

1. Criação do banco de dados

```
CREATE DATABASE hospital
WITH
  OWNER = postgres
  ENCODING = 'UTF8'
  LC_COLLATE = 'Portuguese_Brazil.1252'
  LC_CTYPE = 'Portuguese_Brazil.1252'
  LOCALE_PROVIDER = 'libc'
  TABLESPACE = pg_default
  CONNECTION LIMIT = -1
  IS_TEMPLATE = False;
```

2. Criações das tabelas

2.1. criação da tabela paciente

```
CREATE TABLE paciente(
  codigo SERIAL PRIMARY KEY,
  nome VARCHAR(100) not null,
  idade INTEGER not null,
  email VARCHAR(100),
  telefone VARCHAR(20)
)
```

2.2. Criação da tabela plano_saude

```
CREATE TABLE plano_saude (
  codigo SERIAL PRIMARY KEY,
  categoria VARCHAR(50) NOT NULL,
  carencia_dias INTEGER,
  data_inicio DATE NOT NULL,
  data_desativacao DATE,
  paciente_codigo INTEGER NOT NULL,
  FOREIGN KEY (paciente_codigo) REFERENCES paciente(codigo)
);
```

2.3. Criação da tabela dependente

```
CREATE TABLE dependente (  
    numSeq INTEGER NOT NULL,  
    paciente_codigo INTEGER NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    idade INTEGER NOT NULL,  
    email VARCHAR(100),  
    telefone VARCHAR(20),  
  
    PRIMARY KEY (numSeq, paciente_codigo),  
  
    FOREIGN KEY (paciente_codigo)  
        REFERENCES paciente(codigo)  
        ON DELETE CASCADE  
);
```

2.4. Criação da tabela medico

```
CREATE TABLE medico(  
    codigo SERIAL PRIMARY KEY,  
    crm VARCHAR(9) NOT NULL,  
    nome VARCHAR(100) NOT NULL,  
    especialidade VARCHAR(60),  
    telefone VARCHAR(20),  
    num_cons INTEGER NOT NULL,  
    corredor_cons INTEGER NOT NULL,  
    cod_computador_cons INTEGER NOT NULL  
);
```

2.4.1 alteração consultorio unico

```
ALTER TABLE medico  
ADD CONSTRAINT unico_consultorio  
UNIQUE (num_cons, cod_computador_cons);
```

2.5. Criação da tabela consulta

```
CREATE TABLE consulta(  
    dataCons DATE not null,  
    horario TIME not null,  
    medico_codigo INTEGER NOT NULL,  
    paciente_codigo INTEGER NOT NULL,  
    PRIMARY KEY(dataCons, horario, medico_codigo, paciente_codigo),  
    senha VARCHAR(4),  
    Emergencia BOOLEAN,  
    FOREIGN KEY (medico_codigo) REFERENCES medico(codigo),
```

```
FOREIGN KEY (paciente_codigo) REFERENCES paciente(codigo)
);
```

2.6. Criação da tabela medicamento

```
CREATE TABLE medicamento(
  codigo SERIAL PRIMARY KEY,
  laboratorio VARCHAR(20),
  nome VARCHAR(100) NOT NULL,
  rec_controle_especial BOOLEAN NOT NULL DEFAULT FALSE,
  observacoes TEXT
);
```

2.7. Criação da tabela prescricao

```
CREATE TABLE prescricao(
  medico_codigo INTEGER not null,
  paciente_codigo INTEGER not null,
  horario TIME not null,
  dataCons DATE not null,
  medicamento_codigo INTEGER not null,
  senha VARCHAR(4),
  emergencia BOOLEAN,
  FOREIGN KEY (dataCons, horario, medico_codigo, paciente_codigo)
    REFERENCES consulta (dataCons, horario, medico_codigo, paciente_codigo),
  FOREIGN KEY(medicamento_codigo)
    REFERENCES medicamento(codigo),
  PRIMARY KEY(medico_codigo,paciente_codigo,horario,dataCons,
medicamento_codigo)
)
```

3. Inserção

3.1. Inserção de Dados: Paciente

```
INSERT INTO paciente(nome, idade, telefone) /*Como email é opcional não é
necessário para inserir*/
VALUES ('João', 28, '8599987778');
```

3.2. Inserção de Dados: Plano de Saúde

```
INSERT INTO plano_saude(categoria, data_inicio, paciente_codigo, data_desativacao,
carenacia_dias)
```

```
VALUES ('Basico', '2025-01-01', 1, '2025-12-31', 0);
```

