Projeto - Gestão de agend. de Lab. CIn

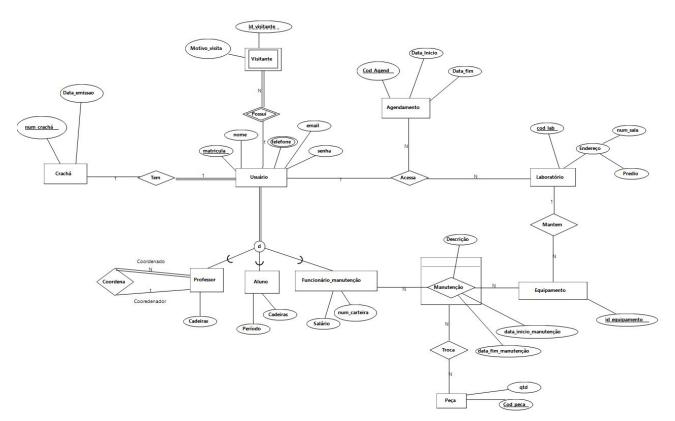
Fernanda de Araújo Lima Pascoal, Gabriel Laroche Borba, Maria Geyzianny de Sousa Silva, Valter José da Silva Junior;







Projeto Conceitual







Projeto Lógico

cracha(num_cracha, data_emissao)

usuario(<u>matricula</u>, nome!, email!, senha!, [num_cracha]!) telefone(<u>matricula</u>, telefone) matricula -> usuario(matricula) num cracha -> cracha(num cracha)

agendamento (cod_agend, data_inicio, data_fim)

laboratorio (cod lab, endereco_num_sala, endereco_predio)

equipamento(<u>id_equipamento,</u>cod_lab) cod_lab -> laboratorio(cod_lab)

peca(<u>cod_peca</u>, qtd)
visitante (<u>matricula</u>, id_<u>visitante</u>, motivo_visita)
matrícula -> usuário(matricula)

professor(<u>matricula</u>, cadeiras, prof_coordenador!) matrícula -> usuario(matricula) prof_coordenador -> professor(matricula) aluno(<u>matricula</u>, periodo, cadeiras) matrícula -> usuario(matricula)

funcionario_manuntencao(<u>matricula</u>, salário, num_carteira) matricula -> usuario(matrícula)

manutencao(<u>matricula</u>, <u>id_equipamento</u>, data_inicio_manutencao, data_fim_manutencao, descricao) matricula -> funcionario_manuntencao(matricula) id equipamento -> Equipamento(id equipamento)

troca(<u>matricula</u>, <u>id equipamento</u>, <u>cod peca</u>) matricula, <u>id_equipamento</u> -> manutencao(matricula, id_equipamento) cod peca -> Peca(cod peca)

acessa(matrícula!, cod_agend, cod_lab)
matrícula -> usuario(Matricula)
cod_agend -> agendamento(cod_agend)
cod_lab -> laboratório(cod_lab)





Projeto Físico

```
CREATE TABLE CRACHA (
num_cracha NUMBER(4),
data emissao DATE.
CONSTRAINT PK_CRACHA PRIMARY KEY(num_cracha)
CREATE TABLE USUARIO (
matricula NUMBER(10),
nome VARCHAR2(80) NOT NULL,
email VARCHAR2(100) NOT NULL,
senha VARCHAR2(100) NOT NULL,
num_cracha NUMBER(4) NOT NULL UNIQUE,
CONSTRAINT PK_USUARIO PRIMARY KEY (matricula),
CONSTRAINT FK_USU_CRA FOREIGN KEY (num_cracha)
REFERENCES CRACHA (num_cracha)
```

```
matricula NUMBER(10),
telefone VARCHAR2(20),
CONSTRAINT PK TELEFONES PRIMARY KEY (matricula.
telefone),
CONSTRAINT FK_TEL_USU FOREIGN KEY (matricula)
REFERENCES USUARIO (matricula)
CREATE TABLE AGENDAMENTO (
cod_agend NUMBER(4),
data_inicio DATE NOT NULL,
data fim DATE NOT NULL.
CONSTRAINT PK_AGENDAMENTO PRIMARY KEY (cod_agend)
CREATE TABLE LABORATORIO (
 cod_lab NUMBER(4),
 endereco_num_sala VARCHAR2(50),
 endereco_predio VARCHAR2(50),
 CONSTRAINT PK_LABORATORIO PRIMARY KEY (cod_lab)
CREATE TABLE EQUIPAMENTO (
 id_equipamento NUMBER(4),
 cod_lab NUMBER(4),
 CONSTRAINT PK EQUIPAMENTO PRIMARY KEY
(id_equipamento),
 CONSTRAINT FK_COD_LAB FOREIGN KEY (cod_lab)
REFERENCES LABORATORIO(cod_lab)
```

