

BANCO DE DADOS

AULA 6

O que são Schemas?

Os Schemas são uma coleção de objetos dentro de um determinado database (banco de dados), servem para agrupar objetos no nível de aplicação como também para simplesmente fazer divisões departamentais. Schemas são bastante utilizados em padrões de sistema de banco de dados. São muito importantes para a performance e segurança.

```
CREATE SCHEMA [nome_esquema]
```

```
CREATE TABLE FUNCIONARIO.CONTABILIDADE (  
  COD INT,  
  NOME VARCHAR (GO)  
)  
GO
```

```
INSERT INTO FUNCIONARIO.CONTABILIDADE VALUES (01 'ALGUEM DA SILVA')
```

A empresa RTL, no intuito de otimizar seu banco de dados, deseja eliminar apenas o SCHEMA de banco de dados PORTIFOLIO, excluindo também todas as suas tabelas, domínios e quaisquer outros elementos. Nesse caso, que comando seria o mais adequado?

- (A) DROP SCHEMA "PORTIFOLIO" DISTINCT;
- (B) DELETE SCHEMA "PORTIFOLIO" RESTRICT;
- (C) DROP SCHEMA "PORTIFOLIO" CASCADE;
- (D) DELETE SCHEMA "PORTIFOLIO" CASCADE;
- (E) DROP ALL SCHEMA "PORTIFOLIO";

DROP SCHEMA -- remove um esquema

O comando **DROP SCHEMA** remove esquemas do banco de dados.

O esquema somente pode ser removido pelo seu dono ou por um superusuário. Deve ser observado que o dono pode remover o esquema (e, portanto, todos os objetos que este contém), mesmo que não seja o dono de alguns objetos contidos no esquema.

IF EXISTS

Não lança um erro se o esquema não existir. Neste caso é emitida uma notificação.

nome

O nome do esquema.

CASCADE

Remove automaticamente os objetos (tabelas, funções, etc.) contidos no esquema.

RESTRICT

Recusa remover o esquema caso este contenha algum objeto. Este é o padrão.

44) Observe a tabela abaixo chamada COMPONENTE.

Id	Nome	Valor	Disponibilidade	Categoria
1	roda	R\$200,00	S	A
2	pneu	R\$300,00	S	A
3	lanterna	R\$150,00	S	B
4	farol	R\$200,00	N	B

* S-Sim N-Não

Com base na tabela acima, considerando a linguagem SQL, assinale a opção que apresenta o comando sintaticamente INCORRETO.

- (A) `SELECT Categoria, AVG(Valor), Nome FROM COMPONENTE GROUP BY Categoria;`
- (B) `SELECT * FROM COMPONENTE WHERE Valor IS NOT NULL;`
- (C) `SELECT Categoria, AVG(Valor) FROM COMPONENTE GROUP BY Categoria;`
- (D) `SELECT Categoria, COUNT(*) FROM COMPONENTE GROUP BY Categoria HAVING COUNT(*) > 3;`
- (E) `SELECT Nome, Valor FROM COMPONENTE WHERE Valor > (SELECT AVG(Valor) FROM COMPONENTE);`

Com relação às variáveis, assinale a opção correta.

- (A) As variáveis persistentes não existem mesmo após a execução dos programas. X
- ~~(B)~~ O valor de uma variável local não é retido após o término da ativação do bloco de execução no qual ela está inserida.
- (C) Para as variáveis locais e intermitentes as células de memória são alocadas e desalocadas via comando específico. X
- ~~(D)~~ As variáveis globais são aquelas vinculadas às células de memória durante a execução do programa e assim permanecem até que a execução do programa se encerre.
- (E) As variáveis intermitentes ou HEAP são criadas e destruídas em tempo de compilação.



