

PROJETO DE BANCO DE DADOS

Sistema para biblioteca

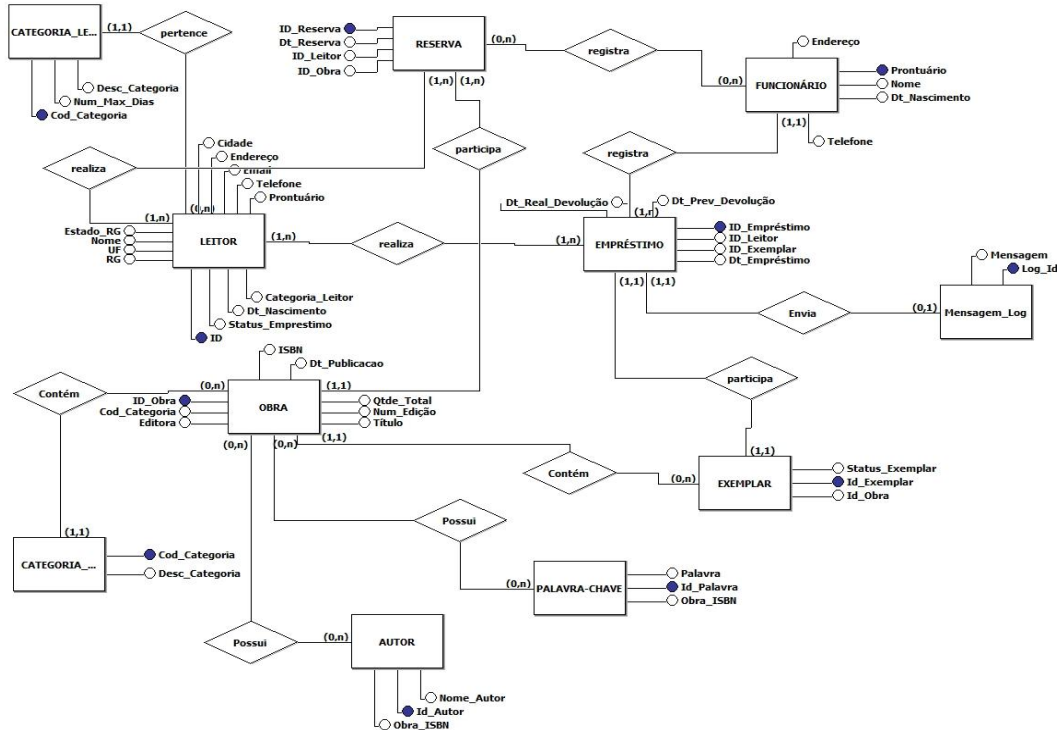
Gabriel Stahlberg Oliveira - 171015x
Marcos A. C. Francischini - 1710338

AGENDA

- Apresentar DER
- Apresentar Modelo Lógico
- Parte de Criação
- Parte de Inserção

- Packages
 - ◆ Procedimento
 - ◆ Relatório
 - ◆ Cat_Leitor
 - ◆ Cat_Obra
 - ◆ Funcionário
 - ◆ Leitor
 - ◆ Obra Literária

Diagrama Entidade Relacionamento(DER)



MODELO LÓGICO

Leitor (Id, Status_Emprestimo, Dt_Nascimento, Cod_Categoria, Prontuario, Telefone, Email, Endereco, Cidade, Estado_RG, Nome, UF, RG);

Cod_Categoria referencia Categoria_Leitor

Nome, Prontuario e Email únicos

Categoria_Leitor (Cod_Categoria, Num_Max_Dias, Desc_Categoria);

Num_Max_Dias checked > 6

Categoria_Obra (Cod_Categoria, Desc_Categoria);

Obra (Id_Obra, ISBN, Dt_Publicacao, Qtde_Total, Num_Edicao, Titulo, Editora, Cod_Categoria);

Cod_Categoria referencia Categoria_Obra

Exemplar (Id_Exemplar, Status_Exemplar, Id_Obra);

Id_Obra referencia Obra

MODELO LÓGICO

Emprestimo (Id_Emprestimo, Id_Leitor, Id_Exemplar, Dt_Emprestimo, Dt_Real_Devolucao, Dt_Prev_Devolucao);
 Id_Leitor referencia Leitor
 Id_Exemplar referencia Exemplar

