

## Banco de Dados II: Trabalho 3 – M2

Acadêmicos: Ellen Junker e Gabriel Leonardo de Almeida

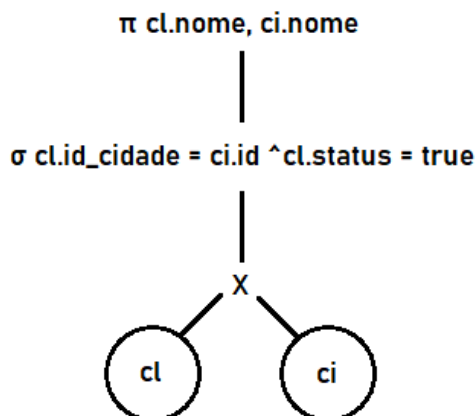
### Questão 1:

As tabelas HEAP apresentam como vantagem um melhor desempenho em operações de INSERT, UPDATE e DELETE, pois o SGBD não precisará procurar pelo índice do dado para realizar a operação (o dado é inserido em um espaço livre que o banco de dados encontra). Como desvantagem se destaca o pior desempenho em operações de SELECT, pois o SGBD passa por cada registro na tabela até encontrar o solicitado.

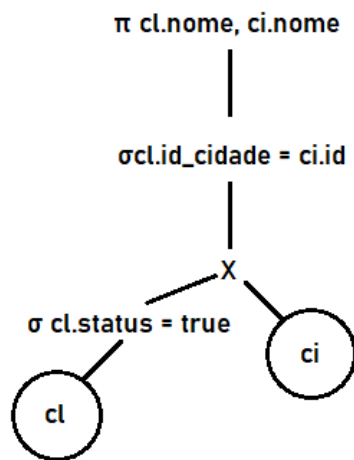
As tabelas com índices, por sua vez, possuem como vantagens um melhor desempenho em operações de SELECT, pois o SGBD procura pelo dado no índice do dado, e a facilidade de garantir que determinadas informações chaves não estejam duplicadas (como a busca é rápida e simples, descobrir se uma chave já existe é uma operação muito barata), etc. Dentre suas desvantagens pode-se citar o pior desempenho em operações de INSERT, UPDATE e DELETE, pois o SGBD interpreta o índice como uma tabela adicional, sendo que se o dado estiver em mais de um índice, o SGBD tem que acessar cada uma dessas tabelas. Além disso, possuem um maior consumo de memória e disco pelo banco de dados (a tabela adicional do índice ocupa mais um espaço no BD), etc.

### Questão 2:

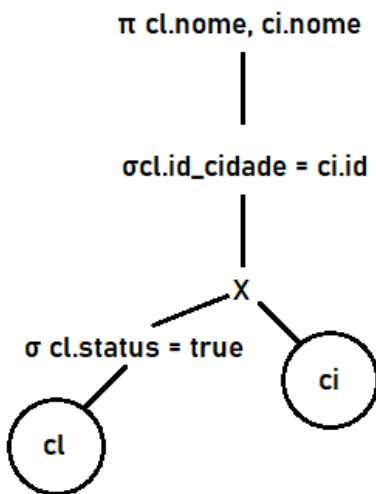
a) Fase 0



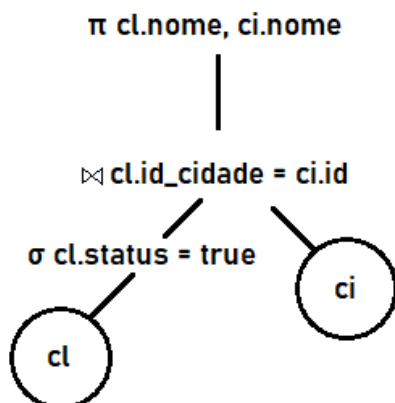
- b) Fase 1, utilizando a heurística de “executar as operações de seleção tão cedo quanto possível”