CREATE TABLE TELEFONES (

```
CREATE TABLE PECA (
cod_peca NUMBER(4),
qtd NUMBER(4) NOT NULL,
CONSTRAINT PK_PECA PRIMARY KEY (cod_peca)
);

CREATE TABLE VISITANTE (
matricula NUMBER(10),
id_visitante VARCHAR2(10),
motivo_visita VARCHAR2(100),
CONSTRAINT PK_VISITANTE PRIMARY KEY(matricula, id_visitante),
CONSTRAINT FK_VIS_USU FOREIGN KEY (matricula) REFERENCES
USUARIO(matricula)
);
```







Projeto Físico

CREATE TABLE PROFESSOR (matricula NUMBER(10), cadeiras VARCHAR2(100), prof_coordenador NUMBER(10) NOT NULL, CONSTRAINT PK PROFESSOR PRIMARY KEY (matricula), CONSTRAINT FK_PRO_USU FOREIGN KEY (matricula) REFERENCES USUARIO (matricula), CONSTRAINT FK PRO PROF FOREIGN KEY (prof_coordenador) REFERENCES PROFESSOR (matricula) CREATE TABLE ALUNO (matricula NUMBER(10), periodo VARCHAR2(50), cadeira VARCHAR2(100), CONSTRAINT PK_ALUNO PRIMARY KEY (matricula), CONSTRAINT FK_ALU_USU FOREIGN KEY (matricula) REFERENCES USUARIO (matricula)

CREATE TABLE FUNCIONARIO_MANUTENCAO (matricula NUMBER(10), salario NUMBER(4), num_carteira VARCHAR2(50), CONSTRAINT PK_FUNC_MANUT PRIMARY KEY (matricula), CONSTRAINT FK_FUNC_MANUT_USU FOREIGN KEY (matricula) REFERENCES USUARIO (matricula) CREATE TABLE MANUTENCAO (matricula NUMBER(10), id_equipamento NUMBER(4), data inicio manutencao DATE. data fim manutencao DATE. descrição VARCHAR2(200), CONSTRAINT PK_MANUTENCAO PRIMARY KEY (matricula, id_equipamento), CONSTRAINT FK_MANU_EQUIP FOREIGN KEY (id_equipamento) REFERENCES EQUIPAMENTO (id_equipamento), CONSTRAINT FK_MANU_FUNC_MANUT FOREIGN KEY (matricula) REFERENCES FUNCIONARIO_MANUTENCAO (matricula)

CREATE TABLE TROCA (matricula NUMBER(10), id_equipamento NUMBER(4), cod_peca NUMBER(4), CONSTRAINT PK_TROCA PRIMARY KEY (matricula, id_equipamento, cod_peca), CONSTRAINT FK_TROCA_MANU_MAT FOREIGN KEY (matricula, id_equipamento) REFERENCES MANUTENCAO (matricula, id_equipamento), CONSTRAINT FK TROCA PECA FOREIGN KEY (cod peca) REFERENCES PECA (cod_peca) CREATE TABLE TROCA (matricula NUMBER(10), id_equipamento NUMBER(4), cod_peca NUMBER(4), CONSTRAINT PK_TROCA PRIMARY KEY (matricula, id_equipamento, cod_peca), CONSTRAINT FK TROCA MANU MAT FOREIGN KEY (matricula. id_equipamento) REFERENCES MANUTENCAO (matricula, id_equipamento), CONSTRAINT FK_TROCA_PECA FOREIGN KEY (cod_peca) REFERENCES PECA (cod_peca) cin.ufpe.br