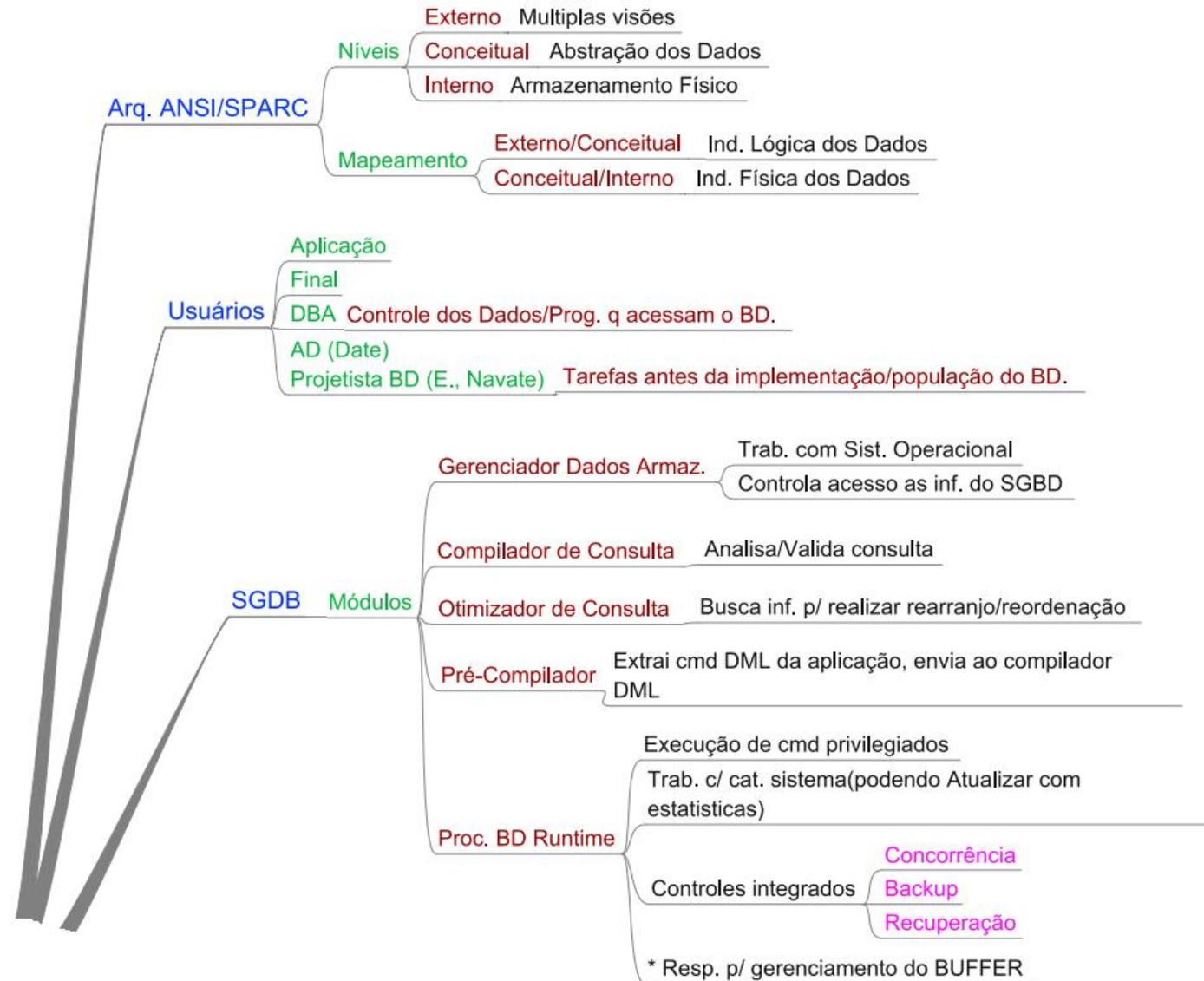
Observe o vetor VET especificado abaixo.

