



Faculdade de Informática e Administração Paulista

PS13SI

ENTREGÁVEL DA DISCIPLINA

2ª ENTREGA

INTEGRANTES

RM (SOMENTE NÚMEROS)	NOME COMPLEMENTO (SEM ABREVIAR)
79922	Andre Luiz Maia Tapioca Brenneisen
83422	Gabriel Domingues de Almeida
82126	Giuliano Zanetti
82358	Guilherme Hideki Takara Chalita
82813	Marcos Lopes da Silva Junior

1. Crie uma visão chamada `TOTAL_CASOS_ACUMULADOS` que exibe os municípios do estado de São Paulo que tem o total de casos acumulados (`casos_acumulados`) maior que 5.000.000. Ordene o resultado pelo total de casos acumulados de maneira decrescente

```
Select * from pf0645.dados_covid;
```

```
create synonym dados
```

```
for dados_covid;
```

```
create synonym rgs_saude
```

```
for regioes_saude;
```

```
create synonym std
```

```
for estados;
```

```
create synonym rgs
```

```
for regioes;
```

```
create synonym mnc
```

```
for municipios;
```

```
create view total_casos_acumulados
```

```
as
```

```
select  std.nome AS nome_estado,mnc.nome AS nome_municipio,  
SUM(dados.CASOS_ACUMULADOS) Casos_Acumulados
```

```
from pf0645.estados std
```

```
join pf0645.dados_covid dados
```

```
on std.COD_ESTADO = dados.COD_ESTADO
```

```
join pf0645.municipios mnc
```

```
on mnc.cod_municipio = dados.cod_municipio
```

```
WHERE dados.COD_ESTADO = 26
```

```
GROUP BY std.nome,mnc.nome;
```

```
create view top10_casos
```

```
AS
```

```
select *
```

```
FROM (
```

```
SELECT *
```

```
from total_casos_acumulados
```

```
ORDER BY Casos_acumulados DESC
```

```
)
```

```
WHERE ROWNUM <= 10;
```

```
select * from top10_casos
```

```
drop view total_dados_acumulados
```

2. Crie uma visão chamada `MEDIA_CASOS_POR_REGIAO` que seleciona os estados agrupados por regiões onde a média de casos (`casos_novos`) está acima da média nacional. Ordene o resultado de forma ascendente pelo nome da região.

```
create view MEDIA_CASOS_POR_REGIAO

AS

select      std.nome      AS      regioao, mnc.nome      AS      estados,
TO_CHAR(AVG(dados.casos_novos), 'FM9990.00') media_Casos_Novos

from pf0645.regioes std

join pf0645.dados_covid dados
on std.COD_regiao = dados.COD_regiao

join pf0645.estados mnc
on mnc.cod_estado = dados.cod_estado

GROUP BY std.nome, mnc.nome

HAVING AVG(dados.casos_novos) > 6.2144

ORDER BY std.nome;

SELECT *

FROM MEDIA_CASOS_POR_REGIAO
```

Para prevenir uma possível lentidão na execução dessa query, que providência(s) você pode tomar para diminuir o custo de execução?

RESPOSTA-INSTRUÇÃO . . .

Se vocês entendem que uma melhoria pode ser implementada escreva o(s) comando(s) para essa implementação logo após o comando de criação da visão. Se vocês julgarem que nada pode ser melhorado não precisam responder esse item.

