);  
INSERT INTO "public"."contratacao" VALUES (8, '2013-01-11', NULL, 496);  
INSERT INTO "public"."contratacao" VALUES (9, '2011-09-10', NULL, 270);  
INSERT INTO "public"."contratacao" VALUES (10, '2005-04-25', NULL, 619);  
INSERT INTO "public"."contratacao" VALUES (11, '2014-03-21', '2014-10-11', 620);
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

-- -----
-- Table structure for empregado
-- -----

```
DROP TABLE IF EXISTS "public"."empregado";
CREATE TABLE "public"."empregado" (
  "codemp" int4 NOT NULL DEFAULT nextval('seq_empregado'::regclass),
  "nome" varchar(30) COLLATE "pg_catalog"."default" NOT NULL,
  "codprof" int4
)
;
```

-- -----
-- Records of empregado
-- -----

```
INSERT INTO "public"."empregado" VALUES (101, 'Daniel Rand', 3);
INSERT INTO "public"."empregado" VALUES (204, 'Emma Frost', 5);
INSERT INTO "public"."empregado" VALUES (330, 'Alan Scott', 2);
INSERT INTO "public"."empregado" VALUES (134, 'Matt Murdock', 1);
INSERT INTO "public"."empregado" VALUES (245, 'Jay Garrick', 2);
INSERT INTO "public"."empregado" VALUES (508, 'Helena Bertinelli', 1);
INSERT INTO "public"."empregado" VALUES (832, 'Dinah Lance', 3);
INSERT INTO "public"."empregado" VALUES (496, 'Connor Hawke', 3);
INSERT INTO "public"."empregado" VALUES (270, 'Bart Allen', 5);
INSERT INTO "public"."empregado" VALUES (619, 'Carter Hall', 4);
INSERT INTO "public"."empregado" VALUES (620, 'Abin Sur', NULL);
```

-- -----
-- Table structure for empresa
-- -----

```
DROP TABLE IF EXISTS "public"."empresa";
CREATE TABLE "public"."empresa" (
  "codempresa" int4 NOT NULL DEFAULT nextval('seq_empresa'::regclass),
  "nome" varchar(60) COLLATE "pg_catalog"."default" NOT NULL,
  "codcid" int4 NOT NULL
)
;
```

