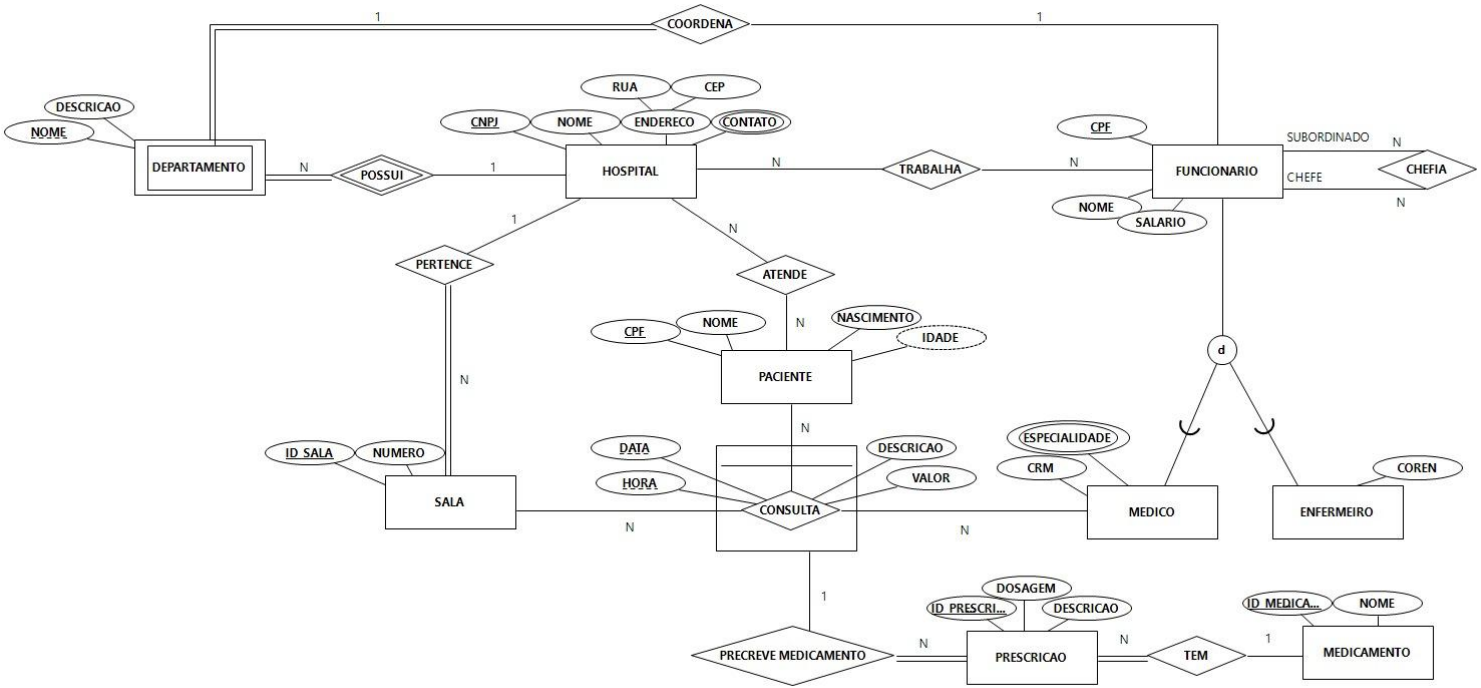
Projeto Banco de Dados Hospital

Caroline Almeida
Diogo Nogueira
Douglas Araújo
Hítalo Nascimento
Ingrid Freire
Otávio Cavalcanti

Centro de Informática UFPE
Universidade Federal de Pernambuco

Modelo Conceitual



Modelo Lógico

HOSPITAL(CNPJ, NOME!, END_RUA, END_CEP)

CONTATO(CNPJ, CONTATO)
CNPJ -> *HOSPITAL*(CNPJ)

FUNCIONARIO(CPF, NOME!, SALARIO!)

