TRABALHO PRÁTICO

PROVA DE VINHOS

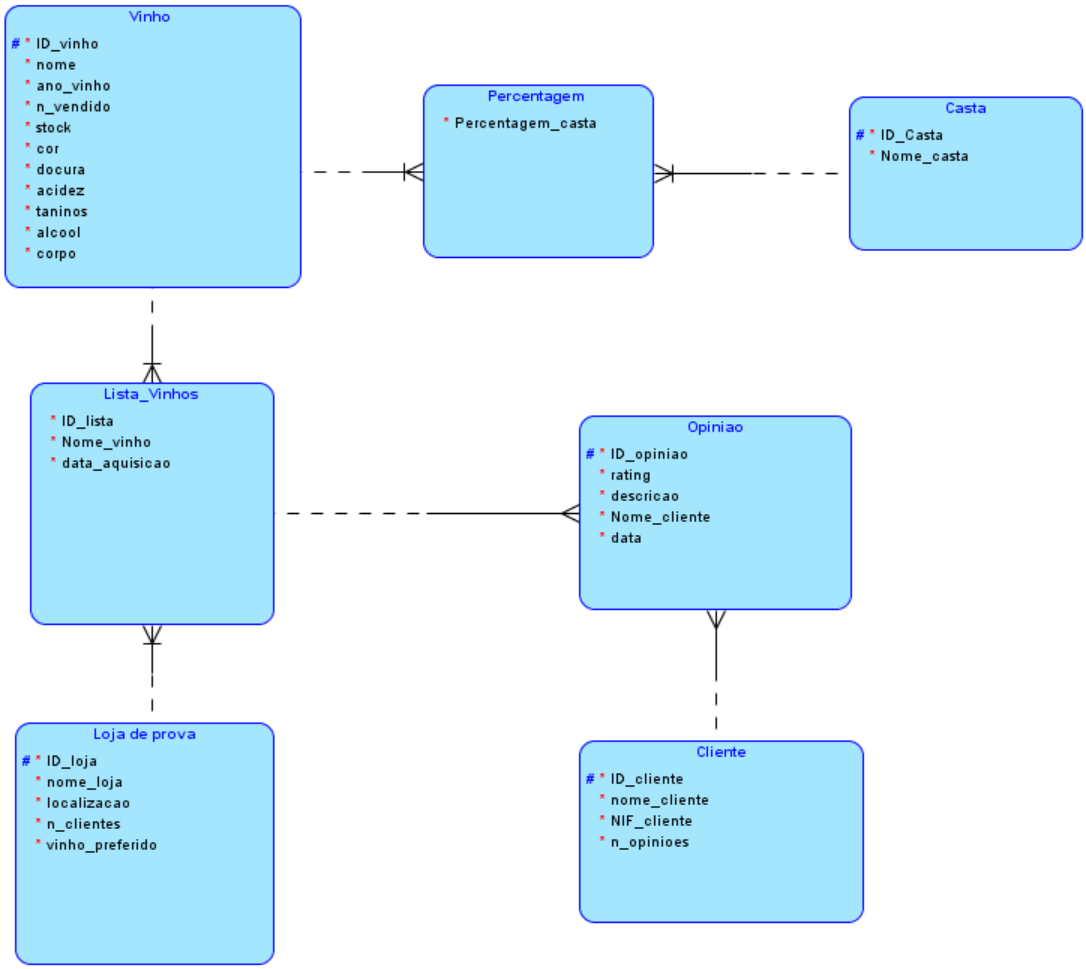
Curso(s):	Engenharia Informática
Unidade Curricular:	Base de Dados I
Ano Letivo:	2021/2022
Docente:	José Carlos Fonseca
Aluno(s):	Luís Barros 1700331
	Diogo Fernandes 1703638
	João Costa 1703762
Data:	14-FEV-2022

ÍNDICE

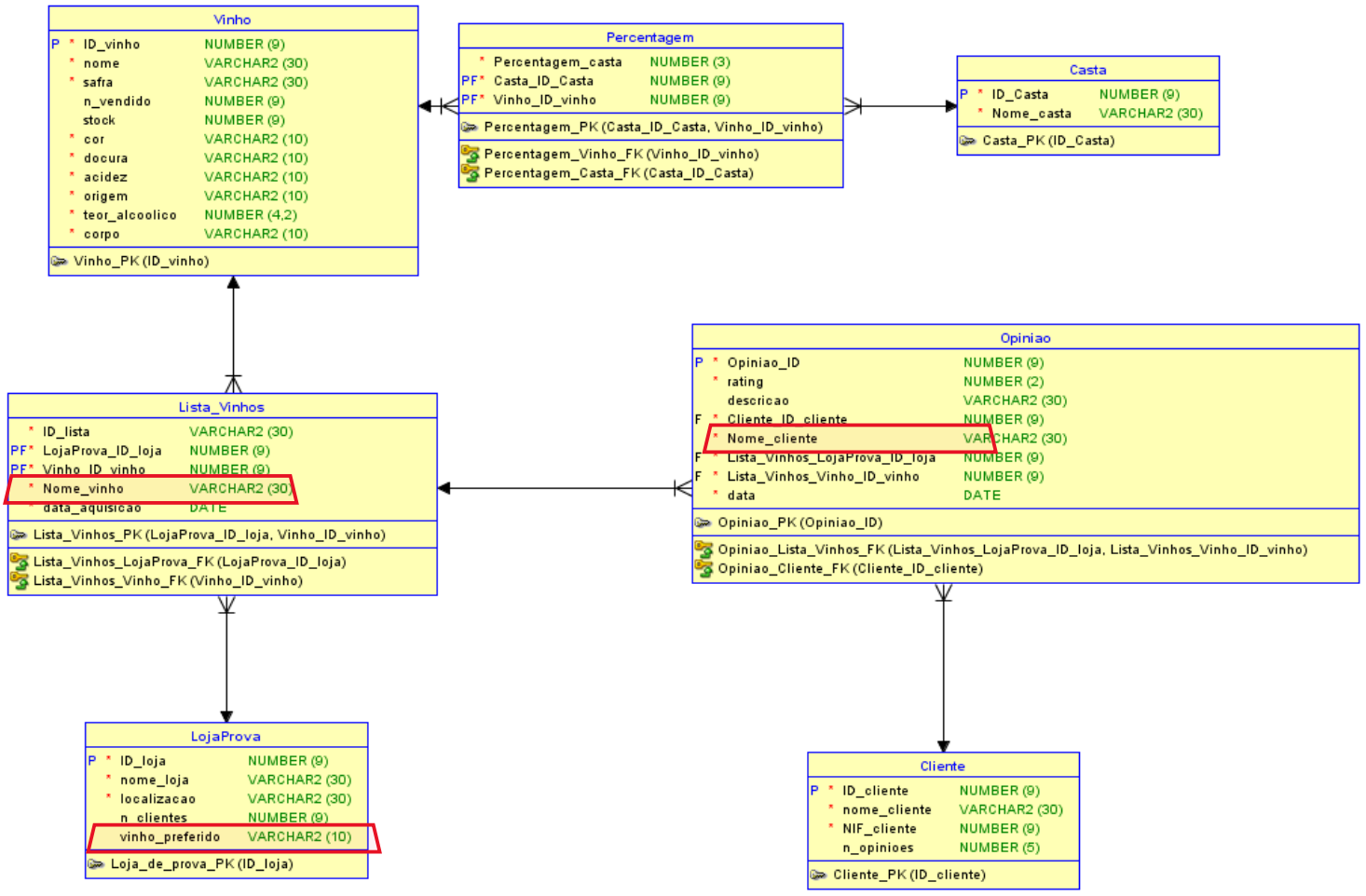
INTRODUÇÃO

Neste documento vai ser apresentado a modulação de uma base de dados para uso de comerciantes proprietários de lojas de vinhos.
O objetivo deste produto é facilitar a gestão de vinhos, e classificar os mesmos, caracterizando-os e registando as opiniões dos clientes sobre os mesmos.
Nesta base de dados vamos ter acesso aos vinhos, as características dos mesmos, as castas usadas na produção dos vinhos, aos clientes das diversas lojas de prova e às opiniões que os clientes fornecem de determinados vinhos consumidos.