3.3. Inserção de Dados: Dependente

3.3.1. Informando numSeq

```
INSERT INTO dependente(numSeq, paciente_codigo ,nome, idade, telefone, email)
VALUES (10, 1, 'Maria', 5, '8599987779', 'mariazinha@protonmail.com');/*Valor alto
para numSeq para garantir unicidade*/
```

3.3.2. Sem informar, usando o trigger

```
INSERT INTO dependente(paciente_codigo ,nome, idade, telefone, email)
VALUES ( 1, 'João Júnior', 3, '8588887777', 'joaojr@protonmail.com');
```

3.4 Inserção de Dados: Medico

```
INSERT INTO medico(crm, nome, especialidade, telefone, num_cons, corredor_cons,
cod_computador_cons)
VALUES ('123456/CE','Francisco de Souza', 'Otorrinolaringologista', '85987654321',
1, 1, 1);
```

3.5 Inserção de Dados: Consulta

```
INSERT INTO consulta(dataCons, horario, medico_codigo, paciente_codigo, senha,
emergencia)
VALUES ('2025-03-20','18:29', 1, 1, '99', FALSE);
```

3.6 Inserção de Dados: medicamento

```
INSERT INTO medicamento(laboratorio ,nome, rec_controle_especial, observacoes)
VALUES ('europharma', 'eurotenol', TRUE, '2 comprimidos ao dia, 1 a cada 12h');
```

3.7 Inserção de Dados: Prescricao

```
INSERT INTO prescricao(dataCons, horario, medico_codigo, paciente_codigo,
medicamento_codigo)
VALUES ('2025-03-20','18:29', 1, 1, '1');
```

4. Leitura

4.1. leitura de Dados: Pacientes em ordem alfabética do nome

```
SELECT nome, codigo FROM paciente  
ORDER BY nome ASC
```

4.2. leitura de Dados: Plano de Saúde

```
SELECT categoria, codigo, data_inicio FROM plano_saude  
ORDER BY codigo ASC
```

4.3. leitura de Dados: Dependente

```
SELECT numSeq, nome FROM dependente  
WHERE paciente_codigo = 1  
ORDER BY numSeq ASC
```

4.4 leitura de Dados: Medico

```
SELECT codigo, nome, crm FROM medico  
ORDER BY codigo ASC
```

4.5 leitura de Dados: Consulta

```
SELECT dataCons, horario ,paciente_codigo,medico_codigo FROM consulta  
ORDER BY dataCons, horario ASC
```

4.6. leitura de Dados: medicamento

```
SELECT laboratorio,nome, codigo FROM medicamento  
ORDER BY laboratorio ASC
```

4.7. leitura de Dados: prescricao

```
SELECT paciente_codigo, medico_codigo FROM prescricao
WHERE medicamento_codigo = 1
ORDER BY dataCons, horario, paciente_codigo, medico_codigo ASC
```

5 Atualização:

```
UPDATE paciente
SET email = 'joaoprimeiropaciente@gmail.com',
WHERE codigo = 1;
```

6 DELETE:

```
DELETE FROM paciente
WHERE codigo = 2;
```

7. Pesquisa por substring case-insensitive:

```
SELECT * FROM medico
WHERE nome ILIKE '%a%';
```

8. 2 Consultas com tipos diferentes de JOIN

União externa à esquerda com os nomes dos medicos e id dos pacientes atendidos ou null

```
SELECT m.nome, c.paciente_codigo
FROM medico m
LEFT JOIN consulta c ON m.codigo=c.medico_codigo
```

União externa total com os nome dos pacientes e medicamentos prescritos a eles

```
SELECT p.nome, pr.medicamento_codigo
FROM paciente p
FULL JOIN prescricao pr ON p.codigo = pr.paciente_codigo
```

9. Consulta com ALL: todos os pacientes mais velhos que seus dependentes

```
SELECT p.nome
FROM paciente p
WHERE idade > ALL (
    SELECT d.idade
    FROM dependente d
    WHERE d.paciente_codigo = p.codigo
);
```

10. Consulta com ANY: Médico que atende no consultorio mais próximo (menor número):

```
SELECT nome
FROM medico m1
WHERE num_cons < ANY (
    SELECT num_cons
    FROM medico m2
    WHERE m2.codigo != m1.codigo
);
```

11. Consulta com funções agregadas, GROUP BY e HAVING:

```
/*Consulta todos os médicos com apenas 1 prescrição no total*/
SELECT m.nome, COUNT(*) AS total_prescicoes
FROM prescricao p
JOIN medico m ON p.medico_codigo = m.codigo
GROUP BY m.codigo
HAVING COUNT(*) = 1;
```

