O PostgreeSQL utiliza o modelo MVCC (Multiversion Concurrency Control).

Esse modelo é utilizado para melhorar a performance das transações do banco de dados no ambiente multi-usuário. Cada transação nesse modelo trabalha com uma imagem dos dados no estado que eles estavam antes da transação. Assim, quando uma transação é iniciada não é levado em consideração as alterações das outras transações simultâneas, realizando um isolamento para cada sessão do banco de dados.

Exemplo: Uma sessão pode executar um update na tabela enquanto outra sessão tenta lê-los, dessa forma, a leitura irá retornar os dados como estavam antes do update.

Para cada registro de cada tabela sempre existem dois campos utilizados no controle de transações. O campo xmin informa o id (auto-sequenciado) da transação que gerou este registro. Já xmax determina o id da transação que a deletou ou alterou seu conteúdo, gerando um novo registro com xmin atualizado.

MVCC e a necessidade de Vacuum

O MVCC traz uma série de benefícios sobre os bloqueios:
Processos de leitura não bloqueiam processos de escrita e vice-versa
Praticamente elimina a incidência de contenções de transações
Reduz drasticamente o risco de DEAD LOCK
Possui um desempenho bem superior
Mas também é possível detectar um ponto fraco no MVCC. Os registros alterados e excluídos não ficam mais acessíveis mas não são imediatamente riscados do mapa. Ou seja, continuam inchando o disco. Para evitar que o banco tome proporções além do aceitável, é necessário executar regularmente a instrução VACUUM, que além de apagar os registros, também atualiza o ID das transações para evitar estouro e consequente reinício automático dos IDs de transações (4 bilhões de transações).

Modos de Bloqueio

O PostgreSQL trabalha com modos de bloqueios vinculados a ações específicas ao programa para controlar o acesso simultâneo aos dados e tabelas. Esses modos podem ser automáticos com o controle de transações no MVCC ou de forma explícita em que MVCC não possui um bloqueio adequado.

Como o MVCC impede que a leitura bloqueie a escrita e vice-versa, as operações que alteram a estrutura da tabela precisarão bloquear essas operações que rolarem concomitantemente, garantindo assim a integridade dos dados durante a alteração da tabela.

Bloqueios no nível de tabela

ACCESS SHARE

O comando SELECT obtém um bloqueio deste modo nas tabelas referenciadas. Em geral, qualquer comando que apenas leia a tabela (sem modificá-la) obtém este modo de bloqueio.
Entra em conflito com: ACCESS EXCLUSIVE

ROW SHARE

O comando SELECT FOR UPDATE obtém o bloqueio neste modo na(s) tabela(s) de destino. Entra em conflito com: EXCLUSIVE e ACCESS EXCLUSIVE

ROW EXCLUSIVE

Os comandos UPDATE, DELETE e INSERT obtêm este modo de bloqueio na tabela de destino (além do modo de bloqueio ACCESS SHARE nas outras tabelas referenciadas). Em geral, este modo de bloqueio é obtido por todos os comandos que modificam os dados da tabela. Entra em conflito com: SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE, ACCESS EXCLUSIVE

SHARE UPDATE EXCLUSIVE

Obtida pelo comando VACUUM (sem a opção FULL). Protege a tabela contra mudanças simultâneas no esquema durante a execução do comando VACUUM. Entra em conflito com: SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.

SHARE

Obtido pelo comando CREATE INDEX. Entra em conflito com: ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.

SHARE ROW EXCLUSIVE

Este modo de bloqueio não é obtido automaticamente por nenhum comando do PostgreSQL. Entra em conflito com: ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE

EXCLUSIVE

Este modo de bloqueio não é obtido automaticamente por nenhum comando do PostgreSQL. Entra em conflito com: ROW SHARE, ROW EXCLUSIVE, SHARE UPDATE EXCLUSIVE, SHARE, SHARE ROW EXCLUSIVE, EXCLUSIVE e ACCESS EXCLUSIVE.

ACCESS EXCLUSIVE

Obtido pelos comandos ALTER TABLE, DROP TABLE e VACUUM FULL. Este é também o modo de bloqueio padrão para o comando LOCK TABLE sem a especificação explícita do modo. Entra em conflito com todos os modos de bloqueio Este modo garante que a transação que o obteve seja a única que esteja acessando a tabela.

Esses bloqueios servem para travar a tabela de realizar alterações em sua estrutura. O bloquio por conflito é quando uma operação/bloqueio não pode ser executada/gerado porquê outro bloqueio já foi obtido, isso ocorre pois duas transações não podem obter modos de bloqueios conflitantes.

Alguns modos de bloqueio também são conflitantes, como no ACCESS EXCLUSIVE na qual o DROP Table não é permitido ser executado na mesma tabela por duas transações

simultaneamente.

Já bloqueios não-conflitantes podem ser obtidos várias vezes e de forma simultânea por muitas transações.