-- -----
-- Records of empresa
-- -----

```
INSERT INTO "public"."empresa" VALUES (1, 'Votorantim', 1);
INSERT INTO "public"."empresa" VALUES (2, 'BASF', 2);
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
INSERT INTO "public"."empresa" VALUES (3, 'General Eletric', 3);
INSERT INTO "public"."empresa" VALUES (4, 'Tigre', 4);
INSERT INTO "public"."empresa" VALUES (5, 'OAS', 5);
INSERT INTO "public"."empresa" VALUES (6, 'Odebrecht', 5);
INSERT INTO "public"."empresa" VALUES (7, 'White Martins', 10);
INSERT INTO "public"."empresa" VALUES (8, 'Costa Eo do Santinho', 9);
INSERT INTO "public"."empresa" VALUES (9, 'Alcoa', 1);
INSERT INTO "public"."empresa" VALUES (10, 'Braskem', 7);
INSERT INTO "public"."empresa" VALUES (11, 'Petrobr s', 5);
```

```
-- -----
-- Table structure for estado
-- -----
```

```
DROP TABLE IF EXISTS "public"."estado";
CREATE TABLE "public"."estado" (
  "codest" int4 NOT NULL DEFAULT nextval('seq_estado '::regclass),
  "nome" varchar(60) COLLATE "pg_catalog"."default" NOT NULL
)
;
```

```
-- -----
-- Records of estado
-- -----
```

```
INSERT INTO "public"."estado" VALUES (1, 'SP');
INSERT INTO "public"."estado" VALUES (2, 'RJ');
INSERT INTO "public"."estado" VALUES (3, 'SC');
INSERT INTO "public"."estado" VALUES (4, 'BA');
INSERT INTO "public"."estado" VALUES (5, 'RS');
```

```
-- -----
-- Table structure for profissao
-- -----
```

```
DROP TABLE IF EXISTS "public"."profissao";
CREATE TABLE "public"."profissao" (
  "codprof" int4 NOT NULL DEFAULT nextval('seq_profissao '::regclass),
  "nome" varchar(50) COLLATE "pg_catalog"."default" NOT NULL
)
;
```

```
-- -----
-- Records of profissao
-- -----
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
INSERT INTO "public"."profissao" VALUES (1, 'Eletricista');
INSERT INTO "public"."profissao" VALUES (2, 'Pedreiro');
INSERT INTO "public"."profissao" VALUES (3, 'Pintor');
INSERT INTO "public"."profissao" VALUES (4, 'Encanador');
INSERT INTO "public"."profissao" VALUES (5, 'Engenheiro');
```

```
-- -----
-- Table structure for projeto
-- -----
```

```
DROP TABLE IF EXISTS "public"."projeto";
CREATE TABLE "public"."projeto" (
  "codproj" char(2) COLLATE "pg_catalog"."default" NOT NULL DEFAULT nextval('seq_projeto)::regclass),
  "nome" varchar(50) COLLATE "pg_catalog"."default" NOT NULL,
  "codempresa" int4
)
;
```

```
-- -----
-- Records of projeto
-- -----
```

```
INSERT INTO "public"."projeto" VALUES ('FS', 'Fortaleza da Solidao', 4);
INSERT INTO "public"."projeto" VALUES ('MX', 'Mansao Xavier', 5);
INSERT INTO "public"."projeto" VALUES ('PD', 'Jornal Planeta Diario', 6);
INSERT INTO "public"."projeto" VALUES ('TK', 'Triskellion', 7);
```

```
-- -----
-- Table structure for salario
-- -----
```

```
DROP TABLE IF EXISTS "public"."salario";
CREATE TABLE "public"."salario" (
  "codsal" int4 NOT NULL DEFAULT nextval('seq_salario)::regclass),
  "salario_hora" float8 NOT NULL,
  "codprof" int4
)
;
```

```
-- -----
-- Records of salario
-- -----
```

```
INSERT INTO "public"."salario" VALUES (1, 13, 1);
INSERT INTO "public"."salario" VALUES (2, 15, 2);
INSERT INTO "public"."salario" VALUES (3, 12, 3);
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
INSERT INTO "public"."salario" VALUES (4, 10, 4);
```

```
INSERT INTO "public"."salario" VALUES (5, 30, 5);
```

```
-- -----  
-- Function structure for anosservico  
-- -----
```

```
DROP FUNCTION IF EXISTS "public"."anosservico"("nome" text);
```

```
CREATE OR REPLACE FUNCTION "public"."anosservico"("nome" text)
```

```
    RETURNS "pg_catalog"."int4" AS $BODY$
```

```
DECLARE soma INTEGER;
```

```
BEGIN
```

```
    SELECT (DATE_PART('year', c.dtdem) - DATE_PART('year', c.dtadm)) FROM CONTRATACAO c JOIN EMPREGADO e  
ON c.codemp = e.codemp WHERE UPPER(e.nome) = UPPER($1) INTO soma;
```

```
    RETURN soma;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql VOLATILE
```

```
COST 100;
```

```
-- -----  
-- Function structure for custototalprojeto  
-- -----
```

```
DROP FUNCTION IF EXISTS "public"."custototalprojeto"("projeto" bpchar);
```

```
CREATE OR REPLACE FUNCTION "public"."custototalprojeto"("projeto" bpchar)
```

```
    RETURNS "pg_catalog"."int4" AS $BODY$
```

```
DECLARE resultado INTEGER;
```

```
BEGIN
```

```
    SELECT p.codproj, SUM(salario_hora * qtdhoras)
```

```
    FROM alocao as a, projeto as p, empregado as e,
```

```
         profissao as prof, salario as s
```

```
WHERE
```

```
    a.codproj = p.codproj AND
```

```
    a.codemp = e.codemp AND
```

```
    e.codprof = prof.codprof AND
```

```
    s.codprof = prof.codprof AND
```

```
    p.codproj = $1
```

```
GROUP BY
```

```
    p.codproj INTO resultado;
```

```
RETURN resultado;
```


