

Documento de Dados do Projeto

Sistema de gerenciamento de vendas de livraria

Giovanna Nucci Matos
Ruan Campos Silva

Sumário

1.	Descrição do Sistema	3
2.	Modelagem Lógica do Banco de Dados Transacional	4
3.	Modelagem Relacional do Banco de Dados Transacional	5
4.	Dicionário de Dados	6
5.	Tabelas Temporais	10
5.1.	Modelo Relacional para Tabela de Historiamento	10
5.1.1.	HFUNCIONARIOS	10
5.2.	Triggers, Stored Procedures e Functions de historiamento	10
6.	Tabelas de Auditoria	12
6.1.	Modelo Relacional para Tabela de Auditoria	12
6.2.	Triggers, Stored Procedures e Functions de Auditoria	12
6.2.1.	Triggers	12
6.2.2.	Procedure	19
7.	Tabela Stage	20
7.1.	Tabela Stage	20
7.2.	Rotina para Armazenamento de Dados na Stage	21
8.	Queries de Informações Estratégicas	22
8.1.	Vendedor que vendeu menos livros	22
8.2.	Vendedor que vendeu mais livros	22
8.3.	Os três livros mais vendidos	23
8.4.	Os três livros menos vendidos	23
8.5.	Dia da semana que possui mais vendas	24
9.	Queries para Gráficos de Análise	25
9.1.	Categoria vendida por faixa-etária	25
9.2.	Quantidade de livros vendidos por mês	26
9.3.	Quantidade de livros vendidos por autor	27
9.4.	Total de vendas por dia da semana	28
9.5.	Lucro total de cada mês	29

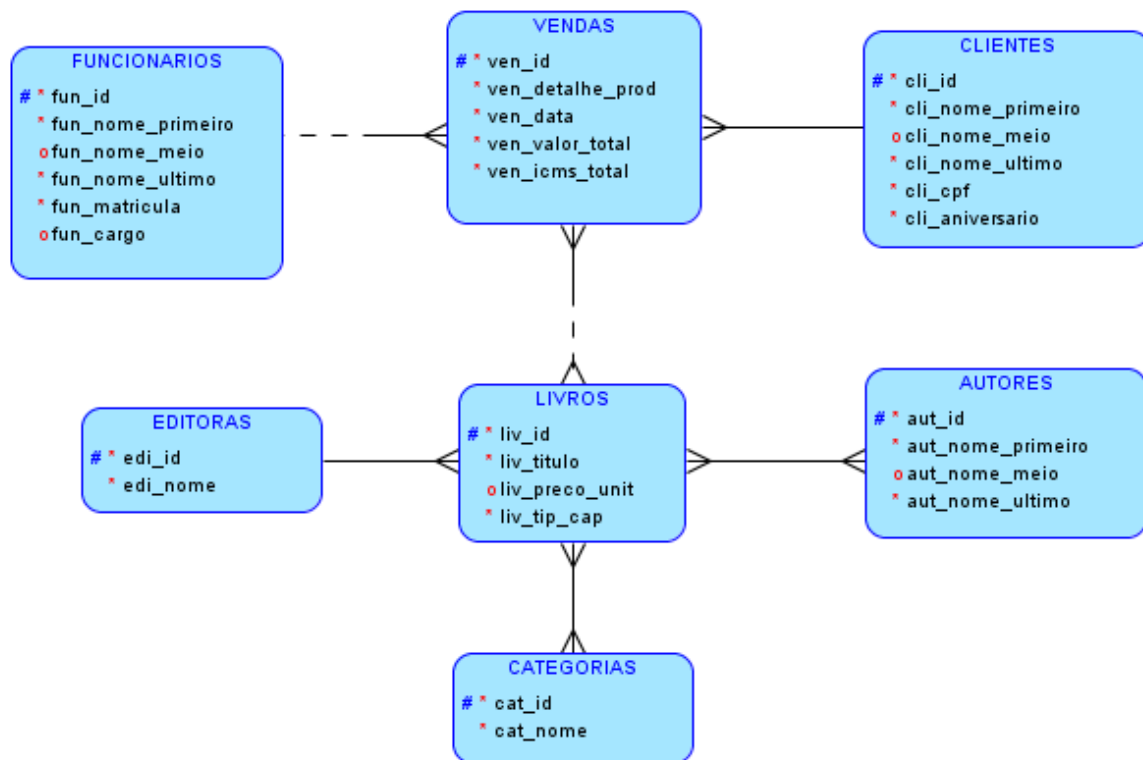
1. Descrição do Sistema

Sistema de gerenciamento de venda de livros de varejo, responsável por gerar registros de venda dos produtos.

