

Resposta Completa / Modelo de Entregáveis da Tarefa 02 – Bootamp DBA

Observações gerais :

1) Como citado no texto, comentários (inteligíveis, que realmente Representam/Apontam características Importantes do objeto) podem ser inseridas tanto no database (via comando COMMENT) quanto no próprio código-fonte SQL - usaremos todas essas opções para fins de Demonstração/Exemplo, mas qqer uma delas seria aceita como resposta

2) Idealmente, TODOS os objetos lógicos e/ou físicos que serão criados em conjunto com as tabelas (tais como Constraints, Índices, Sequences, etc) DEVERIAM ser criados com nomes específicos, seguindo uma regra qualquer de nomenclatura, mesmo uma simples como ter um PREFIXO indicando o tipo do objeto + um nome lógico , similar ao que foi feito por exemplo com as tablespaces e nomes de usuário na tarefa anterior - por uma questão de tempo e para que a resposta não fique muito longa, isso não será feito MAS mostrarei as sintaxes/comandos necessários na criação da tabela de Clientes, para conhecimento.

3) Estamos usando o RDBMS XE (que basicamente é similar a um banco Standard Edition) : nem todas as Editions em todas as versões aceitam Todas as opções de Performance, Segurança e Compressão de dados... Como <https://docs.oracle.com/en/database/oracle/oracle-database/18/xelc/licensing-information.html#GUID-0F2574A6-360F-4237-8098-17B02FFC3BB3> documenta, o XE 18c aceita a feature de Advanced Compression (onde a tabela indicada criada com a feature é compactada on-the-fly, cfrme os dados entram e saem dela), Partition e de In-Memory , mas não outras, como Parallel SQL, heat map, etc - fosse uma situação Produção, o procedimento seria consultar <https://docs.oracle.com/> são encontrados links pra documentação de todas as Editions nas últimas versões, consultar o manual de Licensing para saber se uma determinada feature é permissível ou não na Edition e versão em uso.

NESTE DOCUMENTO, para fins de Demonstração, mesmo sem termos dados completos de aplicabilidade, iremos usar a funcionalidade de Partitioning na tabela de Vendas (assumindo partição por Mês, já que esse é um método comum em dados comerciais) , e de IN-MEMORY na tabela de Produtos (só para servir de Exemplo, não há como saber, apenas com as informações fornecidas, sobre Aplicabilidade - até por esse fato, como dito .

IMPORTANTE : como dito no texto, CASO o participante não esteja seguro e certo da Aplicabilidade de uma dada Feature no modelo fornecido, ao menos Relatar a possibilidade já é suficiente...

4) Para que seja possível reproduzir em qualquer situação, será mostrado aqui o texto completo dos SQLs de Criação das tabelas, Adição de Constraints e códigos PL/SQL : Claro que existem ferramentas gráficas onde vc edita/cria o objeto numa tela gráfica e o software já gera e executa os SQLs e PL/SQLs necessários, o próprio freeware Oracle SQL Developer é uma delas...

Texto Original da Tarefa

O texto original da tarefa era :

"Tarefa 02 - Iniciando os trabalhos : através do usuário APP_OWNER, criar os objetos a serem usados pela aplicação, cfrme Modelagem que será fornecida, E através do usuário de Administração, criar os usuários de banco das pessoas que usarão

a Aplicação, com as pessoas recebendo privilégios em tabelas cfrme os departamentos em que trabalham - preferencialmente, usar ROLES pra agrupar os usuários.

IMPORTANTE :

a) serão MUITO bem-vindos comentários (de boa qualidade, não comentar "Essa tabela é de Clientes", ou "Essa coluna admite 20 caracteres", essas coisas são Óbvias), E aderência à alguma regra de Nomenclatura para triggers, Constraints, Sequences e qquer objetos lógicos que forem necessários.

b) como dificilmente a pessoa que solicita uma implantação de sistemas é um Especialista em RDBMS Oracle, faz parte das atribuições de um DBA conhecer as principais features da versão/release/Edition em uso E sugerir implementar os recursos que julgar importante : no nosso caso, a documentação da nossa versão está em <https://docs.oracle.com/en/database/oracle/oracle-database/18/xelic/licensing-information.html> . Para maximizar a pontuação na entrega dessa tarefa, é solicitado implementar na criação dos objetos as features que julgar pertinente (principalmente Compactação de Dados, Particionamentos, Tempo de permanência de um dado no banco online (há informações, principalmente Comerciais, que não precisam ficar online eternamente), etc... CASO o participante não esteja seguro e certo da Aplicabilidade de uma dada Feature no modelo fornecido, ao menos Relatar a possibilidade