CHEFIA(CPF_CHEFE, CPF_SUBORDINADO)
CPF_CHEFE -> *FUNCIONARIO*(CPF)
CPF_SUBORDINADO -> *FUNCIONARIO*(CPF)

TRABALHA(CNPJ, CPF)
CNPJ -> *HOSPITAL*(CNPJ)
CPF -> *FUNCIONARIO*(CPF)

DEPARTAMENTO(CNPJ, NOME, DESCRICAO, [CHEFE_CPF]!)
CNPJ -> *HOSPITAL*(CNPJ)
CHEFE_CPF -> *FUNCIONARIO*(CPF)

MEDICO(CPF_MEDICO, [CRM]!)
CPF_MEDICO -> *FUNCIONARIO*(CPF)

ESPECIALIDADE(CPF_MEDICO, ESPECIALIDADE)
CPF_MEDICO -> *MEDICO*(CPF_MEDICO)

ENFERMEIRO(CPF_ENFERMERIO, [COREN]!)
CPF_ENFERMERIO -> *FUNCIONARIO*(CPF)

PACIENTE(CPF, NOME!, DATA_NASCIMENTO!, IDADE)

ATENDE(CNPJ, CPF)
CNPJ -> *HOSPITAL*(CNPJ)
CPF -> *PACIENTE*(CPF)

SALA(ID_SALA, NUMERO!, CNPJ!)
CNPJ -> *HOSPITAL*(CNPJ)

CONSULTA(ID_SALA, CPF_MEDICO, CPF_PACIENTE, DATA, HORA, VALOR!, DESCRICAO)
ID_SALA -> *SALA*(ID_SALA)
CPF_MEDICO -> *MEDICO*(CPF_MEDICO)
CPF_PACIENTE -> *PACIENTE*(CPF)

MEDICAMENTO(ID_MEDICAMENTO, NOME!)

PRESCRICAO(ID_PRESCRICAO,ID_MEDICAMENTO!, DOSAGEM!, DESCRICAO, (ID_SALA, CPF_MEDICO, CPF_PACIENTE, DATA, HORA)!)
ID_MEDICAMENTO -> *MEDICAMENTO*(ID_MEDICAMENTO)
(ID_SALA, CPF_MEDICO, CPF_PACIENTE, DATA, HORA) -> *CONSULTA*(ID_SALA, CPF_MEDICO, CPF_PACIENTE, DATA, HORA)

Modelo Físico - Criação

```
CREATE TABLE HOSPITAL (
    CNPJ VARCHAR2(14) PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    END_RUA VARCHAR2(100),
    END_CEP VARCHAR2(10)
);

CREATE TABLE CONTATO (
    CNPJ VARCHAR(14),
    CONTATO VARCHAR(100),
    PRIMARY KEY (CNPJ, CONTATO),
    FOREIGN KEY (CNPJ) REFERENCES HOSPITAL (CNPJ)
);

CREATE TABLE FUNCIONARIO (
    CPF VARCHAR2(11) PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    SALARIO NUMBER(10, 2) NOT NULL
);

CREATE TABLE CHEFIA (
    CPF_CHEFE VARCHAR(11),
    CPF_F VARCHAR(11),
    PRIMARY KEY (CPF_CHEFE, CPF_F),
    FOREIGN KEY (CPF_CHEFE) REFERENCES FUNCIONARIO(CPF),
    FOREIGN KEY (CPF_F) REFERENCES FUNCIONARIO(CPF)
);


CREATE TABLE TRABALHA(
    CNPJ VARCHAR2(14),
    CPF VARCHAR2(11),
    FOREIGN KEY (CNPJ) REFERENCES HOSPITAL (CNPJ),
    FOREIGN KEY (CPF) REFERENCES  FUNCIONARIO (CPF),
    PRIMARY KEY (CNPJ, CPF)
);
```

```
CREATE TABLE DEPARTAMENTO (
    CNPJ VARCHAR2(14),
    NOME VARCHAR2(100),
    DESCRICAO VARCHAR2(255),
    CHEFE_CPF VARCHAR2 (11) UNIQUE NOT NULL,
    PRIMARY KEY (CNPJ,NOME),
    FOREIGN KEY (CNPJ) REFERENCES HOSPITAL (CNPJ) ON
DELETE CASCADE
);

CREATE TABLE MEDICO (
    CPF_MEDICO VARCHAR2(11) PRIMARY KEY,
    CRM VARCHAR2(50) UNIQUE NOT NULL,
    FOREIGN KEY (CPF_MEDICO) REFERENCES FUNCIONARIO
(CPF)
);


CREATE TABLE ESPECIALIDADE (
    CPF_MEDICO VARCHAR2(11),
    ESPECIALIDADE VARCHAR2(100),
    PRIMARY KEY(CPF_MEDICO, ESPECIALIDADE),
    FOREIGN KEY (CPF_MEDICO) REFERENCES MEDICO
(CPF_MEDICO)
);

CREATE TABLE ENFERMEIRO (
    CPF_ENFERMEIRO VARCHAR2(11) PRIMARY KEY,
    COREN VARCHAR2(50) UNIQUE NOT NULL,
    FOREIGN KEY (CPF_ENFERMEIRO) REFERENCES
FUNCIONARIO (CPF)
);
```