Consulta - Junção Interna

 Mostre o nome e data emissão do cracha de todos os usuarios que são alunos e estão no segundo período

```
-- CONSULTAS SELECT

-- Mostre o nome e data emissão do cracha de todos os usuarios que são alunos e estão no segundo período

SELECT U.NOME, C.DATA_EMISSAO

FROM USUARIO U INNER JOIN CRACHA C ON U.NUM_CRACHA = C.NUM_CRACHA INNER JOIN ALUNO A ON U.MATRICULA = A.MATRICULA

WHERE PERIODO = '2';

10
11
12
13
14
15
```

NOME	DATA_EMISSAO		
Valter Junior	04-APR-22		
Vinicius Monitor	05-MAY-22		

cin.ufpe.br





Consulta - Junção Interna com Subconsulta Tabela

 Deve-se mostrar o nome de todos os alunos que agendaram laboratórios na data 02-JAN-22

```
-- Deve-se mostrar o nome de todos os alunos que agendaram laboratórios na data 02-JAN-22
9 SELECT U.NOME
    FROM USUARIO U INNER JOIN ACESSA A
11
                ON U.MATRICULA = A.MATRICULA INNER JOIN AGENDAMENTO AG
12
            ON A.COD AGEND = AG.COD AGEND
    WHERE AG.DATA INICIO = '02-JAN-22' AND U.MATRICULA IN (
14
            SELECT A.MATRICULA
15
                FROM ALUNO A);
16
17
18
19
20
21
22
```

NOME
Vinicius Monitor



Consulta - Group By/ Having

Mostre os usuários que levaram mais de um 1 visitante

```
21
22
     -- Mostre os usuarios que levaram mais de um 1 visitante
 24 SELECT U.NOME, COUNT(*)
     FROM USUARIO U INNER JOIN VISITANTE V ON U.MATRICULA = V.MATRICULA
 26
     GROUP BY U.NOME
     HAVING COUNT(*) > 1;
 27
 28
 29
 30
 31
 32
 33
 34
 35
 76
```

NOME	COUNT(*)		
Olivia Taylor	3		
Sophie Brown	2		



Consulta - Junção Externa

Mostre o Num_Sala dos laboratórios que nunca tiveram um agendamento

```
-- Mostre o Num_Sala dos laboratorios que nunca tiveram um agendamento

SELECT L.ENDERECO_NUM_SALA
FROM LABORATORIO L LEFT JOIN ACESSA A ON L.COD_LAB = A.COD_LAB

WHERE A.COD_LAB IS NULL;

WHERE A.COD_LAB IS NULL;

43
44
45
46
47
48
49
```

```
ENDERECO_NUM_SALA
Lab PET
```





Consulta - Semi Join

Mostre o nome e email dos usuários que ja levaram algum visitante, não pode fazer a consulta com JOIN

```
49
    -- Mostre o nome e email dos usuários que ja levaram algum visitante, não pode fazer a consulta com JOIN
51 V SELECT U.NOME, U.EMAIL
52 FROM USUARIO U
53 WHERE EXISTS (
54
         SELECT *
55
         FROM VISITANTE V
56
         WHERE V.MATRICULA = U.MATRICULA);
57
58
59
60
61
62
63
      NOME
                             EMATL
 Olivia Parker
                  oliviaparker@outlook.com
                  sophiebrown@gmail.com
 Sophie Brown
                  adamscott@yahoo.com
 Adam Scott
 Olivia Taylor
                  oliviataylor@hotmail.com
                                                      2023 Oracle - Live SQL 23.1.5, running Oracle Database 19c EE Extreme Perf - 19.17.0.0.0 - E
```

cin.ufpe.br



Consultas - Anti Join

 Mostre o nome e o número do crachá dos usuários que nunca levaram algum visitante, não pode fazer a consulta com JOIN, e faça ordenado pelo número

do crachá







Consultas - Subconsulta Escalar

 Mostre o código das peça que tem mais ou um número igual de peças que a média total de todos os tipos de peças

```
77
    -- Mostre o codigo das peça que tem mais ou um numero igual de peças que a media total de todos os tipos de peças
79 , SELECT P.COD PECA
    FROM PECA P
    WHERE P.QTD >= (SELECT AVG(QTD)
        FROM PECA)
82
83
84
85
87
90
 COD_PECA
```