MODELO LÓGICO



MODELO RELACIONAL



DICIONÁRIO DE DADOS

Cliente: Pessoa que compra ou prova vinhos na loja de prova

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Cliente	ID_cliente	Numérico	9	Nº de identificação do cliente	-	PK	Sim	Sim
	Nome Cliente	Caracter	30	Nome do cliente	-	-	Sim	-
	NIF_cliente	Numérico	9	Nº de identificação fiscal do Cliente	-	-	Sim	-
	N_opinioes	Numérico	5	Nº de opiniões dado pelo cliente	-	-	Sim	-

Opinião: Opinião dada pelo cliente de um determinado vinho

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório
Opinião	ID_opinião	Numérico	9	Nº de identificação da opinião	-	PK	Sim
	Rating	Numérico	2	Rating dado pelo cliente	-	-	Sim
	Descrição	Caracter	30	Descrição dada pelo cliente	-	-	Não
	Cliente id_cliente	Numérico	9	Nº de identificação do cliente	-	FK	Sim
	Nome_cliente	Caracter	30	Nome do cliente	-	-	Sim
	Lista_Vinhos_LojaProva_ID_Loja	Numérico	9	Nº de identificação da loja de prova	-	FK	Sim
	Lista_vinhos_Vinho_ID_Vinho	Numérico	9	Nº de identificação do vinho	-	FK	Sim
	Data	Data	-	Data em que a opinião foi dada	-	-	Sim

Lista Vinhos: Lista dos vinhos disponíveis para compra ou prova na loja de prova

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Lista_Vinhos	ID_lista	Caracter	30	Nº de identificação de lista	-	-	Sim	Sim
	LojaProva_ID_loja	Numérico	9	Nº de identificação da loja de prova	-	PF	Sim	Sim
	Vinho_ID_vinho	Numérico	9	Nº de identificação do vinho	-	PF	Sim	Sim
	Nome_vinho	Caracter	30	Nome do vinho	-	-	Sim	Não
	Data_aquisição	Data	-	Data em que o vinho foi adquirido pela loja de prova	-	-	Sim	Não

Vinho: Vinhos possíveis de serem adquiridos pela loja de prova

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Vinho	Id_vinho	Numérico	9	Nº de identificação do vinho	-	PK	Sim	Sim
	Nome	Caracter	30	Nome do vinho	-	-	Sim	Não
	Safra	Caracter	30	Ano de colheita do vinho	-	-	Sim	Não
	N_vendido	Numérico	9	Nº de unidades vendidas	-	-	Não	Não
	Stock	Numérico	9	Nº de unidades disponíveis	-	-	Não	Não
	Cor	Caracter	10	Cor do vinho (Característica)	-	-	Sim	Não
	Docura	Caracter	10	Doçura do vinho (Característica)	-	-	Sim	Não
	Acidez	Caracter	10	Acidez do vinho (Característica)	-	-	Sim	Não
	Origem	Caracter	10	Local de origem de colheita e produção do vinho	-	-	Sim	Não
	Teor_alcoolico	Numérico	4.2	Teor alcoólico do vinho	-	-	Sim	Não
	Corpo	Caracter	10	Corpo do vinho (Característica)	-	-	Sim	Não

Percentagem: Percentagem de casta utilizada na produção de cada vinho

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Percentagem	Percentagem_casta	Numérico	3	Percentagem da casta utilizada para produzir o vinho	-	-	Sim	Não
	Casta_ID_Casta	Numérico	9	Nº de identificação da Casta	-	PF	Sim	Não
	Vinho_ID_vinho	Numérico	9	Nº de identificação do vinho	-	PF	Sim	Não

Casta: Castas ou tipos de uva utilizadas na produção dos vinhos

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Casta	ID_Casta	Numérico	9	Nº de identificação da casta	-	PK	Sim	Sim
	Nome_casta	Caracter	30	Nome da casta	-	-	Sim	Não

Loja de Prova: Apenas uma loja de prova, que será o local de compra e prova de vinhos

Tabela	Atributos	Tipo de Dados	Tamanho	Descrição	Restrições	Chave	Obrigatório	Único
Loja Prova	ID_loja	Numérico	9	Nº de identificação da loja	-	PK	Sim	Sim
	Nome_loja	Caracter	30	Nome da loja	-	-	Sim	Sim
	Localização	Caracter	30	Localização da loja	-	-	Sim	Sim
	N_clientes	Numérico	9	Nº de clientes com ficha na loja	-	-	Não	Sim
	Vinho_preferido	Caracter	10	Vinho preferido dos clientes	-	-	Não	Sim

DESNORMALIZAÇÕES E ANOMALIAS

Por Luís Barros: Primeira desnormalização

Na primeira desnormalização (por coluna derivada) adicionámos coluna vinho preferido na tabela loja de provas.

Insert na tabela Vinhos: Quando feito um insert na tabela Vinho é necessário inserir o nº vendido do mesmo, logo é necessário fazer "recálculo" de qual o vinho preferido.

Update na tabela Vinhos: Quando feito um update na tabela Vinho é necessário, um update no vinho preferido.

Delete na tabela Vinhos: Quando feito um delete na tabela Vinhos é necessário um update no vinho preferido.

Insert na tabela Loja de prova: Quando feito um insert na tabela loja de prova, é necessário que o vinho preferido seja o vinho cujo nº vendido seja o mais elevado de todos na tabela Vinho.

Update na tabela Loja de Prova: Quando feito um update na loja de prova**,** o vinho preferido terá de ser o vinho cujo nº vendido seja o maior.

Delete na tabela Loja de Prova: No nosso caso só existe uma loja de prova, logo na prática nunca se iria proceder ao delete das informações da mesma.

Por João Costa: Segunda desnormalização

Na segunda desnormalização (por coluna redundante) adicionámos a coluna nome de cliente na tabela opinião.

Insert na tabela Opinião: Quando inserida um registo na tabela Opinião o nome do cliente deve ser exatamente igual ao da coluna na tabela Cliente.

Delete na tabela Opinião: Quando feito delete na tabela Opinião, o n de opiniões da tabela cliente deve levar update.

Update na tabela Opinião: Quando feito update na tabela opinião é necessário que o ID do cliente e o nome do cliente coincidam com os dados da tabela Cliente.

Insert na tabela Cliente: Quando feito um insert na tabela Cliente, o n de opiniões terá de ser = 0, visto que na prática a ficha do cliente é feita antes do mesmo poder conseguir dar a sua opinião.

Delete na tabela Cliente: Quando feito um delete na tabela Cliente, as opiniões com o ID desse mesmo cliente devem sofrer um delete também de modo a apagar todos os dados desse cliente.

Update na tabela Cliente: Quando feito um update na tabela Cliente, o n de opiniões devem ser de acordo com a tabela opinião cujo id seja desse cliente.

Por Diogo Fernandes: Terceira desnormalização

Na primeira desnormalização (por coluna redundante) adicionámos a coluna nome de vinho na tabela lista de vinhos

Insert na tabela Vinhos: Quando inserida uma coluna na tabela Vinhos é necessário a adição de uma coluna na lista de vinhos com o nome de vinho exatamente igual.

Update na tabela de Vinhos: Quando feito um update na tabela Vinhos é necessário o update na tabela lista de vinhos cujo ID seja o mesmo.

Delete na tabela de Vinhos: Quando feito o delete na tabela Vinhos é necessário também o delete na tabela Lista Vinhos cujo ID do vinho seja o mesmo.

Insert na tabela Lista Vinhos: Quando feito insert na tabela Lista de Vinhos é necessário também o insert com o mesmo ID de vinho e o mesmo nome de vinho na tabela Vinho.

Update na tabela Lista Vinhos: Quando feito update na tabela Lista de Vinhos é necessário que os dados do ID do vinho e o nome do vinho coincidam com os que estão na tabela Vinho.

Delete na tabela Lista Vinhos: Quando feito o delete na tabela Lista de Vinhos é necessário também fazer o delete na tabela Vinhos.

RESTRIÇÕES DAS TABELAS

NOT NULL – Diz que essa coluna não pode ter valores igual a null ;

Ex: Entidade Cliente: “nome_cliente VARCHAR2(30) NOT NULL,”.

UNIQUE – Diz que cada valor dessa tabela seja únicos, ou seja, que não haja repetição;

Ex: Entidade Cliente “NIF_CLIENTE NUMBER(9,0) NOT NULL UNIQUE”.

PRIMARY KEY – uma combinação de NOT NULL e UNIQUE onde unicamente, identifica cada linha de tabela;

Ex: Entidade Vinho:” CONSTRAINT vinho_PK PRIMARY KEY (vinho_id);

FOREIGN KEY – identifica unicamente uma linha que existe noutra tabela;

Ex : Entidade Opiniao : “ FOREIGN KEY (cliente_id_cliente) REFERENCES cliente(cliente_id).

