**SQL: Funções de agregação**

* **MAX** - analisa um conjunto de valores e **retorna o maior entre eles**.
* **MIN** - analisa um grupo de valores e **retorna o menor entre eles.**
* **SUM** - realiza a **soma dos valores em uma única coluna** e retorna esse resultado.
* **AVG** - podemos calcular a **média aritmética dos valores em uma única coluna.**
* **STDDEV** - retorna o **desvio padrão.**
* **COUNT** - retorna o **total de linhas selecionadas.**
* **GROUP BY** - dividimos os **registros que serão agregados em grupos de valores.**
* **HAVING** - A cláusula HAVING só pode existir se houver anteriormente a cláusula GROUP BY. Serve para **filtrar os resultados que serão submetidos a agregação.** Ela**filtra o GROUP BY.**
* **ALIAS** - Podemos utilizar a palavra-chave "as" para criar um apelido para uma coluna. Uma vez que as funções de agregação são retornadas como tal.

**Segue um resumo sobre as cláusulas INNER JOIN, LEFT JOIN e RIGHT JOIN:**

* A cláusula **JOIN** permite que os dados de várias tabelas sejam combinados com base na relação existente entre elas. Por meio dessa cláusula, os dados de uma tabela são usados para selecionar os dados pertencentes à outra tabela.
* A cláusula **INNER JOIN** permite usar um operador de comparação para comparar os valores de colunas provenientes de tabelas associadas. Por meio desta cláusula, os registros de duas tabelas são usados para que sejam gerados os dados relacionados de ambas. Usamos as cláusulas WHERE e FROM para especificar esse tipo de associação.
* A cláusula **LEFT JOIN ou LEFT OUTER** **JOIN** permite obter não apenas os dados relacionados de duas tabelas, mas também os dados não relacionados encontrados na tabela à esquerda da cláusula JOIN. Caso não existam dados relacionados entre as tabelas à esquerda e a direita do JOIN, os valores resultantes de todas as colunas da lista de seleção da tabela à direita serão nulos.
* Ao contrário do LEFT JOIN, a cláusula **RIGHT JOIN ou RIGHT OUTER JOIN** retorna todos os dados encontrados na tabela à direita de JOIN. Caso não existam dados associados entre as tabelas à esquerda e à direita de JOIN, serão retornados valores nulos.

Todas as linhas de dados da tabela à esquerda de JOIN e da tabela à direita serão retornadas pela cláusula **FULL JOIN ou FULL OUTER JOIN.** Caso uma linha de dados não esteja associada a qualquer linha da outra tabela, os valores das colunas a lista de seleção serão nulos. Caso contrário, os valores obtidos serão baseados nas tabelas usadas como referência.

Considerando uma variável de relação R, diz-se que K é uma **chave candidata** de R se ela apresentar as propriedades de **unicidade e irredutibilidade**. A primeira propriedade garante que não há duas tuplas de R com o mesmo valor de K. Já a segunda, que nenhum subconjunto de K apresenta a propriedade de unicidade.

***Tipos de índices – TransactSQL***

* **UNIQUE**: cria um índice exclusivo em uma tabela ou exibição. Um índice exclusivo é aquele no qual duas linhas não podem ter o mesmo valor de chave de índice.
* **CLUSTERED:** cria um índice no qual a ordem lógica dos valores de chave determina a ordem física das linhas correspondentes em uma tabela. No oracle o equivalente é Index-Organized.
* **NÃO CLUSTERIZADO:** cria um índice que especifica a ordem lógica de uma tabela. Com um índice não clusterizado, a ordem física das linhas de dados é independente de sua ordem indexada.

Um dos tipos de trigger disponíveis no Microsoft SQL Server 2012 é denominado **INSTEAD OF**. Nesse tipo de trigger, a ação especificada por ele é executada   
no lugar do evento que deu origem ao disparo do trigger.

O gatilho INSTEAD OF é uma exibição baseada em várias tabelas base, devendo ser usado para oferecer suporte a inserções, atualizações e exclusões que referenciem dados em mais de uma tabela.