Modelo Físico - Criação

```
CREATE TABLE PACIENTE (
    CPF VARCHAR2(11) PRIMARY KEY,
    NOME VARCHAR2(100) NOT NULL,
    DATANASCIMENTO DATE NOT NULL,
    IDADE NUMBER
);


CREATE TABLE ATENDE (
    CNPJ VARCHAR2(14),
    CPF VARCHAR2(11),
    FOREIGN KEY (CNPJ) REFERENCES HOSPITAL (CNPJ),
    FOREIGN KEY (CPF) REFERENCES PACIENTE (CPF),
    PRIMARY KEY (CNPJ, CPF)
);


CREATE TABLE SALA (
    ID_SALA VARCHAR2(50) PRIMARY KEY,
    NUMERO VARCHAR2(50) NOT NULL,
    CNPJ VARCHAR2(14) NOT NULL,
    FOREIGN KEY (CNPJ) REFERENCES HOSPITAL (CNPJ)
);


CREATE TABLE CONSULTA (
    ID_SALA VARCHAR2(50),
    CPF_M VARCHAR2(11),
    CPF_P VARCHAR2(11),
    HORACONSULTA TIMESTAMP,
    VALOR NUMBER(10, 2) NOT NULL,
    DESCRICAO VARCHAR2(255),
    PRIMARY KEY (ID_SALA, CPF_M, CPF_P, HORACONSULTA),
    FOREIGN KEY (ID_SALA) REFERENCES SALA (ID_SALA),
    FOREIGN KEY (CPF_M) REFERENCES MEDICO (CPF_MEDICO),
    FOREIGN KEY (CPF_P) REFERENCES PACIENTE (CPF)
);
```

```
CREATE TABLE MEDICAMENTO (
    ID_MEDICAMENTO VARCHAR2(50) PRIMARY KEY,
    NOME VARCHAR2(255) NOT NULL
);


CREATE TABLE PRESCRICAO (
    ID_PRESCRICAO VARCHAR2(50) PRIMARY KEY,
    ID_MEDICAMENTO VARCHAR2(50) NOT NULL,
    DOSAGEM VARCHAR2(100) NOT NULL,
    DESCRICAO VARCHAR2(255),
    ID_SALA VARCHAR2(50) NOT NULL,
    CPF_M VARCHAR2(11) NOT NULL,
    CPF_P VARCHAR2(11) NOT NULL,
    HORACONSULTA TIMESTAMP NOT NULL,
    FOREIGN KEY (ID_MEDICAMENTO) REFERENCES
MEDICAMENTO (ID_MEDICAMENTO),
    FOREIGN KEY (ID_SALA, CPF_M, CPF_P, HORACONSULTA)
REFERENCES CONSULTA (ID_SALA, CPF_M, CPF_P,
HORACONSULTA)
);
```

Consultas

- Group by/Having

Projeta as salas que tiveram pelo menos uma prescrição e a quantidade

```
SELECT p.ID_SALA, COUNT(p.ID_PRESCRICAO)
AS total_prescricoes
FROM PRESCRICAO p
GROUP BY p.ID_SALA
HAVING COUNT(p.ID_PRESCRICAO) > 0;
```

- Junção interna

Projeta o nome dos funcionários que são médicos

```
SELECT DISTINCT F.NOME
   FROM FUNCIONARIO F
   INNER JOIN MEDICO M
   ON F.CPF = M.CPF_MEDICO
```

Consultas

- Junção externa

– Projeta os hospitais e os números de suas salas

```
SELECT H.NOME, S.NUMERO
   FROM HOSPITAL H
   LEFT OUTER JOIN SALA S
   ON H.CNPJ = S.CNPJ
```

- Semi Junção

– Projeta as salas onde houve prescrição de medicamentos

```
SELECT S.ID_SALA
   FROM SALA S
   WHERE EXISTS
   (
      SELECT P.ID_SALA
      FROM PRESCRICAO P
      WHERE S.ID_SALA = P.ID_SALA
   );
```