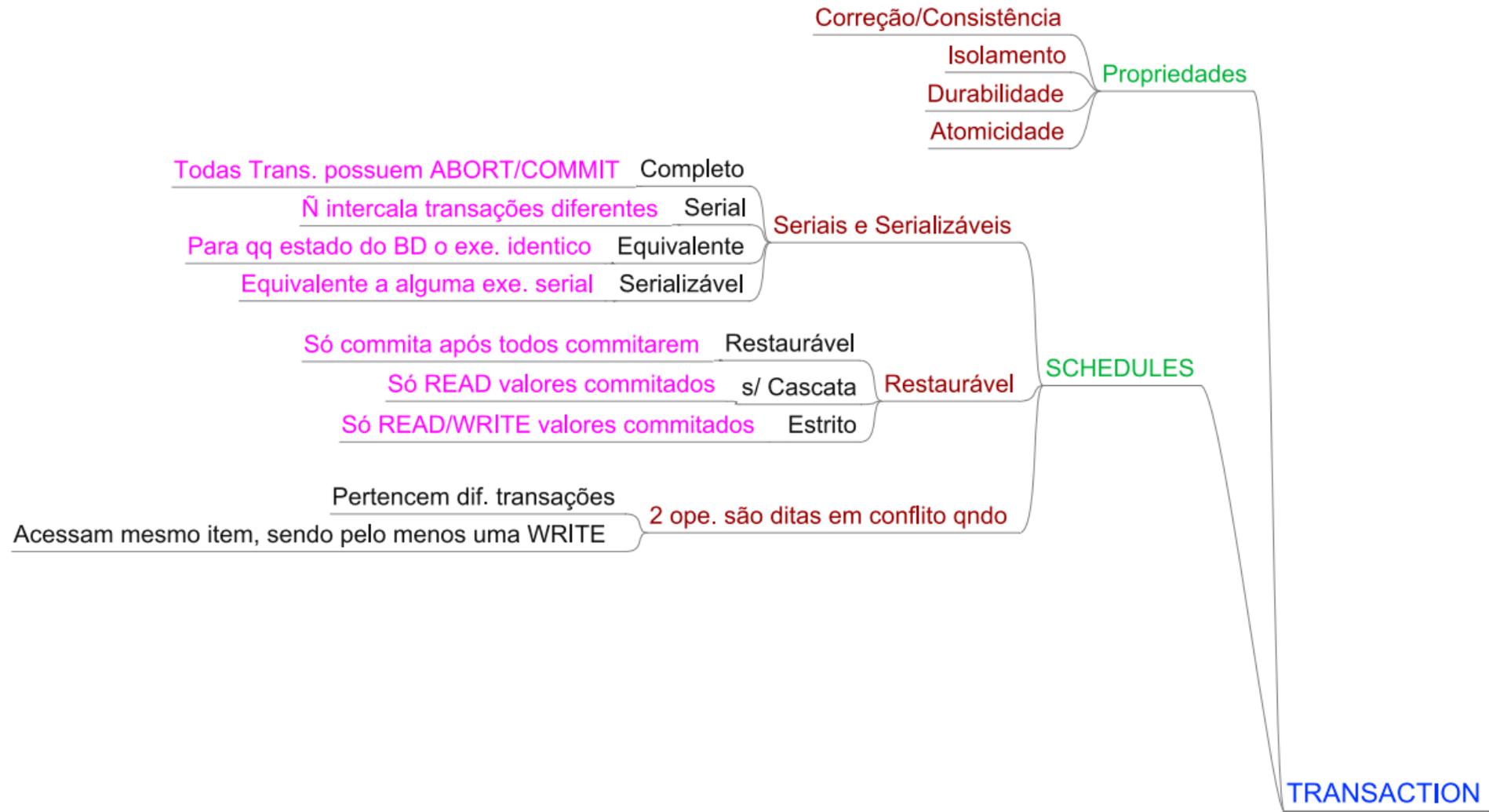
```
tipo v = vetor[1:128]inteiro;  
v: VET;
```

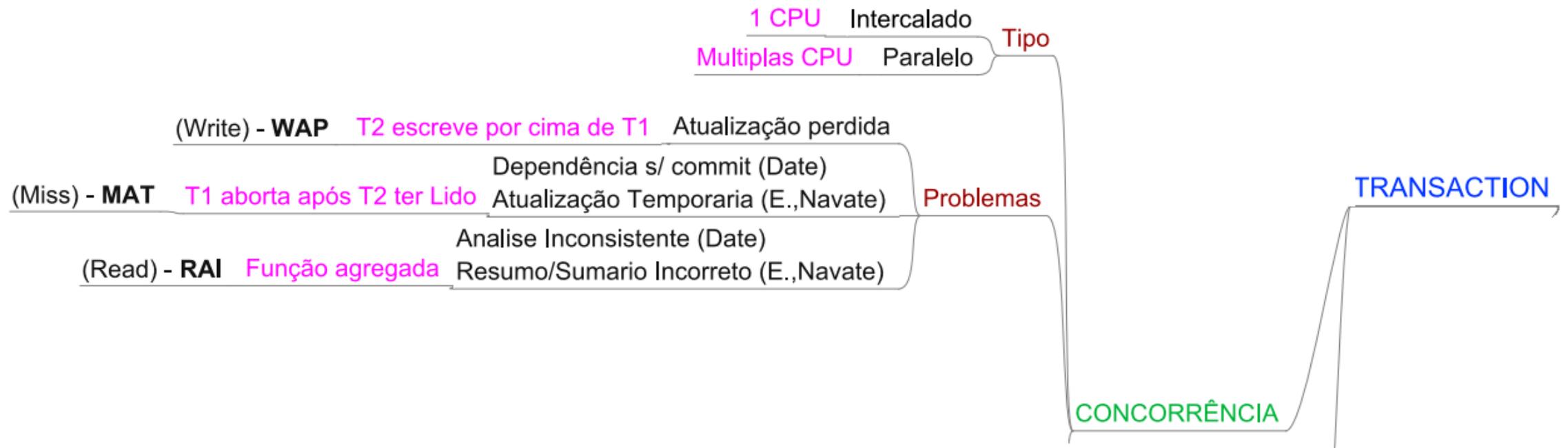
Sendo K a chave a ser localizada dentre os elementos do vetor acima, assinale a opção INCORRETA.