Funcionario (Prontuario, Telefone, Dt_Nascimento, Nome, Endereco);

Reserva (Id_Reserva, Dt_Reserva, Id_Leitor, Id_Obra);
 Id_Leitor referencia Leitor
 Id_Obra referencia Obra

Autor (Id_Autor, Nome_Autor, Obra_ISBN);

Palavra-chave (Id_Palavra, Palavra, Obra_ISBN);

Mensagem_Log (Log_Id, Mensagem);

PARTE DE CRIAÇÃO

```
create user admin_biblioteca  
identified by adm  
default tablespace users  
quota 10m on users  
temporary tablespace temp;  
  
grant all privileges to admin_biblioteca;
```

PARTE DE CRIAÇÃO

```
create table categoria_leitor(  
    cat_leitor_cod varchar(15) primary key,  
    cat_leitor_desc varchar(256) not null,  
    cat_leitor_max_dias number(2) not null,  
  
    constraint dias_check check(cat_leitor_max_dias > 6)  
);
```

PARTE DE CRIAÇÃO

```
create table leitores(  
    leitor_id number(6) primary key,  
    leitor_prontuario varchar(20), -- Ou deixar um valor default, caso seja usuário externo e não tenha o prontuário.  
    leitor_nome varchar(50) not null,  
    leitor_rg_estado varchar(10) not null,  
    leitor_rg varchar(20) not null,  
    leitor_uf varchar(5),  
    leitor_cidade varchar(100),  
    leitor_endereco varchar(100) not null,  
    leitor_email varchar(50),  
    leitor_telefone varchar(50) not null,  
    leitor_status_emprestimo number(1,0) default 1, -- 1 = TRUE(ATIVO)    0 = FALSE(SUSPENSO)  
    leitor_data_nasc date not null,  
    cat_leitor_cod varchar(15) not null,  
  
    constraint cat_leitor_fk foreign key(cat_leitor_cod) references categoria_leitor(cat_leitor_cod) on delete cascade,  
    constraint nome_unico unique(leitor_nome),  
    constraint email_unico unique(leitor_email),  
    constraint prontuario_unico unique(leitor_prontuario)  
);
```


PARTE DE CRIAÇÃO

```
create sequence leitor_seq
increment by 10
start with 10
maxvalue 100000
nocache
nocycle;

create table categoria_obra(
  cat_obra_cod varchar(10) primary key,
  cat_obra_desc varchar(256) not null
);

create table obras(
  obra_id number(6) primary key,
  obra_isbn varchar(20) not null,
  obra_editora varchar(50) not null,
  obra_titulo varchar(50) not null,
  obra_num_edicao number,
  obra_qtde_total number not null,
  cat_obra_cod varchar(10) not null,

  constraint cat_obra_fk foreign key(cat_obra_cod)
    references categoria_obra(cat_obra_cod) on delete cascade
);

create sequence obra_seq
increment by 1
start with 1
maxvalue 100000
nocache
nocycle;
```

PARTE DE CRIAÇÃO

```
start with 1
maxvalue 100000
nocache
nocycle;

create table autores(
  autor_nro number primary key,
  autor_nome varchar(256) not null,
  obra_isbn varchar(20) not null
);

create sequence autores_seq
increment by 1
start with 1
maxvalue 100000
nocache
nocycle;

create table palavras_chave(
  palavra_id number primary key,
  palavra varchar(100) not null,
  obra_isbn varchar(20) not null
);

create sequence palavras_chave_seq
increment by 1
start with 1
maxvalue 100000
nocache
nocycle;
```

PARTE DE CRIAÇÃO

```
create table exemplar(  
    exemplar_id number primary key,  
    exemplar_status varchar(20) default 'Disponivel',  
    obra_id number(6) not null,  
  
    constraint obra_id_fk foreign key(obra_id) references obras(obra_id) on delete cascade  
);  
  
create sequence exemplar_seq  
increment by 1  
start with 1  
maxvalue 100000  
nocache  
nocycle;  
  
create table funcionario(  
    func_prontuario varchar(20) primary key,  
    func_endereco varchar(100) not null,  
    func_nome varchar(50) not null,  
    func_data_nasc date not null,  
    func_telefone varchar(50)  
);
```

