

PROJETO BANCO DE DADOS

Barbara Vaz
Dante Santana
Ismael Benjamim
Liliane Carla
Marco Andrade
Tiago Bello

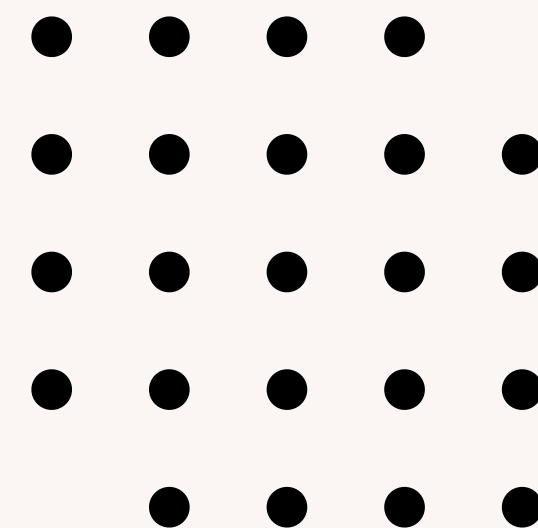
MINI MUNDO

RESUMO

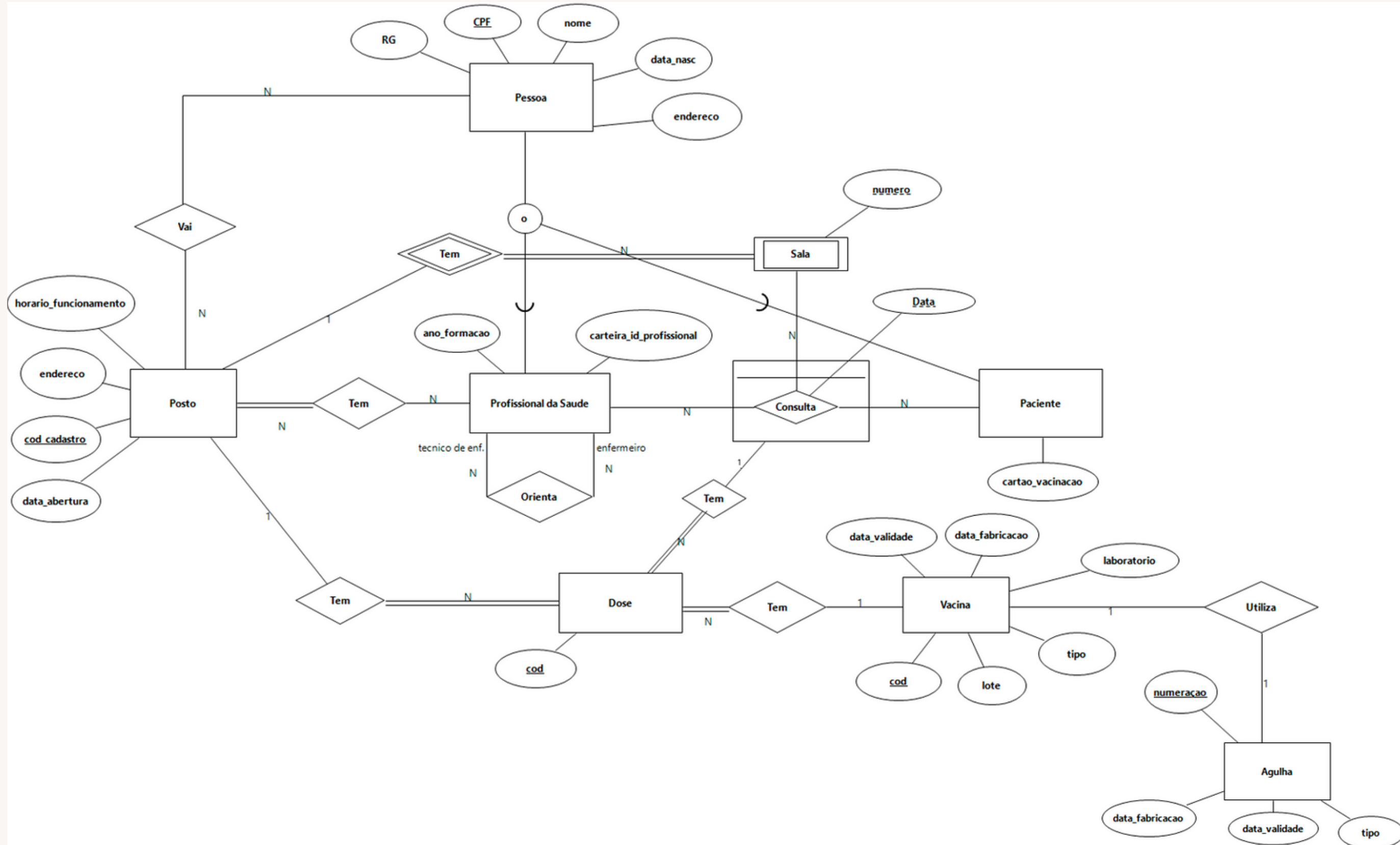
O tema central do nosso Banco de Dados é o Programa Nacional de Vacinação do SUS.

Nessa linha, para manter um controle de dados, é necessário que haja o cadastro do paciente a ser vacinado, do profissional de saúde que irá realizar a consulta, da vacina e da dose a ser aplicada, da agulha a ser utilizada e do posto de saúde em que foi realizada a vacinação.

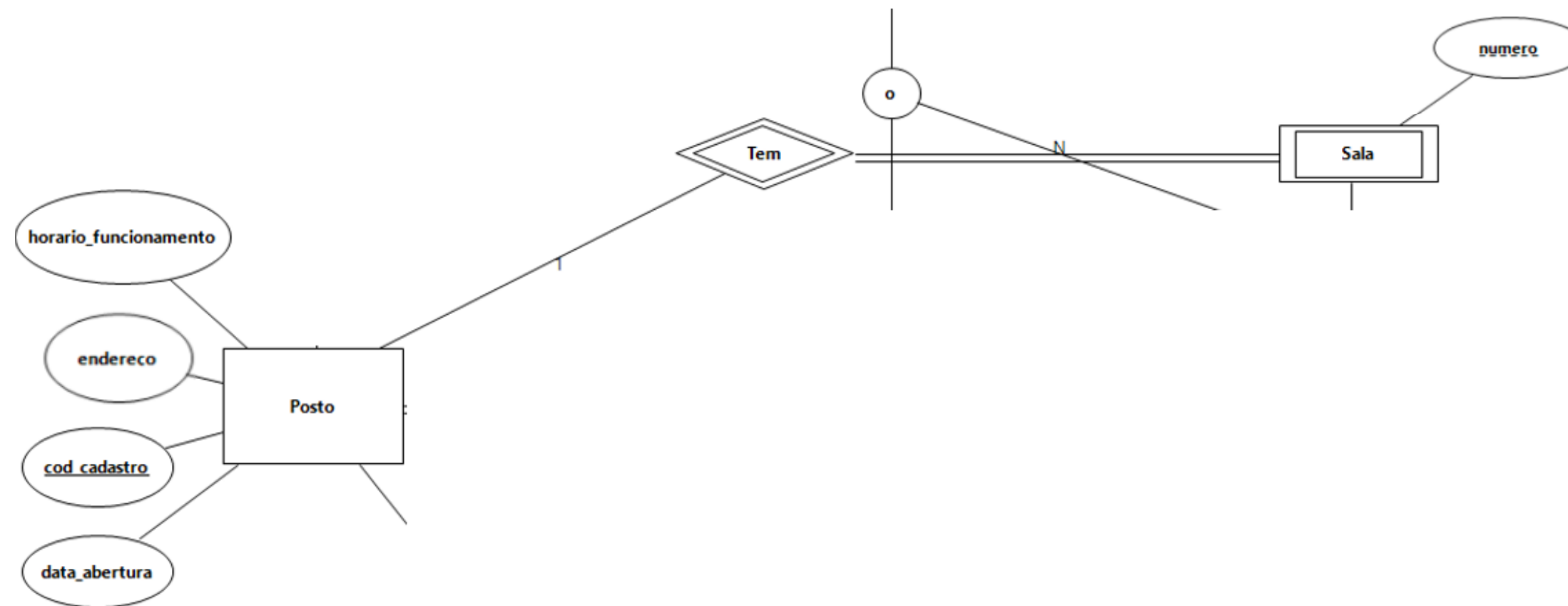
O posto de saúde e sala, em questão, tem profissionais de saúde, diferentes tipos de vacinas e os materiais necessários para a aplicação delas. Com isso, o paciente vai até o local esperando que esse posto, de fato, apresente todas as vacinas e materiais, porém, ele também sabe que nem sempre é o que acontece.