c) hoje em dia um tema muito exigido é adequação à LGPD : para isso, Supondo que os Analistas que estão implementando a App não se atentaram à isso, é solicitado um relatório indicando (pelo nome e conteúdo indicado no modelo) uma lista das colunas que eventualmente poderiam ter dados sensíveis, que sirvam para Identificar unicamente uma pessoa - a Auditoria em si será implementada em tarefa Posterior

d) implementar (através de ROLES, preferencialmente) os acessos restritos Exigidos

e) Como o usuário solicitante não conhece bem as features de preenchimento de dados (como colunas IDENTITY/Sequenciais, triggers, etc) e nem as constraints de checagem de dados e Obrigatoriedade de preenchimento de colunas, as Implementar onde for julgado Adequado

Modelagem (A indica string, 9 indica número, DH indica datas com horas, T indica texto longo, os limites máximos para strings e números serão especificados em parêntesis), PK indica coluna-chave, FK indica coluna-filha) :

"

=> preparação do usuário APP_OWNER, que cfrme dito na tarefa 01 vai ser o donos das tabelas :

```
SYS@db_app::CNTNR=DB_APP> grant create session, create table, create trigger, create sequence to APP_OWNER;
```

Concessão bem-sucedida.

=> criação das tabelas, COM os melhores comentários e práticas

----- tabela de Tipos de Documentos , TB_TIPOS_DOCUMENTOS -----

```
APP_OWNER@db_app::CNTNR=DB_APP> CREATE TABLE TB_TIPOS_DOCUMENTOS(  
  ID_TIPO_DOCUMENTO VARCHAR2(10 CHAR) , -- coluna 1/2 da PK  
  TIPO_PESSOA_DOC CHAR(01 CHAR) , -- coluna 2/2 da PK Pode conter F=Física ou J=Jurídica  
  DESCR_DOCUMENTO VARCHAR2(30 CHAR) -- a descrição pode ter qualquer combinação de caracteres  
  )  
TABLESPACE TS_APP_DADOS ;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> ALTER TABLE TB_TIPOS_DOCUMENTOS  
  ADD CONSTRAINT PK_TIPOS_DOCS PRIMARY KEY (ID_TIPO_DOCUMENTO, TIPO_PESSOA_DOC)  
  USING INDEX TABLESPACE TS_APP_INDEX;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> ALTER TABLE TB_TIPOS_DOCUMENTOS  
  ADD CONSTRAINT CHK_TIPO_PESSOA_DOC CHECK (TIPO_PESSOA_DOC in ('F', 'J'));
```

APP_OWNER@db_app::CNTNR=DB_APP> *comment on table TB_TIPOS_DOCUMENTOS is 'Esta tabela conterà uma linha para cada Tipo de Documento possível de ser Associado tanto a uma pessoa Física (F) quanto a uma pessoa Jurídica (J) e não é previsto nem Particionamento nem compactação, dado o pequeno volume de dados Previsto';*

OBSERVAÇÃO IMPORTANTE : tal como indicado no texto original da tarefa, as colunas strings vão conter a quantidade especifica EM CARACTERES, e NÃO em Bytes - como o database XE que usamos usa um Characterset aonde os caracteres especiais / não-latinos PODEM usar mais que um byte para armazenamento, é Crucial indicar que a quantidade a limitar é em CARACTERES, daí o CHAR usado na especificação das colunas string, tal como por exemplo ID_TIPO_DOCUMENTO VARCHAR2(10 CHAR)