FICHEIRO DDL

```
DROP TABLE casta CASCADE CONSTRAINTS
;

DROP TABLE cliente CASCADE CONSTRAINTS
;

DROP TABLE lista_vinhos CASCADE CONSTRAINTS
;

DROP TABLE lojaprova CASCADE CONSTRAINTS
;

DROP TABLE opiniao CASCADE CONSTRAINTS
;

DROP TABLE percentagem CASCADE CONSTRAINTS
;

DROP TABLE vinho CASCADE CONSTRAINTS
;

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE
CREATE TABLE casta
    (
        id_casta   NUMBER(9) NOT NULL
      , nome_casta VARCHAR2(30) NOT NULL
    )
;

ALTER TABLE casta ADD CONSTRAINT casta_pk PRIMARY KEY (id_casta)
;

CREATE TABLE cliente
    (
        id_cliente   NUMBER(9) NOT NULL
      , nome_cliente VARCHAR2(30) NOT NULL
      , nif_cliente  NUMBER(9) NOT NULL
      , n_opinioes   NUMBER(5)
    )
;

ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY
(id_cliente)
;

CREATE TABLE lista_vinhos
    (
        id_lista          VARCHAR2(30) CONSTRAINT
nnc_lista_vinhos_id_lista NOT NULL
      , lojaprova_id_loja NUMBER(9)
        --  ERROR: Column Lista_Vinhos.LojaProva_ID_loja check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_lista_vinhos_loja_de_prova_id_loja NOT NULL
      , vinho_id_vinho NUMBER(9)
        --  ERROR: Column Lista_Vinhos.Vinho_ID_vinho check
constraint name length exceeds maximum allowed length(30)
```

```
        CONSTRAINT nnc_lista_vinhos_vinho_id_vinho NOT NULL
      , nome_vinho     VARCHAR2(30) CONSTRAINT
nnc_lista_vinhos_nome_vinho NOT NULL
      , data_aquisicao DATE
        --  ERROR: Column Lista_Vinhos.data_aquisicao check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_lista_vinhos_data_aquisicao NOT NULL
    )
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_pk PRIMARY KEY (
lojaprova_id_loja ,vinho_id_vinho )
;

CREATE TABLE lojaprova
    (
        id_loja         NUMBER(9) NOT NULL
      , nome_loja       VARCHAR2(30) NOT NULL
      , localizacao     VARCHAR2(30) NOT NULL
      , n_clientes      NUMBER(9)
      , vinho_preferido VARCHAR2(10)
    )
;

ALTER TABLE lojaprova ADD CONSTRAINT loja_de_prova_pk PRIMARY KEY
(id_loja)
;

CREATE TABLE opiniao
    (
        opiniao_id                     NUMBER(9) CONSTRAINT
nnc_opiniao_opiniao_id NOT NULL
      , rating                         NUMBER(2) CONSTRAINT
nnc_opiniao_rating NOT NULL
      , descricao                      VARCHAR2(30)
      , cliente_id_cliente             NUMBER(9) CONSTRAINT
nnc_opiniao_cliente_id_cliente NOT NULL
      , nome_cliente                   VARCHAR2(30) CONSTRAINT
nnc_opiniao_nome_cliente NOT NULL
      , lista_vinhos_lojaprova_id_loja NUMBER(9)
        --  ERROR: Column Opiniao.Lista_Vinhos_LojaProva_ID_loja
check constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_opiniao_lista_vinhos_lojaprova_id_loja NOT
NULL
      , lista_vinhos_vinho_id_vinho NUMBER(9)
        --  ERROR: Column Opiniao.Lista_Vinhos_Vinho_ID_vinho check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_opiniao_lista_vinhos_vinho_id_vinho NOT NULL
      , data DATE CONSTRAINT nnc_opiniao_data NOT NULL
    )
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_pk PRIMARY KEY
(opiniao_id)
;

CREATE TABLE percentagem
    (
        percentagem_casta NUMBER(3) NOT NULL
      , casta_id_casta    NUMBER(9) NOT NULL
      , vinho_id_vinho    NUMBER(9) NOT NULL
```

```
    )
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_pk PRIMARY KEY (
casta_id_casta ,vinho_id_vinho )
;

CREATE TABLE vinho
    (
        id_vinho       NUMBER(9) NOT NULL
      , nome           VARCHAR2(30) NOT NULL
      , safra          VARCHAR2(30) NOT NULL
      , n_vendido      NUMBER(9)
      , stock          NUMBER(9)
      , cor            VARCHAR2(10) NOT NULL
      , docura         VARCHAR2(10) NOT NULL
      , acidez         VARCHAR2(10) NOT NULL
      , origem         VARCHAR2(10) NOT NULL
      , teor_alcoolico NUMBER(4, 2) NOT NULL
      , corpo          VARCHAR2(10) NOT NULL
    )
;

ALTER TABLE vinho ADD CONSTRAINT vinho_pk PRIMARY KEY (id_vinho)
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_lojaprova_fk
FOREIGN KEY (lojaprova_id_loja) REFERENCES lojaprova (id_loja)
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_vinho_fk FOREIGN
KEY (vinho_id_vinho) REFERENCES vinho (id_vinho)
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_cliente_fk FOREIGN KEY
(cliente_id_cliente) REFERENCES cliente (id_cliente)
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_lista_vinhos_fk FOREIGN
KEY ( lista_vinhos_lojaprova_id_loja ,lista_vinhos_vinho_id_vinho )
REFERENCES lista_vinhos ( lojaprova_id_loja ,vinho_id_vinho )
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_casta_fk FOREIGN
KEY (casta_id_casta) REFERENCES casta (id_casta)
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_vinho_fk FOREIGN
KEY (vinho_id_vinho) REFERENCES vinho (id_vinho)
;
```

Por Luís Barros

Query para saber quais as castas pertencentes aos vinhos, selecionando o nome dos vinhos, o nome das castas e a percentagem das mesmas.

```
SELECT PERCENTAGEM_CASTA, NOME, NOME_CASTA
FROM PERCENTAGEM, VINHO, CASTA
WHERE ID_VINHO = VINHO_ID_VINHO AND ID_CASTA = CASTA_ID_CASTA;
```

PERCENTAGEM_CASTA	NOME	NOME_CASTA
100	CASAL GARCIA VINHO VERDE SWEET	TOURIGA NACIONAL
100	COLOSSAL	CASTELAO
50	CONFIDENCIAL	TOURIGA NACIONAL
50	PLANALTO	BAGA

Query com funcionalidade de apresentar o ID da opinião, Rating, Nome de cliente e data apresentada em formato caracteres com o formato ´DD-MM-YY'

```
SELECT ID_OPINIAO, RATING, NOME_CLIENTE, TO_CHAR(DATA, 'DD-MM-YY')
FROM OPINIAO;
```

ID_OPINIAO	RATING	NOME_CLIENTE	TO_CHAR(DATA,'DD-MM-YY')
1	1	JOAO	15-01-22
2	10	JOAO	01-01-22
3	3	DUARTE	18-12-21
4	4	MANUEL	19-11-22
5	7	RODRIGO	18-11-22

Query com funcionalidade de selecionar todos os vinhos com teor alcoólico superior a 10, ordenando numericamente por teor alcoólico de maneira predefinida pelo software (ascendente).

```
SELECT ID_VINHO, NOME, TEOR_ALCOOLICO
FROM VINHO
GROUP BY ID_VINHO, NOME,
TEOR_ALCOOLICO
HAVING TEOR_ALCOOLICO > 10
ORDER BY TEOR_ALCOOLICO;
```

ID_VINHO	NOME	TEOR_ALCOOLICO
2	ALVARINHO 2017	12.5
10	PLANALTO	12.5
6	ANIMUS TINTO	13
9	CONFIDENCIAL	13
12	ESTEVA DOURO	13
4	RED BLEND 2018	13
5	TINTO 2018	13
3	PAPA FIGOS DOURO 2017	13.5
7	RESERVA RIOJA	13.5
8	COLOSSAL	14

Query com funcionalidade de apresentar o nome dos vinhos e há quantos dias os mesmos foram comprados pela loja de prova, ordenando por data de aquisição descendentemente de modo a aparecer os vinhos adquiridos mais recentemente primeiro.

```
SELECT NOME_VINHO, TRUNC(SYSDATE-DATA_AQUISICAO)
FROM LISTAVINHOS
ORDER BY DATA_AQUISICAO DESC;
```