MODELO CONCEITUAL



MODELO CONCEITUAL + RELACIONAL

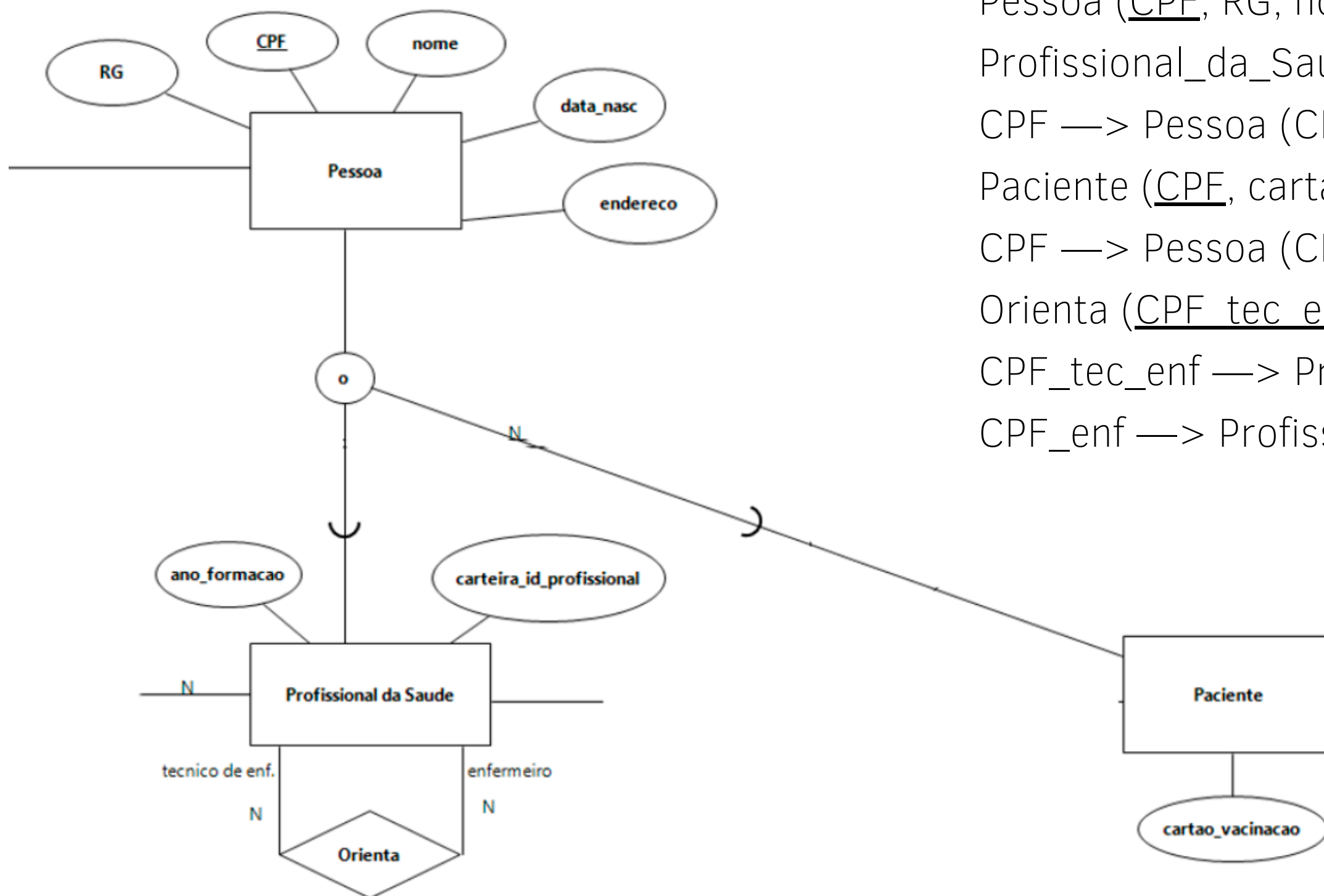


Posto (cod cadastro, horario funcionamento, endereco, data abertura)

Sala (cod cadastro, numero)

cod_cadastro —> Posto(cod_cadastro)

MODELO CONCEITUAL + RELACIONAL



Pessoa (CPF, RG, nome, data_nasc, endereco)

Profissional_da_Saude (CPF, carteira_id_profissional, ano_formacao)

CPF —> Pessoa (CPF)

Paciente (CPF, cartao_vacinacao)

CPF —> Pessoa (CPF)

Orienta (CPF_tec_enf, CPF_enf)

CPF_tec_enf —> Profissional_da_Saude (CPF)

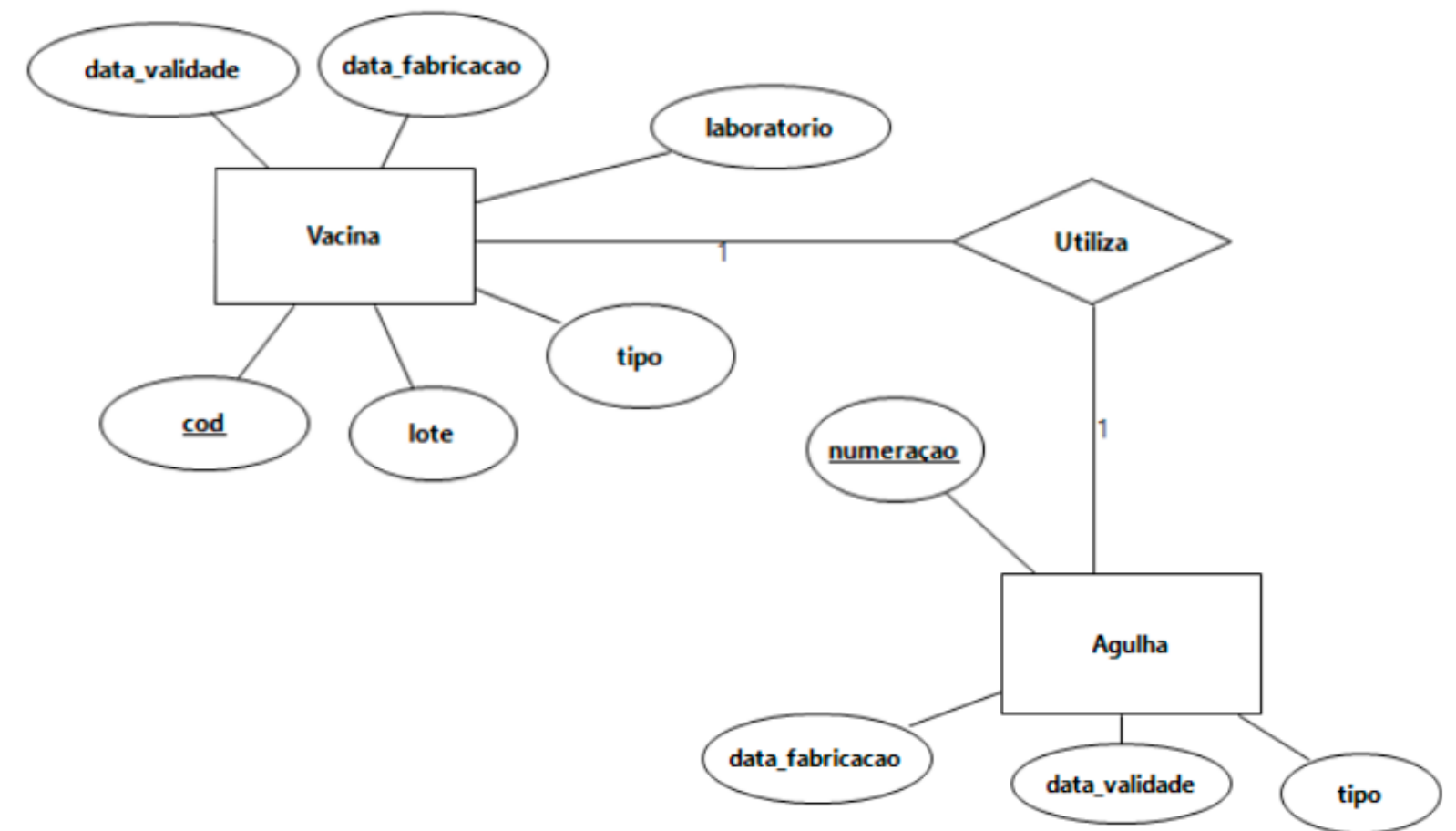
CPF_enf —> Profissional_da_Saude (CPF)

MODELO CONCEITUAL + RELACIONAL

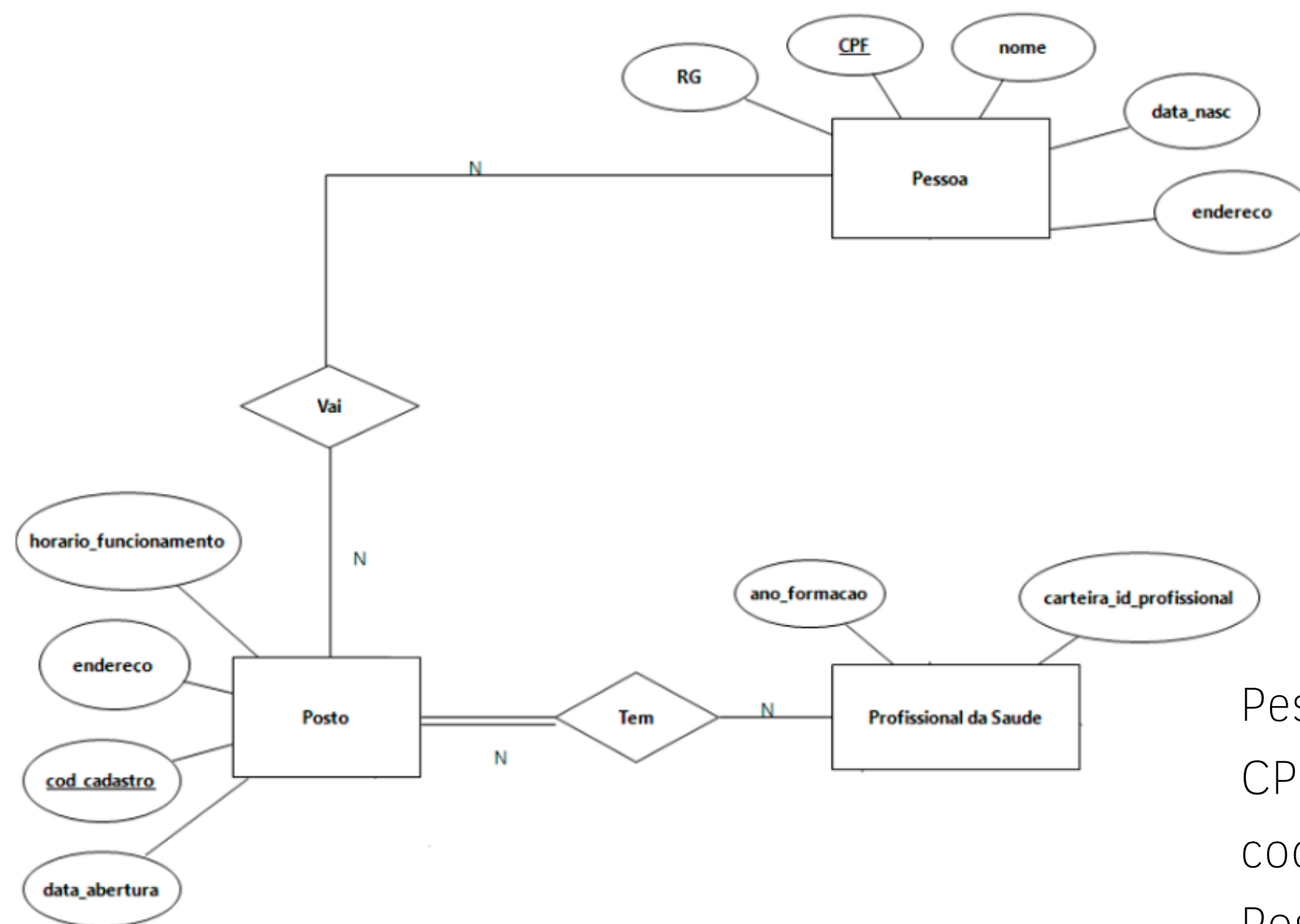
Vacina (cod, lote, tipo, laboratorio, data_fabricacao, data_validade)

