

GDI 2023.1

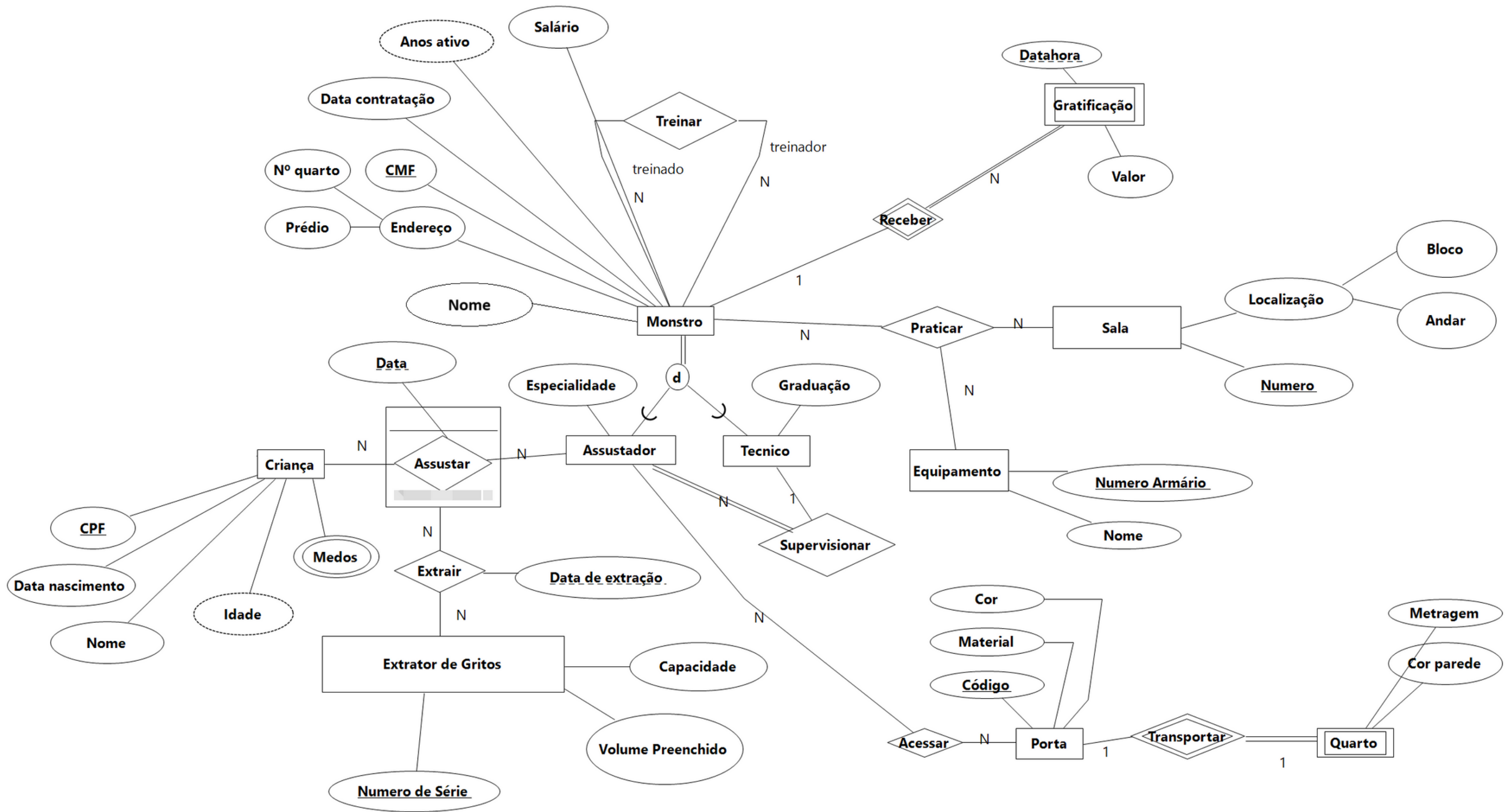
# PROJETO MONSTROS S.A.