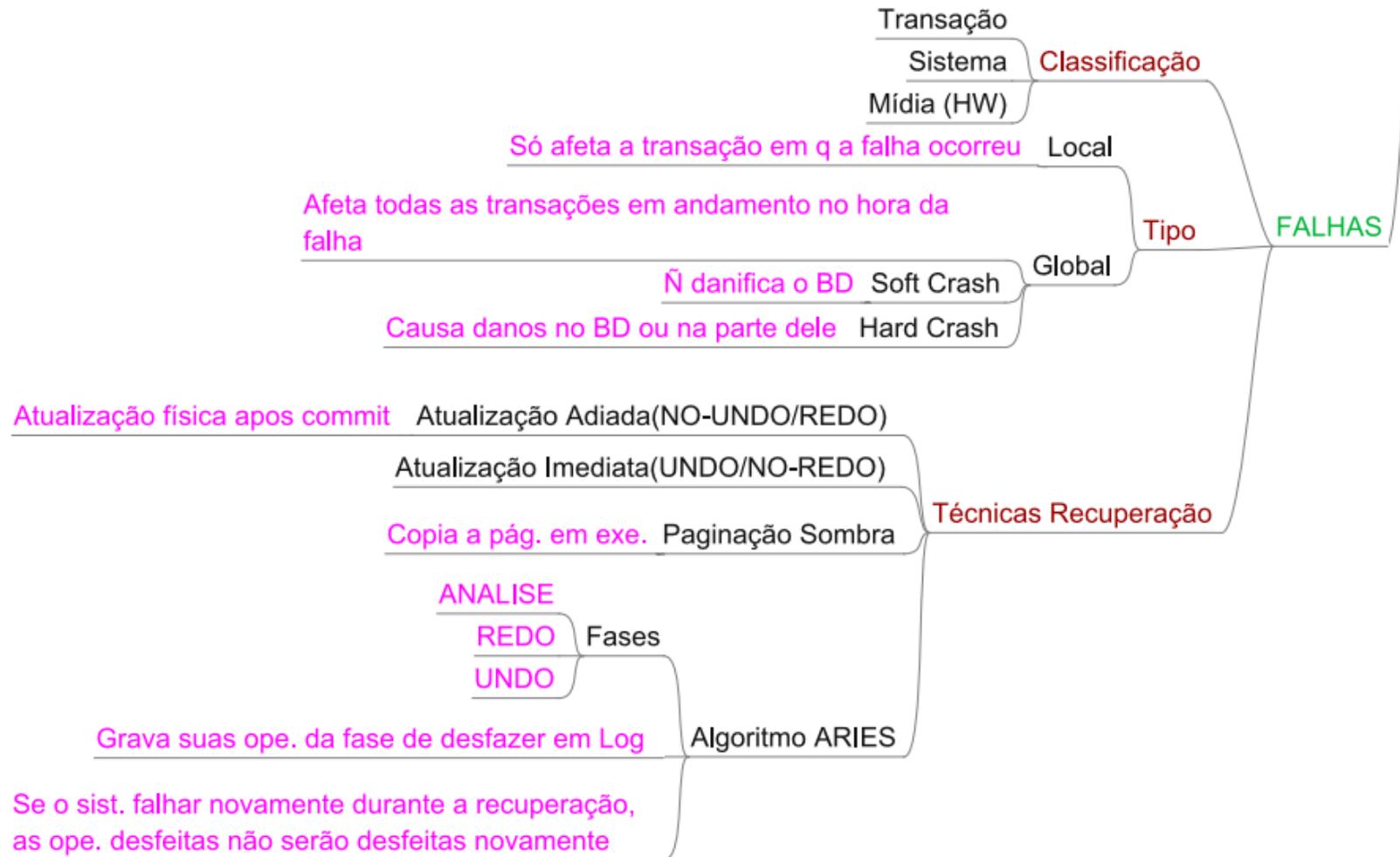
- (A) Na pesquisa sequencial simples, o número médio de comparações que devem ser feitas no vetor VET até encontrar a chave K é 64.
- (B) Na pesquisa binária, o vetor VET deve estar ordenado.
- (C) Na pesquisa binária, divide-se sucessivamente o vetor VET em duas partes, comparando a chave K com o elemento do meio do segmento.
- (D) Na pesquisa sequencial simples, é realizada a comparação da chave com cada elemento de VET, sequencialmente.
- (E) Na pesquisa binária, o número médio de comparações que devem ser feitas no vetor VET até encontrar a chave K é 7.











