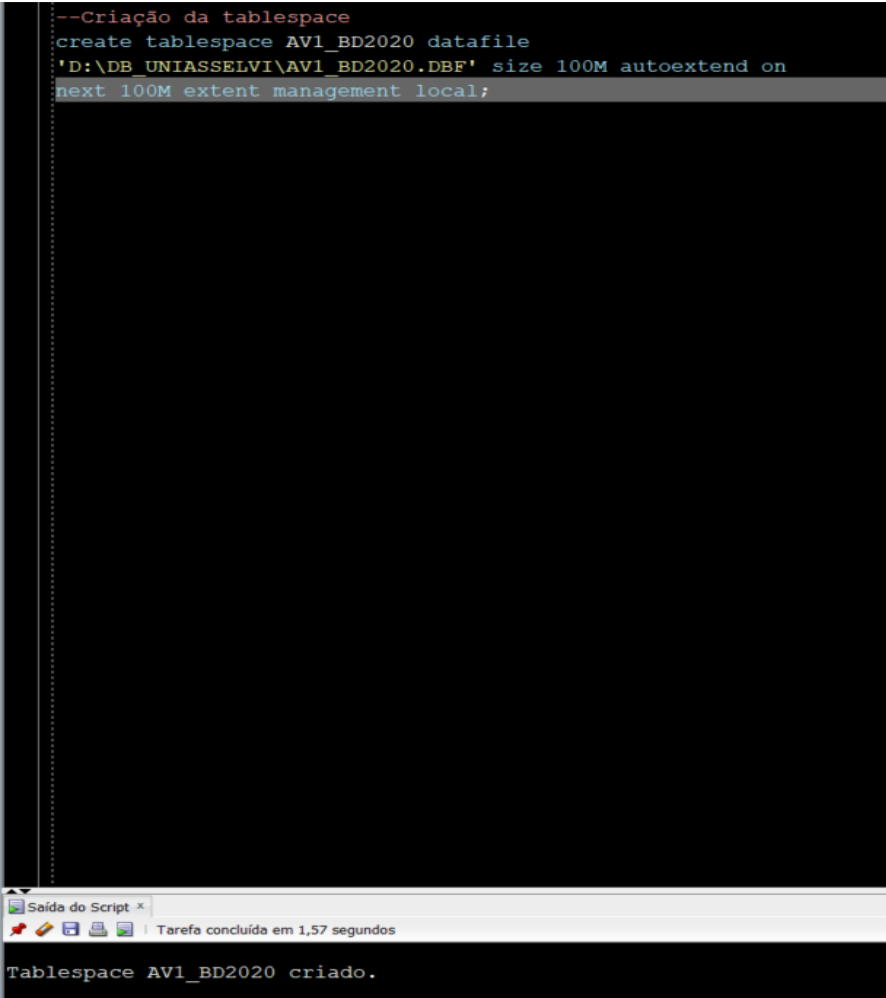


AV1 Banco de Dados Avançado

Nome: Matheus Ludtke da Silva

1 – Criação da tablespace



The screenshot shows a SQL script execution window with a black background and white text. The script contains the following commands:

```
--Criação da tablespace  
create tablespace AV1_BD2020 datafile  
'D:\DB_UNIASSELVI\AV1_BD2020.DBF' size 100M autoextend on  
next 100M extent management local;
```

Below the script, a status bar indicates the task is completed in 1.57 seconds. At the bottom, the output of the script is displayed:

```
Tablespace AV1_BD2020 criado.
```

2 – Criação do usuário “adm” e liberação de acesso.

```
create user adm identified by 1234
default tablespace AV1_BD2020
quota unlimited on AV1_BD2020;

--Liberação de acesso para o user adm
grant create session, alter session,
create table, create procedure,
create view, create trigger,
create synonym, create sequence
to adm;
```

Saída do Script x | Tarefa concluída em 0,117 segundos

User ADM criado.

Grant bem-sucedido.