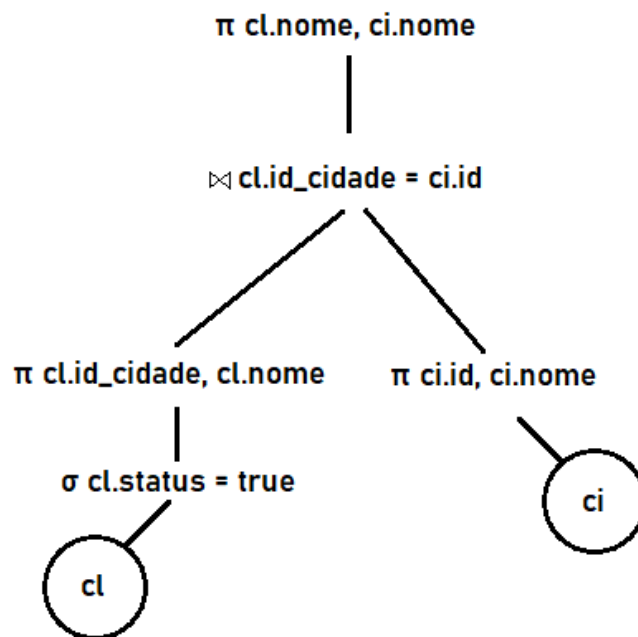
- c) Fase 1, utilizando a heurística de “diminuir os tamanhos das relações a serem utilizadas no produto cartesiano”



- d) Fase 1, utilizando a heurística de “substituir operações de produto cartesiano seguidas pelos respectivos critérios de seleção por operações de junção”

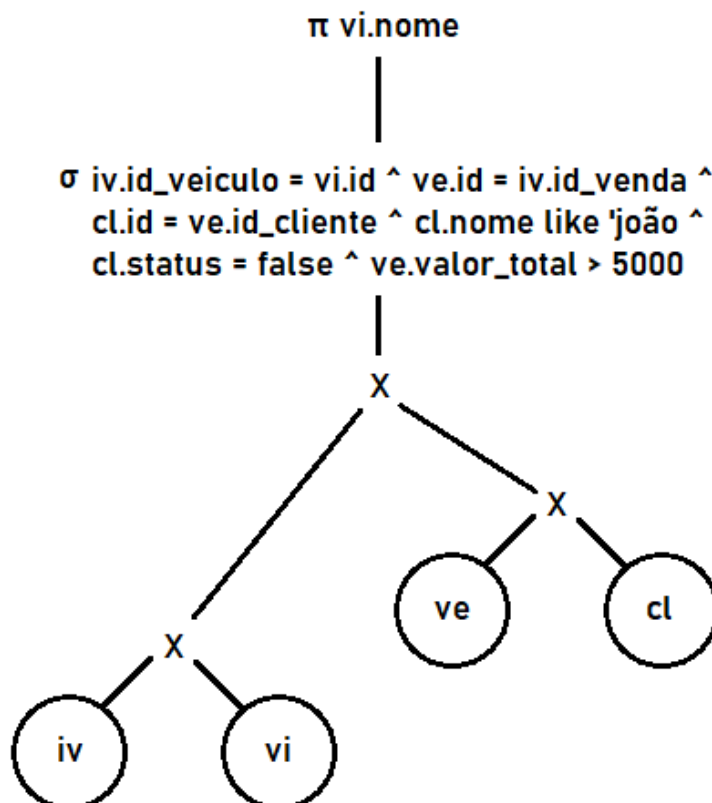


- e) Fase 1, utilizando a heurística de “executar as operações de projeção tão cedo quanto possível”