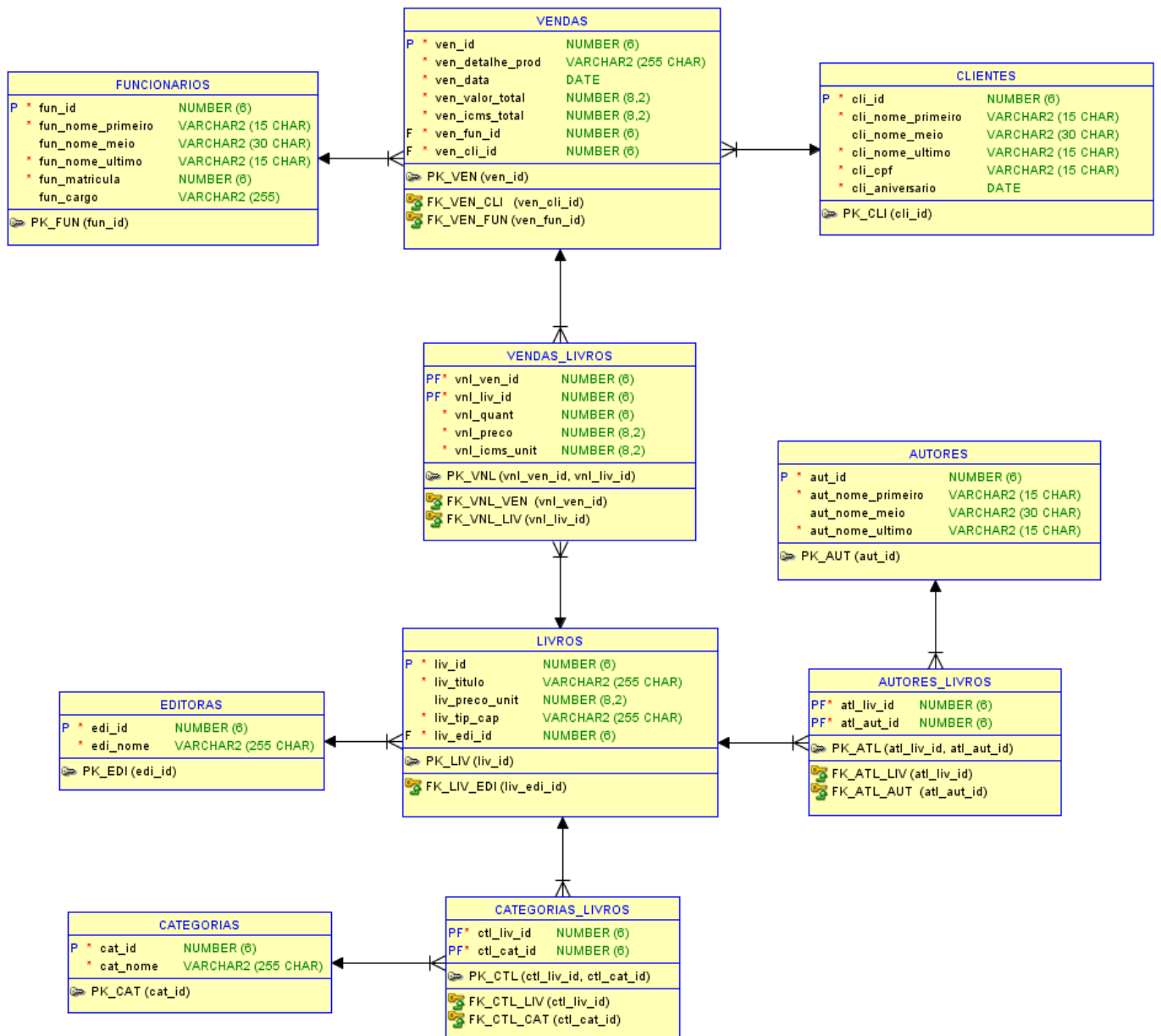
Esses registros contêm os detalhes do produto, data de venda, cpf ou cnpj do comprador, valor final da venda, valor do ICMS de cada produto e ICMS total.

Além disso, existe um cadastro de funcionários para identificar aqueles que realizaram as vendas.

2. Modelagem Lógica do Banco de Dados Transacional



3. Modelagem Relacional do Banco de Dados Transacional



4. Dicionário de Dados

Tabela CLIENTES:

Tabela com os dados dos clientes.

Coluna	Tipo (tamanho)	Constraints	Descrição
cli_id	Number(6,0)	PK_CLI referenciando CLIENTES(cli_id), CK_CLI_NN_01	Coluna identificadora da tabela clientes
cli_nome_primeiro	Varchar2(15)	CK_CLI_NN_02	Coluna com o primeiro nome do cliente
cli_nome_meio	Varchar2(30)	-	Coluna opcional com o nome do meio do cliente
cli_nome_ultimo	Varchar2(15)	CK_CLI_NN_03	Coluna com o último nome do cliente
cli_cpf	Varchar2(15)	CK_CLI_NN_04	Coluna com o CPF do cliente
cli_aniversario	Varchar2(11)	CK_CLI_NN_05	Coluna com a data de aniversário do cliente

Tabela FUNCIONARIOS:

Tabela com os dados dos funcionários.

Coluna	Tipo (tamanho)	Constraints	Descrição
fun_id	Number(6,0)	PK_FUN referenciando FUNCIONARIOS(fun_id), CK_FUN_NN_01	Coluna identificadora da tabela funcionarios
fun_nome_primeiro	Varchar2(15)	CK_FUN_NN_02	Coluna com o primeiro nome do funcionário
fun_nome_meio	Varchar2(30)	-	Coluna opcional com o nome do meio do funcionário
fun_nome_ultimo	Varchar2(15)	CK_FUN_NN_03	Coluna com o último nome do funcionário
fun_matricula	Number(6,0)	CK_FUN_NN_04	Coluna com o número da matrícula do funcionário
fun_cargo	Varchar2(255)	CK_FUN_NN_05	Coluna com o nome do cargo do funcionário

Tabela CATEGORIAS:

Tabela com os dados de categorias.

Coluna	Tipo (tamanho)	Constraints	Descrição
cat_id	Number(6,0)	PK_CAT referenciando CATEGORIAS(cat_id), CK_CAT_NN_01	Coluna identificadora da tabela categorias
cat_nome	Varchar2(255)	CK_CAT_NN_02	Coluna com o nome da categoria

Tabela EDITORAS:

Tabela com os dados das editoras.

Coluna	Tipo (tamanho)	Constraints	Descrição
edi_id	Number(6,0)	PK_EDIT referenciando EDITORAS(edi_id), CK_EDIT_NN_01	Coluna identificadora da tabela editoras
edi_nome	Varchar2(255)	CK_EDIT_NN_02	Coluna com o nome da editora

Tabela AUTORES:

Tabela com os dados dos autores.