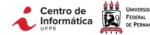


Consultas - Subconsulta Linha

 Mostre a Nome dos alunos que cursam a mesma cadeira e estão no mesmo período do aluno com matrícula 12350

```
90
     -- Mostre a Nome dos alunos que cursam a mesma cadeira e estão no mesmo período do aluno com matricula
     SELECT U.NOME
     FROM ALUNO A, USUARIO U
     WHERE A.MATRICULA = U.MATRICULA AND
 95
         (A.CADEIRA, A.PERIODO) = (
         SELECT A1.CADEIRA, A1.PERIODO
 97
         FROM ALUNO A1
         WHERE A1.MATRICULA = 12350
 99
         AND A.MATRICULA <> 12350
100
101
102
103
104
        NOME
 Fernanda Pascoal
 Miguel Diaz
 Download CSV
```

cin.ufpe.br



Consultas - Subconsulta Tabela

 Mostre a matrícula dos funcionários que começaram e terminaram uma manutenção no mesmo dia do funcionário com matrícula 12356

```
104
     -- Mostre a matricula dos funcionários que começaram e terminaram uma manutenção no mesmo dia do funcionário com matricula 12356
106 . SELECT M1.MATRICULA
     FROM MANUTENCAO M1
     WHERE (M1.DATA INICIO MANUTENCAO, M1.DATA FIM MANUTENCAO) IN (
109
         SELECT M2.DATA INICIO MANUTENCAO, M2.DATA FIM MANUTENCAO
110
         FROM MANUTENCAO M2
111
         WHERE M2.MATRICULA = 12356
112
         AND M1.MATRICULA <> 12356);
113
114
115
116
117
118
  MATRICULA
  12358
  12357
```





Consultas - Operação de conjunto - INTERSECT

Mostre os usuários que já levaram algum visitante, usando intersect

703	Wiles	-	(2.60)	67 69	W W			11.	1000
1 2 _v 3 4 5 6 7	Mostre os SELECT MATRI FROM USU INTERSEC (SELECT FROM VIS	CULA JARIO T MATRICULA	que ja	ı levaram	algum	visitante,	usando	intersect	
M	ATRICULA								
1	2351								
1	2357								
1.	2358								
1:	2359								



Procedimentos

Retorna a quantidade total de Laboratórios

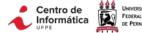
```
-- Retorna a gauntidade total de Laboratórios
 2 V CREATE OR REPLACE
    PROCEDURE
         exibir_qtd_lab
    IS
         qtdLab number;
 7 , BEGIN
         SELECT COUNT(COD_LAB) INTO qtdLab
        FROM LABORATORIO;
10
        DBMS OUTPUT.PUT LINE('Quantidade : '||qtdLab);
11
    END;
12
13 , BEGIN
         exibir qtd lab;
14
15 END;
Statement processed.
Quantidade : 5
```



Retorna qual o período em que um aluno está

```
2 CREATE OR REPLACE FUNCTION
        mostra periodo aluno(codU IN NUMBER)
        RETURN VARCHAR2
    IS
        v periodo VARCHAR2(40);
 7 , BEGIN
 8
        SELECT PERIODO INTO v periodo
 9
        FROM ALUNO
10
        WHERE MATRICULA = codU;
11
12
        RETURN v periodo;
13 , EXCEPTION
14
        WHEN NO_DATA_FOUND THEN
        RETURN 'Aluno não foi encontrado';
15
16
   END;
Statement processed.
Período: 4
```

```
18
19
20
21
22
23
24
25
26
27 DECLARE
        periodo VARCHAR2(40);
29 , BEGIN
        periodo := mostra periodo aluno(12360);
30
        DBMS OUTPUT.PUT LINE('Período: '||periodo);
31
    END;
Statement processed.
Período: 4
```



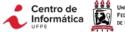
Procedimentos

Altera o email cadastrado de um usuário

```
-- Altera o email cadastrado de um usuário
 2 , CREATE OR REPLACE
    PROCEDURE
         altera_email(
        codU NUMBER,
        novo email USUARIO.EMAIL%TYPE
    IS
 9
    BEGIN
10
        UPDATE USUARIO
        SET EMAIL = novo_email
11
12
        WHERE MATRICULA = codU;
13
        COMMIT;
    END;
15
```

```
15
16 BEGIN
17 altera_email(1234, 'glarocheborba@gmail.com');
18 END;
19
20 SELECT * FROM USUARIO;
```

MATRICULA	NOME	EMAIL	SENHA	NUM_CRACHA
12345	Paulo Vitor	paulo@gmail.com	pass444	6
12346	Fernanda Pascoal	glarocheborba@gmail.com	pass456	2
12347	Gabriel Laroche	glarocheborba@gmail.com	secure789	3



Deve - se mostrar os laboratórios disponíveis.