PARTE DE CRIAÇÃO

```
create table reserva(  
    reserva_id number(6) primary key,  
    reserva_data date not null,  
    leitor_id number(6) not null,  
    obra_id number(6) not null,  
    func_prontuario varchar(20) not null,  
  
    foreign key(leitor_id) references leitores(leitor_id) on delete cascade,  
    foreign key(obra_id) references obras(obra_id) on delete cascade,  
    foreign key(func_prontuario) references funcionario(func_prontuario) on delete cascade  
);  
  
create sequence reserva_seq  
increment by 1  
start with 1  
maxvalue 100000  
nocache  
nocycle;
```

PARTE DE CRIAÇÃO

```
create or replace trigger atualiza_status
  after insert on emprestimo
  for each row
declare

begin
  case
    when inserting then
      update exemplar
      set exemplar_status = 'Indisponivel'
      where exemplar_id = :NEW.exemplar_id;
  end case;
end;
/
```

PARTE DE CRIAÇÃO

```
create or replace view atrasados_view as
  select l.leitor_nome, l.leitor_telefone, l.leitor_email, em.emp_data, em.emp_data_prev_dev
  from emprestimo em
  inner join leitores l
  on l.leitor_id = em.leitor_id
  where em.emp_data_prev_dev < sysdate
  and em.emp_data_real_dev is null
  with read only;
```

PARTE DE INSERÇÃO

```
insert into categoria_leitor(cat_leitor_cod, cat_leitor_desc, cat_leitor_max_dias)
values ('ALU_TEC_GRAD', 'Aluno de curso técnico e de graduação', 15);

insert into categoria_leitor(cat_leitor_cod, cat_leitor_desc, cat_leitor_max_dias)
values ('ALU_POS', 'Aluno de pós-graduação', 30);

insert into categoria_leitor(cat_leitor_cod, cat_leitor_desc, cat_leitor_max_dias)
values ('PROF', 'Professor', 30);

insert into categoria_leitor(cat_leitor_cod, cat_leitor_desc, cat_leitor_max_dias)
values ('FUNC', 'Funcionário', 30);

insert into categoria_leitor(cat_leitor_cod, cat_leitor_desc, cat_leitor_max_dias)
values ('USR_EXT', 'Usuário externo', 7);

-- Inserir em Leitor

insert into leitores
(leitor_id, leitor_prontuario, leitor_nome, leitor_rg_estado, leitor_rg, leitor_uf, leitor_cidade, leitor_endereco, leitor_email,
leitor_telefone, leitor_status_emprestimo, leitor_data_nasc, cat_leitor_cod)
values(leitor_seq.nextval, '1710111', 'Gabriel', 'SP', '11.111.111-1', 'SP', 'Araraquara', 'Rua Um, 1, Jardim Primeiro', 'ga@gmail.com',
'(16)11111111', 1, '02/09/1998', 'ALU_TEC_GRAD');

insert into leitores
(leitor_id, leitor_prontuario, leitor_nome, leitor_rg_estado, leitor_rg, leitor_uf, leitor_cidade, leitor_endereco, leitor_email,
leitor_telefone, leitor_status_emprestimo, leitor_data_nasc, cat_leitor_cod)
values(leitor_seq.nextval, '1711222', 'Marcos', 'SP', '22.222.222-2', 'SP', 'Rincão', 'Rua Dois, 2, Jardim Secundo', 'ma@gmail.com',
'(16)22222222', 1, '21/04/1998', 'ALU_POS');

insert into leitores
(leitor_id, leitor_prontuario, leitor_nome, leitor_rg_estado, leitor_rg, leitor_uf, leitor_cidade, leitor_endereco, leitor_email,
leitor_telefone, leitor_status_emprestimo, leitor_data_nasc, cat_leitor_cod)
values(leitor_seq.nextval, '1713333', 'Dênis', 'SC', '33.333.333-3', 'SC', 'Florianópolis', 'Rua Três, 3, Jardim Terceiro', 'de@ifsp.edu.br',
'(16)33333333', 1, '11/06/1985', 'PROF');

insert into leitores
(leitor_id, leitor_prontuario, leitor_nome, leitor_rg_estado, leitor_rg, leitor_uf, leitor_cidade, leitor_endereco, leitor_email,
leitor_telefone, leitor_status_emprestimo, leitor_data_nasc, cat_leitor_cod)
values(leitor_seq.nextval, '1714421', 'João', 'RJ', '44.444.444-4', 'RJ', 'Niterói', 'Rua Quatro, 4, Jardim Quarto', 'jo@ifsp.edu.br',
'(16)44448757', 1, '13/08/1992', 'FUNC');

insert into leitores
(leitor_id, leitor_nome, leitor_rg_estado, leitor_rg, leitor_uf, leitor_cidade, leitor_endereco, leitor_email,
leitor_telefone, leitor_status_emprestimo, leitor_data_nasc, cat_leitor_cod)
values(leitor_seq.nextval, 'Joana', 'MG', '55.555.555-5', 'MG', 'Belo Horizonte', 'Rua Cinco, 5, Jardim Quinto', 'joa@ifsp.edu.br',
'(16)58742255', 1, '28/09/1990', 'USR_EXT');
```