Consultas

- Anti-junção

– Projeta as salas onde não houve a prescrição de medicamentos

```
SELECT S.ID_SALA
   FROM SALA S
   WHERE NOT EXISTS
   (
        SELECT P.ID_SALA
        FROM PRESCRICAO P
        WHERE S.ID_SALA = P.ID_SALA
   );
```

- Subconsulta do tipo escalar

– Projeta os nomes dos funcionários com o salário maior do que a média de todos os funcionários

```
SELECT F.NOME
FROM FUNCIONARIO F
WHERE SALARIO > (
   SELECT AVG(SALARIO)
   FROM FUNCIONARIO
);
```

Consultas

- Subconsulta do tipo linha

– Projeta o CPF dos pacientes que tiveram prescrição na mesma sala e mesmo médico da prescrição 'PRE01'

```
SELECT DISTINCT C.CPF_P
        FROM CONSULTA C
        WHERE (C.ID_SALA, C.CPF_M) = (
        SELECT P3.ID_SALA, P3.CPF_M
                FROM PRESCRICAO P3
                WHERE P3.ID_PRESCRICAO = 'PRE01'
   )
```

- Subconsulta do tipo tabela

– Projeta nome que fizeram consultas acima de 400 reais.

```
SELECT F.NOME
   FROM FUNCIONARIO F
   WHERE F.CPF IN (
        SELECT C.CPF_M
                FROM CONSULTA C
                WHERE C.VALOR > 400
   );
```

Consultas

- Operação de conjunto

– Seleciona o nome de todos os funcionários que não são médicos

```
SELECT F.NOME
   FROM MEDICO M
   RIGHT JOIN FUNCIONARIO F
   ON F.CPF = M.CPF_MEDICO

MINUS

SELECT F.NOME
   FROM FUNCIONARIO F
   RIGHT JOIN MEDICO M
   ON F.CPF = M.CPF_MEDICO
```

PL/SQL

```
-- Quantidade de funcionários que trabalham em mais de um hospital
CREATE OR REPLACE FUNCTION
CONTAR_FUNCIONARIOS_MULTIPLOS_HOSPITAIS
RETURN NUMBER
IS
   V_QTD_FUNCIONARIOS NUMBER;
BEGIN
   SELECT COUNT(*) INTO V_QTD_FUNCIONARIOS
   FROM (
      SELECT F.CPF
      FROM FUNCIONARIO F
      JOIN TRABALHA T ON F.CPF = T.CPF
      GROUP BY F.CPF
      HAVING COUNT(DISTINCT T.CNPJ) > 1
   );

   RETURN V_QTD_FUNCIONARIOS;
END CONTAR_FUNCIONARIOS_MULTIPLOS_HOSPITAIS;
--
SELECT CONTAR_FUNCIONARIOS_MULTIPLOS_HOSPITAIS() AS
QTD_FUNCIONARIOS_MULTIPLOS_HOSPITAIS FROM DUAL;
```

```
-- Quantidade de medicamentos que foram prescritos em uma determinada
sala
CREATE OR REPLACE FUNCTION contar_medicamentos_sala(p_id_sala IN
VARCHAR2)
RETURN NUMBER
IS
   v_total_medicamentos NUMBER;
BEGIN
   SELECT COUNT(DISTINCT pr.ID_MEDICAMENTO)
   INTO v_total_medicamentos
   FROM PRESCRICAO pr
   WHERE pr.ID_SALA = p_id_sala;

   RETURN v_total_medicamentos;
EXCEPTION
   WHEN NO_DATA_FOUND THEN
      RETURN 0;
END;
/
--
DECLARE
   v_total_medicamentos NUMBER;
BEGIN
   v_total_medicamentos := contar_medicamentos_sala('SALA01');

   DBMS_OUTPUT.PUT_LINE('Total de medicamentos na sala: ' ||
v_total_medicamentos);
END;
/
```