Agulha (numeraçao, tipo, data_fabricacao, data_validade, [cod])

cod —> Vacina (cod)



MODELO CONCEITUAL + RELACIONAL



Pessoa_vai_Posto (CPF, cod_cadastro)

CPF —> Pessoa (CPF)

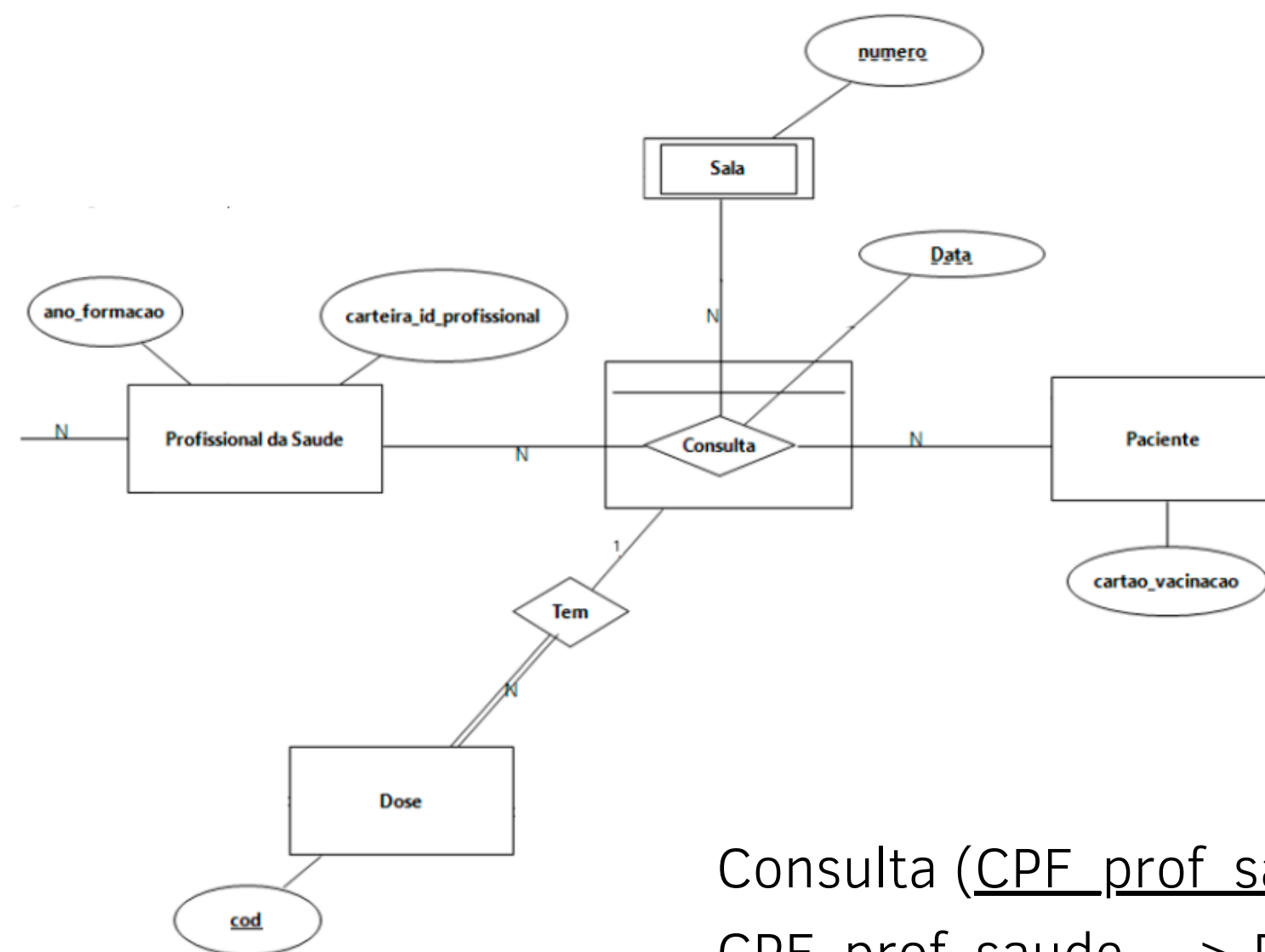
cod_cadastro —> Posto (cod_cadastro)

Posto_tem_PS (cod_cadastro, CPF_prof_saude)

cod_cadastro —> Posto (cod_cadastro)

CPF_prof_saude —> Profissional_da_Saude (CPF)

MODELO CONCEITUAL + RELACIONAL



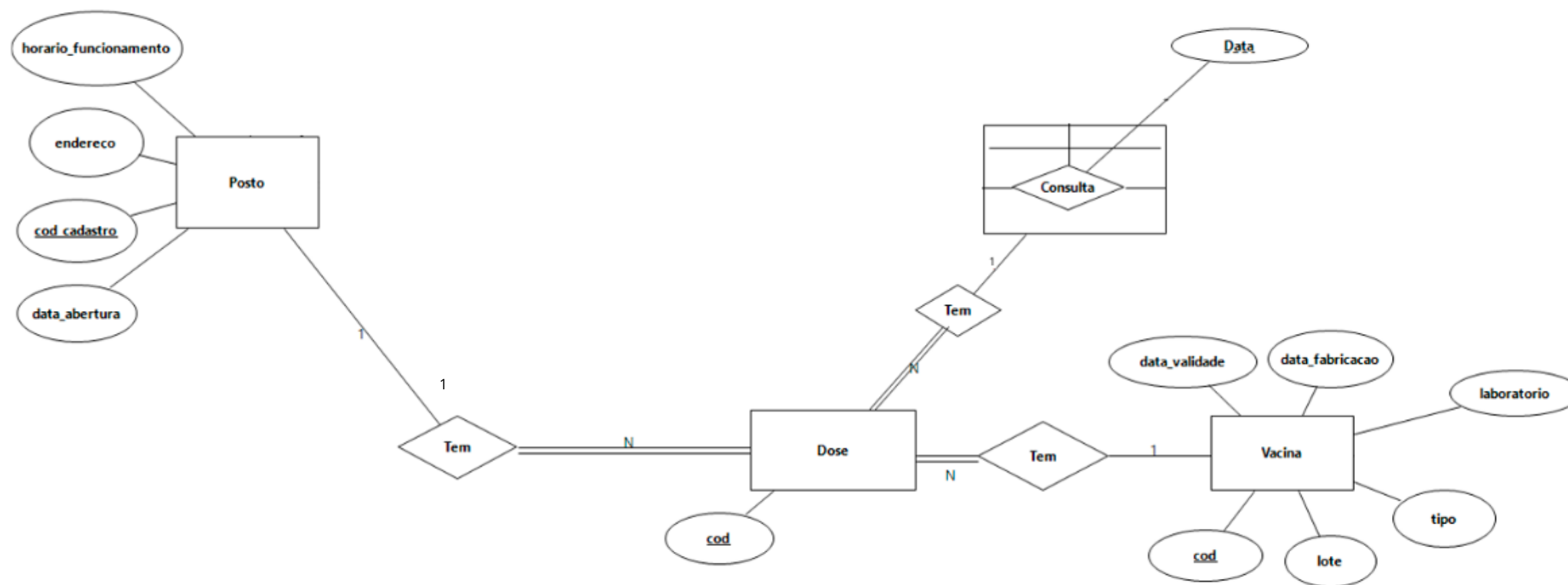
Consulta (CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta)

CPF_prof_saude —> Profissional_da_Saude (CPF)

CPF_paciente —> Paciente (CPF)

cod_cadastro —> Sala (cod_cadastro)

MODELO CONCEITUAL + RELACIONAL



Dose (cod, cod_vacina!, cod_cadastro!, CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta)

cod_vacina —> Vacina (cod)

cod_cadastro —> Posto(codigo_cadastro)

(CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta) → Consulta (CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta)

MODELO RELACIONAL

Posto (cod_cadastro, horario_funcionamento, endereco, data_abertura)

Sala (cod_cadastro, numero)

cod_cadastro —> Posto(cod_cadastro)

Pessoa (CPF, RG, nome, data_nasc, endereco)

Profissional_da_Saude (CPF, carteira_id_profissional, ano_formacao)

CPF —> Pessoa (CPF)

Paciente (CPF, cartao_vacinacao)

CPF —> Pessoa (CPF)

Orienta (CPF_tec_enf, CPF_enf)

CPF_tec_enf —> Profissional_da_Saude (CPF)

CPF_enf —> Profissional_da_Saude (CPF)

Vacina (cod, lote, tipo, laboratorio, data_fabricacao, data_validade)

Agulha (numeracao, tipo, data_fabricacao, data_validade, [cod])

cod —> Vacina (cod)

MODELO RELACIONAL

Pessoa_vai_Posto (CPF, cod_cadastro)

CPF —> Pessoa (CPF)

cod_cadastro —> Posto (cod_cadastro)

Posto_tem_PS (cod_cadastro, CPF_prof_saude)

cod_cadastro —> Posto (cod_cadastro)

CPF_prof_saude —> Profissional_da_Saude (CPF)

Consulta(CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta, cod)

CPF_prof_saude —> Profissional_da_Saude (CPF)

CPF_paciente —> Paciente (CPF)

cod_cadastro —> Sala (cod_cadastro)

cod —> Dose (cod)