PARTE DE INSERÇÃO

```
-- Inserir em Categoria_obra

insert into categoria_obra(cat_obra_cod, cat_obra_desc) values('ROMANC', 'Romance');
insert into categoria_obra(cat_obra_cod, cat_obra_desc) values('CT_FADAS', 'Conto de Fadas');
insert into categoria_obra(cat_obra_cod, cat_obra_desc) values('CURIO', 'Curiosidades');
insert into categoria_obra(cat_obra_cod, cat_obra_desc) values('PROG', 'Programação');
insert into categoria_obra(cat_obra_cod, cat_obra_desc) values('BD', 'Banco de Dados');

-- Inserir em palavras_chave

insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'romance', '147-74-147-0554-1');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'fada', '258-85-258-8888-2');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'conto', '258-85-258-8888-2');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'curiosidades', '369-96-369-9999-3');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'guia', '369-96-369-9999-3');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'programação', '985-58-558-0000-8');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'logica', '985-58-558-0000-8');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'banco', '020-11-332-2012-0');
insert into palavras_chave(palavra_id, palavra, obra_isbn) values(PALAVRAS_CHAVE_SEQ.nextval, 'dados', '020-11-332-2012-0');

-- Inserir em autores

insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Miguel de Cervantes', '147-74-147-0554-1');
insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Charles Perrault', '258-85-258-8888-2');
insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Marcelo Duarte', '369-96-369-9999-3');
insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Marcelo Gomes', '985-58-558-0000-8');
insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Ricardo Concilio', '985-58-558-0000-8');
insert into autores(autor_nro, autor_nome, obra_isbn) values(autores_seq.nextval, 'Carlos A. Heuser', '020-11-332-2012-0');

-- Inserir em obra

insert into obras(obra_id, obra_isbn, obra_editora, obra_titulo, obra_num_edicao, obra_qtde_total, cat_obra_cod)
values(obra_seq.nextval, '147-74-147-0554-1', 'Francisco de Robles', 'Dom Quixote', 102, 10, 'ROMANC');

insert into obras(obra_id, obra_isbn, obra_editora, obra_titulo, obra_num_edicao, obra_qtde_total, cat_obra_cod)
values(obra_seq.nextval, '258-85-258-8888-2', 'Ática', 'O Pequeno Polegar', 201, 10, 'CT_FADAS');

insert into obras(obra_id, obra_isbn, obra_editora, obra_titulo, obra_num_edicao, obra_qtde_total, cat_obra_cod)
values(obra_seq.nextval, '369-96-369-9999-3', 'CIA das Letras', 'O guia dos curiosos', 505, 10, 'CURIO');

insert into obras(obra_id, obra_isbn, obra_editora, obra_titulo, obra_num_edicao, obra_qtde_total, cat_obra_cod)
values(obra_seq.nextval, '985-58-558-0000-8', 'CENGAGE Learning', 'Algoritmos e lógica de programação', 404, 10, 'PROG');

insert into obras(obra_id, obra_isbn, obra_editora, obra_titulo, obra_num_edicao, obra_qtde_total, cat_obra_cod)
values(obra_seq.nextval, '020-11-332-2012-0', 'BooksMark', 'Projeto de Banco de Dados', 999, 10, 'BD');
```