3 – Criação das tabelas

3.1 – Criação da tabela empresa_conveniada

```
create table empresa_conveniada ( codempresa Number(5) not null,
bairro Varchar(25),
cep Number(9),
cidade Varchar(25),
cnpj Number(18),
email Varchar(100),
endereco Varchar(35),
estado Varchar(2),
fax Number(16),
insc_est Number(18),
nome Varchar(80),
site Varchar(100),
tel Number(16),
constraint pk_empresa primary key
```

Saída do Script x | Resultado da Consulta x | Tarefa concluída em 0,274 segundos

Table EMPRESA_CONVENIADA criado.

3.2 – Criação da tabela cliente

```
create table cliente ( codcli Number(5) not null,
                      codempresa Number(5),
                      constraint fk_empresa
                      foreign key (codempresa)
                      references empresa_conveniada(codempresa),
                      constraint pk_cliente primary key (codcli),
                      bairro Varchar(25),
                      cel Number(16),
                      cidade Varchar(30),
                      cpf Number(14),
                      data_nasc Date,
                      email Varchar(100),
                      endereco Varchar(35),
                      estado Varchar(2),
                      nome Varchar(50),
                      obs Varchar(50),
                      rg Number(12),
                      sexo Varchar(1),
                      tel Number(16));
```

Saída do Script * Resultado da Consulta *
Tarefa concluída em 0,069 segundos

Table CLIENTE criado.

3.3 – Criação da tabela usuario

```
create table usuario ( codusuario Number(15) not null,
                      bairro Varchar(25),
                      cel Number(16),
                      cep Number(9),
                      cidade Varchar(30),
                      data_nasc Date,
                      email Varchar(100),
                      endereco Varchar(35),
                      estado Varchar(2),
                      nome Varchar(50),
                      tel Number(16),
                      constraint pk_usuario primary key
```

Saída do Script * Resultado da Consulta *
Tarefa concluída em 0,06 segundos

Table USUARIO criado.

3.4 Criação das tabelas classe,secao e portaria

```
create table classe (  codclasse Number(5) not null,  
                       nome Varchar(30),  
                       constraint pk_classe primary key (codclasse));  
  
create table secao (  codsecao Number(5) not null,  
                     nome Varchar(30),  
                     constraint pk_secao primary key (codsecao));  
  
create table portaria ( codportaria Number(5) not null,  
                       nome Varchar(80),  
                       reter_receita Varchar(20),  
                       constraint pk_portaria primary key (codportaria));
```

Table PORTARIA criado.

Table SECAO criado.

Table CLASSE criado.

3.5 Criação das tabelas fornecedor e fabricante

```
create table fornecedor ( codforn Number(5) not null,  
    bairro Varchar(25),  
    cep Number(9),  
    cidade Varchar(25),  
    cnpj Number(18),  
    email Varchar(100),  
    endereco Varchar(35),  
    estado Varchar(2),  
    fax Number(16),  
    insc_est Number(18),  
    nome Varchar(80),  
    site Varchar(100),  
    tel Number(16),  
    constraint pk_fornecedor primary key (codforn));  
  
create table fabricante ( codfab Number(5) not null,  
    bairro Varchar(25),  
    cep Number(9),  
    cidade Varchar(25),  
    cnpj Number(18),  
    email Varchar(100),  
    endereco Varchar(35),  
    estado Varchar(2),  
    fax Number(16),  
    insc_est Number(18),  
    nome Varchar(80),  
    site Varchar(100),  
    tel Number(16),  
    constraint pk_fabricante primary key (codfab));
```

Saída do Script x Resultado da Consulta x
Tarefa concluída em 0,057 segundos

Table FORNECEDOR criado.

Table FABRICANTE criado.