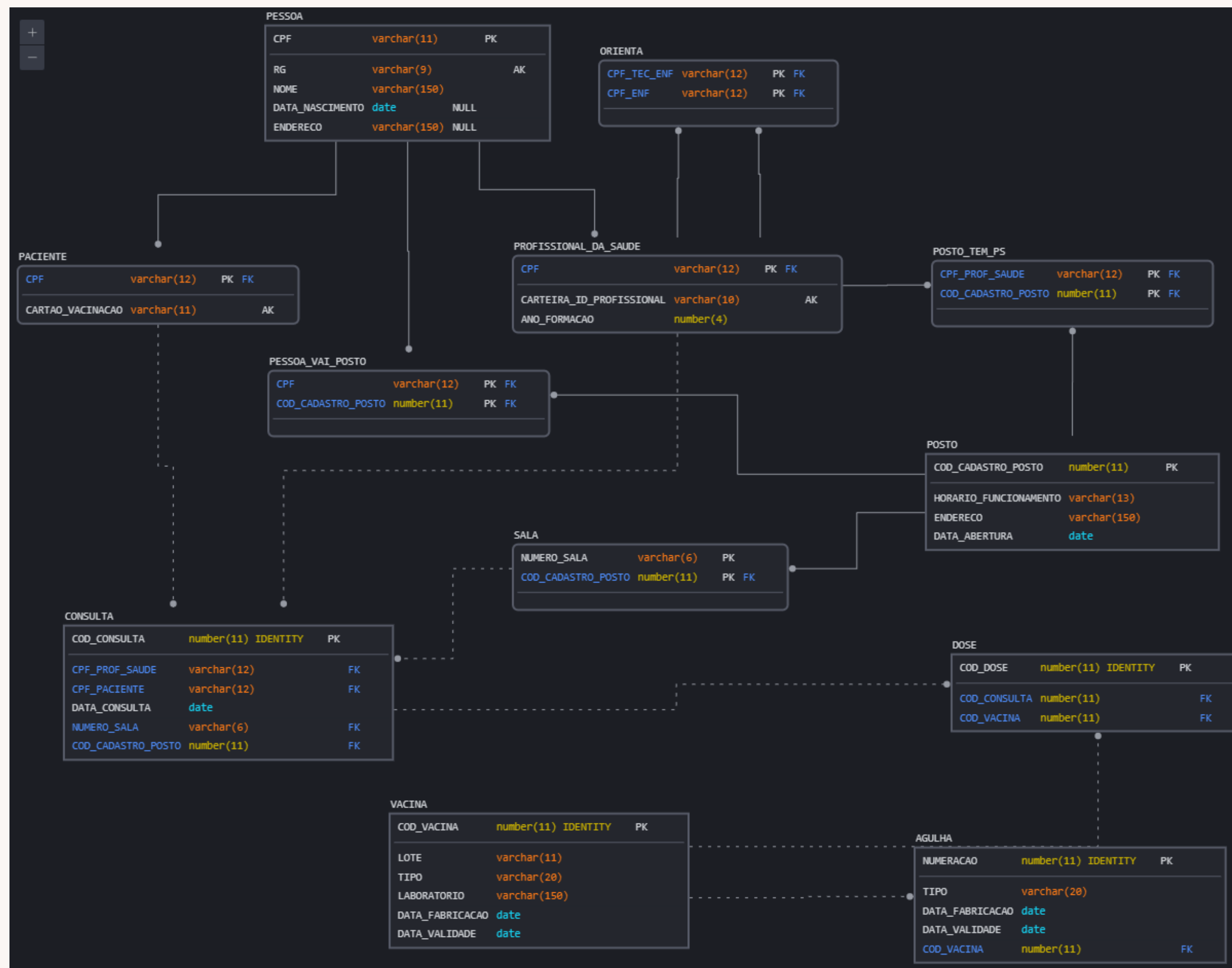
Dose (cod, cod_vacina!, cod_cadastro!, CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta)

cod_vacina —> Vacina (cod)

cod_cadastro —> Posto(codigo_cadastro)

(CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta) → Consulta (CPF_prof_saude, CPF_paciente, cod_cadastro, numero, data_consulta)

SQL



SQL

TABELA POSTO

COD_CADASTRO_POSTO	HORARIO_FUNCIONAMENTO	ENDEREÇO	DATA_ABERTURA
1	08:00-18:00	Rua Barão de Souza Leão, 123, Boa Viagem, Recife - PE, CEP 51030-300	01-JAN-20
2	08:30-18:00	Rua Barão de Souza Leão, 13, Boa Viagem, Recife - PE, CEP 51030-300	02-FEB-20
3	07:30-17:30	Avenida Engenheiro Domingos Ferreira, 789, Pina, Recife - PE, CEP 51011-051	02-FEB-20
4	09:00-19:00	Rua da Hora, 123, Espinheiro, Recife - PE, CEP 52020-010	03-MAR-20
5	08:30-18:00	Rua Ribeiro de Brito, 123, Boa Viagem, Recife - PE, CEP 51021-310	04-APR-20

```
-- CRIA TABELA POSTO
```

```
CREATE TABLE POSTO
```

```
(
```

```
  COD_CADASTRO_POSTO NUMBER(11) NOT NULL,
```

```
  HORARIO_FUNCIONAMENTO VARCHAR(13) NOT NULL,
```

```
  ENDEREÇO VARCHAR(150) NOT NULL,
```

```
  DATA_ABERTURA DATE NOT NULL,
```

```
  CONSTRAINT PK_POSTO PRIMARY KEY (COD_CADASTRO_POSTO)
```

```
);
```

```
-- POVOA TABELA POSTO
```

```
INSERT INTO POSTO VALUES ('1', '08:00-18:00', 'Rua Barão de Souza Leão, 123, Boa Viagem, Recife - PE, CEP 51030-300', TO_DATE('01/01/2020', 'DD/MM/YYYY'));
```

```
...
```

SQL

TABELA SALA

```
-- CRIA TABELA SALA
CREATE TABLE SALA
(
  NUMERO_SALA VARCHAR(6) NOT NULL,
  COD_CADASTRO_POSTO NUMBER(1 1) NOT NULL,
  CONSTRAINT PK_SALA PRIMARY KEY (NUMERO_SALA, COD_CADASTRO_POSTO),
  CONSTRAINT FK_SALA FOREIGN KEY (COD_CADASTRO_POSTO) REFERENCES
  POSTO (COD_CADASTRO_POSTO)
);

-- POVOA TABELA SALA
INSERT INTO SALA (COD_CADASTRO_POSTO, NUMERO_SALA) VALUES ('1', '001');

...
```

NUMERO_SALA	COD_CADASTRO_POSTO
001	1
001	6
001	10
002	1
002	3
002	7
002	11
003	3
003	4
003	8
004	5
004	9

SQL

TABELA PESSOA

```
-- CRIA TABELA PESSOA
```

```
CREATE TABLE PESSOA
```

```
(
```

```
CPF VARCHAR(11) NOT NULL,
```

```
RG VARCHAR(9) NOT NULL,
```

```
NOME VARCHAR(150) NOT NULL,
```

```
DATA_NASCIMENTO DATE NULL,
```

```
ENDERECO VARCHAR(150) NULL,
```

```
CONSTRAINT PK_PESSOA PRIMARY KEY (CPF),
```

```
CONSTRAINT AK1_PESSOA UNIQUE (RG)
```

```
);
```

```
-- POVOA TABELA PESSOA
```

```
INSERT INTO PESSOA VALUES ('98765432109', '363261977', 'Barbara Vaz Ferreira', TO_DATE('01/05/2002', 'DD/MM/YYYY'), 'Avenida Domingos Ferreira, 123');
```

```
...
```

CPF	RG	NOME	DATA_NASCIMENTO	ENDERECO
98765432109	363261977	Barbara Vaz Ferreira	01-MAY-02	Avenida Domingos Ferreira, 123
12345678909	765432123	Ismael Benjamim da Silva Barros	02-FEB-00	Avenida Caxangá, 456
98765432108	123456610	Liliane Carla	03-MAR-01	Avenida Sá e Souza, 789
12345678908	76543205	Dante Santana	04-APR-00	Avenida Caxangá, 1011
98765432107	12345652	Tiago Bello	05-MAY-02	Avenida Caxangá, 456
12345678907	76543195	Fernanda Costa	06-JUN-92	Rua F, 1415 - Piedade, Recife

SQL

TABELA PROF DA SAÚDE

```
-- CRIA TABELA PROFISSIONAL_DA_SAUDE
CREATE TABLE PROFISSIONAL_DA_SAUDE
(
  CPF VARCHAR(12) NOT NULL,
  CARTEIRA_ID_PROFISSIONAL VARCHAR(10) NOT NULL,
  ANO_FORMACAO NUMBER(4) NOT NULL,
  CONSTRAINT PK_PROFISSIONAL_DA_SAUDE PRIMARY KEY (CPF),
  CONSTRAINT CARTEIRA_ID_PROFISSIONAL_AK UNIQUE (CARTEIRA_ID_PROFISSIONAL),
  CONSTRAINT FK_PF_PESSOA FOREIGN KEY (CPF) REFERENCES PESSOA (CPF)
);
-- POVOA TABELA PROFISSIONAL_DA_SAUDE
INSERT INTO PROFISSIONAL_DA_SAUDE VALUES ('98765432109', 'HK3E89G7M2', 2015);
...
```

CPF	CARTEIRA_ID_PROFISSIONAL	ANO_FORMACAO
98765432109	HK3E89G7M2	2015
12345678909	6R9X2N5K1D	2010
98765432108	L4S2P7Q5CJ	2005
12345678908	8Y6F3D9V7B	2020
98765432107	T1W7G4Z0JF	2012

SQL

TABELA PACIENTE

```
-- CRIA TABELA PACIENTE
CREATE TABLE PACIENTE
(
  CPF VARCHAR(12) NOT NULL,
  CARTAO_VACINACAO VARCHAR(11) NOT NULL,
  CONSTRAINT PK_PACIENTE PRIMARY KEY (CPF),
  CONSTRAINT CARTAO_VACINACAO_AK UNIQUE (CARTAO_VACINACAO),
  CONSTRAINT FK_PF_SAUDE FOREIGN KEY (CPF) REFERENCES PESSOA (CPF)
);

-- POVOA TABELA PACIENTE
INSERT INTO PACIENTE (CPF, CARTAO_VACINACAO) VALUES ('98324167431', 'S7Q8N1R0L5M');

...
```

CPF	CARTAO_VACINACAO
98324167431	S7Q8N1R0L5M
52489926877	F3W7G8H1K9V
71293846501	A5B9J7Z0D3
28903457263	X6C2T1N8M0
43789126341	H9K1Q2D0J8
09283745609	T4P6Z0V7F1

SQL

TABELA ORIENTA

```
-- CRIA TABELA ORIENTA
CREATE TABLE ORIENTA
(
  CPF_TEC_ENF VARCHAR(12) NOT NULL,
  CPF_ENF VARCHAR(12) NOT NULL,
  CONSTRAINT CPF_TEC_ENF PRIMARY KEY (CPF_TEC_ENF, CPF_ENF),
  CONSTRAINT FK_CPF_ENF FOREIGN KEY (CPF_ENF) REFERENCES PROFISSIONAL_DA_SAUDE (CPF),
  CONSTRAINT FK_CPF_TEC_ENF FOREIGN KEY (CPF_TEC_ENF) REFERENCES PROFISSIONAL_DA_SAUDE (CPF)
);

-- POVOA TABELA ORIENTA
INSERT INTO ORIENTA (CPF_TEC_ENF, CPF_ENF) VALUES ('98765432109', '12345678905');

...
```