	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
END;

$BODY$
    LANGUAGE plpgsql VOLATILE
    COST 100;

-- -----
-- Function structure for custototalprojeto
-- -----

DROP FUNCTION IF EXISTS "public"."custototalprojeto"("projeto" text);
CREATE OR REPLACE FUNCTION "public"."custototalprojeto"("projeto" text)
    RETURNS "pg_catalog"."int4" AS $BODY$
DECLARE resultado INTEGER;
BEGIN

    SELECT SUM(salario_hora * qtdhoras)
    FROM alocao as a, projeto as p, empregado as e,
         profissao as prof, salario as s
    WHERE
        a.codproj = p.codproj AND
        a.codemp = e.codemp AND
        e.codprof = prof.codprof AND
        s.codprof = prof.codprof AND
        p.codproj = $1
    GROUP BY
        p.codproj INTO resultado;

    RETURN resultado;

END;

$BODY$
    LANGUAGE plpgsql VOLATILE
    COST 100;

-- -----
-- Function structure for custototalprojeto
-- -----

DROP FUNCTION IF EXISTS "public"."custototalprojeto"("projeto" int4);
CREATE OR REPLACE FUNCTION "public"."custototalprojeto"("projeto" int4)
    RETURNS "pg_catalog"."int4" AS $BODY$
DECLARE resultado INTEGER;
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

BEGIN

```
SELECT p.codproj, SUM(salario_hora * qtdhoras)
FROM alocao as a, projeto as p, empregado as e,
     profissao as prof, salario as s
```

WHERE

```
a.codproj = p.codproj AND
a.codemp = e.codemp AND
e.codprof = prof.codprof AND
s.codprof = prof.codprof AND
p.codproj = $1
```

GROUP BY

```
p.codproj INTO resultado;
```

RETURN resultado;

END;

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

-- Function structure for getanosdeservico

-- -----

DROP FUNCTION IF EXISTS "public"."getanosdeservico"("codigoemp" int4);

CREATE OR REPLACE FUNCTION "public"."getanosdeservico"("codigoemp" int4)

RETURNS "pg_catalog"."int4" AS \$BODY\$

DECLARE resultado INTEGER;

BEGIN

```
SELECT (DATE_PART('year', c.dtdem) - DATE_PART('year', c.dtadm)) FROM empregado AS e JOIN contratacao AS c
ON (e.codemp = c.codemp) WHERE e.codemp = $1 INTO resultado;
```

RETURN resultado;

END;

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

-- Function structure for getprofissao

-- -----

DROP FUNCTION IF EXISTS "public"."getprofissao"("codigoempregado" int4);

CREATE OR REPLACE FUNCTION "public"."getprofissao"("codigoempregado" int4)

RETURNS "pg_catalog"."text" AS \$BODY\$

DECLARE resultado TEXT;

BEGIN

-- Buscando a descricao da profissao do codemp

SELECT p.nome FROM empregado AS e JOIN profissao AS p ON (e.codprof=p.codprof) WHERE e.codemp =
codigoempregado INTO resultado;

RETURN resultado;

END;

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

-- Function structure for getstatusempregado

-- -----

DROP FUNCTION IF EXISTS "public"."getstatusempregado"("codigoemp" int4);

CREATE OR REPLACE FUNCTION "public"."getstatusempregado"("codigoemp" int4)

RETURNS "pg_catalog"."text" AS \$BODY\$

DECLARE datademissao DATE;

BEGIN

-- Buscamos a data de demissao no banco de dados

SELECT c.dtdem FROM empregado as e, contratacao as c

WHERE

e.codemp = c.codemp AND

e.codemp = \$1 INTO datademissao;