PL/SQL

```
-- Calcular o salário médio dos funcionários de um departamento
CREATE OR REPLACE FUNCTION salario_medio_departamento
(p_nome_departamento VARCHAR2) RETURN NUMBER IS
   v_salario_medio NUMBER;
BEGIN
   SELECT AVG(SALARIO) INTO v_salario_medio
   FROM FUNCIONARIO
   WHERE CPF IN (
      SELECT DISTINCT CPF
      FROM TRABALHA
      WHERE CNPJ IN (
         SELECT CNPJ
         FROM DEPARTAMENTO
         WHERE NOME = p_nome_departamento
      )
   );

   RETURN v_salario_medio;
END;
/
--
DECLARE
   salario_medio NUMBER;
BEGIN
   salario_medio := salario_medio_departamento('Cardiologia');
   DBMS_OUTPUT.PUT_LINE('O salário médio do departamento de Cardiologia
é: ' || salario_medio);
END;
/
```

```
-- Consultar quantas consultas um determinado paciente teve
CREATE OR REPLACE PROCEDURE consulta_paciente
(p_cpf_paciente VARCHAR2) AS
   v_nome_paciente VARCHAR2(100);
   v_numero_consultas NUMBER;
BEGIN

   SELECT NOME INTO v_nome_paciente FROM PACIENTE
WHERE CPF = p_cpf_paciente;

   SELECT COUNT(*) INTO v_numero_consultas FROM
CONSULTA WHERE CPF_P = p_cpf_paciente;

   DBMS_OUTPUT.PUT_LINE('O paciente ' || v_nome_paciente || '
teve ' || v_numero_consultas || ' consultas.');
END;
/
--
BEGIN
   consulta_paciente('11111111111');
END;
/
```

PL/SQL

```
--  Dado um MEDICO (CPF), quais medicamentos ele prescreveu
CREATE OR REPLACE PROCEDURE medicamentos_por_medico (
   cpf_medico IN VARCHAR2,
   cursor_out OUT SYS_REFCURSOR
) AS
BEGIN
   OPEN cursor_out FOR
   SELECT DISTINCT ME.ID_MEDICAMENTO, ME.NOME
   FROM MEDICAMENTO ME
   JOIN PRESCRICAO P ON ME.ID_MEDICAMENTO = P.ID_MEDICAMENTO
   WHERE P.CPF_M = cpf_medico;
END medicamentos_por_medico;
/
--
DECLARE
   medicamento_cursor SYS_REFCURSOR;
   v_cpf_medico VARCHAR2(11) := '55578667778';
   v_id_medicamento VARCHAR2(50);
   v_nome_medicamento VARCHAR2(255);
BEGIN
   medicamentos_por_medico(v_cpf_medico, medicamento_cursor);

   LOOP
      FETCH medicamento_cursor INTO v_id_medicamento, v_nome_medicamento;
      EXIT WHEN medicamento_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('ID do Medicamento: ' || v_id_medicamento || ', Nome do
Medicamento: ' || v_nome_medicamento);
   END LOOP;

   CLOSE medicamento_cursor;
END;
/
```

```
-- Dado um MEDICAMENTO (NOME), quais Médicos e Pacientes ele foi prescrito
CREATE OR REPLACE PROCEDURE medicos_por_medicamento (
   nome_medicamento IN VARCHAR2,
   cursor_out OUT SYS_REFCURSOR
) AS
BEGIN
   OPEN cursor_out FOR
   SELECT DISTINCT P.ID_PRESCRICAO, M.CPF_MEDICO, F.NOME
   FROM MEDICAMENTO ME
   JOIN PRESCRICAO P ON ME.ID_MEDICAMENTO = P.ID_MEDICAMENTO
   JOIN MEDICO M ON P.CPF_M = M.CPF_MEDICO
   JOIN FUNCIONARIO F ON M.CPF_MEDICO = F.CPF
   WHERE ME.NOME = nome_medicamento;
END medicos_por_medicamento;
/
--
DECLARE
   medicamento_cursor SYS_REFCURSOR;
   v_nome_medicamento VARCHAR2(255) := 'Omeprazol';
   v_id_prescricao VARCHAR2(50);
   v_cpf_medico VARCHAR2(11);
   v_nome_medico VARCHAR2(100);
BEGIN
   medicos_por_medicamento(v_nome_medicamento, medicamento_cursor);

   LOOP
      FETCH medicamento_cursor INTO v_id_prescricao, v_cpf_medico,
v_nome_medico;
      EXIT WHEN medicamento_cursor%NOTFOUND;
      DBMS_OUTPUT.PUT_LINE('ID da Prescrição: ' || v_id_prescricao || ', CPF do
Médico: ' || v_cpf_medico || ', Nome do Médico: ' || v_nome_medico);
   END LOOP;

   CLOSE medicamento_cursor;
END;
/
```

Obrigado!