CPF_TEC_ENF	CPF_ENF
12345678905	98765432109
12345678906	98765432108
12345678907	98765432107
12345678908	98765432106

SQL

TABELA VACINA

COD_VACINA	LOTE	TIPO	LABORATORIO	DATA_FABRICACAO	DATA_VALIDADE
1	153AB-456CD	Pfizer/BioNTech	Laboratório São Marcos	15-JUN-22	12-FEB-23
2	246XY-908ZT	Moderna	Laboratório Leme	29-NOV-22	03-JUL-23
3	342DE-789FG	Johnson & Johnson	Laboratório Maurílio de Almeida	01-APR-22	28-SEP-23
4	597HI-230JK	AstraZeneca	Laboratório de Análises Clínicas Dr. Paulo Frassinetti	23-SEP-22	16-NOV-23

```
CREATE TABLE VACINA
```

```
(  
  COD_VACINA NUMBER(1 1) GENERATED BY DEFAULT ON NULL AS IDENTITY,  
  LOTE VARCHAR(1 1) NOT NULL,  
  TIPO VARCHAR(20) NOT NULL,  
  LABORATORIO VARCHAR(150) NOT NULL,  
  DATA_FABRICACAO DATE NOT NULL,  
  DATA_VALIDADE DATE NOT NULL,  
  CONSTRAINT PK_VACINA PRIMARY KEY (COD_VACINA)  
);
```

```
-- POVOA TABELA VACINA
```

```
INSERT INTO VACINA (COD_VACINA, LOTE, TIPO, LABORATORIO, DATA_FABRICACAO, DATA_VALIDADE) VALUES ('1',  
'153AB-456CD', 'Pfizer/BioNTech', 'Laboratório São Marcos', TO_DATE('15/06/2022', 'DD/MM/YYYY'),  
TO_DATE('12/02/2023', 'DD/MM/YYYY'));
```

```
...
```

SQL

TABELA AGULHA

```
CREATE TABLE AGULHA
(
  NUMERACAO NUMBER(1 1) GENERATED BY DEFAULT ON NULL AS IDENTITY,
  TIPO VARCHAR(20) NOT NULL,
  DATA_FABRICACAO DATE NOT NULL,
  DATA_VALIDADE DATE NOT NULL,
  COD_VACINA NUMBER(1 1) NOT NULL,
  CONSTRAINT PK_AGULHA PRIMARY KEY (NUMERACAO),
  CONSTRAINT FK_VACINA FOREIGN KEY (COD_VACINA) REFERENCES VACINA (COD_VACINA)
);

-- POVOA TABELA AGULHA
INSERT INTO AGULHA (NUMERACAO, TIPO, DATA_FABRICACAO, DATA_VALIDADE, COD_VACINA) VALUES ('1', '13 x 4,5 cm',
TO_DATE('15/03/2018', 'DD/MM/YYYY'), TO_DATE('15/03/2023', 'DD/MM/YYYY'), '1');
...
```

NUMERACAO	TIPO	DATA_FABRICACAO	DATA_VALIDADE	COD_VACINA
1	13 x 4,5 cm	15-MAR-18	15-MAR-23	1
2	30 x 8 mm	28-JUL-19	28-JUL-23	2
3	25 x 7 mm	02-NOV-20	02-NOV-23	3
4	21 x 4 cm	14-MAY-21	14-MAY-24	4
5	18 x 3,8 cm	18-SEP-18	18-SEP-24	5
6	16 x 1,5 cm	05-FEB-19	05-FEB-23	6
7	14 x 2 cm	30-JUN-20	30-JUN-23	7
8	12 x 3,5 cm	12-OCT-21	12-OCT-24	8
9	10 x 3 mm	25-DEC-18	25-DEC-23	9
10	8 x 1,5 cm	08-APR-19	08-APR-24	10

SQL

TABELA PESSOA
VAI POSTO

```
CREATE PESSOA_VAI_POSTO
(
  CPF VARCHAR(12) NOT NULL,
  COD_CADASTRO_POSTO NUMBER(1 1) NOT NULL,
  CONSTRAINT PK_PESSOA_VAI_POSTO PRIMARY KEY (CPF, COD_CADASTRO_POSTO),
  CONSTRAINT FK_PESSOA FOREIGN KEY (CPF) REFERENCES PESSOA (CPF),
  CONSTRAINT FK_POSTO FOREIGN KEY (COD_CADASTRO_POSTO) REFERENCES POSTO (COD_CADASTRO_POSTO)
);
-- POVOA TABELA PESSOA_VAI_POSTO
INSERT INTO PESSOA_VAI_POSTO (CPF, COD_CADASTRO_POSTO) VALUES ('12345678906', '1');
...
```

CPF	COD_CADASTRO_POSTO
09283745609	10
12345678905	4
12345678906	1
28903457263	8
35689012456	11
43789126341	9
52489926877	6
71293846501	7
98324167431	5
98765432105	3

SQL

TABELA POSTO TEM PS

```
CREATE TABLE POSTO_TEM_PS
(
  CPF_PROF_SAUDE VARCHAR(12) NOT NULL,
  COD_CADASTRO_POSTO NUMBER(1 1) NOT NULL,
  CONSTRAINT PK_POSTO_TEM_PS PRIMARY KEY (CPF_PROF_SAUDE, COD_CADASTRO_POSTO),
  CONSTRAINT FK_PF_SAUDE_CPF FOREIGN KEY (CPF_PROF_SAUDE) REFERENCES PROFISSIONAL_DA_SAUDE (CPF),
  CONSTRAINT FK_POSTO_COD FOREIGN KEY (COD_CADASTRO_POSTO) REFERENCES POSTO
(COD_CADASTRO_POSTO)
);
-- POVOA TABELA POSTO_TEM_PS
INSERT INTO POSTO_TEM_PS (COD_CADASTRO_POSTO, CPF_PROF_SAUDE) VALUES ('1', '98765432109');
```

CPF_PROF_SAUDE	COD_CADASTRO_POSTO
12345678905	11
12345678906	9
12345678907	7
12345678908	5

SQL

TABELA CONSULTA

```
CREATE TABLE CONSULTA
```

```
(
```

```
  COD_CONSULTA NUMBER(11) GENERATED BY DEFAULT ON NULL AS IDENTITY,
```

```
  CPF_PROF_SAUDE VARCHAR(12) NOT NULL,
```

```
  CPF_PACIENTE VARCHAR(12) NOT NULL,
```

```
  DATA_CONSULTA DATE NOT NULL,
```

```
  NUMERO_SALA VARCHAR(6) NOT NULL,
```

```
  COD_CADASTRO_POSTO NUMBER(11) NOT NULL,
```

```
  CONSTRAINT PK_CONSULTA PRIMARY KEY (COD_CONSULTA),
```

```
  CONSTRAINT FK_PACIENTE_CPF FOREIGN KEY (CPF_PACIENTE) REFERENCES PACIENTE (CPF),
```

```
  CONSTRAINT FK_PF_DA_SAUDE_CPF FOREIGN KEY (CPF_PROF_SAUDE) REFERENCES PROFISSIONAL_DA_SAUDE (CPF),
```

```
  CONSTRAINT FK_SALA_CODIGO_CADASTRO FOREIGN KEY (NUMERO_SALA, COD_CADASTRO_POSTO) REFERENCES SALA (NUMERO_SALA, COD_CADASTRO_POSTO),
```

```
  CONSTRAINT CK_CPF_PFSAUDE_PACIENTE CHECK (CPF_PROF_SAUDE <> CPF_PACIENTE)
```

```
);
```

COD_CONSULTA	CPF_PROF_SAUDE	CPF_PACIENTE	DATA_CONSULTA	NUMERO_SALA	COD_CADASTRO_POSTO
1	98765432109	98324167431	02-NOV-22	001	1
2	12345678909	52489926877	22-MAY-23	002	3
3	98765432108	71293846501	16-SEP-22	003	4
4	12345678908	28903457263	12-APR-23	004	5
5	98765432107	43789126341	29-DEC-22	001	6

SQL

TABELA CONSULTA

COD_CONSULTA	CPF_PROF_SAUDE	CPF_PACIENTE	DATA_CONSULTA	NUMERO_SALA	COD_CADASTRO_POSTO
1	98765432109	98324167431	02-NOV-22	001	1
2	12345678909	52489926877	22-MAY-23	002	3
3	98765432108	71293846501	16-SEP-22	003	4
4	12345678908	28903457263	12-APR-23	004	5
5	98765432107	43789126341	29-DEC-22	001	6

```
--POVOA TABELA CONSULTA
```

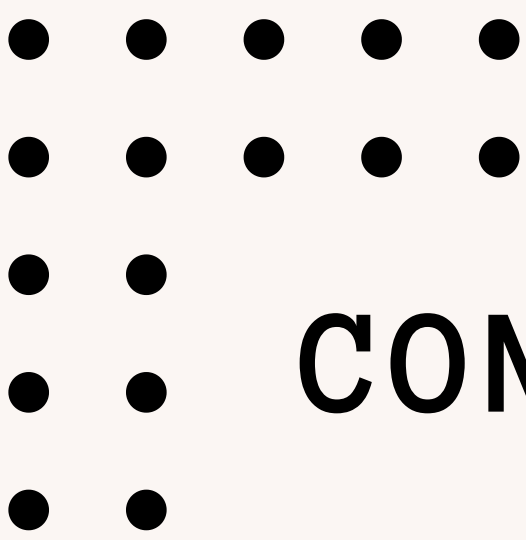
```
INSERT INTO CONSULTA (COD_CONSULTA, CPF_PROF_SAUDE, CPF_PACIENTE,  
COD_CADASTRO_POSTO, NUMERO_SALA, DATA_CONSULTA) VALUES ('1', '98765432109',  
'98324167431', '1', '001', TO_DATE('02/11/2022', 'DD/MM/YYYY'));
```


SQL

TABELA DOSE

```
CREATE TABLE DOSE
(
  COD_DOSE NUMBER(1 1) GENERATED BY DEFAULT ON NULL AS IDENTITY,
  COD_CONSULTA NUMBER(1 1) NOT NULL,
  COD_VACINA NUMBER(1 1) NOT NULL,
  CONSTRAINT PK_DOSE PRIMARY KEY (COD_DOSE),
  CONSTRAINT FK_DOSE FOREIGN KEY (COD_VACINA) REFERENCES VACINA (COD_VACINA),
  CONSTRAINT FK_DOSE_CONSULTA FOREIGN KEY (COD_CONSULTA) REFERENCES CONSULTA (COD_CONSULTA)
);
--POVOA TABELA DOSE
INSERT INTO DOSE (COD_DOSE, COD_VACINA, COD_CONSULTA) VALUES ('1', '2', '8');
```