Coluna	Tipo (tamanho)	Constraints	Descrição
aut_id	Number(6,0)	PK_AUT referenciando AUTORES(aut_id), CK_AUT_NN_01	Coluna identificadora da tabela autores
aut_nome_primeiro	Varchar2(15)	CK_AUT_NN_02	Coluna com o primeiro nome do autor
aut_nome_meio	Varchar2(30)	-	Coluna opcional com o nome do meio do autor
aut_nome_ultimo	Varchar2(15)	CK_AUT_NN_03	Coluna com o último nome do autor

Tabela LIVROS:

Tabela com os dados de livros.

Coluna	Tipo (tamanho)	Constraints	Descrição
liv_id	Number(6,0)	PK_LIV referenciando LIVROS(liv_id), CK_LIV_NN_01	Coluna identificadora da tabela livros
liv_titulo	Varchar2(255)	CK_LIV_NN_02	Coluna com o nome do título do livro
liv_preco_unit	Number(8,2)	CK_LIV_NN_03	Coluna com o preço unitário do livro

liv_tip_cap	Varchar2(255)	CK_LIV_NN_04	Coluna com o tipo da capa do livro
liv_edi_id	Number(6,0)	FK_LIV_EDI referenciando EDITORAS(edi_id), CK_LIV_NN_05	Coluna com a chave estrangeira da tabela editoras

Tabela VENDAS:

Tabela com os dados das vendas.

Coluna	Tipo (tamanho)	Constraints	Descrição
ven_id	Number(6,0)	PK_VEN referenciando VENDAS(ven_id), CK_VEN_NN_01	Coluna identificadora da tabela vendas
ven_detalhe_prod	Varchar2(255)	CK_VEN_NN_02	Coluna com os detalhes dos produtos contidos na venda
ven_data	Varchar2(11)	CK_VEN_NN_03	Coluna com a data em que foi realizada a venda
ven_valor_total	Number(8,2)	CK_VEN_NN_04	Coluna com o valor total da venda
ven_icms_total	Number(8,2)	CK_VEN_NN_05	Coluna o valor total do ICMS dos produtos contidos na venda
ven_fun_id	Number(6,0)	FK_VEN_FUN referenciando FUNCIONARIOS(fun_id), CK_VEN_NN_06	Coluna com a chave estrangeira da tabela funcionarios
ven_cli_id	Number(6,0)	FK_VEN_CLI referenciando CLIENTES(cli_id), CK_VEN_NN_07	Coluna com a chave estrangeira da tabela clientes

Tabela CATEGORIAS LIVROS:

Tabela com as chaves estrangeiras das tabelas autores e livros.

Coluna	Tipo (tamanho)	Constraints	Descrição
ctl_liv_id	Number(6,0)	PK_CTL referenciando CATEGORIAS_LIVROS(ctl_liv_id, ctl_aut_id), FK_CTL_LIV referenciando LIVROS(liv_id), CK_CTL_NN_01	Coluna com a chave estrangeira da tabela livros
ctl_cat_id	Number(6,0)	PK_CTL referenciando CATEGORIAS_LIVROS (ctl_liv_id, ctl_aut_id), FK_CTL_CAT referenciando CATEGORIAS(cat_id), CK_CTL_NN_02	Coluna com a chave estrangeira da tabela categorias

Tabela AUTORES LIVROS:

Tabela com as chaves estrangeiras das tabelas autores e livros.

Coluna	Tipo (tamanho)	Constraints	Descrição
atl_liv_id	Number(6,0)	PK_ATL referenciando AUTORES_LIVROS(atl_liv_id, atl_aut_id), FK_ATL_LIV referenciando LIVROS(liv_id), CK_ATL_NN_01	Coluna com a chave estrangeira da tabela livros
atl_aut_id	Number(6,0)	PK_ATL referenciando AUTORES_LIVROS(atl_liv_id, atl_aut_id), FK_ATL_AUT referenciando AUTORES(aut_id), CK_ATL_NN_02	Coluna com a chave estrangeira da tabela autores

Tabela VENDAS LIVROS:

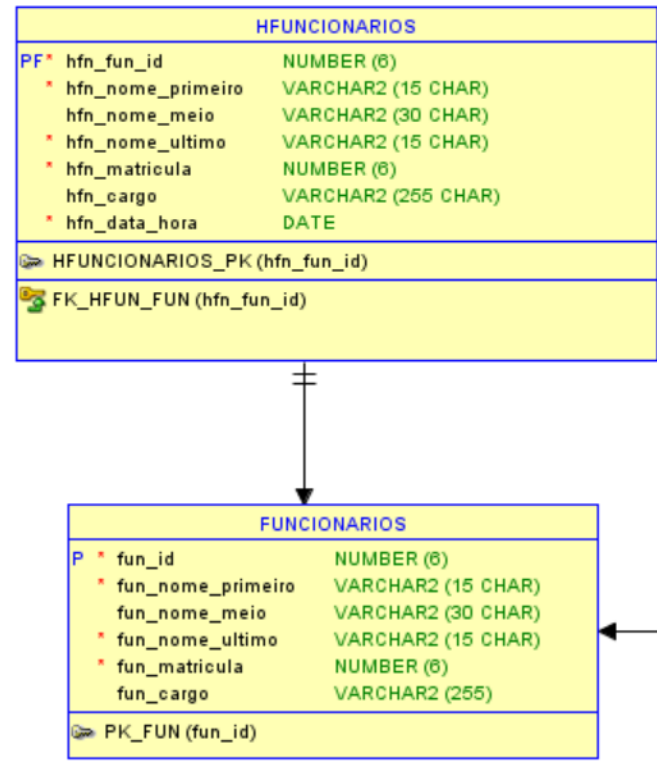
Tabela com as chaves estrangeiras das tabelas livros e vendas.