----- tabela de Clientes , TB_CLIENTES -----

==> prossegue a criação das tabelas :

```
APP_OWNER@db_app::CNTNR=DB_APP> CREATE TABLE TB_CLIENTES  
  (  
  CD_CLIENTE NUMBER(6)  
  GENERATED ALWAYS AS IDENTITY  
  CONSTRAINT "NN_CD_CLIENTE" NOT NULL ENABLE,  
  /* Esta coluna CD_CLIENTE, além de ser PK, muito embora não Identificado claramente,  
  pelo tipo de Utilização é facilmente dedutível que é uma Chave Artificial,  
  ie, um valor atribuído PELO SISTEMA, e não digitado pelo usuário : assim  
  sendo, PODE ser usado uma coluna IDENTITY, auto-preenchida pelo Oracle  
  */  
  NOME_COMPLETO VARCHAR2(100 char),  
  NOME_FANTASIA VARCHAR2(40 char) , --
```

```

LOGRADOURO    VARCHAR2(20 char) , --
ENDERECO      VARCHAR2(50 char) , --
NUMERO        NUMBER(6,0)    , --
COMPLEMENTO  VARCHAR2(50 char) , --
BAIRRO        VARCHAR2(30 char) , --
CIDADE        VARCHAR2(60 char) , --
UF            VARCHAR2(02 char) , --
CEP           VARCHAR2(08 char) , --
TIPO_PESSOA   CHAR(01 char) , -- Criada como CHAR para demonstrar que, PARA colunas de um caracter
              -- de comprimento, não há diferença para VARCHAR2
ID_TIPO_DOCUMENTO VARCHAR2(10 char) , -- FK da tabela TB_TIPOS_DOCUMENTOS
DOCUMENTO     VARCHAR2(30 char) , -- Pode conter número de RG, CPF, CNPJ/CGC ou qqer identificador similar
              -- que sirva pra Identificar um Único Cliente
DT_DESATIVACAO DATE          , -- Preenchido se / quando cliente for Inativado
DT_CADASTRO   DATE          , -- Data do Cadastramento original do Cliente
DT_ULT_COMPRA DATE          , -- Data da Última compra feita pelo Cliente, NULL se nunca comprou
QTD_COMPRAS   NUMBER(6,0)
              DEFAULT 0
              CONSTRAINT "NN_QTD_COMPRAS" NOT NULL ENABLE
              )
TABLESPACE TS_APP_DADOS;

```

==> prosseguem o restante das constraints nomeadas na tabela de Clientes e dos índices a serem usados :

```

APP_OWNER@db_app::CNTNR=DB_APP> CREATE INDEX I_CLIENTES_PK on TB_CLIENTES (CD_CLIENTE) TABLESPACE
TS_APP_INDEX ;

```

```

APP_OWNER@db_app::CNTNR=DB_APP> ALTER TABLE TB_CLIENTES ADD CONSTRAINT PK_CLIENTES PRIMARY KEY
(CD_CLIENTE) USING INDEX;

```

```

APP_OWNER@db_app::CNTNR=DB_APP> ALTER TABLE TB_CLIENTES ADD CONSTRAINT FK_TIPO_PESSOA_DOC FOREIGN
KEY (ID_TIPO_DOCUMENTO, TIPO_PESSOA )
REFERENCES TB_TIPOS_DOCUMENTOS (ID_TIPO_DOCUMENTO, TIPO_PESSOA_DOC);

```

```

APP_OWNER@db_app::CNTNR=DB_APP> COMMENT ON TABLE TB_CLIENTES is 'contém toda a carteira de
clientes, Ativos ou Não, que já realizaram compras ou não';

```

```

APP_OWNER@db_app::CNTNR=DB_APP> COMMENT ON COLUMN TB_CLIENTES.QTD_COMPRAS is 'A coluna
QTD_COMPRAS deve conter sempre a qtdade Atualizada de compras do Cliente ou
ZERO se cliente nunca fez compras, servindo Futuramente para alimentar os Relatórios de Clientes que nunca
Compraram
nada ou compraram Pouco, sendo de baixa qualidade. Por causa dessa necessidade de ser mantido Automaticamente, a
trigger
TR_ALIMENTA_QTD_COMPRAS na tabela TB_VENDAS alimentará essa coluna';

```

----- tabela de Produtos a serem Vendidos , TB_PRODUTOS -----

/* Não foi especificado, mas será Assumido, para servir de Exemplo, que a tabela de Produtos é pequena o suficiente MAS acessada intensamente o suficiente para Justificar o uso de IN-MEMORY

Como dito na especificação da tarefa, porém, em Produção talvez o DBA não quisesse se Arriscar sem ter os

VOLUMES de dados preciso, aí poderia ser indicado isso num comentário, o Participante já ganharia os pontos por Conhecer a Feature e saber Indicar a possibilidade de usar ou não

*/

```
CREATE TABLE TB_PRODUTOS
(
  ID_PRODUTO VARCHAR2(10 CHAR),
  DESCR_PRODUTO VARCHAR2(50 CHAR),
  PRIMARY KEY (ID_PRODUTO) USING INDEX PCTFREE 0 TABLESPACE TS_APP_INDEX ENABLE
) SEGMENT CREATION DEFERRED PCTFREE 0 PCTUSED 99
ROW STORE COMPRESS ADVANCED LOGGING TABLESPACE TS_APP_DADOS INMEMORY PRIORITY NONE
MEMCOMPRESS FOR DML
DISTRIBUTE AUTO NO DUPLICATE ;
```