COD_DOSE	COD_CONSULTA	COD_VACINA
1	8	2
2	6	3
3	1	3
4	3	5
5	9	4
6	5	8
7	4	6
8	9	1
9	5	8
10	7	3
11	10	10

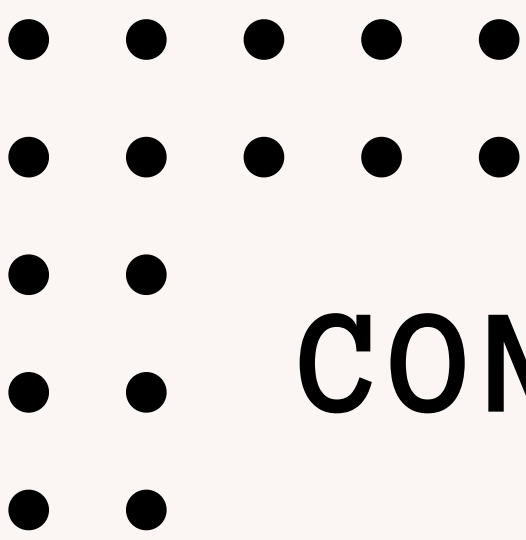


CONSULTAS

Projetar os postos que tem mais de 1 sala.

```
SELECT COD_CADASTRO_POSTO, COUNT(*)  
FROM SALA  
GROUP BY COD_CADASTRO_POSTO  
HAVING COUNT(*) > 1;
```

COD_CADASTRO_POSTO	COUNT(*)
1	2
3	2

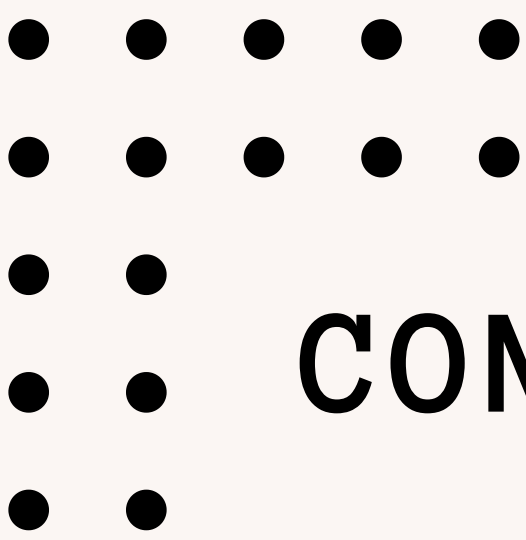


CONSULTAS

Projetar o tipo das vacinas e os tipos de agulhas usadas nelas.

```
SELECT V.TIPO, A.TIPO
FROM VACINA V INNER JOIN
AGULHA A ON V.COD_VACINA = A.COD_VACINA;
```

TIPO	TIPO
Pfizer/BioNTech	13 x 4,5 cm
Moderna	30 x 8 mm
Johnson & Johnson	25 x 7 mm
AstraZeneca	21 x 4 cm
Sinovac	18 x 3,8 cm
Sputnik V	16 x 1,5 cm
Covovax	14 x 2 cm
Novavax	12 x 3,5 cm
Bharat Biotech	10 x 3 mm
CanSino	8 x 1,5 cm

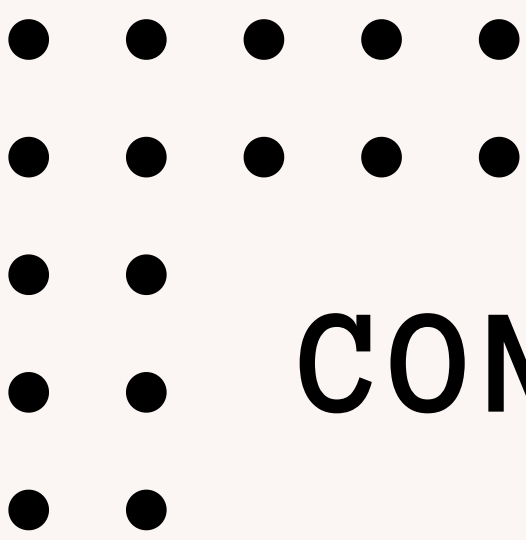


CONSULTAS

Projetar todas as vacinas que não possuem
agulhas no banco.

```
SELECT V.LOTE, A.NUMERACAO
FROM VACINA V LEFT JOIN
  AGULHA A ON V.COD_VACINA = A.COD_VACINA
WHERE A.COD_VACINA IS NULL;
```

LOTE	NUMERACAO
395GH-901IP	-
645GH-009YJ	-

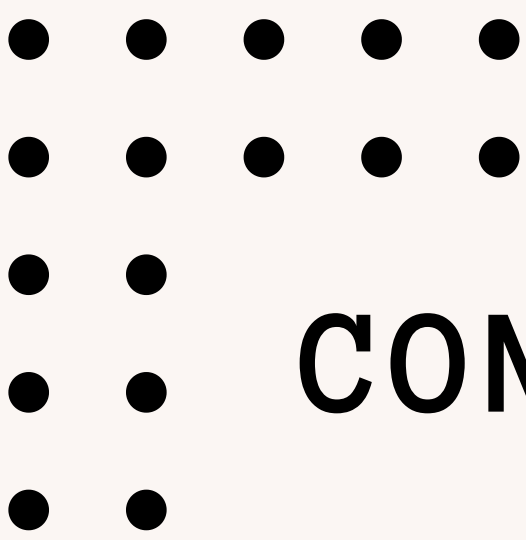


CONSULTAS

Projetar o endereço dos postos com sala.

```
SELECT P.ENDERECO
FROM POSTO P
WHERE EXISTS (SELECT *
FROM SALA S
WHERE P.COD_CADASTRO_POSTO =
S.COD_CADASTRO_POSTO);
```

ENDERECO
Rua Barão de Souza Leão, 123, Boa Viagem, Recife - PE, CEP 51030-300
Avenida Conde da Boa Vista, 123, Boa Vista, Recife - PE, CEP 50060-004
Avenida Caxangá, 123, Cordeiro, Recife - PE, CEP 50640-000
Avenida Engenheiro Domingos Ferreira, 789, Pina, Recife - PE, CEP 51011-051
Rua Joaquim Nabuco, 123, Graças, Recife - PE, CEP 52011-000
Rua da Aurora, 123, Boa Vista, Recife - PE, CEP 50050-000
Rua da Hora, 123, Espinheiro, Recife - PE, CEP 52020-010
Avenida Beira Rio, 123, Olinda - PE, CEP 53030-010
Rua Ribeiro de Brito, 123, Boa Viagem, Recife - PE, CEP 51021-310
Rua Imperatriz Tereza Cristina, 123, Santo Amaro, Recife - PE, CEP 50100-280

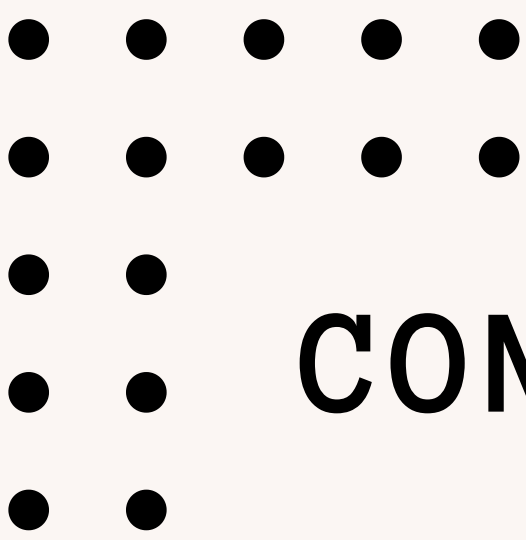


CONSULTAS

Projetar o tipo das vacinas que não tem doses.

```
SELECT V.TIPO
FROM VACINA V
WHERE NOT EXISTS (SELECT *
FROM DOSE D
WHERE V.COD_VACINA = D.COD_VACINA);
```

TIPO
Covovax
CanSino
CanSino
Bharat Biotech

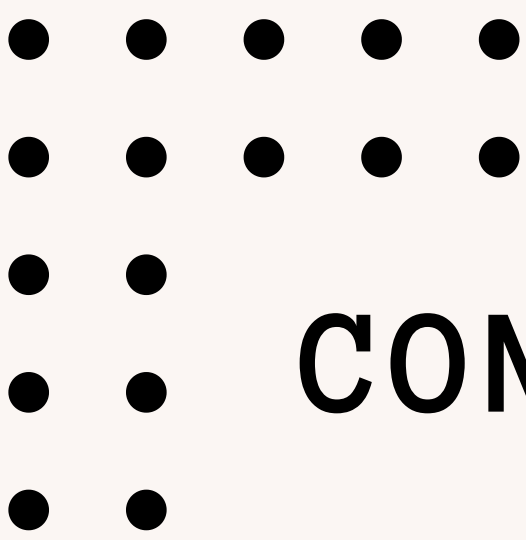


CONSULTAS

Projetar o código dos postos tem mais salas do que a média de salas.

```
SELECT COD_CADASTRO_POSTO, COUNT(*)
FROM SALA
GROUP BY COD_CADASTRO_POSTO
HAVING COUNT(*) > (SELECT AVG(QTD)
FROM (SELECT COD_CADASTRO_POSTO, COUNT(*) AS QTD
FROM SALA
GROUP BY COD_CADASTRO_POSTO));
```

COD_CADASTRO_POSTO	COUNT(*)
1	2
3	2



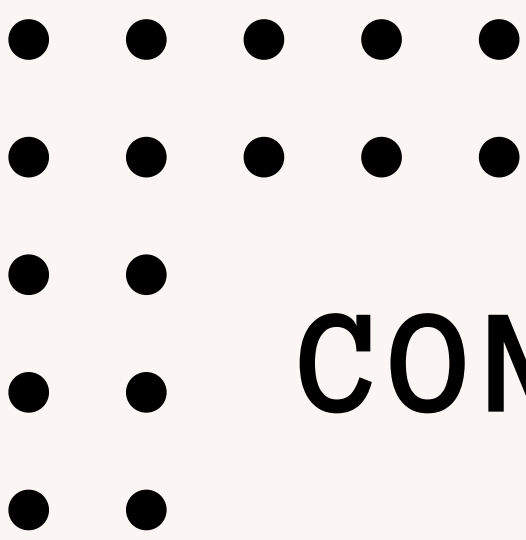
CONSULTAS

Projetar o código das vacinas que sejam do mesmo tipo e laboratória da vacina '31058947623'.

```
SELECT VI.LABORATORIO, V1.COD_VACINA AS "Código da Vacina"
FROM VACINA V1
WHERE (V1.TIPO, V1.LABORATORIO) = (SELECT
V2.TIPO, V2.LABORATORIO
FROM VACINA V2
WHERE V2.COD_VACINA = '11');
```