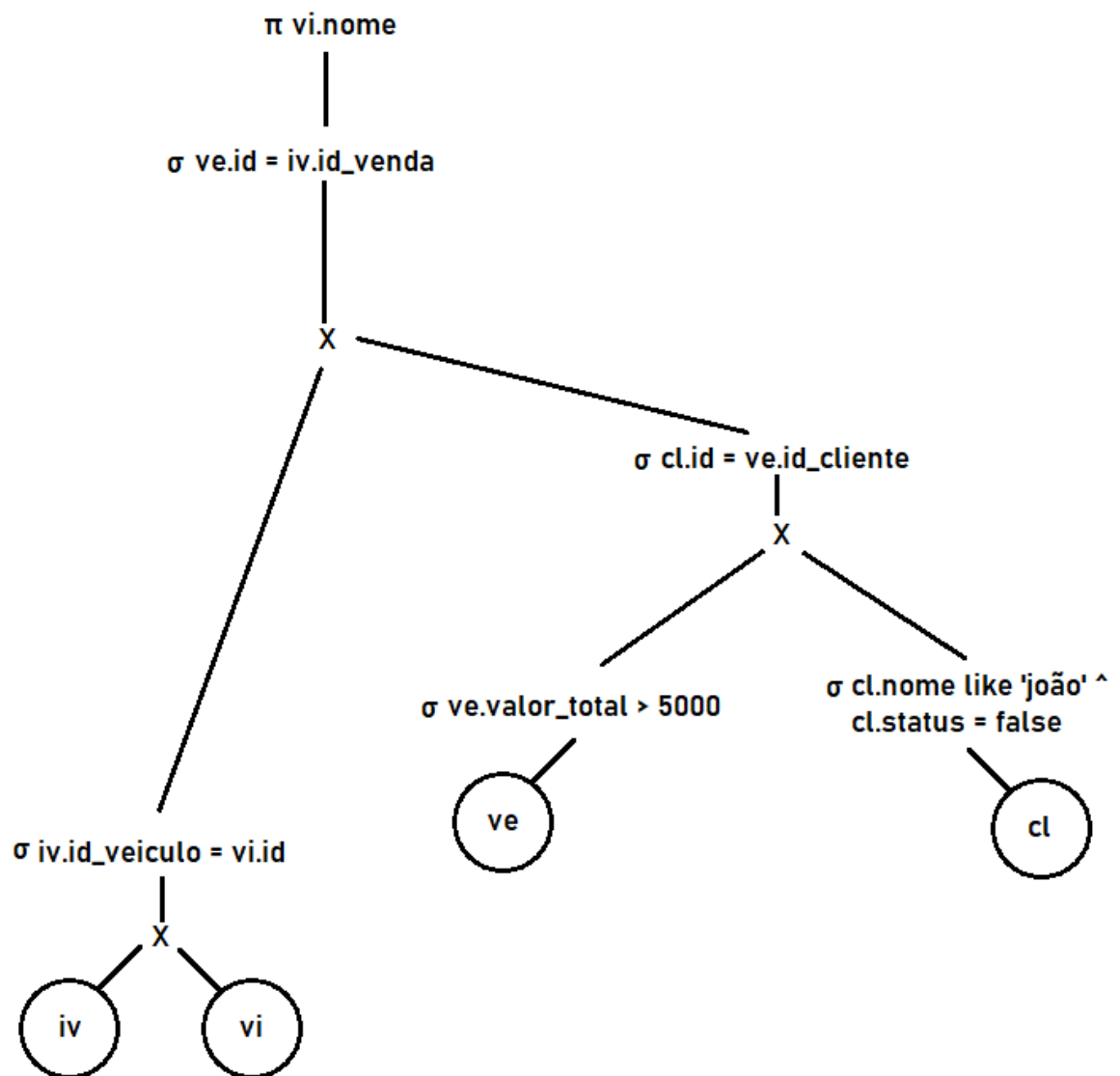


### Questão 3:

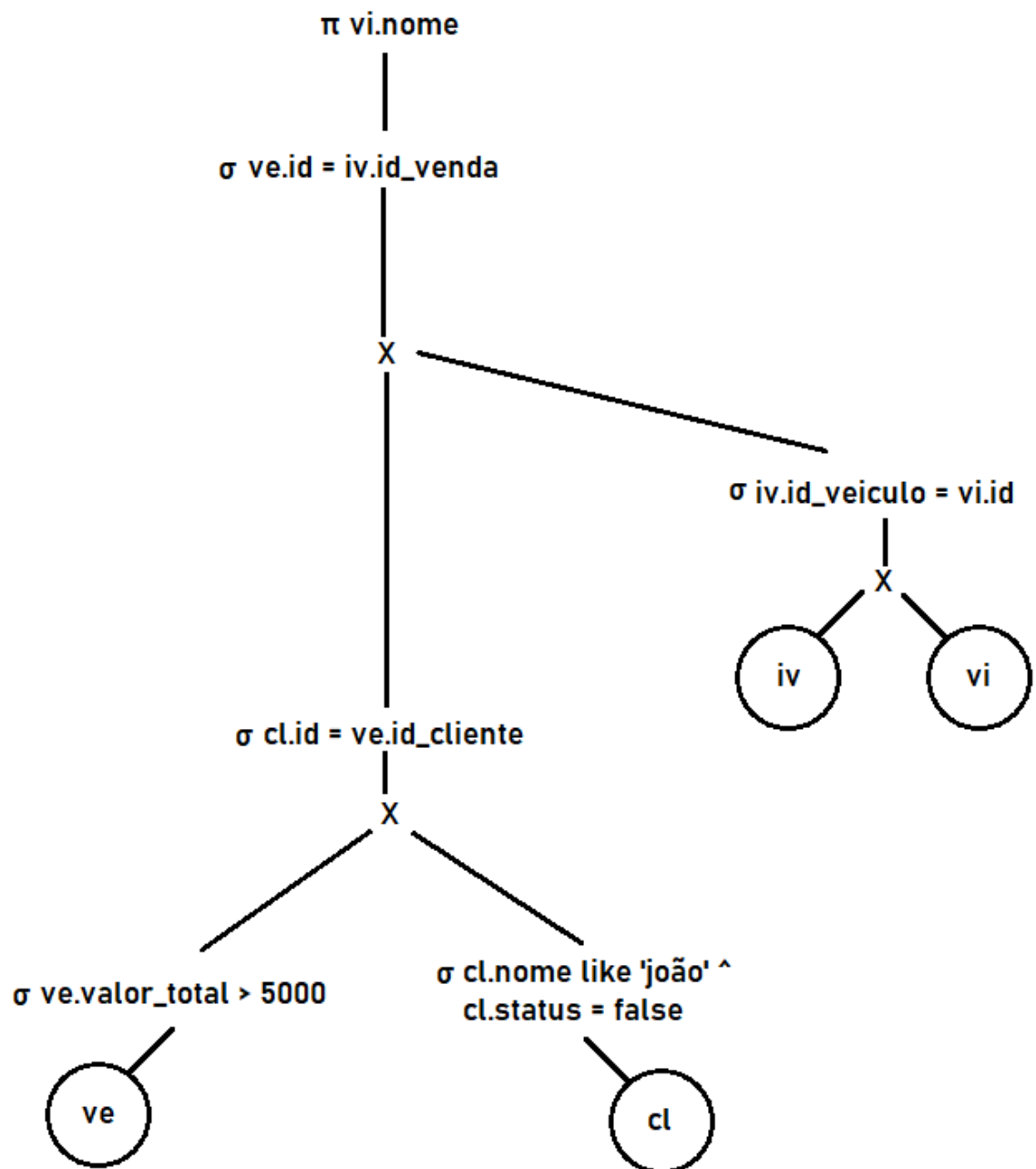