--

COMMENT ON TABLE TB_PRODUTOS IS 'Tabela de Produtos passíveis de Vendas, deve ter Pequeno volume de dados, e só sofre INSERTs';

----- tabela de Vendas , TB_VENDAS -----

/* Obs : embora não indicado especificamente, a tabela de vendas deve ser a maior (já que uma Empresa lucrativa tem muito mais vendas do que produtos em estoque, ou clientes, etc) , portanto NELA serão usados os recursos de particionamento (particionamento Automático por Ano/Mês na data de venda, é um critério que faz Sentido), visando principalmente a poder Arquivar / Remover dados antigos, E será usada a Compactação avançada, que o XE permite

*/

-- Ainda para fins Organizacionais, o CLOB terá tablespace própria (criar e permissionar como usuário Administrador) :

```
SYSTEM@db_app::CNTNR=DB_APP> CREATE TABLESPACE TS_CLOB_OBS_VENDAS datafile
'/opt/oracle/oradata/XE/DB_APP/ts_clob_obs_vendas_01.dbf' size 300M;
```

```
SYSTEM@db_app::CNTNR=DB_APP> alter user APP_OWNER QUOTA UNLIMITED ON TS_CLOB_OBS_VENDAS;
```

-- prossegue a criação das tabelas, como o usuário APP_OWNER - como intes extras nesse CREATE, serão usados :

-- => LOB em tablespace próprio

-- => particionamento Automático, onde o próprio RDBMS cria as novas partições cfrme Preciso, mas com nome

-- gerado pelo sistema, usar ALTER TABLE RENAME PARTITION posteriormente se desejado

-- => tabela será compactada Automaticamente

```
APP_OWNER@db_app::CNTNR=DB_APP> create sequence SEQ_VENDAS;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> CREATE TABLE TB_VENDAS (
  NUM_VENDA NUMBER(8,0) DEFAULT SEQ_VENDAS.NEXTVAL,
  DT_VENDA DATE,
  CD_CLIENTE NUMBER(6,0),
```

```

NUM_NF    NUMBER(5,0),
OBS      CLOB,
CONSTRAINT PK_VENDAS PRIMARY KEY (NUM_VENDA)
USING INDEX PCTFREE 0 TABLESPACE TS_APP_INDEX ENABLE,
CONSTRAINT FK_CLIENTE FOREIGN KEY (CD_CLIENTE)
REFERENCES TB_CLIENTES (CD_CLIENTE) ENABLE
)
PCTFREE 0 PCTUSED 99 ROW STORE COMPRESS ADVANCED TABLESPACE TS_APP_DADOS
LOB (OBS) STORE AS SECUREFILE SF_CLOB_OBS_VENDAS (TABLESPACE TS_CLOB_OBS_VENDAS DISABLE STORAGE IN
ROW RETENTION AUTO CACHE LOGGING COMPRESS HIGH )
PARTITION BY RANGE (DT_VENDA) INTERVAL (NUMTOYMINTERVAL('1','MONTH'))
(PARTITION P_VENDA_01 VALUES LESS THAN (TO_DATE(' 2021-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN'))PCTFREE 0 PCTUSED 99 INTRANS 1 MAXTRANS 255 ROW STORE COMPRESS ADVANCED
TABLESPACE TS_APP_DADOS LOB (OBS) STORE AS SECUREFILE (TABLESPACE TS_CLOB_OBS_VENDAS DISABLE STORAGE
IN ROW RETENTION AUTOCACHE LOGGING COMPRESS HIGH ) );

```

APP_OWNER@db_app::CNTNR=DB_APP> COMMENT ON TABLE TB_VENDAS IS 'Tabela que vai registrar Todas as vendas Efetuadas - será criada com PCTFREE 0 pois nunca haverá UPDATES ';

-- trigger que Atualiza a coluna QTD_VENDAS do cliente que fez a venda

```

APP_OWNER@db_app::CNTNR=DB_APP> create or replace trigger TR_ALIMENTA_QTD_COMPRAS AFTER INSERT OR
DELETE on TB_VENDAS for each row
BEGIN
  if INSERTING then
    update TB_CLIENTES set QTD_COMPRAS = QTD_COMPRAS + 1,
                        DT_ULT_COMPRA = SYSDATE
    WHERE CD_CLIENTE = :new.CD_CLIENTE;
  elsif DELETING then
    update TB_CLIENTES set QTD_COMPRAS = QTD_COMPRAS -1 WHERE CD_CLIENTE = :old.CD_CLIENTE;
  end if;
END;