acn2 – Ariovaldo Capuano Neto

chmfc – Carlos Henrique Matos Fontaine Costa

iybp – Iara Yasmin Batista Pereira

lfc4 – Lorenzo Fontenelle Chaves



# EER Monstros SA

**Sala** (Número, Localização\_Bloco, Localização\_Andar);

**Equipamento** (Numero Armário, Nome)

**ExtratorDeGritos** (Numero Serie, Capacidade, Volume\_Preenchido)

**Criança** (CPF, Nome!, Data\_nascimento);

**Medos** (CPF Criança, Medo)

CPF\_Criança -> Criança (CPF)

**Porta** (Código, Material, Cor);

**Quarto** (Código, Cor\_parede, Metragem)

Código -> Porta (Código)

**Monstro** (CMF, Nome!, Salário, Data\_contratação, Endereço\_Quarto, Endereço\_Prédio)

**Gratificação** (CMF, Datahora, Valor)

CMF -> Monstro (CMF)

**Assustador** (CMF, CMF\_Supervisor!, Especialidade)

CMF -> Monstro (CMF)

CMF\_Supervisor -> Tecnico (CMF)

**Técnico** (CMF, Graduação)

CMF -> Monstro (CMF)

**Treinar** (CMF Treinado, CMF Treinador)

CMF\_Treinado -> Monstro (CMF)

CMF\_Treinador -> Monstro (CMF)

**Praticar** (CMF, Número, Numero Armário)

CMF -> Monstro (CMF)

Número -> Sala (Número)

Numero\_Armário -> Equipamento (Numero\_Armario)

**Assustar** (CPF, CMF, Data)

CPF -> Criança (CPF)

CMF -> Assustador (CMF)

**Extrair** (Numero Serie, CPF, CMF, Data, Data Extração)

(CPF, CMF, Data) -> Assustar (CPF, CMF, Data)

Numero\_Serie -> ExtratorDeGritos (Numero\_Serie)

**Assustador\_Acessa\_Porta** (CMF, Código)

CMF -> Assustador (CMF)

Código -> Porta (Código)

# Exemplo de população

```
INSERT INTO monstro VALUES ('BYF', 'joseph', 19, 6222.35, '1156', '2188');
INSERT INTO sala VALUES (46, 'A', 9);
INSERT INTO equipamento VALUES (132, 'foto feia');
INSERT INTO extrator_de_gritos VALUES ('MJX-864', 71, 69);
INSERT INTO crianca VALUES ('978', 'pedrinho', to_date('2016-05-30', 'YYYY-MM-DD'));
INSERT INTO porta VALUES ('HHY', 'metal', 'preto');
INSERT INTO quarto VALUES ('PGS', 'vermelho', 46);
INSERT INTO medo VALUES ('978', 'banco de dados');
INSERT INTO gratificacao VALUES ('VRC', to_date('2023-06-24 12:23:57', 'yyyy/mm/dd hh24:mi:ss'), 528.28);
INSERT INTO tecnico VALUES ('JBK', 'teoria do susto');
INSERT INTO treinar VALUES ('OUL', 'MXD');
INSERT INTO praticar VALUES ('QWB', 42, 640);
INSERT INTO assustar VALUES ('378', 'MXD', to_date('2011-08-14', 'YYYY-MM-DD'));
INSERT INTO assustador_acessa_porta VALUES ('PGD', 'CHM');
INSERT INTO assustador VALUES ('YWH', 'RNU', 'crianças');
INSERT INTO extrair VALUES ('NGT-960', '315', 'WUN', to_date('2018-01-17', 'YYYY-MM-DD'), to_date('2023-05-13 05:45:27', 'yyyy/mm/dd hh24:mi:ss'));
```

PROJETO MONSTROS S.A.  
acn2, chmfc, iybp, lfc4

# Buscas no Banco de Dados

Todas as buscas foram feitas na plataforma LIVESQL da ORACLE



```

1  -- Busca os monstros que acessam mais de uma porta e retorna com seu CMF, NOME e Quantidade de Portas
2
3  SELECT  ap.cmf , m.nome, COUNT(ap.codigo_porta) AS QTD_PORTAS
4          FROM  assustador_acessa_porta ap
5          INNER JOIN  monstro m ON (ap.cmf = m.cmf)
6          GROUP BY  ap.cmf, m.nome
7          HAVING COUNT(ap.codigo_porta) > 1
8          ORDER BY  QTD_PORTAS DESC
9
10 -- Preenchendo os requisitos: GROUP BY / HAVING e JUNÇÃO INTERNA