3. Tomando por base a primeira entrega (o banco de dados de cadastro de pacientes), vocês devem construir um bloco PL/SQL Anônimo que solicite o número do cartão do SUS do paciente e retorne todas as informações relativas a esse paciente.

```
CREATE TABLE ENDERECO
(
id_endereco INTEGER,
rua VARCHAR2(100) NOT NULL,
numero INTEGER NOT NULL,
bairro VARCHAR2(100) NOT NULL,
complemento VARCHAR2(50),
estado VARCHAR2(2) NOT NULL,
cidade VARCHAR2(50) NOT NULL,
cd_ibge NUMBER(7) NOT NULL,
cep NUMBER(9) NOT NULL
);
```

```
ALTER TABLE ENDERECO ADD CONSTRAINT PK_ENDERECO PRIMARY KEY
(id_endereco);
```

```
CREATE TABLE PACIENTE
(
cartao_sus NUMBER(15),
nome_paciente varchar2(100) not null,
raca_cor VARCHAR2(10),
data_nascimento VARCHAR2(10),
nome_mae VARCHAR2(150) NOT NULL ,
nome_pai VARCHAR2(150),
telefone NUMBER (13) NOT NULL,
sexo VARCHAR2(15) NOT NULL,
estado_civil VARCHAR2(15) NOT NULL,
PACIENTE_ENDERECO_FK integer REFERENCES ENDERECO(id_endereco)
);
```

```
ALTER TABLE PACIENTE ADD CONSTRAINT PK_PACIENTE PRIMARY KEY
(cartao_sus);
```

```
CREATE TABLE ESCOLA
(
id_escola INTEGER,
nome_escola VARCHAR2(100) NOT NULL,
serie VARCHAR2(50) NOT NULL,
programa varchar2(50)not null,
ESCOLA_PACIENTE_FK integer REFERENCES PACIENTE(cartao_sus)
```

```
);
```

```
ALTER TABLE ESCOLA ADD CONSTRAINT PK_ESCOLA PRIMARY KEY (id_escola);
```

```
INSERT INTO ENDERECO VALUES(12345  
, 'barcelos', 38, 'Vilas', 'apto31', 'SP', 'sao paulo', 1234567, 03660060);
```

```
INSERT INTO PACIENTE VALUES(1337, 'gabriel', 'negro', '20210608'  
, 'marcia', 'marcio', 989823436, 'masculino', 'casado', 12345);
```

```
INSERT INTO ESCOLA VALUES(1111, 'baroes', 'quinta serie', 'estudo  
mais', 1337);
```

```
----- DROP TABLE
```

```
DROP TABLE escola;  
DROP TABLE paciente;  
DROP TABLE endereco;
```

```
set serveroutput on  
ACCEPT numero_CARTAO_SUS prompt 'Qual o número do cartao SUS?'  
DECLARE  
V_CARTAO_SUS PACIENTE.CARTAO_SUS%TYPE:=&numero_CARTAO_SUS;  
V_NOME_PACIENTE PACIENTE.NOME_PACIENTE%TYPE;  
V_RACA_COR PACIENTE.RACA_COR%TYPE;  
V_DATA_NASCIMENTO PACIENTE.DATA_NASCIMENTO%TYPE;  
V_NOME_MAE PACIENTE.NOME_MAE%TYPE;  
V_NOME_PAI PACIENTE.NOME_PAI%TYPE;  
V_TELEFONE PACIENTE.TELEFONE%TYPE;  
V_SEXO PACIENTE.SEXO%TYPE;  
V_ESTADO_CIVIL PACIENTE.ESTADO_CIVIL%TYPE;  
  
V_ESCOLA ESCOLA.NOME_ESCOLA%TYPE;  
V_SERIE ESCOLA.SERIE%TYPE;  
V_PROGRAMA ESCOLA.SERIE%TYPE;  
V_RUA ENDERECO.RUA%TYPE;  
V_NUMERO ENDERECO.NUMERO%TYPE;  
V_BAIRRO ENDERECO.BAIRRO%TYPE;  
V_COMPLEMENTO ENDERECO.COMPLEMENTO%TYPE;
```



```
V_CIDADE ENDERECO.CIDADE%TYPE;  
V_ESTADO ENDERECO.ESTADO%TYPE;  
V_CEP ENDERECO.CEP%TYPE;
```

```
BEGIN
```

```
SELECT  
NOME_PACIENTE, RACA_COR, DATA_NASCIMENTO, NOME_MAE, NOME_PAI, TELEFONE, SE  
XO, ESTADO_CIVIL, NOME_ESCOLA, SERIE, PROGRAMA, RUA, NUMERO, BAIRRO, COMPLEM  
ENTO, CIDADE, ESTADO, CEP
```

```
INTO V_NOME_PACIENTE, V_RACA_COR, V_DATA_NASCIMENTO, V_NOME_MAE,  
V_NOME_PAI, V_TELEFONE, V_SEXO, V_ESTADO_CIVIL, V_ESCOLA,  
V_SERIE, V_PROGRAMA, V_RUA, V_NUMERO, V_BAIRRO, V_COMPLEMENTO,  
V_CIDADE, V_ESTADO, V_CEP
```

```
FROM PACIENTE, ESCOLA, ENDERECO;
```

```
dbms_output.put_line('A cor é '||V_RACA_COR|| ' A data de nascimento  
é: '|| V_DATA_NASCIMENTO|| ' O nome da mãe: '||V_NOME_MAE||' O nome  
do pai: '|| V_NOME_PAI||' o telefone: '||V_TELEFONE||' o sexo:  
'||V_SEXO||' Estado Civil: '|| V_ESTADO_CIVIL||' a Escola é:  
'||V_ESCOLA||' A série é: '||V_SERIE|| ' Rua: '|| V_RUA||' Número:  
'||V_NUMERO||' Bairro: '||V_BAIRRO||' Complemento: '||  
V_COMPLEMENTO||' Cidade: '||V_CIDADE||' Estado: '||v_ESTADO||' CEP:  
'||V_CEP);
```

```
END;
```