3.6 Criação das tabelas produto e estoque

```
create table produto ( codprod Number(25),
                        codclasse Number(5),
                        codfab Number(5),
                        codportaria Number(5),
                        codsecao Number(5),
                        margem_lucro Number(7,2),
                        nome Varchar(30),
                        preco_fab Number(6,2),
                        preco_venda Number(6,2),
                        qtde_caixa Number(3),
                        constraint fk_classe_produto
                        foreign key (codclasse)
                        references classe(codclasse),
                        constraint fk_fabricante_produto
                        foreign key (codfab)
                        references fabricante(codfab),
                        constraint fk_portaria_produto
                        foreign key (codportaria)
                        references portaria(codportaria),
                        constraint fk_secao_produto
                        foreign key (codsecao)
                        references secao(codsecao),
                        constraint pk_produto primary key (codprod));

create table estoque ( codestoque Number(15) not null,
                        codprod Number(5),
                        constraint fk_produto_estoque
                        foreign key (codprod)
                        references produto(codprod),
                        constraint pk_estoque primary key (codestoque),
                        quantidade Number(15));
```

Salida do Script x Resultado da Consulta x

Tarefa concluída em 0,05 segundos

Table PRODUTO criado.

Table ESTOQUE criado.

3.7 Criação da tabela venda

```
create table venda ( codvenda Number(5) not null,
                    valor_venda Number(9,2),
                    codcli Number(5),
                    codempresa Number(5),
                    codusuario Number(5),
                    dt_venda Date,
                    constraint fk_cliente_venda
                    foreign key (codcli)
                    references cliente(codcli),
                    constraint fk_empresa_venda
                    foreign key (codempresa)
                    references empresa_conveniada(codempresa),
                    constraint fk_usuario_venda
                    foreign key (codusuario)
                    references usuario(codusuario),
                    constraint pk_venda primary key (codvenda));
```

Table VENDA criado.

3.8 Criação das tabelas venda_prod e forn_prod

```
create table prod_venda ( codvenda Number(5) not null,
                        codprod Number(5),
                        valor_prod_venda Number(9,2),
                        quantidade Number(4),
                        codcli Number(5),
                        codempresa Number(5),
                        codusuario Number(5),
                        dt_venda Date,
                        constraint fk_venda_capa
                        foreign key (codvenda)
                        references venda(codvenda),
                        constraint fk_produto_venda
                        foreign key (codprod)
                        references produto(codprod),
                        constraint pk_venda_produto primary key (codvenda));

create table forn_prod ( codforn Number(5) not null,
                        codprod Number(5) not null,
                        constraint fk_prod_forn_forn
                        foreign key (codforn)
                        references fornecedor(codforn),
                        constraint fk_prod_forn_prod
                        foreign key (codprod)
                        references produto(codprod));
```

Table PROD_VENDA criado.

Table FORN_PROD criado.

3.8 Criação das tabelas entrada_nota e prod_nota

```
create table entrada_nota ( codnota Number(5) not null,
                           codforn Number(5),
                           codusuario Number(5),
                           dt_emissao Date,
                           dt_entrada Date,
                           valor_nota Number(9,2),
                           constraint fk_entrada_forn
                           foreign key (codforn)
                           references fornecedor(codforn),
                           constraint fk_entrada_usuario
                           foreign key (codusuario)
                           references usuario(codusuario),
                           constraint pk_entrada primary key (codnota));

create table prod_nota ( codnota Number(5) not null,
                        codprod Number(5),
                        valor_prod_nota Number(9,2),
                        quantidade Number(5),
                        constraint fk_prod_nota_item
                        foreign key (codnota)
                        references entrada_nota(codnota),
                        constraint fk_prod_nota_prod
                        foreign key (codprod)
                        references produto(codprod));
```

Salida do Script x Resultado da Consulta x
Tarefa concluída em 0,073 segundos

Table ENTRADA_NOTA criado.

Table PROD_NOTA criado.

4 – Compilação das procedures para inserção, deleção e atualização dos dados.

4.1 – Compilação da pr_empresa_conveniada

```
CREATE OR REPLACE PROCEDURE pr_empresa_conveniada (
    vOpr CHAR,
    vcodempresa Number,
    vbairro Varchar,
    vcep Number,
    vcidade Varchar,
    vcnpj Number,
    vemail Varchar,
    vendereco Varchar,
    vestado Varchar,
    vfax Number,
    vinsc_est Number,
    vnome Varchar,
    vsite Varchar,
    vtel Number
)
IS
    vExecao EXCEPTION;
BEGIN
    IF (vOpr = 'I') THEN
```