```

CMF	NOME	QTD_PORTAS
URN	gina	3
VUU	william	3
JSG	anthony	2
VBL	renee	2
VWE	sandra	2
PGD	monica	2
NPF	erica	2
OUL	melissa	2
CJZ	amanda	2

```
1 -- Busca todas as crianças que não tem medo e retorna o seu CPF e seu NOME
2
3 SELECT c.cpf, c.nome
4 FROM medo m
5 FULL OUTER JOIN crianca c ON ( m.cpf = c.cpf)
6 WHERE m.cpf IS NULL
7
8 -- Preenchendo os requisitos: JUNÇÃO EXTERNA
```

CPF	NOME
126	brenda
792	nicole
426	adam
795	cody
536	stephen
338	john
949	shelly
507	stephen
383	nicholas
893	justin

```
1  -- Busca os monstros que receberam gratificação
2  SELECT m.nome
3  FROM monstro m
4  WHERE EXISTS (
5      SELECT g.cmf
6      FROM gratificacao g
7      WHERE g.cmf = m.cmf
8  )
9
10 -- SEMI-JOIN
```

NOME
michelle
sara
gina
tammy
nicholas
sandra
william



```
1  -- Busca os assustadores que não assustaram nenhuma criança e retorna com seus nomes
2
3  SELECT m.nome
4  FROM assustador ma
5  INNER JOIN monstro m ON m.cmf = ma.cmf
6  WHERE NOT EXISTS (
7      SELECT a.cmf
8      FROM assustar a
9      WHERE a.cmf = ma.cmf
10 );
11
12 -- Preenchendo os requisitos: ANTI-JOIN
```

NOME
joshua
anthony
kyle
lori
gina
michelle
michele
thomas
richard
sandra
ronald
nicholas
amanda
terri

```
1  -- Busca os monstros que tem Salário maior que a média e retorna com o NOME do monstro e o seu SALARIO
2
3  SELECT m.nome, m.salario
4      FROM monstro m
5  WHERE m.salario > (SELECT AVG(m2.salario) FROM monstro m2)
6  ORDER BY m.salario DESC;
7
8  -- Preenchendo os requisitos: SUBCONSULTA ESCALAR
```

NOME	SALARIO
monica	10000
sandra	9000
erica	9000
jason	9000
jennifer	8000
richard	8000
michele	8000
joshua	8000
tammy	7000
anthony	7000
amanda	7000
joseph	6000
debra	6000

```

1  -- busca o monstro que assustou uma criança em um determinado dia e retorna a CRIANCA_ASSUSTADA e o MONSTRO_ASSUSTADOR
2
3  SELECT c.nome as crianca_assustada,
4         (SELECT m.nome
5          FROM monstro m
6          JOIN assustar a ON m.cmf = a.cmf
7          WHERE a.cpf = c.cpf
8          AND a.data_susto = TO_DATE('2011-08-14', 'YYYY-MM-DD'))
9         as monstro_assustador
10 FROM crianca c
11 WHERE c.cpf = '378'
12
13 -- Prenchendo os requisitos: SUBCONSULTA DO TIPO LINHA

```

CRIANCA_ASSUSTADA	MONSTRO_ASSUSTADOR
heidi	patricia

```
1  -- Busca os monstros que estão praticando nas salas localizadas no bloco A e retorna o CMF e o NOME dos monstros
2
3  SELECT m.cmf, m.nome
4  FROM monstro m
5  WHERE m.cmf IN (
6      SELECT DISTINCT p.cmf
7      FROM praticar p
8      INNER JOIN sala s ON p.numero_sala = s.numero
9      WHERE s.localizacao_bloco = 'A'
10 );
11
12 -- Preenchendo os requisitos: SUBCONSULTA DO TIPO TABELA
```