12. Consulta com order by descendente:

```
SELECT codigo, nome, num_cons FROM medico
ORDER BY num_cons DESC;
```

13. Triggers

Trigger para inserir e autoincrementar dependente.numSeq a partir do ultimo valor para cada paciente

Função do trigger

```
CREATE OR REPLACE FUNCTION gerar_numseq_dependente()
RETURNS TRIGGER AS $$
DECLARE
    ultimo_seq INTEGER;
BEGIN
    SELECT COALESCE(MAX(numSeq), 0) INTO ultimo_seq
    FROM dependente
    WHERE paciente_codigo = NEW.paciente_codigo;

    NEW.numSeq := ultimo_seq + 1;
    RETURN NEW;
END;
$$ LANGUAGE plpgsql;
```

Trigger

```
CREATE TRIGGER trigger_gerar_numseq
BEFORE INSERT ON dependente
FOR EACH ROW
WHEN (NEW.numSeq IS NULL)
EXECUTE FUNCTION gerar_numseq_dependente();
```

14. Carga Inicial de dados

pacientes

	codigo [PK] integer	nome character varying (100)	idade integer	email character varying (100)	telefone character varying (20)
1	1	João	28	joaoprimeiropaciente@gmail.com	8599987778
2	5	Paulo Berna	51	[null]	88997775467
3	4	Ricardo Oliveira	47	ricardopastor@yahoo.com	[null]
4	3	Valdísio	63	valadin@outlook.com	85996857473

dependentes

	numseq [PK] integer	paciente_codigo [PK] integer	nome character varying (100)	idade integer	email character varying (100)	telefone character varying (20)
1	1	1	Maria	5	mariazinha@protonmail.com	8599987779
2	2	1	João Júnior	3	joaojr@protonmail.com	858887777
3	2	3	Valdísio junior	29	[null]	[null]

planos

	codigo [PK] integer	categoria character varying (50)	carencia_dias integer	data_inicio date	data_desativacao date	paciente_codigo integer
1	1	Basico	0	2025-01-01	2025-12-31	1
2	2	Premium	0	2025-01-29	[null]	3
3	3	odontologico	0	2025-01-29	[null]	3
4	4	Plus	60	2024-03-10	[null]	4
5	5	Básico	90	2023-01-01	[null]	5

medicos

	codigo [PK] integer	crm character varying (9)	nome character varying (100)	especialidade character varying (60)	telefone character varying (20)	num_cons integer	corredor_cons integer	cod_computador_cons integer
1	1	123456/CE	Francisco de Souza	Otorrinolaringologista	85987654321	1	1	1
2	2	889956/SP	Dr. José Maria	Cardiologia	11999999999	101	10	101
3	3	987654/SP	Dra. Ana Cordoso	Dermatologia	11888888888	102	10	102
4	4	764589/PI	Dr. Hércules Flu	Ortopedia/Traumatologia	89888888888	25	2	25
5	6	879567/CE	Dra. Rita Santana	Oftalmologista	85999999999	105	10	105

medicamentos

	codigo [PK] integer	laboratorio character varying (20)	nome character varying (100)	rec_controle_especial boolean	observacoes text
1	1	europarma	eurotenol	true	2 comprimidos ao dia, 1 a cada 12h
2	2	globo	cetaconazol 10mg	false	[null]
3	3	ache	Deller 50mg	true	[null]
4	4	neoQuimica	neosoro 0,5mg/ml	false	[null]

consultas

	datacons [PK] date	horario [PK] time without time zone	medico_codigo [PK] integer	paciente_codigo [PK] integer	senha character varying (4)	emergencia boolean
1	2025-03-20	18:29:00	1	1	99	false
2	2025-04-01	18:01:00	2	3	01	true
3	2025-05-11	12:07:00	3	4	99	[null]
4	2025-01-26	07:45:00	2	5	04	true

prescricoes

	medico_codigo [PK] integer	paciente_codigo [PK] integer	horario [PK] time without time zone	datacons [PK] date	medicamento_codigo [PK] integer	senha character varying (4)	emergencia boolean
1	1	1	18:29:00	2025-03-20	1	[null]	[null]
2	2	3	18:01:00	2025-04-01	3	[null]	[null]
3	3	4	12:07:00	2025-05-11	2	[null]	[null]
4	2	5	07:45:00	2025-01-26	1	[null]	[null]