-- Avaliamos se o funcionario estava ativo ou desligado

IF datademissao IS NULL THEN

RETURN 'Ativo';

ELSE

RETURN 'Inativo';

END IF;

END;

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

-- Function structure for isdataadmissao

-- -----

DROP FUNCTION IF EXISTS "public"."isdataadmissao"("codigoemp" int4);

CREATE OR REPLACE FUNCTION "public"."isdataadmissao"("codigoemp" int4)

RETURNS "pg_catalog"."text" AS \$BODY\$

DECLARE dataadmissao DATE;

BEGIN

-- Buscamos a data de admissao no banco de dados

SELECT c.dtadm FROM contratacao as c

WHERE

c.codemp = \$1 AND

DATE_PART('DAY',c.dtadm) = DATE_PART('DAY',CURRENT_DATE) AND

DATE_PART('MONTH',c.dtadm) = DATE_PART('MONTH',CURRENT_DATE)

INTO dataadmissao;

-- Avaliamos se o funcionario estava ativo ou desligado

IF dataadmissao IS NULL THEN

SELECT c.dtadm FROM contratacao as c WHERE c.codemp = \$1 INTO dataadmissao;

RETURN CONCAT('O aniversario de admissao é na data ',TO_CHAR(dataadmissao,'DD/MM/YYYY'), '.') as resultado;

ELSE

RETURN 'Parabens' as resultado;

END IF;

-- -----
--IF <condicao> THEN

--<comando>

--END IF;

-- -----
--IF <condicao> THEN

--<comando>

--ELSIF THEN

--<comando>

--ELSE

--<comando>

--END IF;

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

END;

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

-- Function structure for isfuncionariocar

-- -----

DROP FUNCTION IF EXISTS "public"."isfuncionariocar"("codigoproj" int4);

CREATE OR REPLACE FUNCTION "public"."isfuncionariocar"("codigoproj" int4)

RETURNS "pg_catalog"."text" AS \$BODY\$

DECLARE nomeprojeto TEXT;

DECLARE empregado TEXT;

BEGIN

--

SELECT nome FROM projeto WHERE codproj = \$1 INTO nomeprojeto;

SELECT nome FROM vw_projeto_custo_total

WHERE

nome_projeto LIKE nomeprojeto

ORDER BY

salario_total

DESC LIMIT 1 INTO empregado;

RETURN empregado;

END;

\$BODY\$

LANGUAGE plpgsql VOLATILE

COST 100;

-- -----

-- Function structure for isfuncionariocar

-- -----

DROP FUNCTION IF EXISTS "public"."isfuncionariocar"("codigoproj" bpchar);

CREATE OR REPLACE FUNCTION "public"."isfuncionariocar"("codigoproj" bpchar)

RETURNS "pg_catalog"."text" AS \$BODY\$

DECLARE nomeprojeto TEXT;

DECLARE empregado TEXT;

BEGIN

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

--

```
SELECT nome FROM projeto WHERE codproj = $1 INTO nomeprojeto;
```

```
SELECT nome FROM vw_projeto_custo_total
```

```
WHERE
```

```
    nome_projeto LIKE nomeprojeto
```

```
ORDER BY
```

```
    salario_total
```

```
DESC LIMIT 1 INTO empregado;
```

```
RETURN empregado;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql VOLATILE
```

```
COST 100;
```