Analise as afirmações a seguir a respeito de Store Procedure:

I – Uma Store Procedure é um procedimento armazenado em um arquivo executável que pode ser chamado através de comandos SQL.

II – Uma Store Procedure armazena tarefas repetitivas dentro de um banco de dados e aceita parâmetros de entrada para que a tarefa seja efetuada de acordo com a necessidade individual.

III – Uma Store Procedure é disparada automaticamente após a execução de Insert, ou Update, ou Delete.

Assinale a alternativa correta:

- a) A afirmação I está correta.
- b) A afirmação II está correta.
- c) Nenhuma afirmação está correta.
- d) Todas as afirmações estão corretas.

38 -- Considere a tabela chamada "clientes", com a estrutura e os dados a seguir:

codigo_cliente	nome_cliente
1	Davi
2	Carlos
3	Ana
4	Bruna

Assinale o comando SQL utilizado para listar os clientes, ordenando os nomes dos clientes de forma decrescente.

- a) `SELECT codigo_cliente, nome_cliente FROM clientes ORDER BY nome_cliente DESC;`
- b) `SELECT codigo_cliente, nome_cliente FROM clientes ORDER BY codigo_cliente ASC;`
- c) `SELECT codigo_cliente, nome_cliente FROM clientes ORDER BY nome_cliente DESC;`
- d) `SELECT codigo_cliente, nome_cliente FROM clientes ORDER BY codigo_cliente ASC;`

QUESTÃO 12

Considere os seguintes comandos SQL (padrão ANSI) executados em sequência.

```
CREATE TABLE "ALUNO"  
(  
  "ALUNO_ID" integer NOT NULL,  
  "NOME" text,  
  CONSTRAINT "PK_ALUNO" PRIMARY KEY  
  ("ALUNO_ID")  
);
```

```
INSERT INTO "ALUNO"("ALUNO_ID", "NOME") VALUES  
(1, 'VINICIUS');
```

```
INSERT INTO "ALUNO"("ALUNO_ID", "NOME") VALUES  
(2, 'JOÃO');
```

```
COMMIT;
```

```
INSERT INTO "ALUNO"("ALUNO_ID", "NOME") VALUES  
(3, 'OSMAR');
```

```
ROLLBACK;
```

```
INSERT INTO "ALUNO"("ALUNO_ID", "NOME") VALUES  
(4, 'MARCOS');
```

```
COMMIT;
```

Quais os nomes inseridos na tabela ALUNO?

- (A) VINICIUS, JOÃO, MARCOS
- (B) VINICIUS, JOÃO
- (C) VINICIUS, JOÃO, OSMAR, MARCOS
- (D) OSMAR, MARCOS
- (E) MARCOS, JOÃO

Considere a tabela a seguir.