CMF	NOME
EVC	ronald
SCE	kyle
HBJ	sara
CAM	joshua
URN	gina
CJZ	amanda
YGM	jeffrey
RNU	nicholas
YWH	michele
VWE	sandra
VUU	william

```

1  -- Busca os assustadores que acessaram portas
2  SELECT m.nome
3  FROM monstro m
4  INNER JOIN assustador a on m.cmf = a.cmf
5
6  INTERSECT
7
8  SELECT m.nome
9  FROM monstro m
10 INNER JOIN assustador_acessa_porta ap ON m.cmf = ap.cmf
11
12 -- Operação de conjuntos
13

```

NOME
amanda
curtis
erica
gina
joshua
melissa
michele
monica
nicholas
patricia
sandra
sara
william

PROJETO MONSTROS S.A.  
acn2, chmfc, iybp, lfc4

# PL/SQL

plataforma LIVESQL da ORACLE



```
1  -- Cria um TRIGGER: Exige que o salário do monstro seja pelo menos 1000
2
3  CREATE OR REPLACE TRIGGER salario_minimo_monstro
4  BEFORE INSERT OR UPDATE ON monstro
5  FOR EACH ROW
6  DECLARE
7      salario_minimo NUMBER := 1000; -- salario minimo
8  BEGIN
9      IF :new.salario < salario_minimo THEN
10         DBMS_OUTPUT.PUT_LINE('Salario tem que ser maior que o salario minimo');
11     END IF;
12 END;
13
14
```

```
1  -- Cria uma FUNCTION que calcula a gratificação do monstro
2
3  CREATE OR REPLACE FUNCTION calcular_gratificacao(cmf_in VARCHAR)
4  RETURN FLOAT
5  AS
6      gratificacao_total FLOAT := 0;
7  BEGIN
8      SELECT SUM(valor) INTO gratificacao_total
9      FROM gratificacao
10     WHERE cmf = cmf_in;
11     RETURN gratificacao_total;
12 END;
```

```
1  -- Cria um TRIGGER que impede que a data de nascimento seja no futuro
2
3  CREATE OR REPLACE TRIGGER validar_data_nascimento
4  BEFORE INSERT OR UPDATE ON crianca
5  FOR EACH ROW
6  BEGIN
7      IF :new.data_nascimento > SYSDATE THEN
8          RAISE_APPLICATION_ERROR(-20001, 'A criança ainda nem nasceu!');
9      END IF;
10 END;
```

```
1  -- Cria um PROCEDURE que remove o acesso que foi feita a porta por um monstro
2
3  CREATE OR REPLACE PROCEDURE remover_acesso(cmf_in VARCHAR, codigo_porta_in VARCHAR)
4  AS
5  BEGIN
6      DELETE FROM assustador_acessa_porta
7      WHERE cmf = cmf_in AND codigo_porta = codigo_porta_in;
8  END;
```

```
1  -- Cria um PROCEDURE que aponta um supervisor para um assustador
2
3  CREATE OR REPLACE PROCEDURE apontar_supervisor(cmf_supervisionado IN VARCHAR2, cmf_supervisor IN VARCHAR2)
4  AS
5  BEGIN
6      UPDATE assustador
7      SET cmf_supervisor = cmf_supervisor
8      WHERE cmf = cmf_supervisionado;
9  END;
```

```
-- função para calcular a idade de uma criança
CREATE OR REPLACE FUNCTION calcular_idade(cpf_in IN VARCHAR2) RETURN NUMBER IS
    data_nasc DATE;
    idade NUMBER;
BEGIN
    SELECT data_nascimento INTO data_nasc
    FROM crianca
    WHERE cpf = cpf_in;

    -- calcular idade em anos
    idade := FLOOR(MONTHS_BETWEEN(SYSDATE, data_nasc) / 12);
RETURN idade;
END;
```

```
1 SELECT cpf, calcular_idade(cpf), nome, data_nascimento
2 FROM crianca
```

CPF	CALCULAR_IDADE(CPF)	NOME	DATA_NASCIMENTO
978	7	tina	30-MAY-16
851	10	jessica	02-MAY-13
532	6	brandon	24-AUG-17
126	7	brenda	07-JUN-16
348	9	casey	23-OCT-13
792	8	nicole	11-AUG-15
523	10	megan	23-MAY-13
426	9	adam	02-JUN-14



PROJETO MONSTROS S.A.  
acn2, chmfc, iybp, lfc4

**OBRIGADO!**