NOME_VINHO	TRUNC(SYSDATE-DATA_AQUISICAO)
TINTO 2018	6
ALVARINHO 2017	8
ANIMUS TINTO	9
RED BLEND 2018	12
CASAL GARCIA	41

Query com funcionalidade de calcular o nº total vendido de garrafas

```
SELECT SUM(N_VENDIDO) AS TOTALVENDIDO
FROM VINHO;
```

	TOTALVEN...
1	97

Query com funcionalidade de mostrar os vinhos e as suas cores que têm um ‘A’ no nome ordenando por tamanho de nome por ordem predefinida pelo software (ascendente).

```
SELECT NOME, COR
FROM VINHO
WHERE INSTR (NOME, 'A') > 0
ORDER BY LENGTH(NOME);
```

NOME	COR
PLANALTO	BRANCO
COLOSSAL	TINTO
CASAL GARCIA	VERDE
CONFIDENCIAL	TINTO
ANIMUS TINTO	TINTO
ESTEVA DOURO	TINTO
RESERVA RIOJA	TINTO
ALVARINHO 2017	BRANCO
PAPA FIGOS DOURO 2017	TINTO
CASAL GARCIA VINHO VERDE SWEET	VERDE

Query com funcionalidade de mostrar as caracteristicas do vinho mais vendido.

```
SELECT * FROM VINHO
WHERE N_VENDIDO = (SELECT MAX(N_VENDIDO) FROM VINHO);
```

ID_VINHO	NOME	SAFRA	N_VENDIDO	STOCK	COR	DOCURA	ACIDEZ	ORIGEM	TEOR_ALCOOLICO	CORPO
10	PLANALTO	2017	20	0	BRANCO	SECO	ACIDO	DOURO	12.5	MEDIO

Por Diogo Fernandes

Query com a funcionalidade de organizar os vinhos existentes em stock pela respetiva cor. Após organizar os vinhos pela respetiva cor, organiza-os alfabéticamente apresentando também o stock existente e o respetivo teor alcoolico aredondado.

```
SELECT cor, nome, stock, ROUND(teor_alcoolico) AS ROUND
FROM vinho WHERE stock > 0
ORDER BY cor ASC, nome ASC, stock, teor_alcoolico
```

COR	NOME	STOCK	ROUND
BRANCO	ALVARINHO 2017	2	13
TINTO	ANIMUS TINTO	10	13
TINTO	COLOSSAL	13	14
TINTO	CONFIDENCIAL	14	13
TINTO	ESTEVA DOURO	5	13
TINTO	PAPA FIGOS DOURO 2017	3	14
TINTO	RED BLEND 2018	7	13
TINTO	RESERVA RIOJA	10	14
TINTO	TINTO 2018	10	13
VERDE	CASAL GARCIA	10	10
VERDE	CASAL GARCIA VINHO VERDE SWEET	20	9

Query com a funcionalidade de unir a coluna cor com a coluna acidez da tabela vinho, sem repetição, caso exista, de valores.

```
SELECT DISTINCT cor
FROM vinho
UNION ALL
SELECT DISTINCT acidez
FROM vinho
```

COR
VERDE
BRANCO
TINTO
ACIDO
MEDIO

Query com a funcionalidade de apresentar o nome dos clientes que deixaram uma opinião onde a mesma é "MAU".

```
SELECT nome_cliente FROM opiniao
WHERE descricao IN ('MAU');
```

NOME_CLIENTE
DUARTE
MANUEL

Query com a funcionalidade de apresentar as percentagens de casta onde a casta são iguais áquela que é a maior percentagem de casta.

```
SELECT percentagem_casta FROM percentagem
WHERE percentagem_casta = (SELECT MAX(percentagem_casta) FROM
percentagem);
```

PERCENTAGEM_CASTA
100
100

Query com a funcionalidade de apresentar o id dos clientes que estejam presentes tanto na tabela cliente como na tabela opinião.

```
SELECT id_cliente FROM cliente
INTERSECT
SELECT cliente_id_cliente FROM opiniao
ORDER BY id_cliente
```

ID_CLIENTE
1
2
3
4

Por João Costa

Query com a funcionalidade de apresentar o nome dos vinhos que apresentam teor alcoolico entre 13 e 15.

```
SELECT
        NOME,
        TEOR_ALCOOLICO
FROM
        VINHO
WHERE
        TEOR_ALCOOLICO > 13
AND     TEOR_ALCOOLICO < 15
ORDER BY
        NOME
```

1	COLOSSAL	14
2	PAPA FIGOS DOURO 2017	13,5
3	RESERVA RIOJA	13,5

Query com a funcionalidade de apresentar o nome dos vinhos que apresentam a sua safra entre os anos 2018 e 2020

```
SELECT
        SAFRA ,
        NOME
FROM
        VINHO
WHERE
        SAFRA BETWEEN '2018' AND     '2020'
ORDER BY
        NOME
```

	SAFRA	NOME
1	2020	ALVARINHO 2017
2	2019	ANIMUS TINTO
3	2020	CASAL GARCIA VINHO VERDE SWEET
4	2018	ESTEVA DOURO
5	2018	RED BLEND 2018
6	2018	TINTO 2018

Query com a funcionalidade de apresentar o número de castas existentes.

```
SELECT
        COUNT (ID_CASTA)AS NOME_CASTA
FROM
        CASTA;
```

1	10

Query com a funcionalidade de apresentar todos os vinhos que apresentam origem “Douro”

```
SELECT
        nome,
        origem
FROM
        vinho
WHERE
        origem LIKE 'DOURO'
ORDER BY
        Nome
```

ANIMUS TINTO	DOURO
ESTEVA DOURO	DOURO
PAPA FIGOS DOURO 2017	DOURO
PLANALTO	DOURO

Query com a funcionalidade de apresentar os vinhos que na sua caracteristica de acidez apresentam-se “Acido”.

```
SELECT
        nome ,
        acidez
FROM
        vinho
WHERE
        acidez LIKE'ACIDO'
ORDER BY
        nome
```

```
SELECT nome, acidez
FROM vinho
WHERE acidez LIKE 'ACIDO'
ORDER BY nome
```

VIEWS

Por Luís Barros:

View que permite visualizar os vinhos tintos e o seu teor alcoólico, ordenando por teor alcoólico.

```
CREATE VIEW VINHO_TINTO_TEOR AS
SELECT NOME"VINHO", COR"COR", TEOR_ALCOOLICO"ALCOOL"
FROM VINHO WHERE COR LIKE 'TINTO'
ORDER BY TEOR_ALCOOLICO;
```

VINHO	COR	ALCOOL
ANIMUS TINTO	TINTO	13
CONFIDENCIAL	TINTO	13
ESTEVA DOURO	TINTO	13
RED BLEND 2018	TINTO	13
TINTO 2018	TINTO	13
RESERVA RIOJA	TINTO	13.5
PAPA FIGOS DOURO 2017	TINTO	13.5
COLOSSAL	TINTO	14

Por Diogo Fernandes

View que permite visualizar os vinhos e o seu teor alcoólico, ordenando pelo seu nome.

```
CREATE OR REPLACE VIEW VinhosTeorAlcoolico
AS SELECT nome, teor_alcoolico
FROM vinho
ORDER BY nome;
```

NOME	TEOR_ALCOOLICO
ALVARINHO 2017	12,5
ANIMUS TINTO	13
CASAL GARCIA	9,5
CASAL GARCIA VINHO VERDE SWEET	9
COLOSSAL	14
CONFIDENCIAL	13
ESTEVA DOURO	13
PAPA FIGOS DOURO 2017	13,5
PLANALTO	12,5
RED BLEND 2018	13
RESERVA RIOJA	13,5
TINTO 2018	13

Por João Costa

View que permite visualizar os vinhos que têm a sua origem na região do "Douro"

```
CREATE OR REPLACE  VIEW
        VINHO_ORIGEM AS SLECT NOME"VINHO",
        ORIGEM"ORIGEM"
FROM
        VINHO
WHERE
        ORIGEM LIKE 'DOURO';
```