TABELA: FUNCIONARIO					
Pnome	Unome	Cpf	Sexo	Salario	Departamento
JOAO	MEDEIROS	12345678999	M	10000,00	1
MARIA	SANTOS	98765432112	F	8000,00	3
ANA	SILVA	45678912399	F	12000,00	5
PEDRO	TELES	78912385400	M	5000,00	5
MARCELO	SILVA	14785236988	M	22000,00	5

Com base nos dados da tabela acima, assinale a opção que apresenta a *query* que recupera todos os funcionários no departamento 5, cujo salário esteja entre R\$ 10.000 e R\$ 20.000.

- (A) SELECT * FROM FUNCIONARIO WHERE (SALARIO BETWEEN 10000 AND 20000) AND DEPARTAMENTO < 5
- (B) SELECT * FROM FUNCIONARIO WHERE SALARIO < 20000 AND DEPARTAMENTO = 5
- (C) SELECT * FROM FUNCIONARIO WHERE (SALARIO BETWEEN 10000 AND 20000) AND DEPARTAMENTO = 5
- (D) SELECT * FROM FUNCIONARIO WHERE (SALARIO BETWEEN 10000 AND 20000) AND DEPARTAMENTO > 5
- (E) SELECT * FROM FUNCIONARIO WHERE (SALARIO ≤ 20000) AND DEPARTAMENTO = 5

Considere as Tabelas ALUNO, DISCIPLINA e NOTAS a seguir para responder às questões 1, 2 e 3.

ALUNO	
ALUNO_ID	NOME
1	VINICIUS
2	JOAO
3	OSMAR

DISCIPLINA	
DISCIPLINA_ID	NOME
1	CVGA
2	AD
3	PROG1
4	BD1

NOTAS		
ALUNO_ID	DISCIPLINA_ID	NOTA
1	1	10,0
2	1	5,0
3	2	4,0
1	2	5,0
2	4	8,0
3	1	6,0

QUESTÃO 1

Considerando as Tabelas ALUNO, DISCIPLINA e NOTAS, qual comando SQL (padrão ANSI) calcula, corretamente, somente a média de NOTA do aluno 'VINICIUS' ?

- (A) `SELECT MED(N.NOTA) FROM NOTAS N JOIN ALUNO A ON N.ALUNO_ID=A.ALUNO_ID`
- (B) `SELECT AVG(N.NOTA) FROM NOTAS N JOIN ALUNO A ON N.ALUNO_ID=A.ALUNO_ID WHERE A.NOME='VINICIUS'`
- (C) `SELECT MED(N.NOTA) FROM NOTAS N JOIN ALUNO A ON N.ALUNO_ID=A.ALUNO_ID WHERE A.NOME='VINICIUS'`
- (D) `SELECT AVG(N.NOTA) FROM NOTAS N JOIN ALUNO A ON N.ALUNO_ID=A.ALUNO_ID`
- (E) `SELECT AVG(N.NOTA) FROM NOTAS N WHERE N.NOME='VINICIUS'`

Considere as Tabelas ALUNO, DISCIPLINA e NOTAS a seguir para responder às questões 1, 2 e 3.

ALUNO	
ALUNO_ID	NOME
1	VINICIUS
2	JOAO
3	OSMAR

DISCIPLINA	
DISCIPLINA_ID	NOME
1	CVGA
2	AD
3	PROG1
4	BD1

NOTAS		
ALUNO_ID	DISCIPLINA_ID	NOTA
1	1	10,0
2	1	5,0
3	2	4,0
1	2	5,0
2	4	8,0
3	1	6,0

QUESTÃO 2

Com relação às Tabelas ALUNO, DISCIPLINA e NOTAS, qual comando SQL (padrão ANSI) lista as disciplinas pelo NOME com a respectiva média de NOTA na disciplina?

- (A) `SELECT AVG(N.NOTA) AS MEDIA FROM NOTAS N JOIN DISCIPLINA D ON N.DISCIPLINA_ID = D.DISCIPLINA_ID GROUP BY D.NOME`
- (B) `SELECT D.NOME, AVG(N.NOTA) AS MEDIA FROM NOTAS N JOIN DISCIPLINA D ON N.DISCIPLINA_ID = D.DISCIPLINA_ID`
- (C) `SELECT D.NOME, MED(N.NOTA) AS MEDIA FROM NOTAS N JOIN DISCIPLINA D ON N.DISCIPLINA_ID = D.DISCIPLINA_ID GROUP BY D.NOME`
- (D) `SELECT D.NOME, SUM(N.NOTA) AS MEDIA FROM NOTAS N JOIN DISCIPLINA D ON N.DISCIPLINA_ID = D.DISCIPLINA_ID AVG BY D.NOME`
- (E) `SELECT D.NOME, AVG(N.NOTA) AS MEDIA FROM NOTAS N JOIN DISCIPLINA D ON N.DISCIPLINA_ID = D.DISCIPLINA_ID GROUP BY D.NOME`