```

----- tabela de Itens de Vendas , TB_ITENS_VENDAS -----

```

APP_OWNER@db_app::CNTNR=DB_APP> CREATE TABLE TB_ITENS_VENDAS (
  DT_VENDA    DATE,
  NUM_VENDA   NUMBER(8,0) NOT NULL ENABLE,
  NUM_ITEM_VENDA NUMBER(3,0) NOT NULL ENABLE,
  ID_PRODUTO  VARCHAR2(10 CHAR),
  VALOR_DO_ITEM NUMBER(6,2),
  CONSTRAINT TB_ITENS_VENDAS_PK PRIMARY KEY (NUM_VENDA, NUM_ITEM_VENDA)
USING INDEX PCTFREE 10 INTRANS 2 MAXTRANS 255 COMPUTE STATISTICS
TABLESPACE TS_APP_DADOS ENABLE,
CONSTRAINT FK_ITENS_VENDAS FOREIGN KEY (NUM_VENDA)
REFERENCES TB_VENDAS (NUM_VENDA) ENABLE,
CONSTRAINT FK_PRODUTOS FOREIGN KEY (ID_PRODUTO)
REFERENCES TB_PRODUTOS (ID_PRODUTO)
)
PCTFREE 0 PCTUSED 99 ROW STORE COMPRESS ADVANCED TABLESPACE TS_APP_DADOS
PARTITION BY RANGE (DT_VENDA) INTERVAL (NUMTOYMINTERVAL('1','MONTH'))
(PARTITION P_VENDA_ITEM_01 VALUES LESS THAN (TO_DATE(' 2021-01-01 00:00:00', 'SYYYY-MM-DD HH24:MI:SS',
'NLS_CALENDAR=GREGORIAN'))
PCTFREE 0 PCTUSED 99 ROW STORE COMPRESS ADVANCED

```

```
TABLESPACE TS_APP_DADOS
);
;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> COMMENT ON TABLE TB_ITENS_VENDAS IS 'Esta tabela guardará os Itens das vendas, seguindo os mesmos ditamos de Armazenamento da tabela de VENDAS.';
```