LABORATORIO	Código da Vacina
Laboratório São Francisco	10
Laboratório São Francisco	11
Laboratório São Francisco	12

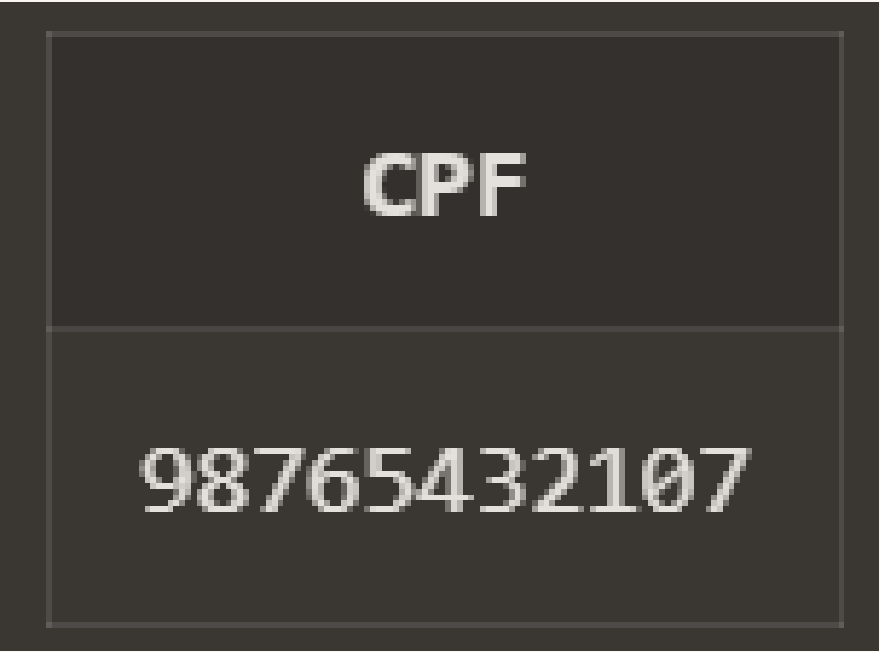




CONSULTAS

Projetar todos os pacientes que também são profissionais da saúde.

```
SELECT CPF
FROM PACIENTE
INTERSECT
SELECT CPF
FROM PROFISSIONAL_DA_SAUDE;
```



PROCEDIMENTO E FUNÇÃO

EXIBIR A QUANTIDADE DE SALAS NO
POSTO 3

Statement processed.
Quantidade de salas no posto 3: 2

MOSTRA_QTD_SALAS

2

```
CREATE OR REPLACE PROCEDURE EXIBE_QTD_SALAS IS
    QTD NUMBER;
BEGIN
    SELECT COUNT(*) INTO QTD FROM SALA WHERE
    COD_CADASTRO_POSTO = '3';
    DBMS_OUTPUT.PUT_LINE('Quantidade de salas no posto
3: '|| QTD);
END;
/
EXECUTE EXIBE_QTD_SALAS;
```

```
CREATE OR REPLACE FUNCTION MOSTRA_QTD_SALAS
RETURN NUMBER IS
    QTD NUMBER;
BEGIN
    SELECT COUNT(*) INTO QTD FROM SALA WHERE
    COD_CADASTRO_POSTO = '3';
    RETURN QTD;
END;
/
SELECT MOSTRA_QTD_SALAS FROM DUAL;
```

PROCEDIMENTO E FUNÇÃO

EXIBIR A QUANTIDADE DE SALAS EM
QUALQUER POSTO (PARÂMETRO):

```
CREATE OR REPLACE PROCEDURE EXIBE_QTD_SALAS
(COD_POSTO IN NUMBER) IS
    QTD NUMBER;
BEGIN
    SELECT COUNT(*) INTO QTD FROM SALA WHERE
COD_CADASTRO_POSTO = COD_POSTO;
    DBMS_OUTPUT.PUT_LINE('Quantidade de salas no posto
'|| COD_POSTO ||': '|| QTD);
END;
/
EXECUTE EXIBE_QTD_SALAS('11');
```

```
Statement processed.
Quantidade de salas no posto 11: 1
```

COD_CADASTRO_POSTO	QTD_SALAS
3	2

```
CREATE OR REPLACE FUNCTION MOSTRA_QTD_SALAS
(COD_POSTO NUMBER) RETURN NUMBER IS
    QTD NUMBER;
BEGIN
    SELECT COUNT(*) INTO QTD FROM SALA WHERE
COD_CADASTRO_POSTO = COD_POSTO;
    RETURN QTD;
END;
/
SELECT MOSTRA_QTD_SALAS('3') FROM DUAL;
/
SELECT COD_CADASTRO_POSTO, MOSTRA_QTD_SALAS('3')
AS QTD_SALAS FROM POSTO
WHERE COD_CADASTRO_POSTO = '3';
```

FUNÇÃO

```
Statement processed.  
O Profissional de saude com CPF "98765432109" realizou um total de: 3 consultas.
```

TOTAL DE CONSULTAS REALIZADAS POR UM PROFISSIONAL

```
CREATE OR REPLACE FUNCTION  
consultasRealizadas(cpf_profissional VARCHAR)  
RETURN NUMBER IS  
    quantidade_consultas NUMBER(11);  
BEGIN  
    SELECT COUNT(*) into quantidade_consultas  
    FROM CONSULTA WHERE CPF_PROF_SAUDE =  
    cpf_profissional;  
    RETURN quantidade_consultas;  
END;  
/
```

```
DECLARE  
    num_cpf VARCHAR(12);  
    qtd_consultas NUMBER(11);  
BEGIN  
    num_cpf := '98765432109';  
    qtd_consultas := consultasRealizadas(num_cpf);  
    dbms_output.put_line('O Profissional de saude com CPF "' ||  
    num_cpf || '" realizou um total de: ' || qtd_consultas || '  
consultas.');
```

PROCEDIMENTO

```
Statement processed.  
Barbara Vaz Ferreira (98765432109) realizou 3 consultas  
Tiago Bello (98765432107) realizou 1 consultas  
Juliana Santos (12345678905) realizou 1 consultas  
Marcela Carvalho (98765432106) realizou 1 consultas  
Liliane Carla (98765432108) realizou 1 consultas
```

RANKING DE CONSULTAS REALIZADAS POR PROFISSIONAIS DE SAUDE EM UM PERIODO

```
CREATE OR REPLACE PROCEDURE RankingConsultasProfissionaisPorData(dt_ini IN DATE, dt_fim IN DATE)  
IS  
    CURSOR profissionais_saude IS  
        SELECT P.NOME, P.CPF, COUNT(*) AS qtd_consulta FROM CONSULTA, PESSOA P WHERE ((DATA_CONSULTA BETWEEN dt_ini AND dt_fim)  
AND (P.CPF = CPF_PROF_SAUDE)) GROUP BY CPF_PROF_SAUDE, P.NOME, P.CPF ORDER BY COUNT(*) DESC;  
BEGIN  
    FOR professional IN profissionais_saude  
    LOOP  
        dbms_output.put_line(professional.nome || ' (' || professional.cpf || ') realizou ' || professional.qtd_consulta || ' consultas');  
    END LOOP;  
END;  
  
EXECUTE RankingConsultasProfissionais(TO_DATE('02/10/2018', 'DD/MM/YYYY'), TO_DATE('01/01/2023', 'DD/MM/YYYY'));
```

FUNÇÃO

Statement processed.

As agulhas fabricadas no dia "15-MAR-18" já foram utilizadas em 1 vacinas.

QUANTAS VACINAS JÁ FORAM UTILIZADAS PELAS
AGULHAS FABRICADAS NO DIA 15/03/2018?

```
CREATE OR REPLACE FUNCTION
quantidadeVacinas(dt_fabricacao IN DATE)
RETURN NUMBER IS
quantidade_vacina NUMBER(11);
BEGIN
SELECT COUNT(*) into quantidade_vacina
FROM VACINA V WHERE V.COD_VACINA IN (SELECT
A.COD_VACINA
FROM AGULHA A
WHERE A.DATA_FABRICACAO = dt_fabricacao);
RETURN quantidade_vacina;
END;
/
```

```
DECLARE
data_fabricacao DATE;
quantidade_vacina NUMBER(11);
BEGIN
data_fabricacao := TO_DATE('15/03/2018', 'DD/MM/YYYY');
quantidade_vacina := quantidadeVacinas(data_fabricacao);
dbms_output.put_line('As agulhas fabricadas no dia "' ||
data_fabricacao || '" já foram utilizadas em ' ||
quantidade_vacina || ' vacinas.');
```

PROCEDIMENTO

INSERIR A SALA 9 NO POSTO 11

NUMERO_SALA	COD_CADASTRO_POSTO
002	11
9	11

```
CREATE OR REPLACE PROCEDURE InserirSalaNoPosto(num_sala VARCHAR, cod_posto VARCHAR)
IS
BEGIN
  INSERT INTO SALA (NUMERO_SALA, COD_CADASTRO_POSTO) VALUES(num_sala, cod_posto);
  COMMIT;
END;

EXECUTE InserirSalaNoPosto('9', '11');
```

FUNÇÃO

Function created.

Statement processed.