- a) Fase 0



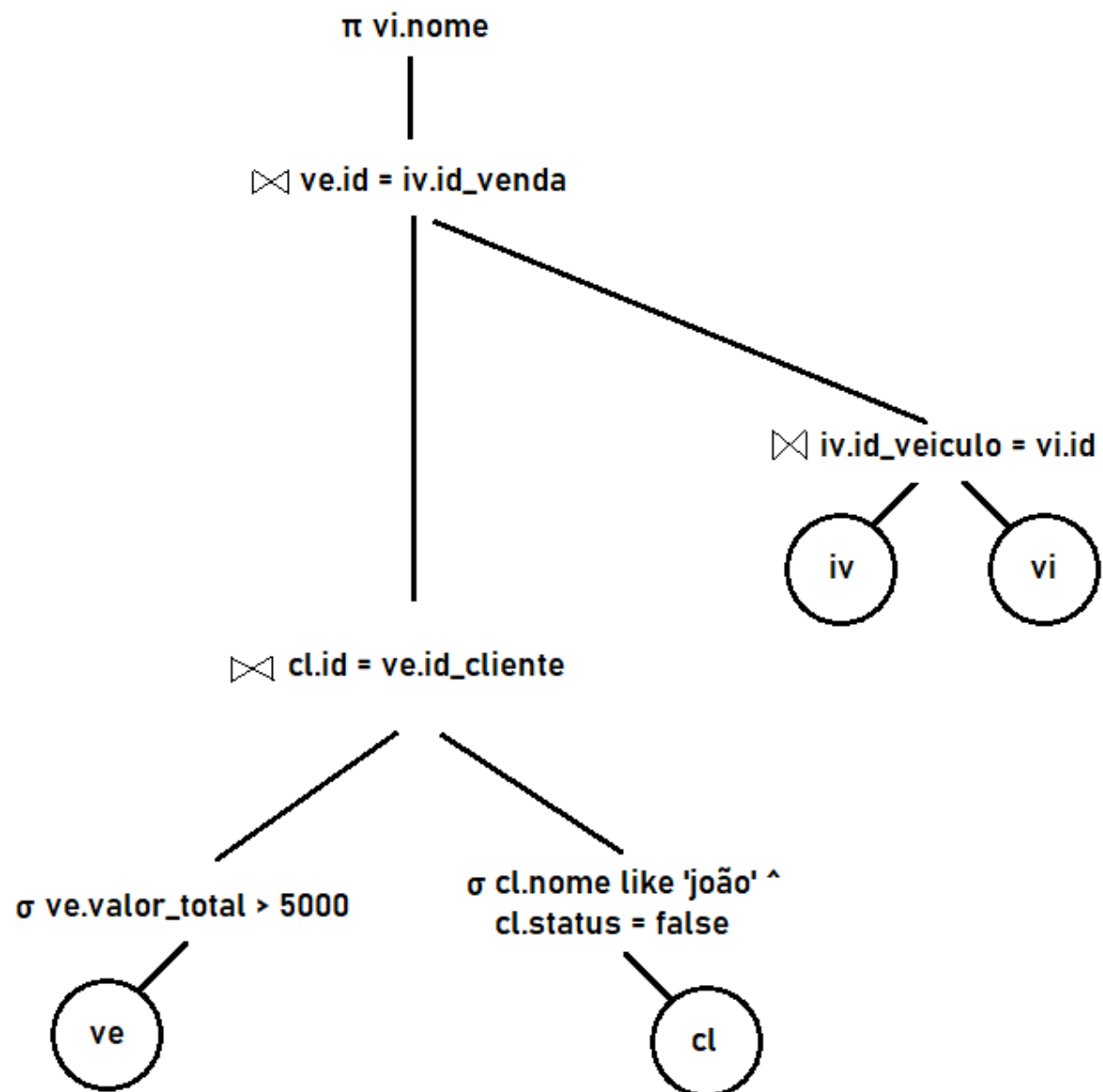
- b) Fase 1, utilizando a heurística de “executar as operações de seleção tão cedo quanto possível”