PAPA FIGOS DOURO 2017	DOURO
ANIMUS TINTO	DOURO
PLANALTO	DOURO
ESTEVA DOURO	DOURO

SEQUÊNCIAS

Por Luís Barros

Sequência "VI", sequência que vai começar com o valor da variável (id_vinho) inica em 1, e a cada inserção é incrementado 1.

```
-- SEQUENCIA DE VINHOS
CREATE SEQUENCE VI
START WITH 1
INCREMENT BY 1;

--INSERIR VINHOS
INSERT INTO VINHO VALUES
(VI.CURRVAL, 'CASAL GARCIA VERDE SWEET', 2020, NULL, NULL, 'VERDE',
'DOCE', 'ACIDO', 'VERDE', 9.0, 'LEVE');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'ALVARINHO 2017', 2020, NULL, NULL, 'BRANCO', 'SECO',
'ACIDO', 'MONCAO', 12.5, 'LEVE');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'PAPA FIGOS DOURO 2017', 2017, NULL, NULL, 'TINTO',
'SECO', 'MEDIO', 'DOURO', 13.5, 'INTENSO');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'RED BLEND 2018', 2018, NULL, NULL, 'TINTO', 'SECO',
'MEDIO', 'PORTUGAL', 13.0, 'INTENSO');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'TINTO 2018', 2018, NULL, NULL, 'TINTO', 'SECO',
'MEDIO', 'LISBOA', 13.0, 'INTENSO');
```

Por Diogo Fernandes

Sequência “CIL”, sequência que vai começar com o valor da variável (id_ciente) inica em 1, e a cada inserção é incrementado 1.

```
CREATE SEQUENCE cil
    START WITH 1
    INCREMENT BY 1;

INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'LUIS BARROS', '123456789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'JOAO', '231481934', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'DUARTE', '123456789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, ' JULIO MANUEL', '12546789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'RODRIGO', '748423413', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'JOAO', '111333222', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'MANUEL', '222111333', NULL);
```

Por João Costa

```
CREATE SEQUENCE CAST
START WITH 1 INCREMENT BY 1;

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'TOURIGA NACIONAL'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'BAGA'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'CASTELAO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'TOURIGA FRANCA'
        );

INSERT INTO
        CASTA VALUES
```

```
        (
                CASTA.NEXTVAL,
                'TRINCADEIRA'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ALVARINHO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'LOUREIRO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ARINTO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ENCRUZADO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'BICAL'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'FERNAO PIRES'
        );
```

SINÓNIMOS

Por Luís Barros
Sinónimo para tabela da Lista de Vinhos:

```
CREATE SYNONYM LISTAVINHOS FOR LISTA
```

Por Diogo Fernandes
Sinónimo para tabela da Loja de Prova:

```
CREATE SYNONYM lojadeprova FOR loja;
```

Por João Costa
Sinónimo para tabela Cliente:

```
CREATE SYNONYM Comprador FOR Cl;
```

PRIVILÉGIOS

Dar privilégios a dois administradores:

```
GRANT SELECT ANY TABLE, SELECT ANY TABLE TO BDI_EI_1703638;
GRANT SELECT ANY TABLE, SELECT ANY TABLE TO BDI_EI_1703762;
GRANT SELECT ON CLIENTE TO BDI_EI_1703638 WITH GRANT OPTION;
GRANT SELECT ON CLIENTE TO BDI_EI_1703762 WITH GRANT OPTION;
```

Privilégios de insert e update:

```
GRANT UPDATE(ID_CLIENTE, NOME_CLIENTE, NIF_CLIENTE, N_OPINIOES)
ON CLIENTE TO BDI_EI_1703638;
GRANT UPDATE(ID_CLIENTE, NOME_CLIENTE, NIF_CLIENTE, N_OPINIOES)
ON CLIENTE TO BDI_EI_1703762;
GRANT INSERT(ID_CLIENTE, NOME_CLIENTE, NIF_CLIENTE, N_OPINIOES)
ON CLIENTE TO BDI_EI_1703638;
GRANT INSERT(ID_CLIENTE, NOME_CLIENTE, NIF_CLIENTE, N_OPINIOES)
ON CLIENTE TO BDI_EI_1703638;
```

MATRIZ CRUD

Utilizador	Cliente	Vinho	LojadeProva	ListadeVinhos	Opinião	Percentagem	Casta
Administrador	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD	CRUD
Comprador	CRU	-	R	R	CR	R	R
Convidado	-	-	R	R	-	-	-

ROLES

```
CREATE ROLE ADMIN_ROLE;

GRANT SELECT, INSERT, UPDATE, DELETE ON CLIENTE TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON CASTA TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON PERCENTAGEM TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON VINHO TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON LOJADEPROVA TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON OPINIAO TO ADMIN_ROLE;
GRANT SELECT, INSERT, UPDATE, DELETE ON LISTADEVINHOS TO ADMIN_ROLE;

GRANT ADMIN_ROLE TO JOSE_FONSECA;


CREATE ROLE CLIENTE_ROLE;

GRANT SELECT, INSERT, UPDATE ON CLIENTE TO CLIENTE_ROLE;
GRANT SELECT ON CASTA TO CLIENTE_ROLE;
GRANT SELECT ON PERCENTAGEM TO CLIENTE_ROLE;
GRANT SELECT ON LOJADEPROVA TO CLIENTE_ROLE;
GRANT SELECT, INSERT, UPDATE ON OPINIAO TO CLIENTE_ROLE;
GRANT SELECT ON LISTADEVINHOS TO CLIENTE_ROLE;

GRANT CLIENTE_ROLE TO CLIENTE;


CREATE ROLE CONVIDADO_ROLE;

GRANT SELECT ON LOJADEPROVA TO CONVIDADO_ROLE;
GRANT SELECT ON LISTAVINHOS TO CONVIDADO_ROLE;

GRANT CONVIDADO_ROLE TO CONVIDADO;
```

TRANSAÇÕES

Transação é o agrupamento lógico das operações levadas a cabo por um ou mais comandos DML ou um comando DDL, sendo estas executadas de uma forma atómica com o objetivo de preservar a integridade e consistência dos dados. Estas operações só podem se tornar permanentes na base de dados forem executadas com sucesso.

Por Luís Barros

Na primeira desnormalização, por redundância , ao fazer-se um update no n_vendido na tabela vinho, teriamos de recalcular se esse vinho com o update feito seria o novo vinho preferido da loja de prova.

```
create or replace PROCEDURE UPDT_VINHOPREFERIDO(V_NOVOVINHO VARCHAR2,
V_NVENDIDO NUMBER) IS
V_NOME VINHO.NOME%TYPE;
V_COUNT NUMBER(1);
N_VENDIDO NUMBER(9);

BEGIN

    SELECT COUNT(*)
    INTO V_COUNT
    FROM VINHO
    WHERE N_VENDIDO = (SELECT MAX(N_VENDIDO) FROM VINHO);



    IF V_COUNT = 1  THEN
        V_NOME:= V_NOVOVINHO;
        DBMS_OUTPUT.PUT_LINE(V_NOME);

        UPDATE VINHO SET N_VENDIDO = V_NVENDIDO WHERE NOME = V_NOME;
        UPDATE LOJADEPROVA SET VINHO_PREFERIDO = V_NOME;

    ELSE

        DBMS_OUTPUT.PUT_LINE(' NAO HA ATUALIZACAO DE VINHO
PREFERIDO');
        END IF;

COMMIT;
END;
```

Procedimento de teste

```
CREATE OR REPLACE PROCEDURE TEST_UPDT_VINHOPREFERIDO IS
BEGIN
UPDT_VINHOPREFERIDO('PAPA FIGOS', 0);
END;
/
EXEC TEST_UPDT_VINHOPREFERIDO;
```

PL/SQL

Procedimentos por Luís Barros

Procedimento para determinar se é necessário fazer "restock" a algum tipo de vinho.

```
CREATE OR REPLACE PROCEDURE RESTOCK
IS V_NUMERO NUMBER(4);
V_RESTOCK VARCHAR2(30);
BEGIN
SELECT COUNT(VINHO.NOME)
INTO V_NUMERO
FROM VINHO WHERE VINHO.STOCK = 0;
IF:V_NUMERO = 0 THEN V_RESTOCK:= 'NAO E NECESSARIO RESTOCK';
ELSIF:V_NUMERO > 0 THEN V_RESTOCK:= 'E NECESSARIO RESTOCK';
END IF;
DBMS_OUTPUT.PUT_LINE(V_RESTOCK);
END;
```

	NOME	SAFRA	N_VENDIDO	STOCK	COR
2	ALVARINHO 2017	2020	10	2	BRANCO
3	PAPA FIGOS DOURO 2017	2017	5	3	TINTO
4	RED BLEND 2018	2018	5	7	TINTO
5	TINTO 2018	2018	6	10	TINTO
1	CASAL GARCIA VINHO VERDE SWEET	2020	2	20	VERDE
6	ANIMUS TINTO	2019	3	10	TINTO
7	RESERVA RIOJA	2012	4	10	TINTO
8	COLOSSAL	2015	8	13	TINTO
9	CONFIDENCIAL	2014	9	14	TINTO
10	PLANALTO	2017	20	0	BRANCO
11	CASAL GARCIA	2017	10	10	VERDE
12	ESTEVA DOURO	2018	15	5	TINTO

```
Connecting to the database bdi_ei_1700331.
 E NECESSARIO RESTOCK
Process exited.
Disconnecting from the database bdi_ei_1700331.
```