Coluna	Tipo (tamanho)	Constraints	Descrição
vnl_ven_id	Number(6,0)	PK_VNL referenciando VENDAS_LIVROS(vnl_ven_id, vnl_liv_id), FK_VNL_VEN referenciando VENDAS(ven_id), CK_VNL_NN_01	Coluna com a chave estrangeira da tabela vendas
vnl_liv_id	Number(6,0)	PK_VNL referenciando VENDAS_LIVROS(vnl_ven_id, vnl_liv_id), FK_VNL_LIV referenciando LIVROS(liv_id), CK_VNL_NN_02	Coluna com a chave estrangeira da tabela livros
vnl_quant	Number(6,0)	CK_VNL_NN_03	Coluna com a quantidade de produtos vendidos naquela venda
vnl_preco	Number(8,2)	CK_VNL_NN_04	Coluna com o preço da venda
vnl_icms_unit	Number(8,2)	CK_VNL_NN_05	Coluna com o valor do ICMS de cada produto daquela venda

5. Tabelas Temporais

5.1. Modelo Relacional para Tabela de Historiamento

5.1.1. HFUNCIONARIOS



5.2. Triggers, Stored Procedures e Functions de historiamento

Tabela

```
CREATE TABLE hfuncionarios (  
    hfn_fun_id          NUMBER(6),  
    hfn_nome_primeiro  VARCHAR2(15 CHAR),  
    hfn_nome_meio      VARCHAR2(30 CHAR),  
    hfn_nome_ultimo    VARCHAR2(15 CHAR),  
    hfn_matricula       NUMBER(6),  
    hfn_cargo          VARCHAR2(255),  
    hfn_data_hora       DATE  
);
```

Trigger

```
CREATE TRIGGER TG_HFN BEFORE UPDATE OR DELETE ON FUNCIONARIOS
FOR EACH ROW
BEGIN
    declare data date;

    select SYSDATE
    into data
    from dual;

    INSERT INTO HFUNCIONARIOS
VALUES(:OLD.FUN_ID, :OLD.FUN_NOME_PRIMEIRO, :OLD.FUN_NOME_MEIO, :OLD.FUN_NOME_ULTIMO, :OLD.FUN_MATRICULA, :OLD.FUN_CARGO, SYSDATE);
END;
/
```

6. Tabelas de Auditoria

6.1. Modelo Relacional para Tabela de Auditoria

AUDITORIA		
P	* aud_id	NUMBER (6)
	* aud_dt_registro	DATE
	* aud_user	VARCHAR2 (30 CHAR)
	* aud_tp_acao	VARCHAR2 (6 CHAR)
	* aud_nm_tabela	VARCHAR2 (30 CHAR)
	* aud_id_linha	VARCHAR2 (30 CHAR)
➡ AUDITORIA_PK (aud_id)		

6.2. Triggers, Stored Procedures e Functions de Auditoria

6.2.1. Triggers

ID Auditoria: TG_SEQ_AUD

```
alter table auditoria add constraint aud_pk primary key (aud_id);
create sequence SEQ_AUD nocycle nocache;

create trigger TG_SEQ_AUD before insert on auditoria for each row
begin
    :new.aud_id := SEQ_AUD.nextval;
end;
/
```

Clientes: TG_AUD_CLI

```
create trigger TG_AUD_CLI after insert or delete or update on clientes
for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
```

```

        proc_inserere_audit(v_usuario, 'INSERT', 'CLIENTE',
to_char(:new.cli_id));

        elsif updating then

            proc_inserere_audit(v_usuario, 'UPDATE', 'CLIENTE',
to_char(:new.cli_id));

            else

                proc_inserere_audit(v_usuario, 'DELETE', 'CLIENTE',
to_char(:old.cli_id));

            end if;

        end;

    /

```

Funcionarios: TG_AUD_FUN

```

create trigger TG_AUD_FUN after insert or delete or update on
funcionarios for each row

declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;

    if inserting then
        proc_inserere_audit(v_usuario, 'INSERT', 'FUNCIONARIOS',
to_char(:new.fun_id));
    elsif updating then
        proc_inserere_audit(v_usuario, 'UPDATE', 'FUNCIONARIOS',
to_char(:new.fun_id));
    else
        proc_inserere_audit(v_usuario, 'DELETE', 'FUNCIONARIOS',
to_char(:old.fun_id));
    end if;

end;

/

```

Categorias: TG_AUD_CAT

```
create trigger TG_AUD_CAT after insert or delete or update on
categorias for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
        proc_insere_audit(v_usuario, 'INSERT', 'CATEGORIAS',
to_char(:new.cat_id));
    elsif updating then
        proc_insere_audit(v_usuario, 'UPDATE', 'CATEGORIAS',
to_char(:new.cat_id));
    else
        proc_insere_audit(v_usuario, 'DELETE', 'CATEGORIAS',
to_char(:old.cat_id));
    end if;
end;
/
```

Editoras: TG_AUD_EDI

```
create trigger TG_AUD_EDI after insert or delete or update on editoras
for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
        proc_insere_audit(v_usuario, 'INSERT', 'EDITORAS',
to_char(:new.edi_id));
    elsif updating then
```

```

        proc_inserere_audit(v_usuario, 'UPDATE', 'EDITORAS',
to_char(:new.edi_id));

        else

        proc_inserere_audit(v_usuario, 'DELETE', 'EDITORAS',
to_char(:old.edi_id));

        end if;

end;

/

```

Autores: TG_AUD_AUT

```

create trigger TG_AUD_AUT after insert or delete or update on autores
for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;

    if inserting then
        proc_inserere_audit(v_usuario, 'INSERT', 'AUTORES',
to_char(:new.aut_id));
    elsif updating then
        proc_inserere_audit(v_usuario, 'UPDATE', 'AUTORES',
to_char(:new.aut_id));
    else
        proc_inserere_audit(v_usuario, 'DELETE', 'AUTORES',
to_char(:old.aut_id));
    end if;

end;

/