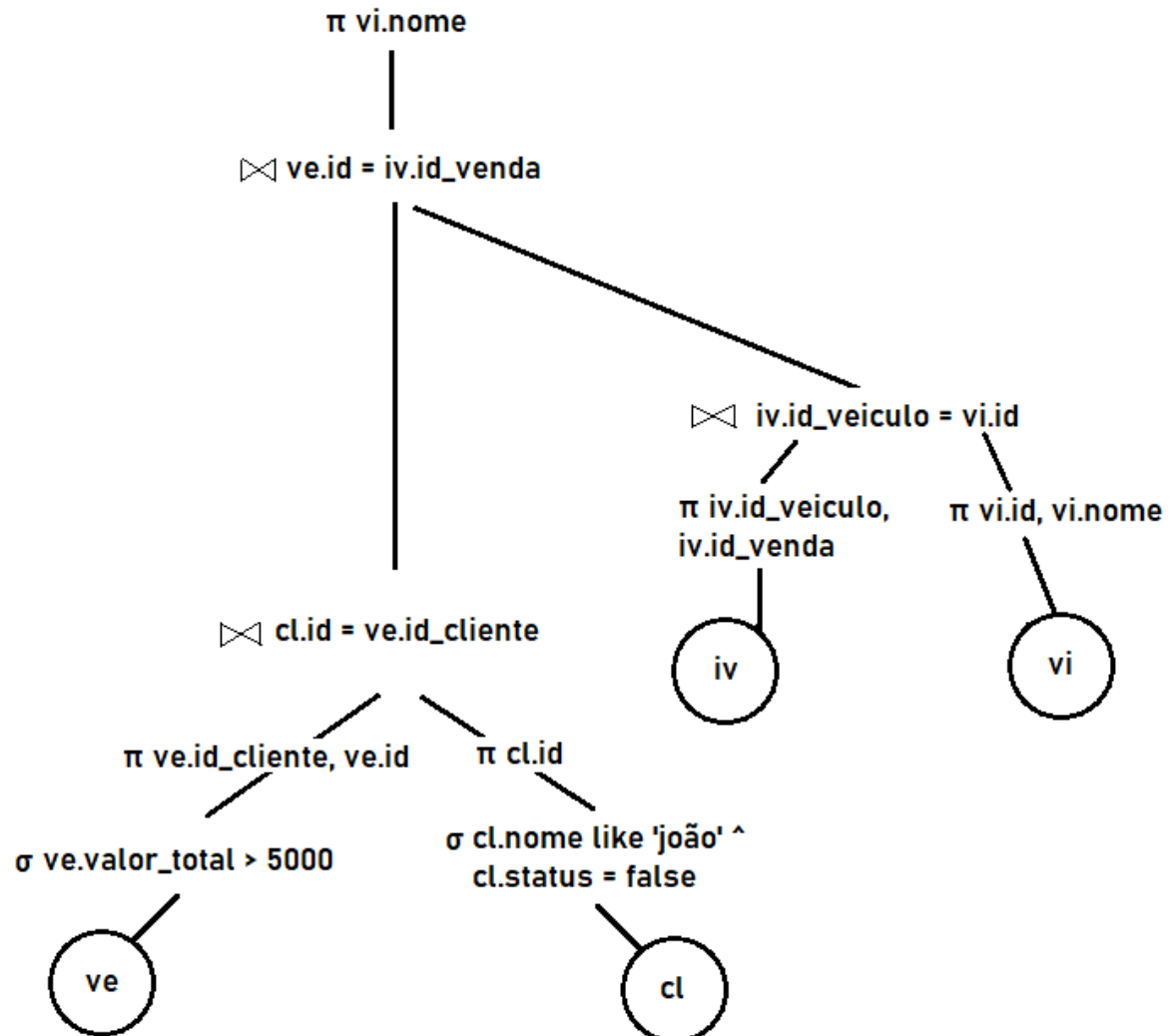
- c) Fase 1, utilizando a heurística de “diminuir os tamanhos das relações a serem utilizadas no produto cartesiano”



- d) Fase 1, utilizando a heurística de “substituir operações de produto cartesiano seguidas pelos respectivos critérios de seleção por operações de junção”



- e) Fase 1, utilizando a heurística de “executar as operações de projeção tão cedo quanto possível”



**Questão 4:**

- Apenas um item está certo.
- Apenas os itens I e II estão certos.
- Apenas os itens I e III estão certos.
- Apenas os itens II e III estão certos.
- Todos os itens estão certos.