-- -----

```
-- Function structure for tiraaccento
```

-- -----

```
DROP FUNCTION IF EXISTS "public"."tiraaccento"("texto" text);
```

```
CREATE OR REPLACE FUNCTION "public"."tiraaccento"("texto" text)
```

```
    RETURNS "pg_catalog"."text" AS $BODY$
```

```
DECLARE resultado TEXT;
```

```
BEGIN
```

```
    -- Retirando os acentos do texto e deixando ele UPPER
```

```
    SELECT UPPER($1) INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Ã', 'A') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'É', 'E') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Ó', 'O') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Õ', 'O') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Ç', 'C') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Á', 'A') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'À', 'A') INTO resultado;
```

```
    SELECT REPLACE(resultado, 'Ú', 'U') INTO resultado;
```

```
RETURN resultado;
```

```
END;
```

```
$BODY$
```

```
LANGUAGE plpgsql VOLATILE
```

```
COST 100;
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

-- -----

-- View structure for vw_dados_profissionais

-- -----

DROP VIEW IF EXISTS "public"."vw_dados_profissionais";

CREATE VIEW "public"."vw_dados_profissionais" AS SELECT e.nome AS nome_completo,

prof.nome AS profissao,

emp.nome AS empresa,

c.nome AS cidade,

est.nome AS estado

FROM empregado e,

alocacao a,

profissao prof,

projeto p,

empresa emp,

cidade c,

estado est

WHERE ((((((e.codemp = a.codemp) AND (e.codprof = prof.codprof)) AND (a.codproj = p.codproj)) AND (emp.codempresa = p.codempresa)) AND (c.codcid = emp.codcid)) AND (est.codest = c.codest));

-- -----

-- View structure for vw_empregado_projetos

-- -----

DROP VIEW IF EXISTS "public"."vw_empregado_projetos";

CREATE VIEW "public"."vw_empregado_projetos" AS SELECT empregado.codemp,

empregado.nome,

empregado.codprof,

profissao.nome AS profissao,

alocacao.codproj,

alocacao.qtdhoras,

projeto.nome AS nome_projeto

FROM empregado,

profissao,

alocacao,

projeto

WHERE (((((alocacao.codemp = empregado.codemp) AND (profissao.codprof = empregado.codprof)) AND (projeto.codproj = alocacao.codproj)) AND (alocacao.qtdhoras >= 100))

ORDER BY empregado.nome;

-- -----

-- View structure for vw_projeto_custo_total

-- -----

DROP VIEW IF EXISTS "public"."vw_projeto_custo_total";

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
CREATE VIEW "public"."vw_projeto_custo_total" AS SELECT empregado.codemp,
    empregado.nome,
    profissao.nome AS profissao,
    (salario.salario_hora * (alocacao.qtdhoras)::double precision) AS salario_total,
    alocacao.qtdhoras,
    projeto.nome AS nome_projeto,
    ( SELECT sum((ssub.salario_hora * (asub.qtdhoras)::double precision)) AS sum
      FROM alocacao asub,
      empregado esub,
      salario ssub
     WHERE (((asub.codproj = projeto.codproj) AND (asub.codemp = esub.codemp)) AND (esub.codprof =
ssub.codprof))) AS "custo total",
    empresa.nome AS empresa,
    cidade.nome AS cidade,
    estado.nome AS estado
FROM empregado,
    profissao,
    salario,
    alocacao,
    projeto,
    empresa,
    cidade,
    estado
WHERE ((((((alocacao.codemp = empregado.codemp) AND (profissao.codprof = empregado.codprof)) AND
(salario.codprof = profissao.codprof)) AND (projeto.codproj = alocacao.codproj)) AND (empresa.codempres =
projeto.codempres)) AND (cidade.codcid = empresa.codcid)) AND (estado.codest = cidade.codest))
ORDER BY empregado.nome;
```

```
-- -----
-- View structure for vw_estado_count_cidade
-- -----
```

```
DROP VIEW IF EXISTS "public"."vw_estado_count_cidade";
CREATE VIEW "public"."vw_estado_count_cidade" AS SELECT est.nome,
    count(*) AS count
FROM ((estado est
    JOIN cidade cid ON ((est.codest = cid.codest)))
    JOIN empresa emp ON ((emp.codcid = cid.codcid)))
GROUP BY est.nome;
```

```
-- -----
-- View structure for vw_cidade_empresa
-- -----
```

```
DROP VIEW IF EXISTS "public"."vw_cidade_empresa";
```