Resultado da Consulta x Salida do Script x
Tarefa concluída em 0,061 segundos

Procedure PR_EMPRESA_CONVENIADA compilado

4.2 – Compilação da pr_cliente

```
CREATE OR REPLACE PROCEDURE pr_cliente (  
    vOpr CHAR,  
    vcodcli Number,  
    vcodempresa Number,  
    vbairro Varchar,  
    vcel Number,  
    vcidade Varchar,  
    vcpf Number,  
    vdata_nasc Date,  
    vemail Varchar,  
    vendereco Varchar,  
    vestado Varchar,  
    vnome Varchar,  
    vobs Varchar,  
    vrg Number,  
    vsexo Varchar,  
    vtel Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN
```

Resultado da Consulta x Saída do Script x

Tarefa concluída em 0,055 segundos

Procedure PR_CLIENTE compilado

4.3 – Compilação da pr_usuario

```
CREATE OR REPLACE PROCEDURE pr_usuario (  
    vOpr CHAR,  
    vcodusuario Number,  
    vbairro Varchar,  
    vcel Number,  
    vcep Number,  
    vcidade Varchar,  
    vdata_nasc Date,  
    vemail Varchar,  
    vendereco Varchar,  
    vestado Varchar,  
    vnome Varchar,  
    vtel Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
        INSERT INTO adm.usuario(  
            vcodusuario
```

Resultado da Consulta x Saída do Script x

Tarefa concluída em 0,05 segundos

Procedure PR_USUARIO compilado

4.4 – Compilação da pr_fornecedor

```
CREATE OR REPLACE PROCEDURE pr_fornecedor (  
    vOpr CHAR,  
    vcodforn Number,  
    vbairro Varchar,  
    vcep Number,  
    vcidade Varchar,  
    vcnpj Number,  
    vemail Varchar,  
    vendereco Varchar,  
    vestado Varchar,  
    vfax Number,  
    vinsc_est Number,  
    vnome Varchar,  
    vsite Varchar,  
    vtel Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,037 segundos

Procedure PR_FORNECEDOR compilado

4.5 – Compilação da pr_classe, pr_secao e pr_portaria

```
CREATE OR REPLACE PROCEDURE pr_classe (  
    vOpr CHAR,  
    vcodclasse Number,  
    vnome Varchar  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
        INSERT INTO adm.classe(  
            codclasse,  
            nome)  
        VALUES(  
            vcodclasse,  
            vnome);  
    ELSE
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,034 segundos

Procedure PR_PORTARIA compilado

Procedure PR_SECAO compilado

Procedure PR_CLASSE compilado

4.6 – Compilação da pr_fabricante

```
CREATE OR REPLACE PROCEDURE pr_fabricante (  
    vOpr CHAR,  
    vcodfab Number,  
    vbairro Varchar,  
    vcep Number,  
    vcidade Varchar,  
    vcnpj Number,  
    vemail Varchar,  
    vendereco Varchar,  
    vestado Varchar,  
    vfax Number,  
    vinsc_est Number,  
    vnome Varchar,  
    vsite Varchar,  
    vtel Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,056 segundos

Procedure PR_FABRICANTE compilado

4.7 – Compilação da pr_produto

```
CREATE OR REPLACE PROCEDURE pr_produto (  
    vOpr CHAR,  
    vcodprod Number,  
    vcodclasse Number,  
    vcodfab Number,  
    vcodportaria Number,  
    vcodsecao Number,  
    vmargem_lucro Number,  
    vnome Varchar,  
    vpreco_fab Number,  
    vpreco_venda Number,  
    vqtde_caixa Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
        INSERT INTO ade_produto (  

```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,044 segundos

Procedure PR_PRODUTO compilado

4.8 – Compilação da pr_estoque

```
CREATE OR REPLACE PROCEDURE pr_estoque (  
    vOpr CHAR,  
    vcodestoque Number,  
    vcodprod Number,  
    vquantidade Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.estoque(  
            codestoque,  
            codprod,  
            quantidade)  
        VALUES(  
            vcodestoque,  
            vcodprod,  
            vquantidade);  
    END IF;  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,051 segundos