PACKAGE PROCEDIMENTO

- **PROCEDURE: efetuar_emprestimo(p_leitor_id number, p_pront_func varchar, p_exemplar_id number)**
- ◆ Verifica o status do leitor (ativo / inativo);
 - ◆ Verifica a quantidade máxima que esse tipo de leitor pode ficar com o exemplar;
 - ◆ A partir dessa quantidade, calcula-se a data prevista para devolução;
 - ◆ Verifica a existência do funcionário;
 - ◆ Recupera o id da obra correspondente ao exemplar solicitado;
 - ◆ Verifica se o leitor havia reservado essa obra e se sim, se o período de 3 dias não foi expirado. Caso contrário, verifica se a o exemplar está disponível para empréstimo;
 - ◆ Registra o empréstimo ou notifica caso haja algum impedimento.

PACKAGE PROCEDIMENTO

- **PROCEDURE: `efetuar_reserva(p_obra_id number, p_leitor_id number, p_pront_func varchar);`**
- ◆ Verifica a possibilidade de ter algum exemplar dessa obra disponível. Caso tenha, informa-o(a) que é possível fazer o empréstimo diretamente;
 - ◆ Verifica se há alguma reserva que tenha expirado os 3 dias de prazo para a realização do empréstimo. Caso tenha, torna o exemplar disponível;
 - ◆ Caso uma das obras que tenha expirado o prazo seja a obra correspondente ao exemplar que deseja-se reservar, efetua-se o empréstimo automaticamente;
 - ◆ Registra a reserva ou notifica caso haja algum impedimento.

PACKAGE PROCEDIMENTO

- **PROCEDURE: efetuar_devolucao(p_exemplar_id number);**
- ◆ Verifica se a devolução está sendo feita dentro do prazo estipulado. Caso não, o leitor torna-se inativo para novos empréstimos e/ou reservas até que regularize sua situação;
 - ◆ Caso essa devolução seja para regularizar a situação do leitor, o mesmo torna-se ativo para empréstimos novamente;
 - ◆ Verifica se a obra correspondente a esse exemplar possui alguma reserva;
 - ◆ Caso tenha a reserva, é enviada uma mensagem ao leitor que efetuou a reserva, avisando-o de que possui 3 dias para realizar o empréstimo do exemplar;
 - ◆ Caso não tenha nenhuma reserva, o exemplar torna-se novamente disponível.

PACKAGE RELATÓRIO

```
create or replace package relatorio as
    procedure obras_emprestadas;
    procedure obras_atrasadas;
    procedure obras_reservadas;
    procedure historico_leitor(p_leitor_id number);
    procedure consulta_obra(p_obra_id number);

    invalid_id exception;
end relatorio;
/
```

PACKAGE RELATÓRIO

```
procedure obras_emprestadas is
  cursor emp_c is
    select l.leitor_nome, o.obra_titulo, e.exemplar_id, em.emp_data, em.emp_data_prev_dev
    from emprestimo em
    inner join leitores l
    on l.leitor_id = em.leitor_id
    inner join exemplar e
    on e.exemplar_id = em.exemplar_id
    inner join obras o
    on o.obra_id = em.obra_id
    where e.exemplar_status != 'Disponivel'
    and em.emp_data_real_dev is null
    order by o.cat_obra_cod;
  emp_rec emp_c%rowtype;
begin
  dbms_output.put_line('***** LISTA DE OBRAS EMPRESTADAS *****');
  dbms_output.put_line('');
  open emp_c;
  loop
    fetch emp_c into emp_rec;
    exit when emp_c%notfound;

    dbms_output.put_line('Nome do leitor: ' || emp_rec.leitor_nome);
    dbms_output.put_line('Título da obra: ' || emp_rec.obra_titulo);
    dbms_output.put_line('ID do exemplar: ' || emp_rec.exemplar_id);
    dbms_output.put_line('Data do empréstimo: ' || emp_rec.emp_data);
    dbms_output.put_line('Data prevista para devolução: ' || emp_rec.emp_data_prev_dev);
    dbms_output.put_line('-----');

  end loop;
  close emp_c;
exception
  when NO_DATA_FOUND then
    dbms_output.put_line('Todos os exemplares de todas as obras estão disponíveis');
  when others then
    dbms_output.put_line(SQLCODE);
    dbms_output.put_line(SQLERRM);
end obras_emprestadas;
```