----- todas as tabelas já foram criadas neste ponto, serão inseridas algumas linhas para testes -----

```
APP_OWNER@db_app::CNTNR=DB_APP> insert into TB_PRODUTOS values('P01', 'Produto 1: tablet LG');
```

```
APP_OWNER@db_app::CNTNR=DB_APP> insert into TB_PRODUTOS values('P02', 'Produto 2 : TV Samsung K6000');
```

```
APP_OWNER@db_app::CNTNR=DB_APP> commit;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_TIPOS_DOCUMENTOS VALUES('RG' , 'F', 'DOCUMENTO RG para PFs');
```

```
APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_TIPOS_DOCUMENTOS VALUES('CNPJ', 'J', 'DOCUMENTO CNPJ, para PJs');
```

```
APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_TIPOS_DOCUMENTOS VALUES('CGC' , 'J', 'DOCUMENTO CGC, para PJs');
```

```
APP_OWNER@db_app::CNTNR=DB_APP> commit;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_CLIENTES (
```

```
  NOME_COMPLETO , -- VARCHAR2(100)
```

```
  NOME_FANTASIA , -- VARCHAR2(40)
```

```
  LOGRADOURO , -- VARCHAR2(20)
```

```
  ENDERECO , -- VARCHAR2(50)
```

```
  NUMERO , -- NUMBER(6)
```

```
  COMPLEMENTO , -- VARCHAR2(50)
```

```
  BAIRRO , -- VARCHAR2(30)
```

```
  CIDADE , -- VARCHAR2(60)
```

```
  UF , -- VARCHAR2(2)
```

```
  CEP , -- VARCHAR2(8)
```

```
  TIPO_PESSOA , -- CHAR(1)
```

```
  ID_TIPO_DOCUMENTO, -- VARCHAR2(10)
```

```
  DOCUMENTO , -- VARCHAR2(30)
```

```
  DT_CADASTRO -- DATE
```

```
)
```

```
VALUES (
```

```
'JOSE LAURINDO CHIAPPA', -- NOME_COMPLETO - VARCHAR2(100)
```

```
'DBA CHIAPPA' , -- NOME_FANTASIA - VARCHAR2(40)
```

```
'RUA' , -- LOGRADOURO - VARCHAR2(20)
```

```
'CHAGÚ' , -- ENDERECO - VARCHAR2(50)
```

```
193 , -- NUMERO - NUMBER(6)
```

```
'Apto 102' , -- COMPLEMENTO - VARCHAR2(50)
```

```
'VILA FORMOSA' , -- BAIRRO - VARCHAR2(30)
```

```
'São Paulo' , -- CIDADE - VARCHAR2(60)
```

```
'SP' , -- UF - VARCHAR2(2)
```

```
'03357020' , -- CEP - VARCHAR2(8)
```

```

'F'          , -- TIPO_PESSOA   - CHAR(1)
'RG'        , -- ID_TIPO_DOCUMENTO - VARCHAR2(10)
'99.123.456-00' , -- DOCUMENTO    - VARCHAR2(30)
SYSDATE      -- DT_CADASTRO     - DATE
);
APP_OWNER@db_app::CNTNR=DB_APP> commit;

-- Confirmar que coluna preenchida :

APP_OWNER@db_app::CNTNR=DB_APP> SELECT * FROM TB_CLIENTES;
CD_CLIENTE NOME_COMPLETO    NOME_FANTASIA LOGRADOURO ENDERECO NUMERO COMPLEMENTO BAIRRO
CIDADE   UF CEP    T ID_TIPO_DO DOCUMENTO          DT_DESATIVACAO      DT_CADASTRO
DT_ULT_COMPRA      QTD_COMPRAS
-----
-----
1 JOSE LAURINDO CHIAPPA DBA CHIAPPA RUA    CHAGÚ    193 Apto 102 VILA FORMOSA São Paulo SP
03357020 F RG      99.123.456-00                23/09/2021 20:24:00                0

-- Criar uma venda para esse Cliente com um dos Produtos cadastrados :

APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_VENDAS (
    DT_VENDA , -- DATE
    CD_CLIENTE, -- NUMBER(6,0)
    NUM_NF , -- NUMBER(5,0)
    OBS     -- CLOB
)
VALUES (
    SYSDATE,
    1,
    00001,
    'Primeira Venda feita para o Cliente DBA CHIAPPA'
);
APP_OWNER@db_app::CNTNR=DB_APP> commit;

-- Confirmar que a trigger Alimentou a Qtdade de compras para o cliente que fez uma compra

APP_OWNER@db_app::CNTNR=DB_APP> select * from TB_CLIENTES where CD_CLIENTE=1;
CD_CLIENTE NOME_COMPLETO    NOME_FANTASIA LOGRADOURO ENDERECO NUMERO COMPLEMENTO BAIRRO
CIDADE   UF CEP    T ID_TIPO_DO DOCUMENTO          DT_DESATIVACAO      DT_CADASTRO
DT_ULT_COMPRA      QTD_COMPRAS
-----
-----
1 JOSE LAURINDO CHIAPPA DBA CHIAPPA RUA    CHAGÚ    193 Apto 102 VILA FORMOSA São Paulo SP
03357020 F RG      99.123.456-00                23/09/2021 20:24:00 23/09/2021 20:39:53    1

-- Inserir os 2 Itens que foram Comprados na Venda

APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_ITENS_VENDAS VALUES(SYSDATE, SEQ_VENDAS.CURRVAL, 1,
'P01', 400.50);

APP_OWNER@db_app::CNTNR=DB_APP> INSERT INTO TB_ITENS_VENDAS VALUES(SYSDATE, SEQ_VENDAS.CURRVAL, 2,
'P02', 1500.13);

```

```
APP_OWNER@db_app::CNTNR=DB_APP> commit;
```

=====

=> OBSERVAÇÃO : quando se usa um indicador de NOT NULL na criação da tabela SEM indicar nome da constraints, é implicitamente criada uma constraint com nome default SYS_xxx, depois das tabelas todas criadas (onde haverá algumas criadas dessa maneira) , conferir isso com a consulta :

=====

```
select table_name, constraint_name, search_condition, constraint_type from user_constraints;
```

===== Relatório de dados Sensíveis/Confidenciais
=====

O item c) , prevendo futura adequação à LGPD, pede que seja providenciado um relatório indicando (pelo nome e conteúdo indicado no modelo) uma lista das colunas que eventualmente poderiam ter dados sensíveis, que sirvam para Identificar unicamente uma pessoa – a Auditoria em si será implementada em tarefa Posterior.

Segue :

coluna DOCUMENTO na tabela TB_CLIENTES : um documento (seja RG, CPF, CNPJ/CGC, qualquer um) necessariamente Identifica Unicamente uma pessoa, portanto é um dado Sensível : deve ser encriptografado, acessado apenas por quem de direito E

necessariamente o Dono do dado NECESSITA haver feito Autorização para isso.