Procedimento para contar clientes com ficha na loja de prova

```
CREATE OR REPLACE PROCEDURE CONTAR_CLIENTES
IS C_NUMBER NUMBER(9);
BEGIN
SELECT COUNT(*)
INTO C_NUMBER
FROM CLIENTE
DBMS_OUTPUT.PUT_LINE(C_NUMBER);
END;
```

```
Connecting to the database bdi_ei_1700331.
9
Process exited.
Disconnecting from the database bdi_ei_1700331.
```

Procedimento para calcular a média do teor alcoólico dos vinhos.

```
CREATE OR REPLACE PROCEDURE MEDIA_TEORALCOOLICO
IS V_TEORALCOOLICO NUMBER(4,2);
BEGIN
SELECT AVG(VINHO.TEOR_ALCOOLICO)
INTO V_TEORALCOOLICO
FROM VINHO;
DBMS_OUTPUT.PUT_LINE('MEDIA TEOR ALCOOLICO DOS VINHOS: ' ||
V_TEORALCOOLICO);
END;
```

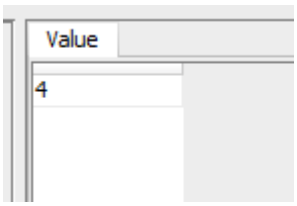
```
Connecting to the database bdi_ei_1700331.
MEDIA TEOR ALCCOLICO DOS VINHOS: 12.46
Process exited.
Disconnecting from the database bdi_ei_1700331.
```

Função

Função para determinar quantos clientes deram a sua opinião na loja

```
CREATE OR REPLACE FUNCTION N_CLIENTESOPINIAO
RETURN NUMBER
IS
NUMERO_CLIENTES NUMBER(4);
BEGIN
SELECT COUNT(CLIENTE.ID_CLIENTE) INTO NUMERO_CLIENTES
FROM CLIENTES WHERE N_OPINIOES !=0;
RETURN NUMERO_CLIENTES;
END;
```

ID_CLIENTE	NOME_CLIENTE	NIF_CLIENTE	N_OPINIOES
1	JOAO	231481934	2
2	DUARTE	123456789	1
3	JULIO MANUEL	12546789	1
4	RODRIGO	748423413	1
6	MANUEL	222111333	0
5	LUIS BARROS	123123123	0
7	JOAO COSTA	122222111	0
8	DIOGO FERNANDES	111222333	0
9	JOAO ANTONIO	111222313	0



Trigger para verificar se a data de aquisição do vinho introduzida pelo administrador é válida, ou seja, se é igual á do sistema na qual está a inserir.

```
CREATE OR REPLACE TRIGGER VALIDA_DATA2
BEFORE INSERT ON LISTAVINHOS
FOR EACH ROW
BEGIN
IF:NEW.DATA_AQUISICAO = SYSDATE THEN
DBMS_OUTPUT.PUT_LINE('DATA VALIDA');
ELSIF:NEW.DATA_AQUISICAO = SYSDATE THEN
RAISE_APPLICATION_ERROR(-2001, 'DATA INVALIDA');
END IF;
END;
```

Teste do Trigger:
SYSDATE = 25-JAN-22

```
SET SERVEROUTPUT ON;

INSERT INTO LISTAVINHOS VALUES
(6, 1, 2, 'ALVARINHO 2017', TO_DATE('26-JAN-22'));
```

```
Error starting at line : 3 in command -
INSERT INTO LISTAVINHOS VALUES
(6, 1, 2, 'ALVARINHO 2017', TO_DATE('26-JAN-22'))
Error report -
ORA-20001: DATA INVALIDA
```

Cursor para apresentar lista de clientes com ficha na loja de prova e o nº de clientes com ficha registada na loja de prova.

```
DECLARE
    CURSOR CLIENTEREG IS SELECT NOME_CLIENTE FROM CLIENTE;
    FORNOME CLIENTE.NOME_CLIENTE%TYPE;
BEGIN
OPEN CLIENTEREG;
LOOP
FETCH CLIENTEREG INTO FORNOME;
IF CLIENTEREG%NOTFOUND THEN EXIT;
END IF;

DBMS_OUTPUT.PUT_LINE('CLIENTE : '||FORNOME);
END LOOP;
DBMS_OUTPUT.PUT_LINE('TOTAL : '||CLIENTEREG%ROWCOUNT);
CLOSE CLIENTEREG;
END;
```

Package que nos indica o corpo do vinho quando feito o input do id do mesmo.

```
CREATE OR REPLACE PACKAGE PKG_VINHO_CORPO AS
--
PROCEDURE PCR_GET_CORPO_VINHO(PI_N_IDVINHO IN NUMBER, CORPO_CINHO OUT
VARCHAR2);
--
END;
--
/
CREATE OR REPLACE PACKAGE BODY PKG_VINHO_CORPO AS
--
PROCEDURE PCR_GET_CORPO_VINHO(PI_N_IDVINHO IN NUMBER, CORPO_VINHO OUT
VARCHAR2) IS
    V_CORPO VARCHAR2(10);
    BEGIN
    --
    BEGIN
        SELECT CORPO INTO V_CORPO FROM VINHO WHERE ID_VINHO =
PI_N_IDVINHO;
        EXCEPTION WHEN NO_DATA_FOUND THEN
        V_CORPO:=NULL;
        END;
        --
        CORPO_VINHO:=V_CORPO;
        --
    END PCR_GET_CORPO_VINHO;
    --
    END;
    /
```

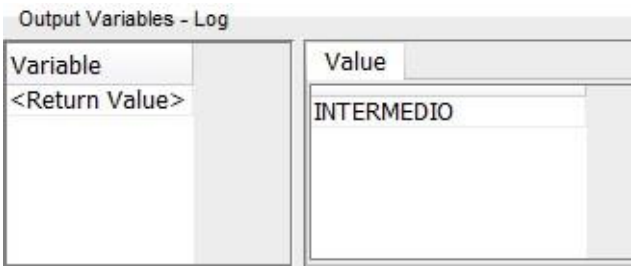
Procedimentos por Diogo Fernandes

Procedimento com a funcionalidade de adicionar uma opinião na tabela opiniao.

```
create or replace PROCEDURE PCR_ADD_OPINIAO
    (rating NUMBER, nova_opiniao VARCHAR2, cliente_id NUMBER,
     id_lista_vinho NUMBER, nome VARCHAR2,
     id_vinho NUMBER)
IS
    new_id_opiniao NUMBER(4);
BEGIN
    SELECT id_opiniao INTO new_id_opiniao FROM opiniao
    WHERE id_opiniao = (SELECT MAX(id_opiniao) FROM opiniao);
    --
    INSERT INTO opiniao VALUES
        (new_id_opiniao, rating, nova_opiniao, cliente_id,
id_lista_vinho,
         nome, 1, id_vinho, SYSDATE);
END;
```

Função por Diogo Fernandes
Função com funcionalidade de consoante a média total do rating atribui uma classificação global das avaliações.

```
create or replace FUNCTION
media_rating
RETURN VARCHAR2 IS
    rating_final VARCHAR2(20);
    aux NUMBER(4);
--
BEGIN
--
    SELECT AVG(rating) INTO aux
    FROM opiniao;

    IF aux < 3 THEN
        rating_final := 'PESSIMO';
    ELSIF aux < 5 THEN
        rating_final := 'MAU';
    ELSIF aux < 7 THEN
        rating_final := 'INTERMEDIO';
    ELSE
        rating_final := 'EXCELENTE';
    END IF;
--
    RETURN rating_final;
END;
```



Trigger por Diogo Fernandes
Trigger acionado ao inserir uma descrição na tabela opinião para fazer a validação da mesma.

```
CREATE OR REPLACE TRIGGER trg_ValidaOpiniao
BEFORE INSERT ON opiniao
FOR EACH ROW
BEGIN
    IF: NEW.descricao NOT IN ('EXCELENTE', 'INTERMEDIO', 'MAU',
'PESSIMO') THEN
        RAISE_APPLICATION_ERROR(-20001, 'AVALIACAO NAO CUMPRE OS
PARAMETROS');
    END IF;
END;
```

```
INSERT INTO opiniao
VALUES
    (6, 9, 'HORRIVEL', 4, 3, 'RODRIGO', 1, 7, TO_DATE('22.01.27'));
```

```
            -
VALUES
    (6, 9, 'HORRIVEL', 4, 3, 'RODRIGO', 1, 7, TO_DATE('22.01.27'))
Error report -
ORA-20001: AVALIACAO NAO CUMPRE OS PARAMETROS
```