COD_AGEND	DATA_INICIO	DATA_FIM
1	02-JAN-22	03-JAN-22
2	03-FEB-22	04-FEB-22
3	04-MAR-22	05-MAR-22
4	05-APR-22	06-APR-22

46				£450
47	SELECT *	FROM	LABORATORIO	L;

COD_LAB	ENDERECO_NUM_SALA	ENDERECO_PREDIO	ID_EQUIPA
1	GRAD4	Laboratório 4	=
2	GRAD1	Laboratório 1	9
3	GRAD2	Laboratório 2	7
4	GRAD3	Laboratório 3	3
5	Lab PET	Laboratório do PET	4

cin.ufpe.br



```
CREATE OR REPLACE FUNCTION verificar_laboratorios_disponiveis (
    p data IN DATE
                                  BEGIN
RETURN SYS_REFCURSOR
                                      OPEN c_resultados FOR
IS
                                          SELECT L.cod_lab, L.endereco_num_sala, L.endereco_predio,
    c resultados SYS REFCURSOR;
                                              CASE
                                                  WHEN EXISTS (
                                                      SELECT *
                                                      FROM ACESSA A
                                                      INNER JOIN AGENDAMENTO AG ON A.cod_agend = AG.cod_agend
                                                      WHERE A.cod_lab = L.cod_lab
                                                      AND (
                                                          AG.data_inicio BETWEEN p_data AND p_data+1
                                                          OR AG.data_fim BETWEEN p_data AND p_data+1
                                                          OR p_data BETWEEN AG.data_inicio AND AG.data_fim
                                                          OR p_data+1 BETWEEN AG.data_inicio AND AG.data_fim
                                                  1 THEN ! shoratória indicagnical para a pariada acrosificada !
```



```
)

) THEN 'Laboratório indisponível para o período especificado.'

ELSE 'Laboratório disponível para o período especificado.'

END AS disponibilidade

FROM LABORATORIO L;

RETURN c_resultados;

END;
```





```
DECLARE
    c SYS REFCURSOR;
    v_cod_lab LABORATORIO.cod_lab%TYPE;
    v endereco num sala LABORATORIO.endereco num sala%TYPE;
    v_endereco_predio LABORATORIO.endereco_predio%TYPE;
    v disponibilidade VARCHAR2(100);
BEGIN
    c := verificar laboratorios disponiveis(TO DATE('2022-01-02', 'YYYY-MM-DD'));
    L00P
        FETCH c INTO v_cod_lab, v_endereco_num_sala, v_endereco_predio, v_disponibilidade;
        EXIT WHEN c%NOTFOUND:
        DBMS OUTPUT.PUT LINE
            (v_cod_lab || ' - ' || v_endereco_num_sala || ', ' || v_endereco_predio || ' - ' || v_disponibilidade);
    END LOOP:
    CLOSE c;
ENID .
```



```
35
        v disponibilidade VARCHAR2(100);
36 BEGIN
37
        c := verificar laboratorios disponiveis(TO DATE('2022-01-02', 'YYYY-MM-DD'));
38 .
        L00P
39
            FETCH c INTO v_cod_lab, v_endereco_num_sala, v_endereco_predio, v_disponibilidade;
40
             EXIT WHEN c%NOTFOUND:
41 ~
            DBMS OUTPUT.PUT LINE
                 (v_cod_lab || ' - ' || v_endereco_num_sala || ', ' || v_endereco_predio || ' - ' || v_disponibilidade);
42
43
        END LOOP;
44
        CLOSE c;
45
    END:
```

```
Statement processed.

1 - GRAD4, Laboratório 4 - Laboratório indisponível para o período especificado.

2 - GRAD1, Laboratório 1 - Laboratório disponível para o período especificado.

3 - GRAD2, Laboratório 2 - Laboratório disponível para o período especificado.

4 - GRAD3, Laboratório 3 - Laboratório disponível para o período especificado.

5 - Lab PET, Laboratório do PET - Laboratório disponível para o período especificado.
```

Obrigada(o).