===== GRANTS das tabelas para os usuários finais
=====

Acesso (LEMBRANDO que as ROLES já foram criadas E concedidas aos usuários na tarefa anterior) :

a1) o usuário PATINHAS (Gerente-geral) acessa/lê e faz DML de qualquer tipo em qualquer tabela da modelagem (não em qualquer tabela do database) :

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE ON TB_PRODUTOS to GERENTE;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE ON TB_TIPOS_DOCUMENTOS to GERENTE;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE ON TB_CLIENTES to GERENTE;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE ON TB_VENDAS to GERENTE;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE ON TB_ITENS_VENDAS to GERENTE;
```

a2) os usuários do setor Contabilidade só consultam tabelas de VENDAS e ITENS DE VENDAS

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT ON TB_VENDAS TO CONTABILIDADE;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT ON TB_ITENS_VENDAS TO CONTABILIDADE;
```

a3) usuários do setor VENDAS/COMERCIAL fazem Queries e DMLs nas tabelas de VENDAS e ITENS DE VENDAS, e Queries + SELECTs + DMLs na tabela TB_PRODUTOS

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE on TB_VENDAS TO VENDAS;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE on TB_ITENS_VENDAS TO VENDAS;
```

```
APP_OWNER@db_app::CNTNR=DB_APP> GRANT SELECT, INSERT, UPDATE, DELETE on TB_PRODUTOS TO VENDAS;
```

==> Comprovantes de privilégios, usando o script privs_by_user.sql anteriormente fornecido :

```
SYSTEM@db_app::CNTNR=DB_APP> @privs_by_user
```

Enter Username : *patinhas*

Roles granted to user

```
GRANTED_ROLE      ADM DEF
```

```
-----  
GERENTE           NO YES
```

1 linha selecionada.

Table Privileges granted to a user through roles