Package por Diogo Fernandes
Package quem contém dois procedimentos, um que ao introduzir o id do vinho desejado devolve o nome do mesmo se existir e NULL caso não exista, outro com a funcionalidade de adicionar um vinho.

```
create or replace PACKAGE PKG_GERENCIA_VINHOS AS
    PROCEDURE PCR_GET_NOME_VINHO(PI_N_ID IN NUMBER, PO_NOME OUT
VARCHAR2);
    PROCEDURE PCR_ADD_VINHO
    (
        nome VARCHAR2, safra VARCHAR2, stock NUMBER,
        cor VARCHAR2, docura VARCHAR2, acidez VARCHAR2,
        origem VARCHAR2, teor_alcoolico NUMBER, corpo VARCHAR2
    );
END;
--
/
create or replace PACKAGE BODY PKG_GERENCIA_VINHOS AS
--
PROCEDURE PCR_GET_NOME_VINHO(PI_N_ID IN NUMBER, PO_NOME OUT VARCHAR2)
IS
    nome_vinho VARCHAR2(30);
BEGIN
--
    BEGIN
        SELECT nome INTO nome_vinho FROM vinho WHERE id_vinho =
PI_N_ID;
    EXCEPTION WHEN NO_DATA_FOUND THEN
        nome_vinho := NULL;
    END;
--
    PO_NOME := nome_vinho;
END PCR_GET_NOME_VINHO;

PROCEDURE PCR_ADD_VINHO
(
    nome VARCHAR2, safra VARCHAR2, stock NUMBER,
    cor VARCHAR2, docura VARCHAR2, acidez VARCHAR2,
    origem VARCHAR2, teor_alcoolico NUMBER, corpo VARCHAR2
)
IS
    --
    vinho_id NUMBER(4);
    nome_novo_vinho VARCHAR2(30);
    --
BEGIN
    --
    SELECT id_vinho INTO vinho_id FROM vinho
    WHERE vinho_id = (SELECT MAX(id_vinho) FROM vinho);
    --
    BEGIN
        INSERT INTO vinho VALUES
                    (vinho_id + 1, nome, safra, 0, stock, cor,
                    docura, acidez, origem, teor_alcoolico, corpo);
```

```
    END;
    --
END PCR_ADD_VINHO;
--
END;
```

Procedimentos por João Costa

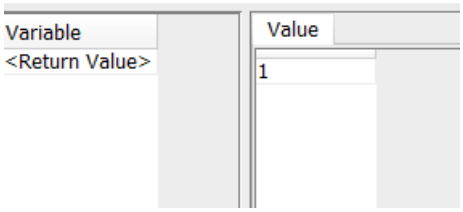
Procedimento com a funcionalidade de mostrar o total de vinhos vendidos.

```
CREATE OR REPLACE PROCEDURE TOTAL_VENDIDOS
IS
        NUMBER_OF_VINHOS          :=0;
        TOTAL_VENDIDOS NUMBER(4)  :=0;
BEGIN
        SELECT
                COUNT(*)
        INTO
                NUMBER_OF_VINHOS
        FROM
                VINHO;

        FOR I IN 1..NUMBER_OF_VINHOS
        LOOP
                SELECT
                        N_VENDIDO
                INTO
                        TOTAL_VENDIDOS
                FROM
                        VINHO
                WHERE
                        ID_VINHO = I;

                AUX := AUX +TOTAL_VENDIDOS;
                TOTAL_VENDIDOS:= AUX;
        END LOOP;
        DBMS_OUTPUT.PUT_LINE(TOTAL_VENDIDOS);
END;
```

Função por João Costa

Função com funcionalidade de mostrar o numero de vinhos que não apresentam qualquer stock .

```
CREATE OR REPLACE FUNCTION N_VINHOSSTOCK
        RETURN NUMBER
IS
        NUMERO_VINHO NUMBER(4);
BEGIN
        SELECT
                COUNT(VINHO.NOME)
        INTO
                NUMERO_VINHO
        FROM
                VINHO
        WHERE
                STOCK = 0;
```



```
        RETURN NUMERO_VINHO;
END;
```

Trigger por João Costa

Trigger acionado ao inserir um tipo de doçura de um vinho na tabela que faz a validação da mesma.

```
CREATE OR REPLACE TRIGGER TRG_DOCURA BEFORE
INSERT
ON
        VINHO FOR EACH ROW BEGIN IF:NEW.DOCURA NOT IN
                                                        (
                                                                'DOCE',
                                                                'SECO',
                                                                'MEDIO'
                                                        )
        THEN RAISE_APLICATION_ERROR
        (
                -2001,
                'A SUA ESCOLHA NÃO VAI DE ENCONTRO AO PEDIDO'
        );

END IF;
END;
```

```
INSERT INTO VINHO VALUES
(13, 'VINHA DO CONTADOR', 2015, 15, 5, 'BRANCO', 'ACIDO', 'MEDIO', 'DAO', 14, 'LEVE');
```

```
Error starting at line : 1 in command -
INSERT INTO VINHO VALUES
(13, 'VINHA DO CONTADOR', 2015, 15, 5, 'BRANCO', 'ACIDO', 'MEDIO', 'DAO', 14, 'LEVE')
Error report -
ORA-21000: argumento de número de erro de raise_application_error de -2001 está fora do inte
ORA-06512: na "BDI_EI_1700331.TRG_DOCURA", linha 3
ORA-04088: erro durante a execução do trigger 'BDI_EI_1700331.TRG_DOCURA'
```

Package por João Costa

```
CREATE OR REPLACE PACKAGE PKG_TOTAL_VENDAS_STOCK
AS

PROCEDURE TOTAL_VENDIDOS;

FUNCTION N_VINHOSSTOCK
        RETURN NUMBER;
END;
```

BIBLIOGRAFIA

No desenvolvimento do relatório foi utilizado os slides fornecidos pelo docente, tendo sido tudo adaptado para este trabalho.

CONCLUSÃO

Neste relatório podemos concluir que a área SQL vai ter benefícios no futuro e principalmente relacionado com o mundo cada vez mais moderno.
Com esta base de dados foram utilizadas várias técnicas que no futuro vão fazer com que muitos erros cometidos neste trabalho não voltaram a ser cometidos. Facilitando assim quando for criada uma base de dados, tanto em ORACLE ou noutra do mesmo tipo.