```

Livros: TG_AUD_LIV

```

create trigger TG_AUD_LIV after insert or delete or update on livros
for each row
declare

```

```

        v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
        proc_inserere_audit(v_usuario, 'INSERT', 'LIVROS',
to_char(:new.liv_id));
    elsif updating then
        proc_inserere_audit(v_usuario, 'UPDATE', 'LIVROS',
to_char(:new.liv_id));
    else
        proc_inserere_audit(v_usuario, 'DELETE', 'LIVROS',
to_char(:old.liv_id));
    end if;
end;
/

```

Vendas: TG_AUD_VEN

```

create trigger TG_AUD_VEN after insert or delete or update on vendas
for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
        proc_inserere_audit(v_usuario, 'INSERT', 'VENDAS',
to_char(:new.ven_id));
    elsif updating then
        proc_inserere_audit(v_usuario, 'UPDATE', 'VENDAS',
to_char(:new.ven_id));
    else
        proc_inserere_audit(v_usuario, 'DELETE', 'VENDAS',
to_char(:old.ven_id));
    end if;
end;

```



```
        end if;
    end;
/
```

Categorias_livros: TG_AUD_CTL

```
create trigger TG_AUD_CTL after insert or delete or update on
categorias_livros for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
    from dual;
    if inserting then
        proc_insere_audit(v_usuario, 'INSERT', 'CATEGORIAS_LIVROS',
to_char(:new.ctl_cat_id) || ',' || to_char(:new.ctl_liv_id));
    elsif updating then
        proc_insere_audit(v_usuario, 'UPDATE', 'CATEGORIAS_LIVROS',
to_char(:new.ctl_cat_id) || ',' || to_char(:new.ctl_liv_id));
    else
        proc_insere_audit(v_usuario, 'DELETE', 'CATEGORIAS_LIVROS',
to_char(:old.ctl_cat_id) || ',' || to_char(:old.ctl_liv_id));
    end if;
end;
/
```

Autores_livros: TG_AUD_ATL

```
create trigger TG_AUD_ATL after insert or delete or update on
autores_livros for each row
declare
    v_usuario varchar(30);
begin
    select user
    into v_usuario
```

```

        from dual;

        if inserting then

            proc_inserere_audit(v_usuario, 'INSERT', 'AUTORES_LIVROS',
to_char(:new.atl_aut_id) || ',' || to_char(:new.atl_liv_id));

            elsif updating then

                proc_inserere_audit(v_usuario, 'UPDATE', 'AUTORES_LIVROS',
to_char(:new.atl_aut_id) || ',' || to_char(:new.atl_liv_id));

            else

                proc_inserere_audit(v_usuario, 'DELETE', 'AUTORES_LIVROS',
to_char(:old.atl_aut_id) || ',' || to_char(:old.atl_liv_id));

            end if;

        end;

    /

```

Vendas_livros: TG_AUD_VNL

```

create trigger TG_AUD_VNL after insert or delete or update on
vendas_livros for each row

declare

    v_usuario varchar(30);

begin

    select user
    into v_usuario
    from dual;

    if inserting then

        proc_inserere_audit(v_usuario, 'INSERT', 'VENDAS_LIVROS',
to_char(:new.vnl_ven_id) || ',' || to_char(:new.vnl_liv_id));

        elsif updating then

            proc_inserere_audit(v_usuario, 'UPDATE', 'VENDAS_LIVROS',
to_char(:new.vnl_ven_id) || ',' || to_char(:new.vnl_liv_id));

        else

            proc_inserere_audit(v_usuario, 'DELETE', 'VENDAS_LIVROS',
to_char(:old.vnl_ven_id) || ',' || to_char(:old.vnl_liv_id));

        end if;

    end;

    /

```

6.2.2. Procedure

Inserção na tabela Auditoria: PROC_INSERE_AUDIT

```
create procedure proc_insere_audit
    (nm_usuario IN varchar, tp_acao IN varchar, nm_tabela IN
    varchar, id_linha IN varchar)
IS
BEGIN
    insert into auditoria values (0, sysdate, nm_usuario, tp_acao,
    nm_tabela, id_linha);
    END proc_insere_audit;
/
```

7. Tabela Stage

7.1. Tabela Stage

Tabela STAGE VENDAS:

Tabela Stage com os dados das vendas realizadas.

Coluna	Tipo (tamanho)	Constraints	Descrição
stg_id_ven	Number(6,0)	PK_STG referenciando STAGE_VENDAS(stg_id_ven, stg_id_liv), CK_STG_NN_01	Coluna identificadora com o id da venda
stg_id_liv	Number(6,0)	PK_STG referenciando STAGE_VENDAS(stg_id_ven, stg_id_liv), CK_STG_NN_02	Coluna identificadora com o id do livro vendido
stg_data	Varchar2(11)	CK_STG_NN_03	Coluna com a data da venda
stg_nome_liv	Varchar2(100)	CK_STG_NN_04	Coluna com o nome do livro vendido
stg_nome_aut	Varchar2(50)	CK_STG_NN_05	Coluna com o nome do autor do livro vendido
stg_nome_edi	Varchar2(50)	CK_STG_NN_06	Coluna com o nome da editora do livro vendido
stg_nome_cat	Varchar2(50)	CK_STG_NN_07	Coluna com o nome da categoria do livro vendido
stg_valor_total	Number(8,2)	CK_STG_NN_08	Coluna com o valor total da compra
stg_icms_total	Number(8,2)	CK_STG_NN_09	Coluna com o valor total do ICMS da compra
stg_preco_unit	Number(8,2)	CK_STG_NN_10	Coluna com o valor unitário do livro vendido
stg_icms_unit	Number(8,2)	CK_STG_NN_11	Coluna com o ICMS unitário do livro vendido
stg_porcentagem_valor	Number(6,2)	CK_STG_NN_12	Coluna com a porcentagem relativa do livro vendido em relação ao valor total da compra
stg_quant_prod	Number(3,0)	CK_STG_NN_13	Coluna com a quantidade de livros presentes na compra
stg_id_fun	Number(6,0)	CK_STG_NN_14	Coluna com o id do funcionário que realizou a venda
stg_id_cli	Number(6,0)	CK_STG_NN_15	Coluna com o id do cliente que efetuou a compra