Bloqueios no nível de linha

O bloqueio no nível de linha é obtido quando a tupla é atualizada ou excluída. Os bloqueios no nível de linha não afetam a consulta aos dados, já que bloqueiam apenas escritas na mesma linha. Como visto anteriormente, o modelo de MVCC impede que bloqueios de leitura afetem operações de escrita e vice-versa, porém é possível forçar um bloqueio de escrita por meio de um comando de leitura utilizando o FOR UPDATE, podendo contornar o MVCC.

Impasses ou DeadLocks

A utilização de bloqueios explícitos pode causar impasses (deadlocks), especialmente quando duas (ou mais) transações mantiverem bloqueios desejados pelas demais. O PostgreSQL detecta automaticamente as situações de impasse e, para solucioná-las, aborta uma das transações envolvidas e permite que a(s) outra(s) prossiga(m).

Um deadlock ocorre quando dois ou mais processos possuem bloqueios em objetos separados, e cada um dos processos estão tentando obter um bloqueio no objeto que o outro processo bloqueou. Assim, o SQL Server, resolve abortando um dos processos permitindo que o outro processo continue a sua transação no banco de dados. O processo abortado envia uma mensagem para o arquivo de error log do SQL Server informando sobre ela. Deadlocks podem causar uma pressão sobre os recursos do servidor SQL Server, principalmente CPU.

Dicas para evitar deadlocks:

--Verifique se o banco de dados está desenhado corretamente.
--Não permita que usuários interfiram durante as transações.
--Procure ter transações no SQL o mais curto possível.
--Reduza a quantidade de vezes que sua aplicação conversa com o SQL Server usando procedures para manter as transações dentro de um único lote.
--Reduza a quantidade de leituras, se você precisa fazer leituras constantes da mesma informação, coloque essa informação em uma variável ou matriz para você consultar por lá.
--Diminua o máximo possível o tempo de bloqueio. Desenvolva aplicativos que fazem bloqueios em último caso e libere-os o mais rápido possível.

Access Share

Bloqueio requerido apenas por instruções SELECT e ANALYZE são simples leitura em geral permitidas, a menos que alguma outra transação tenha requerido Access Exclusive.

Row Share
A instrução SELECT com a cláusula FOR UPDATE obtém este nível de bloqueio. Ele conflita apenas com os níveis mais retritos Exclusive e Access Exclusive.

Row Exclusive
Bloqueio solicitado automaticamente por instruções INSERT, UPDATE, DELETE. Conflita com quem precisar além de Access Exclusive e Exclusive, também com Share Row Exclusive e Share.

Share Update Exclusive
Bloqueio padrão do comando VACUUM sem a opção FULL. Conflita com todos os próximos níveis de bloqueio e com ele próprio. Bloqueia a tabela inteira.

Share
A partir deste nível de bloqueio, já vimos com quem cada um conflita. Bloqueia toda a tabela e é requerido por instruções CREATE INDEX.

Share Row Exclusive
Similar ao Exclusive, mas apenas no nível de linha. Não há instrução SQL que automaticamente solicite este nível de bloqueio.

Exclusive
Este só não conflita com Access Share. É requerido automaticamente apenas em tabelas de catálogo em determinadas operações.

Access Exclusive
Conflita com todo mundo. Ou seja, nenhuma outra transação pode operar na mesma tabela bloqueada por alguma transação que solicitou Access Exclusive. Instruções ALTER TABLE, DROP TABLE, TRUNCATE, REINDEX, CLUSTER e VACUUM FULL necessitam deste modo de bloqueio.

Seção 1 bloqueia a Tabela A

Seção 2 bloqueia a Tabela B

Seção 1 tenta bloquear a Tabela B

Lock da Seção 1 pois a Tabela B está bloqueadapela Seção 2

Seção 2 tenta bloquear a Tabela A

Lock da Seção 2 pois a Tabela A está bloqueadapela Seção 1

DeadLock

Access Share: Solicitado por instruções SELECT e ANALYZE apenas permitindo simples leitura. Conflita com Access Exclusive;
Row Share: Solicitado por instrução SELECT FOR UPDATE. Conflita com: Exclusive e Access Exclusive;
Row Exclusive: Solicitado por instruções INSERT, UPDATE, DELETE.
Share Update Exclusive: Bloqueio do comando VACUUM sem a opção FULL. Bloqueia a tabela inteira.
Share: Bloqueia toda a tabela e é requerido por instruções CREATE INDEX.
Share Row Exclusive: Similar ao Exclusive, mas apenas no nível de linha. Não há instrução que automaticamente solicite este bloqueio.
Exclusive: É requerido automaticamente apenas em tabelas de catálogo em determinadas operações.
Access Exclusive: Conflita com todos os outros. Solicitado por instruções ALTER TABLE, DROP TABLE, TRUNCATE, REINDEX, CLUSTER e VACUUM FULL.