Procedure PR_ESTOQUE compilado

4.9 – Compilação da pr_venda

```
CREATE OR REPLACE PROCEDURE pr_venda (  
    vOpr CHAR,  
    vcodvenda Number,  
    vvalor_venda Number,  
    vcodcli Number,  
    vcodempresa Number,  
    vcodusuario Number,  
    vdt_venda Date  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.venda(  
            codvenda,  
            valor_venda,  
            codcli,  
            codempresa,  
            codusuario,  
            vdt_venda)  
        VALUES(  
            vcodvenda,  
            vvalor_venda,  
            vcodcli,  
            vcodempresa,  
            vcodusuario,  
            vdt_venda);  
    END IF;  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,047 segundos

Procedure PR_VENDA compilado

4.10 – Compilação da pr_prod_venda

```
CREATE OR REPLACE PROCEDURE pr_prod_venda (  
    vOpr CHAR,  
    vcodvenda Number,  
    vcodprod Number,  
    vvalor_prod_venda Number,  
    vquantidade Number,  
    vcodcli Number,  
    vcodempresa Number,  
    vcodusuario Number,  
    vdt_venda Date  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.prod_venda(  
            codvenda,  
            codprod,  
            valor_prod_venda,  
            quantidade,  
            codcli,  
            codempresa,  
            vcodusuario,  
            vdt_venda  
        )  
        VALUES (  
            vcodvenda,  
            vcodprod,  
            vvalor_prod_venda,  
            vquantidade,  
            vcodcli,  
            vcodempresa,  
            vcodusuario,  
            vdt_venda  
        );  
    END IF;  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,037 segundos

Procedure PR_PROD_VENDA compilado

4.11 – Compilação da pr_forn_prod

```
CREATE OR REPLACE PROCEDURE pr_forn_prod (  
    vOpr CHAR,  
    vcodforn Number,  
    vcodprod Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.forn_prod(  
            codforn,  
            codprod  
        )  
        VALUES (  
            vcodforn,  
            vcodprod  
        );  
    ELSE  
        RAISE vExecao;  
    END IF;  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,034 segundos

Procedure PR_FORN_PROD compilado

4.12 – Compilação da pr_entrada_nota

```
CREATE OR REPLACE PROCEDURE pr_entrada_nota (  
    vOpr CHAR,  
    vcodnota Number,  
    vcodforn Number,  
    vcodusuario Number,  
    vdt_emissao Date,  
    vdt_entrada Date,  
    vvalor_nota Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.entrada_nota(  
            codnota,  
            codforn,  
            codusuario,  
            vdt_emissao,  
            vdt_entrada,  
            vvalor_nota  
        )  
        VALUES (  
            vcodnota,  
            vcodforn,  
            vcodusuario,  
            vdt_emissao,  
            vdt_entrada,  
            vvalor_nota  
        );  
    END IF;  
EXCEPTION  
    WHEN vExecao THEN  
        RAISE_APPLICATION_ERROR(-20000, 'Erro ao executar a rotina pr_entrada_nota');  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,05 segundos

Procedure PR_ENTRADA_NOTA compilado

4.13 – Compilação da pr_prod_nota

```
CREATE OR REPLACE PROCEDURE pr_prod_nota (  
    vOpr CHAR,  
    vcodnota Number,  
    vcodprod Number,  
    vvalor_prod_nota Number,  
    vquantidade Number  
)  
IS  
    vExecao EXCEPTION;  
BEGIN  
    IF (vOpr = 'I') THEN  
  
        INSERT INTO adm.prod_nota(  
            codnota,  
            codprod,  
            vvalor_prod_nota,  
            vquantidade  
        )  
        VALUES (  
            vcodnota,  
            vcodprod,  
            vvalor_prod_nota,  
            vquantidade  
        );  
    END IF;  
EXCEPTION  
    WHEN vExecao THEN  
        RAISE_APPLICATION_ERROR(-20000, 'Erro ao executar a rotina pr_prod_nota');  
END;
```

Resultado da Consulta x Saída do Script x
Tarefa concluída em 0,054 segundos

Procedure PR_PROD_NOTA compilado