```
GRANTED_ROLE      OWNER      TABLE_NAME      PRIVILEGE
```

```
-----  
GERENTE           APP_OWNER  TB_CLIENTES      DELETE  
GERENTE           APP_OWNER  TB_CLIENTES      INSERT  
GERENTE           APP_OWNER  TB_CLIENTES      SELECT  
GERENTE           APP_OWNER  TB_CLIENTES      UPDATE
```

GERENTE	APP_OWNER	TB_ITENS_VENDAS	DELETE
GERENTE	APP_OWNER	TB_ITENS_VENDAS	INSERT
GERENTE	APP_OWNER	TB_ITENS_VENDAS	SELECT
GERENTE	APP_OWNER	TB_ITENS_VENDAS	UPDATE
GERENTE	APP_OWNER	TB_PRODUTOS	DELETE
GERENTE	APP_OWNER	TB_PRODUTOS	INSERT
GERENTE	APP_OWNER	TB_PRODUTOS	SELECT
GERENTE	APP_OWNER	TB_PRODUTOS	UPDATE
GERENTE	APP_OWNER	TB_TIPOS_DOCUMENTOS	DELETE
GERENTE	APP_OWNER	TB_TIPOS_DOCUMENTOS	INSERT
GERENTE	APP_OWNER	TB_TIPOS_DOCUMENTOS	SELECT
GERENTE	APP_OWNER	TB_TIPOS_DOCUMENTOS	UPDATE
GERENTE	APP_OWNER	TB_VENDAS	DELETE
GERENTE	APP_OWNER	TB_VENDAS	INSERT
GERENTE	APP_OWNER	TB_VENDAS	SELECT
GERENTE	APP_OWNER	TB_VENDAS	UPDATE

20 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

não há linhas selecionadas

SYSTEM@db_app::CNTNR=DB_APP> *@privs_by_user*

Enter Username : *huguinho*

Roles granted to user

GRANTED_ROLE	ADM DEF
--------------	---------

CONTABILIDADE	NO YES
---------------	--------

1 linha selecionada.

Table Privileges granted to a user through roles

GRANTED_ROLE	OWNER	TABLE_NAME	PRIVILEGE
--------------	-------	------------	-----------

CONTABILIDADE	APP_OWNER	TB_ITENS_VENDAS	SELECT
---------------	-----------	-----------------	--------

CONTABILIDADE	APP_OWNER	TB_VENDAS	SELECT
---------------	-----------	-----------	--------

2 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

PRIVILEGE	ADM
-----------	-----

CREATE SESSION	NO
----------------	----

1 linha selecionada.

SYSTEM@db_app::CNTNR=DB_APP> @privs_by_user

Enter Username : *zezinho*

Roles granted to user

GRANTED_ROLE	ADM DEF

CONTABILIDADE	NO YES

1 linha selecionada.

Table Privileges granted to a user through roles

GRANTED_ROLE	OWNER	TABLE_NAME	PRIVILEGE

CONTABILIDADE	APP_OWNER	TB_ITENS_VENDAS	SELECT
CONTABILIDADE	APP_OWNER	TB_VENDAS	SELECT

2 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

PRIVILEGE	ADM

CREATE SESSION	NO

1 linha selecionada.

SYSTEM@db_app::CNTNR=DB_APP> @privs_by_user

Enter Username : *luisinho*

Roles granted to user

GRANTED_ROLE	ADM DEF
--------------	---------

CONTABILIDADE	NO YES
---------------	--------

1 linha selecionada.

Table Privileges granted to a user through roles

GRANTED_ROLE	OWNER	TABLE_NAME	PRIVILEGE
--------------	-------	------------	-----------

CONTABILIDADE	APP_OWNER	TB_ITENS_VENDAS	SELECT
---------------	-----------	-----------------	--------

CONTABILIDADE	APP_OWNER	TB_VENDAS	SELECT
---------------	-----------	-----------	--------

2 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

PRIVILEGE	ADM
-----------	-----

CREATE SESSION NO

1 linha selecionada.

SYSTEM@db_app::CNTNR=DB_APP> @privs_by_user

Enter Username : *peninha*

Roles granted to user

GRANTED_ROLE	ADM DEF
--------------	---------

VENDAS	NO YES
--------	--------

1 linha selecionada.

Table Privileges granted to a user through roles

GRANTED_ROLE	OWNER	TABLE_NAME	PRIVILEGE
--------------	-------	------------	-----------

VENDAS	APP_OWNER	TB_ITENS_VENDAS	DELETE
VENDAS	APP_OWNER	TB_ITENS_VENDAS	INSERT
VENDAS	APP_OWNER	TB_ITENS_VENDAS	SELECT
VENDAS	APP_OWNER	TB_ITENS_VENDAS	UPDATE
VENDAS	APP_OWNER	TB_PRODUTOS	DELETE
VENDAS	APP_OWNER	TB_PRODUTOS	INSERT
VENDAS	APP_OWNER	TB_PRODUTOS	SELECT
VENDAS	APP_OWNER	TB_PRODUTOS	UPDATE
VENDAS	APP_OWNER	TB_VENDAS	DELETE
VENDAS	APP_OWNER	TB_VENDAS	INSERT
VENDAS	APP_OWNER	TB_VENDAS	SELECT
VENDAS	APP_OWNER	TB_VENDAS	UPDATE

12 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

PRIVILEGE	ADM
CREATE SESSION	NO

1 linha selecionada.

SYSTEM@db_app::CNTNR=DB_APP> @privs_by_user

Enter Username : *donald*

Roles granted to user

GRANTED_ROLE	ADM DEF
VENDAS	NO YES

1 linha selecionada.

Table Privileges granted to a user through roles

GRANTED_ROLE	OWNER	TABLE_NAME	PRIVILEGE
VENDAS	APP_OWNER	TB_ITENS_VENDAS	DELETE
VENDAS	APP_OWNER	TB_ITENS_VENDAS	INSERT
VENDAS	APP_OWNER	TB_ITENS_VENDAS	SELECT

VENDAS	APP_OWNER	TB_ITENS_VENDAS	UPDATE
VENDAS	APP_OWNER	TB_PRODUTOS	DELETE
VENDAS	APP_OWNER	TB_PRODUTOS	INSERT
VENDAS	APP_OWNER	TB_PRODUTOS	SELECT
VENDAS	APP_OWNER	TB_PRODUTOS	UPDATE
VENDAS	APP_OWNER	TB_VENDAS	DELETE
VENDAS	APP_OWNER	TB_VENDAS	INSERT
VENDAS	APP_OWNER	TB_VENDAS	SELECT
VENDAS	APP_OWNER	TB_VENDAS	UPDATE

12 linhas selecionadas.

System Privileges assigned to a user through roles

não há linhas selecionadas

Table privileges assigned directly to a user

não há linhas selecionadas

System privileges assigned directly to a user

PRIVILEGE	ADM

CREATE SESSION	NO

1 linha selecionada.

SYSTEM@db_app::CNTNR=DB_APP>