PACKAGE CAT_LEITOR

```
create or replace package categ_leitor as
  procedure cadastrar_categ_leitor(cat_leitor_cod categoria_leitor.cat_leitor_cod%type,
                                   cat_leitor_desc categoria_leitor.cat_leitor_desc%type,
                                   cat_leitor_max_dias categoria_leitor.cat_leitor_max_dias%type);

  procedure atualizar_categ_leitor_desc(cat_leit_cod categoria_leitor.cat_leitor_cod%type,
                                       cat_leit_desc categoria_leitor.cat_leitor_desc%type);

  procedure excluir_categ_leitor(cat_leit_cod categoria_leitor.cat_leitor_cod%type);

  invalid_cat_leitor_cod exception;
end categ_leitor;
/
```

PACKAGE CAT_OBRA

```
create or replace package categ_obra as
  procedure cadastrar_cat_obra(cat_ob_cod categoria_obra.cat_obra_cod%type,
                                cat_ob_desc categoria_obra.cat_obra_desc%type);

  procedure atualizar_cat_obra_desc(cat_ob_cod categoria_obra.cat_obra_cod%type,
                                     cat_ob_desc categoria_obra.cat_obra_desc%type);

  procedure excluir_cat_obra(cat_ob_cod categoria_obra.cat_obra_cod%type);

  invalid_cat_obra_cod exception;
end categ_obra;
/
```

PACKAGE FUNCIONÁRIO

```
create or replace package func as
    procedure cadastrar_funcionario(func_prontuario funcionario.func_prontuario%type,
                                     func_endereco funcionario.func_endereco%type,
                                     func_nome funcionario.func_nome%type,
                                     func_dt_nasc funcionario.func_data_nasc%type,
                                     func_telefone funcionario.func_telefone%type);

    procedure atualizar_funcionario_tel(func_pront funcionario.func_prontuario%type,
                                       func_tel funcionario.func_telefone%type);

    procedure excluir_funcionario(func_pront funcionario.func_prontuario%type);

    invalid_func_prontuario exception;
end func;
/
```


PACKAGE LEITOR

```
create or replace package leitor as
  procedure cadastrar_leitor(leitor_pront leitores.leitor_prontuario%type,
                             leitor_nome leitores.leitor_nome%type,
                             leitor_rg_est leitores.leitor_rg_estado%type,
                             leitor_rg leitores.leitor_rg%type,
                             leitor_uf leitores.leitor_uf%type,
                             leitor_cidade leitores.leitor_cidade%type,
                             leitor_ender leitores.leitor_endereco%type,
                             leitor_email leitores.leitor_email%type,
                             leitor_tel leitores.leitor_telefone%type,
                             leitor_st_emp leitores.leitor_status_emprestimo%type,
                             leitor_dt_nasc leitores.leitor_data_nasc%type,
                             cat_leitor_cod leitores.cat_leitor_cod%type);

  procedure atualizar_leitor_email(leitorid leitores.leitor_id%type,
                                  leitor_em leitores.leitor_email%type);

  procedure atualizar_leitor_telefone(leitorid leitores.leitor_id%type,
                                      leitor_tel leitores.leitor_telefone%type);

  procedure excluir_leitor(leitorid leitores.leitor_id%type);

  invalid_leitor_id exception;
end leitor;
/
```

PACKAGE OBRA_LITERÁRIA

```
create or replace package obra_literaria as
  procedure cadastrar_obra(obra_isbn obras.obra_isbn%type,
                           obra_editora obras.obra_editora%type,
                           obra_titulo obras.obra_titulo%type,
                           obra_num_edicao obras.obra_num_edicao%type,
                           obra_qtde_total obras.obra_qtde_total%type,
                           cat_obra_cod obras.cat_obra_cod%type);

  procedure atualizar_qtde_total_obras(obraid obras.obra_id%type,
                                       qtde_nova obras.obra_qtde_total%type);

  procedure excluir_obra(obraid obras.obra_id%type);

  invalid_obra_id exception;
end obra_literaria;
/
```

OBRIGADO!