Considere as Tabelas ALUNO, DISCIPLINA e NOTAS a seguir para responder às questões 1, 2 e 3.

ALUNO	
ALUNO_ID	NOME
1	VINICIUS
2	JOAO
3	OSMAR

DISCIPLINA	
DISCIPLINA_ID	NOME
1	CVGA
2	AD
3	PROG1
4	BD1

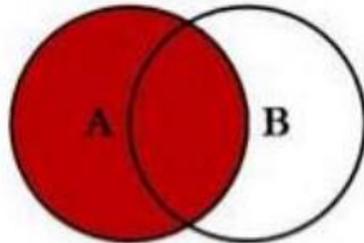
NOTAS		
ALUNO_ID	DISCIPLINA_ID	NOTA
1	1	10,0
2	1	5,0
3	2	4,0
1	2	5,0
2	4	8,0
3	1	6,0

QUESTÃO 3

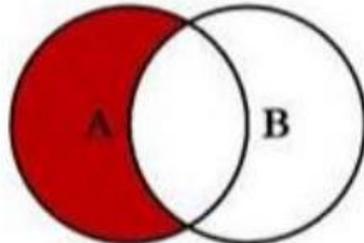
Referente às Tabelas ALUNO, DISCIPLINA e NOTAS, qual comando SQL (padrão ANSI) lista somente os NOMES dos alunos que não fizeram a disciplina "BD1"?

- (A) `SELECT A.NOME FROM ALUNO A
EXCEPT
SELECT DISTINCT A.NOME FROM NOTAS N,
ALUNO A, DISCIPLINA D WHERE
N.ALUNO_ID=A.ALUNO_ID
AND D.NOME='BD1'`
- (B) `SELECT A.NOME
FROM NOTAS N, ALUNO A, DISCIPLINA D
WHERE N.ALUNO_ID=A.ALUNO_ID
AND N.DISCIPLINA_ID=D.DISCIPLINA_ID AND
D.NOME='BD1'`
- (C) `SELECT A.NOME FROM ALUNO A
INTERSECT
SELECT DISTINCT A.NOME FROM NOTAS N,
ALUNO A, DISCIPLINA D WHERE
N.ALUNO_ID=A.ALUNO_ID
AND N.DISCIPLINA_ID=D.DISCIPLINA_ID AND
D.NOME='BD1'`
- (D) `SELECT A.NOME FROM ALUNO A
UNION
SELECT DISTINCT A.NOME FROM NOTAS N,
ALUNO A, DISCIPLINA D WHERE
N.ALUNO_ID=A.ALUNO_ID
AND N.DISCIPLINA_ID=D.DISCIPLINA_ID AND
D.NOME='BD1'`
- (E) `SELECT A.NOME FROM ALUNO A
EXCEPT
SELECT DISTINCT A.NOME FROM NOTAS N,
ALUNO A, DISCIPLINA D WHERE
N.ALUNO_ID=A.ALUNO_ID
AND N.DISCIPLINA_ID=D.DISCIPLINA_ID AND
D.NOME='BD1'`