0 Posto de Saúde "11" recebeu 1 visitas.

TOTAL DE VISITAS A UM POSTO DE SAÚDE

```
CREATE OR REPLACE FUNCTION
numeroVisitas(cadastro_posto NUMBER)
RETURN NUMBER IS
    qtd_visitas NUMBER(11);
BEGIN
    SELECT COUNT(*) into qtd_visitas
    FROM PESSOA_VAI_POSTO
    WHERE COD_CADASTRO_POSTO = cadastro_posto;
    RETURN qtd_visitas;
END;
/
```

```
DECLARE
    cod_posto NUMBER(11);
    numero_visitas NUMBER(11);
BEGIN
    cod_posto := '11';
    numero_visitas := numeroVisitas(cod_posto);
    dbms_output.put_line('O Posto de Saúde "' || cod_posto ||
'" recebeu ' || numero_visitas || ' visitas.');
```

END;

/

PROCEDIMENTO

TOTAL DE VISITAS NOS POSTOS
DE SAÚDE EM UM DETERMINADO
PERÍODO DE TEMPO

```
Statement processed.  
0 posto de código 11 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 10 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 9 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 8 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 7 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 6 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 5 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 4 recebeu 2 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 3 recebeu 4 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 1 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21
```

```
CREATE OR REPLACE PROCEDURE numeroVisitasPorData(data_inicio IN DATE, data_fim IN DATE) IS  
  CURSOR visitas_posto IS  
    SELECT P.COD_CADASTRO_POSTO, COUNT(PVP.CPF) AS "VISITAS"  
    FROM POSTO P INNER JOIN PESSOA_VAI_POSTO PVP ON P.COD_CADASTRO_POSTO =  
    PVP.COD_CADASTRO_POSTO  
    WHERE ((DATA_ABERTURA BETWEEN data_inicio AND data_fim))  
    GROUP BY P.COD_CADASTRO_POSTO  
    ORDER BY P.COD_CADASTRO_POSTO ASC;  
...
```

PROCEDIMENTO

TOTAL DE VISITAS NOS POSTOS
DE SAÚDE EM UM DETERMINADO
PERÍODO DE TEMPO

```
Statement processed.  
0 posto de código 11 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 10 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 9 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 8 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 7 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 6 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 5 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 4 recebeu 2 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 3 recebeu 4 visitas entre 01-JAN-20 e 01-JAN-21  
0 posto de código 1 recebeu 1 visitas entre 01-JAN-20 e 01-JAN-21
```

```
BEGIN
```

```
FOR visitas IN visitas_posto
```

```
LOOP
```

```
dbms_output.put_line('0 posto de código ' || visitas.cod_cadastro_posto || ' recebeu ' || visitas.visitas  
|| ' visitas entre ' || data_inicio || ' e ' || data_fim);
```

```
END LOOP;
```

```
END;
```

```
EXECUTE numeroVisitasPorData(TO_DATE('01/01/2020', 'DD/MM/YYYY'), TO_DATE('01/01/2021',  
'DD/MM/YYYY'));
```

FUNÇÃO

PEGAR O TIPO DA ÚLTIMA VACINA APLICADA NO POSTO 10

```
CREATE OR REPLACE FUNCTION
ultimaVacinaPosto(COD_POSTO IN NUMBER)
RETURN VARCHAR IS
    VACINA_TIPO VARCHAR(20);
BEGIN
    SELECT V.TIPO INTO VACINA_TIPO
    FROM VACINA V
    INNER JOIN DOSE D
    ON V.COD_VACINA = D.COD_VACINA
    INNER JOIN CONSULTA C
    ON D.COD_CONSULTA = C.COD_CONSULTA
    WHERE COD_CADASTRO_POSTO = COD_POSTO
    ORDER BY C.DATA_CONSULTA DESC
    FETCH FIRST 1 ROW ONLY;

    RETURN VACINA_TIPO;

END;
```

Function created.

Statement processed.

A última vacina aplicada no posto 10 foi: AstraZeneca

```
DECLARE
    COD_POSTO NUMBER := 10;
    RESULTADO VARCHAR(20);
BEGIN
    RESULTADO := ultimaVacinaPosto(COD_POSTO);
    DBMS_OUTPUT.PUT_LINE('A última vacina aplicada no
posto ' || COD_POSTO || ' foi: ' || RESULTADO);
END;
/
```

PROCEDIMENTO

Procedure created.

Statement processed.

NUMERACAO	TIPO	DATA_FABRICACAO	DATA_VALIDADE	COD_VACINA
2	30 x 8 mm	28-JUL-19	28-JUL-23	5

SUBSTITUIR A AGULHA NA VACINA
MODERNA PARA AGULHA DE 18 X3,8 CM.

```
CREATE OR REPLACE PROCEDURE alterarTipoAgulhaNaVacina(num_antigo NUMBER, num_novo  
NUMBER)
```

```
IS
```

```
BEGIN
```

```
    UPDATE AGULHA
```

```
    SET cod_vacina = num_novo
```

```
    WHERE cod_vacina = num_antigo;
```

```
    COMMIT;
```

```
END;
```

```
/
```

```
EXECUTE alterarTipoAgulhaNaVacina(2, 5);
```

FUNÇÃO

```
Statement processed.  
A SALA 001 do posto 1 tem 1 consultas.
```

RETORNA A QUANTIDADE DE CONSULTAS QUE ACONTECERAM EM UMA SALA DE UM POSTO

```
CREATE OR REPLACE FUNCTION consultasSala(cod_posto  
VARCHAR, num_sala VARCHAR)  
RETURN NUMBER IS  
    qtd_consulta_sala NUMBER(11);  
BEGIN  
    SELECT COUNT(*) into qtd_consulta_sala  
    FROM CONSULTA WHERE NUMERO_SALA = num_sala AND  
COD_CADASTRO_POSTO = cod_posto;  
    RETURN qtd_consulta_sala;  
END;
```

```
DECLARE  
    cod_posto VARCHAR(11);  
    num_sala VARCHAR(11);  
    qtd_consultas NUMBER;  
BEGIN  
    cod_posto := '1';  
    num_sala := '001';  
    qtd_consultas := consultasSala(cod_posto, num_sala);  
    dbms_output.put_line('A SALA ' || num_sala || ' do posto '  
|| cod_posto || ' tem ' || qtd_consultas || ' consultas.');
```

```
END;
```

PROCEDURE

PROCEDIMENTO PARA ALTERAR UMA LINHA DE POSTO
RECENDO UM OU TODOS OS PARAMETROS DA TABELA

```
CREATE OR REPLACE PROCEDURE altera_posto
(
    p_cod IN NUMBER,
    p_horario_funcionamento IN VARCHAR DEFAULT NULL,
    p_endereco IN VARCHAR DEFAULT NULL,
    p_data_abertura IN DATE DEFAULT NULL
)
AS
```

```
BEGIN
  DECLARE
    v_sql VARCHAR(4000) := 'UPDATE posto SET ';
  BEGIN
    IF p_horario_funcionamento IS NOT NULL THEN
      v_sql := v_sql || 'HORARIO_FUNCIONAMENTO = ' || p_horario_funcionamento || ', ';
    END IF;

    IF p_endereco IS NOT NULL THEN
      v_sql := v_sql || 'ENDERECO = ' || p_endereco || ', ';
    END IF;

    IF p_data_abertura IS NOT NULL THEN
      v_sql := v_sql || 'DATA_ABERTURA = TO_DATE(' || TO_CHAR(p_data_abertura, 'DD/MM/YYYY') || ', ' || 'DD/MM/YYYY'), ';
    END IF;

    v_sql := RTRIM(v_sql, ', ') || ' WHERE COD_CADASTRO_POSTO = ' || p_cod;
    EXECUTE IMMEDIATE v_sql;
  END;
END altera_posto;
```

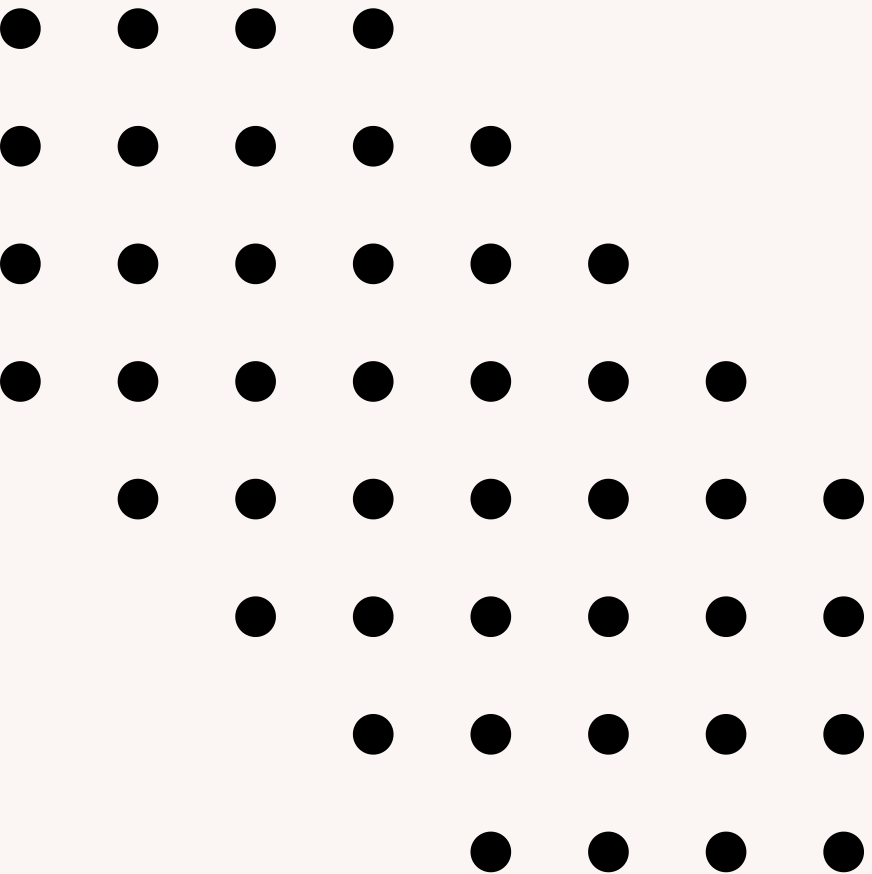

ORA-20205: Limite de profissionais de saúde atingido ORA-06512: at "SQL_NAYGPTPSNJKEQYRNOLQAOHVPL.LIMITA_PS", line 6
ORA-06512: at "SYS.DBMS_SQL", line 1721

TRIGGER

IMPEDIR MAIS DE 5 PROFISSIONAIS DE SAÚDE EM UM POSTO

```
CREATE OR REPLACE TRIGGER LIMITA_PS BEFORE INSERT ON POSTO_TEM_PS
FOR EACH ROW
DECLARE
    COUNT_ NUMBER;
BEGIN
    SELECT COUNT(*) INTO COUNT_ FROM POSTO_TEM_PS WHERE COD_CADASTRO_POSTO = :NEW.COD_CADASTRO_POSTO;
    IF COUNT_ >= 5 THEN
        RAISE_APPLICATION_ERROR(-20205,'Limite de profissionais de saúde atingido');
    END IF;
END;
/
--SELECT * FROM PROFISSIONAL_DA_SAUDE;
--SELECT * FROM POSTO_TEM_PS WHERE COD_CADASTRO_POSTO = 1;
--INSERT INTO POSTO_TEM_PS VALUES ('98765432105', '1');
```

CPF_PROF_SAUDE	COD_CADASTRO_POSTO
12345678908	1
12345678909	1
98765432107	1
98765432108	1
98765432109	1



Obrigado!