ANEXOS

Ficheiro DDL:

```
DROP TABLE casta CASCADE CONSTRAINTS
;

DROP TABLE cliente CASCADE CONSTRAINTS
;

DROP TABLE lista_vinhos CASCADE CONSTRAINTS
;

DROP TABLE lojaprova CASCADE CONSTRAINTS
;

DROP TABLE opiniao CASCADE CONSTRAINTS
;

DROP TABLE percentagem CASCADE CONSTRAINTS
;

DROP TABLE vinho CASCADE CONSTRAINTS
;

-- predefined type, no DDL - MDSYS.SDO_GEOMETRY
-- predefined type, no DDL - XMLTYPE
CREATE TABLE casta
    (
        id_casta   NUMBER(9) NOT NULL
      , nome_casta VARCHAR2(30) NOT NULL
    )
;

ALTER TABLE casta ADD CONSTRAINT casta_pk PRIMARY KEY (id_casta)
;

CREATE TABLE cliente
    (
        id_cliente   NUMBER(9) NOT NULL
      , nome_cliente VARCHAR2(30) NOT NULL
      , nif_cliente  NUMBER(9) NOT NULL
      , n_opinioes   NUMBER(5)
    )
;

ALTER TABLE cliente ADD CONSTRAINT cliente_pk PRIMARY KEY
(id_cliente)
;

CREATE TABLE lista_vinhos
    (
        id_lista          VARCHAR2(30) CONSTRAINT
nnc_lista_vinhos_id_lista NOT NULL
      , lojaprova_id_loja NUMBER(9)
```

```
        --  ERROR: Column Lista_Vinhos.LojaProva_ID_loja check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_lista_vinhos_loja_de_prova_id_loja NOT NULL
      , vinho_id_vinho NUMBER(9)
        --  ERROR: Column Lista_Vinhos.Vinho_ID_vinho check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_lista_vinhos_vinho_id_vinho NOT NULL
      , nome_vinho     VARCHAR2(30) CONSTRAINT
nnc_lista_vinhos_nome_vinho NOT NULL
      , data_aquisicao DATE
        --  ERROR: Column Lista_Vinhos.data_aquisicao check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_lista_vinhos_data_aquisicao NOT NULL
    )
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_pk PRIMARY KEY (
lojaprova_id_loja ,vinho_id_vinho )
;

CREATE TABLE lojaprova
    (
        id_loja         NUMBER(9) NOT NULL
      , nome_loja       VARCHAR2(30) NOT NULL
      , localizacao     VARCHAR2(30) NOT NULL
      , n_clientes      NUMBER(9)
      , vinho_preferido VARCHAR2(10)
    )
;

ALTER TABLE lojaprova ADD CONSTRAINT loja_de_prova_pk PRIMARY KEY
(id_loja)
;

CREATE TABLE opiniao
    (
        opiniao_id                      NUMBER(9) CONSTRAINT
nnc_opiniao_opiniao_id NOT NULL
      , rating                          NUMBER(2) CONSTRAINT
nnc_opiniao_rating NOT NULL
      , descricao                       VARCHAR2(30)
      , cliente_id_cliente              NUMBER(9) CONSTRAINT
nnc_opiniao_cliente_id_cliente NOT NULL
      , nome_cliente                    VARCHAR2(30) CONSTRAINT
nnc_opiniao_nome_cliente NOT NULL
      , lista_vinhos_lojaprova_id_loja NUMBER(9)
        --  ERROR: Column Opiniao.Lista_Vinhos_LojaProva_ID_loja
check constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_opiniao_lista_vinhos_lojaprova_id_loja NOT
NULL
      , lista_vinhos_vinho_id_vinho NUMBER(9)
        --  ERROR: Column Opiniao.Lista_Vinhos_Vinho_ID_vinho check
constraint name length exceeds maximum allowed length(30)
        CONSTRAINT nnc_opiniao_lista_vinhos_vinho_id_vinho NOT NULL
      , data DATE CONSTRAINT nnc_opiniao_data NOT NULL
    )
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_pk PRIMARY KEY
(opiniao_id)
;
```

```
CREATE TABLE percentagem
    (
        percentagem_casta NUMBER(3) NOT NULL
      , casta_id_casta    NUMBER(9) NOT NULL
      , vinho_id_vinho    NUMBER(9) NOT NULL
    )
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_pk PRIMARY KEY (
casta_id_casta ,vinho_id_vinho )
;

CREATE TABLE vinho
    (
        id_vinho       NUMBER(9) NOT NULL
      , nome           VARCHAR2(30) NOT NULL
      , safra          VARCHAR2(30) NOT NULL
      , n_vendido      NUMBER(9)
      , stock          NUMBER(9)
      , cor            VARCHAR2(10) NOT NULL
      , docura         VARCHAR2(10) NOT NULL
      , acidez         VARCHAR2(10) NOT NULL
      , origem         VARCHAR2(10) NOT NULL
      , teor_alcoolico NUMBER(4, 2) NOT NULL
      , corpo          VARCHAR2(10) NOT NULL
    )
;

ALTER TABLE vinho ADD CONSTRAINT vinho_pk PRIMARY KEY (id_vinho)
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_lojaprova_fk
FOREIGN KEY (lojaprova_id_loja) REFERENCES lojaprova (id_loja)
;

ALTER TABLE lista_vinhos ADD CONSTRAINT lista_vinhos_vinho_fk FOREIGN
KEY (vinho_id_vinho) REFERENCES vinho (id_vinho)
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_cliente_fk FOREIGN KEY
(cliente_id_cliente) REFERENCES cliente (id_cliente)
;

ALTER TABLE opiniao ADD CONSTRAINT opiniao_lista_vinhos_fk FOREIGN
KEY ( lista_vinhos_lojaprova_id_loja ,lista_vinhos_vinho_id_vinho )
REFERENCES lista_vinhos ( lojaprova_id_loja ,vinho_id_vinho )
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_casta_fk FOREIGN
KEY (casta_id_casta) REFERENCES casta (id_casta)
;

ALTER TABLE percentagem ADD CONSTRAINT percentagem_vinho_fk FOREIGN
KEY (vinho_id_vinho) REFERENCES vinho (id_vinho)
```

Insert values:

```
-- SEQUENCIA DE VINHOS
```

```
CREATE SEQUENCE VI
START WITH 1
INCREMENT BY 1;

--INSERIR VINHOS
INSERT INTO VINHO VALUES
(VI.CURRVAL, 'CASAL GARCIA VERDE SWEET', 2020, NULL, NULL, 'VERDE',
'DOCE', 'ACIDO', 'VERDE', 9.0, 'LEVE');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'ALVARINHO 2017', 2020, NULL, NULL, 'BRANCO', 'SECO',
'ACIDO', 'MONCAO', 12.5, 'LEVE');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'PAPA FIGOS DOURO 2017', 2017, NULL, NULL, 'TINTO',
'SECO', 'MEDIO', 'DOURO', 13.5, 'INTENSO');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'RED BLEND 2018', 2018, NULL, NULL, 'TINTO', 'SECO',
'MEDIO', 'PORTUGAL', 13.0, 'INTENSO');
INSERT INTO VINHO VALUES
(VI.NEXTVAL, 'TINTO 2018', 2018, NULL, NULL, 'TINTO', 'SECO',
'MEDIO', 'LISBOA', 13.0, 'INTENSO');
    --sequencia de clientes

CREATE SEQUENCE cil
    START WITH 1
    INCREMENT BY 1;


    --insert de clientes


INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'LUIS BARROS', '123456789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'JOAO', '231481934', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'DUARTE', '123456789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, ' JULIO MANUEL', '12546789', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'RODRIGO', '748423413', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'JOAO', '111333222', NULL);
INSERT INTO cliente VALUES
    (cil.NEXTVAL, 'MANUEL', '222111333', NULL);



--sequencia de castas

CREATE SEQUENCE CAST
START WITH 1 INCREMENT BY 1;

--insert de castas


INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'TOURIGA NACIONAL'
```

```
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'BAGA'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'CASTELAO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'TOURIGA FRANCA'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'TRINCADEIRA'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ALVARINHO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'LOUREIRO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ARINTO'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'ENCRUZADO'
        );

INSERT INTO
        CASTA VALUES
        (
```

```
                CASTA.NEXTVAL,
                'BICAL'
        );

INSERT INTO
        CASTA VALUES
        (
                CASTA.NEXTVAL,
                'FERNAO PIRES'
        );
```

```
--INSERT NA TABELA LISTAVINHOS

INSERT INTO LISTAVINHOS VALUES
(1, 1, 11, 'CASAL GARCIA', TO_DATE('23-DEC-21');
INSERT INTO LISTAVINHOS VALUES
(2, 1, 4, 'RED BLEND 2018', TO_DATE('21-JAN-22');
INSERT INTO LISTAVINHOS VALUES
(3, 1, 5, 'TINTO 2018', TO_DATE('27-JAN-22');
INSERT INTO LISTAVINHOS VALUES
(4, 1, 6, 'ANIMUS TINTO', TO_DATE('24-JAN-22');
INSERT INTO LISTAVINHOS VALUES
(5, 1, 2, 'ALVARINHO 2017', TO_DATE('25-JAN-22');
```

```
--INSERT NA TABELA OPINIAO

INSERT INTO OPINIAO VALUES
(1, 1, 'PESSIMO', 1, 1, 'JOAO', 1, TO_DATE('15-JAN-22');
INSERT INTO OPINIAO VALUES
(2, 10, 'EXCELENTE', 1, 2, 'JOAO', 1, TO_DATE('01-JAN-22');
INSERT INTO OPINIAO VALUES
(3, 3, 'MAU', 2, 3, 'DUARTE', 1, TO_DATE('18-DEC-21');
INSERT INTO OPINIAO VALUES
(4, 4, 'MAU', 3, 4, 'MANUEL', 1, TO_DATE('19-NOV-21');
INSERT INTO OPINIAO VALUES
(5, 7, 'INTERMEDIO', 4, 1, 'RODRIGO', 1, TO_DATE('18-NOV-21');
```

```
--INSERT NA TABELA PERCENTAGEM

INSERT INTO PERCENTAGEM VALUES
(100, 1, 1);
INSERT INTO PERCENTAGEM VALUES
(50, 2, 10);
INSERT INTO PERCENTAGEM VALUES
(50, 1, 9);
INSERT INTO PERCENTAGEM VALUES
(100, 3, 8);
```