7.2. Rotina para Armazenamento de Dados na Stage

```
CREATE PROCEDURE PR_CARGA_STAGE
AS
BEGIN
    INSERT INTO STAGE_VENDAS(
        SELECT
            ven_id, liv_id, to_char(ven_data,'dd-mm-yyyy'),
            liv_titulo, aut_nome_primeiro||' '||aut_nome_ultimo,
            edi_nome, cat_nome, ven_valor_total,
            ven_icms_total, vnl_preco, vnl_icms_unit,
            TRUNC((vnl_preco * 100) / ven_valor_total,2),
            vnl_quant, ven_fun_id, ven_cli_id
        FROM
            vendas INNER JOIN vendas_livros ON ven_id = vnl_ven_id
            INNER JOIN livros ON vnl_liv_id = liv_id
            INNER JOIN editoras ON liv_edi_id = edi_id
            INNER JOIN autores_livros ON liv_id = atl_liv_id
            INNER JOIN autores ON atl_aut_id = aut_id
            INNER JOIN categorias_livros ON liv_id = ctl_liv_id
            INNER JOIN categorias ON ctl_cat_id = cat_id
    );
END;
EXECUTE PR_CARGA_STAGE();
```

8. Queries de Informações Estratégicas

8.1. Vendedor que vendeu menos livros

Autor: Giovanna Nucci Matos

SQL:

```
select
    b.fun_nome_primeiro || ' ' || b.fun_nome_ultimo
  Funcionário, count(a.stg_id_ven) Quantidade
from
    stage_vendas a
inner join
    funcionarios b on a.stg_id_fun = b.fun_id
group by
    b.fun_nome_primeiro || ' ' || b.fun_nome_ultimo
order by 2
fetch first 1 row only;
```

Resultado:

	FUNCIONÁRIO	QUANTIDADE
1	Nilce Leon	1

8.2. Vendedor que vendeu mais livros

Autor: Giovanna Nucci Matos

SQL:

```
select
    b.fun_nome_primeiro || ' ' || b.fun_nome_ultimo
  Funcionário, count(a.stg_id_ven) Quantidade
from
    stage_vendas a
inner join funcionarios b on a.stg_id_fun = b.fun_id
group by
    b.fun_nome_primeiro || ' ' || b.fun_nome_ultimo
order by 2 desc
fetch first 1 row only;
```

Resultado:

	FUNCIONÁRIO	QUANTIDADE
1	Bobby Lee	5

8.3. Os três livros mais vendidos

Autor: Giovanna Nucci Matos

SQL:

```
select
    stg_nome_liv Livro, sum(stg_quant_prod) "QUANTIDADE
VENDIDA"
from
    stage_vendas
group by
    stg_nome_liv
order by 2 desc
fetch first 3 rows only;
```

Resultado:

	LIVRO	QUANTIDADE VENDIDA
1	Hunter x Hunter - vol.01	5
2	Guia do Mochileiro das Galáxias	5
3	Evangalion - Edição Especial	4

8.4. Os três livros menos vendidos

Autor: Ruan Campos Silva

SQL:

```
select
    stg_nome_liv Livro, sum(stg_quant_prod) "QUANTIDADE
VENDIDA"
from
    stage_vendas
group by
    stg_nome_liv
order by 2
fetch first 3 rows only;
```

Resultado:

	LIVRO	QUANTIDADE VENDIDA
1	Questões ENEM e FUVEST 2020	1
2	Assassinato no Expresso do Oriente	1
3	JAVA para Leigos	1

8.5. Dia da semana que possui mais vendas

Autor: Ruan Campos Silva

SQL:

```
select
    to_char(to_date(stg_data), 'Day') "DIA DA SEMANA",
    count(stg_id_ven) "QUANTIDADE DE VENDAS"
from
    stage_vendas
group by
    to_char(to_date(stg_data), 'Day')
order by 2 desc
fetch first 2 row only;
```

Resultado:

	DIA DA SEMANA	QUANTIDADE DE VENDAS
1	Sexta-Feira	4
2	Quarta-Feira	4

9. Queries para Gráficos de Análise

9.1. Categoria vendida por faixa-etária

Autor: Giovanna Nucci Matos

SQL:

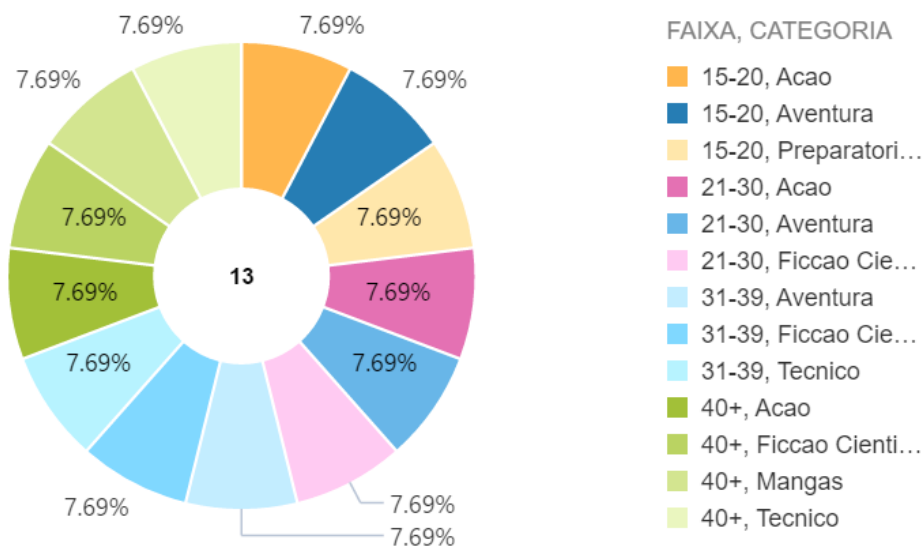
```
select
    a.stg_nome_cat Categoria,
    (CASE WHEN trunc((sysdate- to_date(b.cli_aniversario)
)/365.25) between 15 and 20 then '15-20'
        WHEN trunc((sysdate- to_date(b.cli_aniversario) )/365.25)
between 21 and 30 then '21-30'
        WHEN trunc((sysdate- to_date(b.cli_aniversario) )/365.25)
between 31 and 40 then '31-39'
        else '40+' end) as faixa, count(a.stg_id_ven) "QUANTIDADE
DE LIVROS VENDIDOS"
from
    stage_vendas a
inner join
    clientes b on a.stg_id_cli = b.cli_id
group by
    a.stg_nome_cat, (CASE WHEN trunc((sysdate-
to_date(b.cli_aniversario) )/365.25) between 15 and 20 then
'15-20'
        WHEN trunc((sysdate- to_date(b.cli_aniversario) )/365.25)
between 21 and 30 then '21-30'
        WHEN trunc((sysdate- to_date(b.cli_aniversario) )/365.25)
between 31 and 40 then '31-39'
        else '40+' end)
order by 2;
```

Resultado:

CATEGORIA	FAIXA	QUANTIDADE DE LIVROS VENDIDOS
1 Aventura	15-20	3
2 Ação	15-20	1
3 Preparatório Vestibular - Concurso Público	15-20	2
4 Aventura	21-30	2
5 Ficção Científica	21-30	1
6 Ação	21-30	1
7 Ficção Científica	31-39	1
8 Técnico	31-39	1
9 Aventura	31-39	1
10 Ação	40+	1
11 Técnico	40+	1
12 Ficção Científica	40+	4
13 Mangás	40+	1

Gráfico:

CATEGORIA VENDIDA POR FAIXA-ETÁRIA



9.2. Quantidade de livros vendidos por mês

Autor: Giovanna Nucci Matos

SQL:

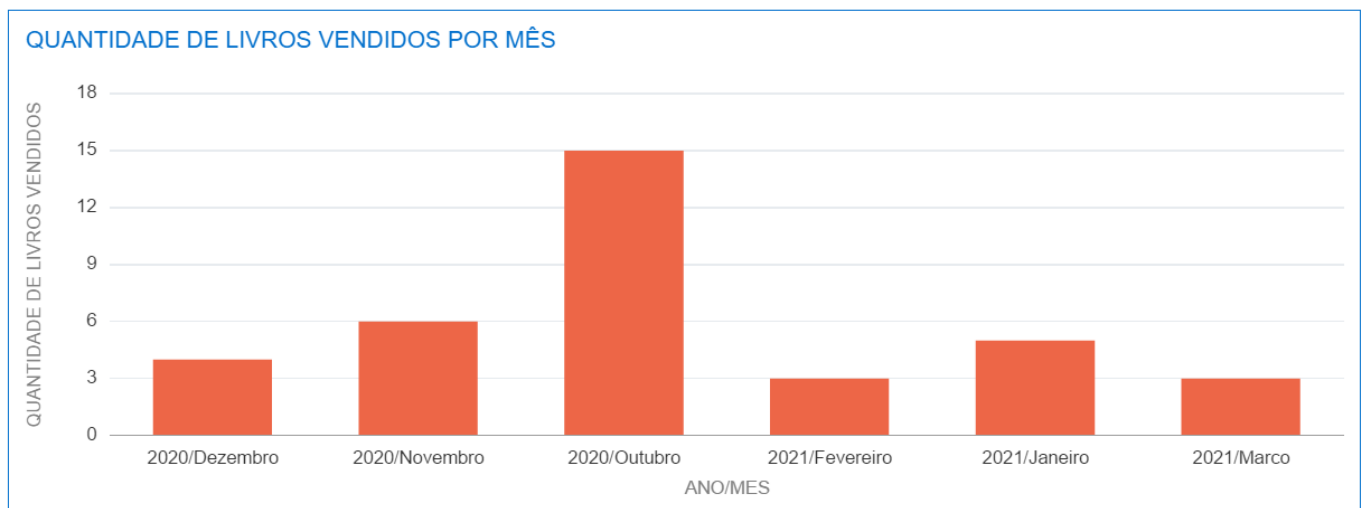
```
select
    to_char(to_date(stg_data), 'yyyy/Month') "ANO/MÊS",
    sum(stg_quant_prod) "QUANTIDADE DE LIVROS VENDIDOS"
from
    stage_vendas
group by
    to_char(to_date(stg_data), 'yyyy/Month')
```

```
order by to_date(to_char(to_date(stg_data), 'yyyy/Month'), 'yyyy-mm');
```

Resultado:

ANO/MES	QUANTIDADE DE LIVROS VENDIDOS
1 2020/Outubro	19
2 2020/Novembro	6
3 2020/Dezembro	4
4 2021/Janeiro	5
5 2021/Fevereiro	3
6 2021/Março	3

Gráfico:



9.3. Quantidade de livros vendidos por autor

Autor: Ruan Campos Silva

SQL:

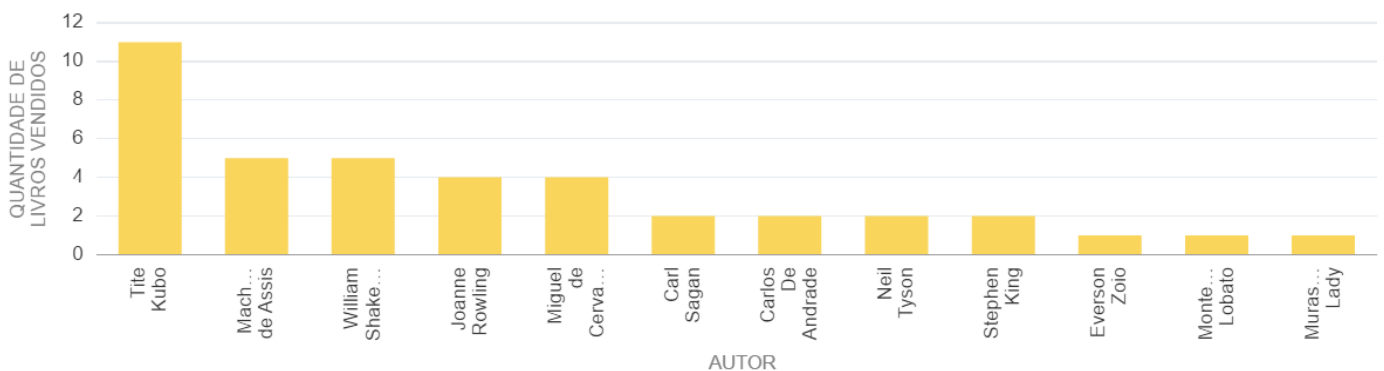
```
select
    stg_nome_aut Autor, sum(stg_quant_prod) "QUANTIDADE DE
LIVROS VENDIDOS"
from
    stage_vendas
group by
    stg_nome_aut
order by 2;
```

Resultado:

	AUTOR	QUANTIDADE DE LIVROS VENDIDOS
1	Everson Zoio	1
2	Monteiro Lobato	1
3	Murasaki Lady	1
4	Carl Sagan	2
5	Stephen King	2
6	Neil Tyson	2
7	Carlos De Andrade	2
8	Joanne Rowling	4
9	Miguel de Cervantes	4
10	William Shakespeare	5
11	Machado de Assis	5
12	Tite Kubo	11

Gráfico:

QUANTIDADE DE LIVROS VENDIDOS POR AUTOR



9.4. Total de vendas por dia da semana

Autor: Ruan Campos Silva

SQL:

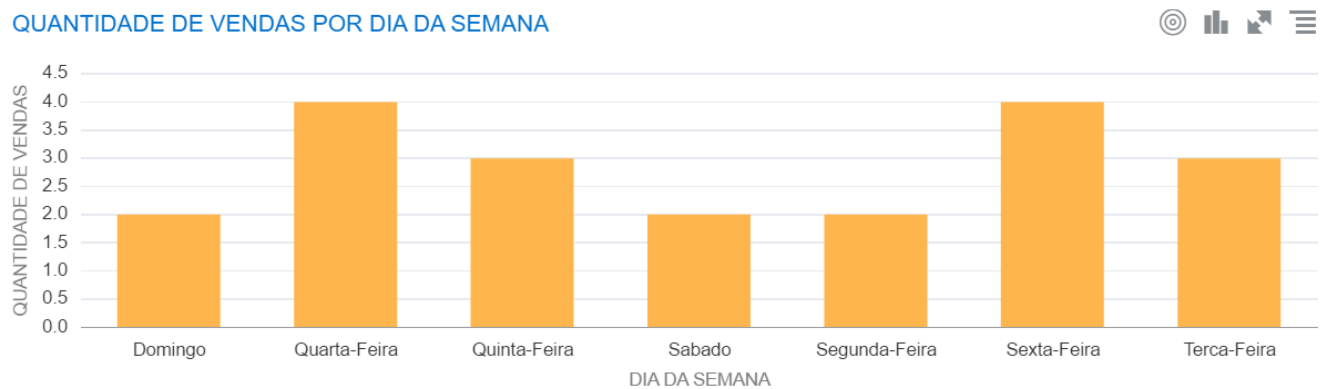
```
select
    to_char(to_date(stg_data), 'Day') "DIA DA SEMANA",
    count(stg_id_ven) "QUANTIDADE DE VENDAS"
from
    stage_vendas
group by
    to_char(to_date(stg_data), 'Day')
order by 1;
```

Resultado:

	DIA DA SEMANA	QUANTIDADE DE VENDAS
1	Domingo	2
2	Quarta-Feira	4
3	Quinta-Feira	3
4	Sábado	2
5	Segunda-Feira	2
6	Sexta-Feira	4
7	Terça-Feira	3

Gráfico:

QUANTIDADE DE VENDAS POR DIA DA SEMANA



9.5. Lucro total de cada mês

Autor: Giovanna Nucci Matos e Ruan Campos Silva

SQL:

```
select
    to_char(to_date(stg_data), 'yyyy/Month') Ano,
    to_char(sum(stg_valor_total), 'L999999.00') "LUCRO TOTAL"
from
    stage_vendas
group by
    to_char(to_date(stg_data), 'yyyy/Month')
order by
    to_date(to_char(to_date(stg_data), 'yyyy/Month'), 'yyyy-mm');
```

Resultado:

	ANO/MES	LUCRO TOTAL
1	2020/Outubro	R\$ 780.12
2	2020/Novembro	R\$ 289.90
3	2020/Dezembro	R\$ 282.08
4	2021/Janeiro	R\$ 295.51
5	2021/Fevereiro	R\$ 128.10
6	2021/Março	R\$ 419.45

Gráfico:

LUCRO TOTAL DE CADA MÊS