	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

```
CREATE VIEW "public"."vw_cidade_empresa" AS SELECT emp.nome AS empresa,
      cid.nome AS cidade
FROM (cidade cid
      RIGHT JOIN empresa emp ON ((emp.codcid = cid.codcid)))
UNION
SELECT emp.nome AS empresa,
      cid.nome AS cidade
FROM (cidade cid
      LEFT JOIN empresa emp ON ((emp.codcid = cid.codcid)));
```

```
-- -----
```

```
-- Alter sequences owned by
```

```
-- -----
```

```
SELECT setval('"public"."seq_cidade"', 23, true);
SELECT setval('"public"."seq_contratacao"', 1001, false);
SELECT setval('"public"."seq_empregado"', 1001, true);
SELECT setval('"public"."seq_empresa"', 1001, false);
SELECT setval('"public"."seq_estado"', 1001, false);
SELECT setval('"public"."seq_profissao"', 1001, false);
SELECT setval('"public"."seq_projeto"', 1001, false);
SELECT setval('"public"."seq_salario"', 1001, false);
```

```
-- -----
```

```
-- Primary Key structure for table cidade
```

```
-- -----
```

```
ALTER TABLE "public"."cidade" ADD CONSTRAINT "cidade_pkey" PRIMARY KEY ("codcid");
```

```
-- -----
```

```
-- Primary Key structure for table contratacao
```

```
-- -----
```

```
ALTER TABLE "public"."contratacao" ADD CONSTRAINT "contratacao_pkey" PRIMARY KEY ("codcont");
```

```
-- -----
```

```
-- Primary Key structure for table empregado
```

```
-- -----
```

```
ALTER TABLE "public"."empregado" ADD CONSTRAINT "empregado_pkey" PRIMARY KEY ("codemp");
```

```
-- -----
```

```
-- Primary Key structure for table empresa
```

```
-- -----
```

```
ALTER TABLE "public"."empresa" ADD CONSTRAINT "empresa_pkey" PRIMARY KEY ("codempresa");
```

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

-- Primary Key structure for table estado

ALTER TABLE "public"."estado" ADD CONSTRAINT "estado_pkey" PRIMARY KEY ("codest");

-- Primary Key structure for table profissao

ALTER TABLE "public"."profissao" ADD CONSTRAINT "profissao_pkey" PRIMARY KEY ("codprof");

-- Primary Key structure for table projeto

ALTER TABLE "public"."projeto" ADD CONSTRAINT "projeto_pkey" PRIMARY KEY ("codproj");

-- Primary Key structure for table salario

ALTER TABLE "public"."salario" ADD CONSTRAINT "salario_pkey" PRIMARY KEY ("codsal");

Estes comandos vão criar um novo banco de dados com tabelas e dados necessários ao restante do exercício.

	<h1>INTRODUÇÃO A BANCO DE DADOS</h1> <h2>Roteiro – Aula 07</h2>
--	---

PASSO 2: TRABALHANDO COM FUNCTIONS

Baseado nas instruções dadas em aulas anteriores, escreva comandos SQL que criem as seguintes triggers e objetos do banco de dados:

1. Crie uma tabela log. Essa tabela deve ser formada pelos seguintes atributos:
 - a. Codlog Integer
 - b. Data_hora DATETIME
 - c. Operacao VARCHAR(1)
 - d. Tabela TEXT
 - e. Valor TEXT
2. Agora que a tabela de log já foi criada, crie para as tabelas alocação, contratação, empregado e folha funções que tem o objetivo de criar um log auditável para essas tabelas. Em seguida implemente elas utilizando a estrutura de triggers.
3. Implemente na tabela empregado a trigger que toda vez que a profissão do empregado for alterada, o último salário da folha também será atualizado.
4. Implemente na tabela alocacao a trigger que toda vez que a quantidade de horas trabalhadas pelo empregado for alterada, o último salário da folha também será atualizado.
5. Todas as vezes que um novo registro for inserido na tabela estado ou cidade o nome precisa estar maiúsculo. Implemente esse requisito em SQL, utilizando triggers.
6. Construa um código usando objetos SQL que caso o funcionário seja desligado será gerada o salário do mês atual com o salário acrescido de 30% do salário normal. Armazene esse salário na tabela folha,