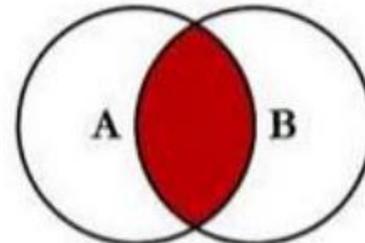
SQL JOINS



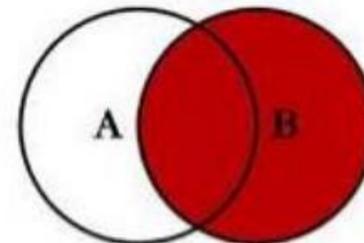
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key
```



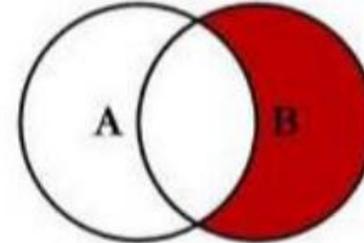
```
SELECT <select_list>  
FROM TableA A  
LEFT JOIN TableB B  
ON A.Key = B.Key  
WHERE B.Key IS NULL.
```



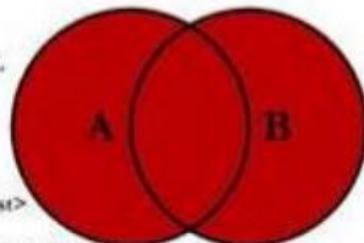
```
SELECT <select_list>  
FROM TableA A  
INNER JOIN TableB B  
ON A.Key = B.Key
```



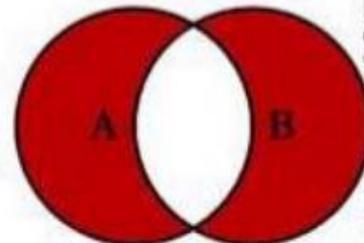
```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
RIGHT JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL.
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key
```



```
SELECT <select_list>  
FROM TableA A  
FULL OUTER JOIN TableB B  
ON A.Key = B.Key  
WHERE A.Key IS NULL  
OR B.Key IS NULL.
```

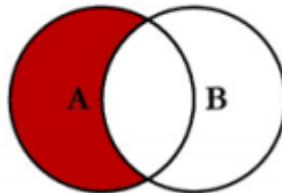
© C.L. Moffat, 2008



ALGUMAS OPERAÇÕES EM COM REGISTROS

EXCEPT (EXCETO) retorna linhas distintas da consulta de entrada à esquerda que não são produzidas pela consulta de entrada à direita.

EXCEPT



```
SQLQuery2.sql - DALILAH.master (Dalilah\Paulo (52))  
--SELECT ID, NOME FROM @A  
EXCEPT  
SELECT ID, NOME FROM @B
```

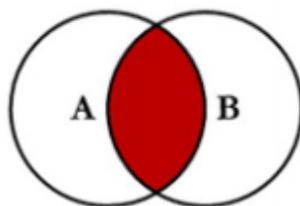
	ID	NOME
1	1	José
2	3	João

DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 2 rows

ALGUMAS OPERAÇÕES EM COM REGISTROS

INTERSECT retorna linhas distintas que são produzidas pelo operador das consultas de entrada à esquerda e à direita.

INTERSECT



```
SQLQuery2.sql - DALILAH.master (Dalilah\Paulo (52))*
```

```
SELECT ID, NOME FROM @A  
INTERSECT  
SELECT ID, NOME FROM @B
```

100 %

Results Messages

ID	NOME
1	Maria

DALILAH (11.0 CTP) Dalilah\Paulo (52) master 00:00:00 1 rows

Função ISNULL

No sistema MySQL, a função **ISNULL()** é utilizada para testar se uma expressão é NULL. Se a expressão for NULL, esta função apresenta o valor 1. Caso contrário, esta função apresenta o valor 0.

Por exemplo,

A **cláusula EXISTS** faz uma verificação se existe algum resultado para a subquery informada.

```
1  SELECT
2      [ coluna1, coluna2, ... | * ]
3  FROM
4      [ tabela1, tabela2, ... ]
5  WHERE
6      EXISTS (
7      SELECT
8          [ coluna1, coluna2, ... | * ]
9      FROM
10         [ tabela1, tabela2, ... ]
11     WHERE [ condicao ]
12     )
```

```
1  SELECT
2      P.ID,
3      P.nome
4  FROM
5      produto P
6  WHERE
7      EXISTS (
8      SELECT
9      V.ID_PRODUTO
10     FROM
11     venda_produto V
12     WHERE
13     V.ID_PRODUTO = P.ID
14     )
```

```
1  SELECT
2      P.ID,
3      P.nome
4  FROM
5      produto P
6  WHERE
7      NOT EXISTS (
8      SELECT
9      V.ID_PRODUTO
10     FROM
11     venda_produto V
12     WHERE
13     V.ID_PRODUTO = P.ID
14     )
```

+55 (21) 99461-8818

@explicadoresnet

www.explicadores.net.br