Resposta: c, apenas os itens I e III estão certos

**Questão 5:**

- a. Atomicidade
- b. Isolamento
- c. Distributividade
- d. Consistência
- e. Durabilidade

Resposta: b, isolamento

**Questão 6:****i) Registro de timestamp**

Basicamente, para cada transação, é associado um timestamp fixo e exclusivo (hora do relógio do sistema (clock) ou um contador lógico). O registro de timestamp precisa garantir que a ordem em que o item está sendo acessado não viole a ordem do timestamp. Para isso, ele associa dois valores de timestamp para cada item do BD (read\_TS() e write\_TS()). Esses timestamps são atualizados sempre que uma nova instrução read\_TS(X) ou write\_TS(X) é executada.

O registro de timestamp previne deadlock também, através de dois esquemas: Wait-die (esperar-morrer) e Wound-wait (ferir-esperar). O Wait-die aborta a primeira operação (supondo que existem apenas 2 operações (A e B)) e a reinicia posteriormente com o mesmo timestamp. E o Wound-wait faz a primeira operação (também supondo que existem apenas 2 operações nesse exemplo) esperar.

**ii) Multiversão**

Basicamente, cada processo conectado ao BD enxerga uma cópia temporária do BD em tempo de execução. Qualquer alteração que esteja sendo feita em determinado momento por um processo, não será vista pelos demais processos operando no banco de dados, até que as alterações tenham sido comitadas.

Quando um BD multiversão precisar atualizar um determinado dado, ele não sobrescreverá esse dado com o novo dado, mas marcará o dado antigo como obsoleto e adicionará a versão mais recente em outro lugar. Isso permite que os processos acessem os dados que estavam lá quando eles começaram a ler, mesmo que este dado tenha sido modificado ou excluído por outro processo.



Esse método também permite que o BD evite a sobrecarga, acumulando vazios na memória ou na estrutura do disco. Mas geralmente requer uma varredura periódica do sistema para limpar e apagar os dados obsoletos.

### iii) Validação

Basicamente, as atualizações feitas por um processo não são comitadas no BD até que a transação termine. É uma técnica otimista. As atualizações são feitas sobre cópia local dos dados. É feita uma validação para verificar a serialização e se ela não for violada, os dados são comitados a partir dessa cópia dos dados.

### Questão 7:

<b>T1 BINÁRIO</b>	<b>T2 BINÁRIO</b>
lock_item(X);	
read_item(X);	
unlock_item(X);	
lock_item(Y);	lock_item(X);
write_item(Y);	read_item(X);
unlock_item(Y);	Unlock_item(X);
	lock_item(Y);
	read_item(Y);
	unlock_item(Y);
	lock_item(X);
	write_item(X);
	unlock_item(X);

<b>T1 COMPARTILHADO/EXCLUSIVO</b>	<b>T2 COMPARTILHADO/EXCLUSIVO</b>
read_lock_item(X);	read_lock_item(X);
read_item(X);	read_item(X);
unlock_item(X);	unlock_item(X);
write_lock_item(Y);	
write_item(Y);	
unlock_item(Y)	

	<code>read_lock_item(Y);</code>
	<code>read_item(Y);</code>
	<code>unlock_item(Y);</code>
	<code>write_lock_item(X);</code>
	<code>write_item(X);</code>
	<code>unlock_item(X);</code>

Como pode ser observado pela implementação das transações 1 e 2 pelos métodos de bloqueio binário e compartilhado/exclusivo, o método binário demonstra possuir uma maior restrição no bloqueio de seus itens. Isso ocorre porque acontece um bloqueio total na leitura e na escrita do item por outras transações quando ele está sendo acessado.

Por outro lado, o bloqueio compartilhado/exclusivo deixa o bloqueio de leitura compartilhado, ou seja, várias transações podem ler ao mesmo tempo. O bloqueio total só é realizado quando a escrita no item se faz necessária, de modo a realizar um bloqueio de escrita.