Um instantâneo de banco de dados (**snapshot**) é uma exibição estática somente leitura de um banco de dados do SQL Server (o banco de dados de origem). Um snapshot de um banco de dados sempre reside na mesma instância de servidor que o banco de dados de origem. Quando o banco de dados de origem é atualizado, o snapshot do banco de dados é atualizado. Portanto, quanto mais tempo o snapshot de banco de dados existir, maior será a probabilidade de ele usar todo o espaço em disco disponível.

***Arquivos de configuração PostgreSQL***

* postgresql.conf => é o principal arquivo de configuração do servidor.
* pg\_hba.conf => é o arquivo de configuração para o host-based authentication (hba).
* pg\_ident.conf => é o arquivo de configuração para o user name mapping.

Os arquivos de dados são formatados em blocos oracle. O tamanho de um bloco oracle pode variar, de 2KB a 16KB(linux/windows), a 32KB(outros sistemas). O tamanho do bloco é controlado pelo parâmetro DB\_BLOCK\_SIZE, e nunca pode ser alterado apos a criação do BD, porque ele é usado para formatar os arquivos de dados que compõe o tablespace system. Se, mais tarde, ficar evidente q o tamanho do bloco é inadequado, é recomendável criar um novo BD e transferir tudo pra ele.

PS: O tamanho dos blocos oracle é fixo para um Tablespace (padrão é 8K).

O DB\_BLOCK\_SIZE é definido na criação do banco de dados e não pode ser alterado depois.

A faixa de valores que pode ser definido para o BD\_BLOCK\_SIZE é: 2048 a 32768 bytes.

O valor default é: 8192 bytes

***Views Materializadas***• Objetos que são utilizados para sumarizar, computar, replicar e distribuir dados;  
• Muito utilizadas em ambiente de data warehousing, suporte a decisão, computação distribuída e mobile;  
• Podem ser atualizadas constantemente;

**Alert Log:** é um log de mensagens que inclui erros de corrupção de blocos pelo Oracle.

Tipos de mensagens de log neste arquivo: Inicialização do banco de dados, Desligamento, log switches, espaço de erros etc.

Oracle automaticamente cria um novo alert log sempre que um antigo é deletado.

**TransactSQL –** Criar usuário e forçar a mudança de senha

CREATE LOGIN <nome> WITH PASSWORD <senha> MUST\_CHANGE

Verificação da sintaxe de consulta em SQL:

"Essas consultas são analisadas e validadas pela exatidão da sintaxe da consulta, os nomes de arquivos e elementos de dados, e assim por diante, por um compilador de consulta, que as compila para um formato interno."

O controle de concorrência em sistemas gerenciadores de bancos de dados pode levar a situações em que uma transação, por exemplo, T1, esteja aguardando a liberação de algum item de dados que esteja sendo bloqueado, por exemplo, pela transação T2. Concomitantemente, a transação T2 está aguardando pela liberação de outro item de dados bloqueado por T1. Tal tipo de situação recebe a denominação de impasse ou deadlock.

***Tunnig***

O enunciado menciona "grande quantidade de registros". Se fosse feito um commit para cada inserção de registro (autocommit), o desempenho geral do processo seria reduzido.

Ao se desabilitar essa função e "executar apenas um commit ao final das inserções de registros", isso seria "uma forma de melhorar o desempenho (tuning)".

**Relacionamento recursivo:** O mesmo tipo entidade  participa mais de uma vez em um tipo relacionamento em papéis diferentes - (Auto Relacionamento)

Um **dicionário de dados** (do inglês data dictionary) é uma coleção de metadados que contêm definições e representações de elementos de dados. Dentro do contexto de SGBD, um dicionário de dados é um grupo de tabelas, habilitadas apenas para leitura ou consulta.

***MySQL***

The EXTRACT() function is used to return a single part of a date/time, such as year, month, day, hour, minute, etc.

***PostgreSQL***

**File System Replication**: Todas as alterações em um sistema de arquivos são espelhadas para um sistema de arquivos que residem em outro computador. A única restrição é que o espelhamento deve ser feito de uma forma que assegura que o servidor espera tem uma cópia consistente do sistema de arquivos.

Existem várias formas de parar o servidor de banco de dados. O tipo de parada pode ser controlado através do envio de sinais diferentes para o processo postmaster.  
**-SIGTERM**: após receber o sinal SIGTERM o servidor não aceita novas conexões, mas deixa as sessões existentes trabalharem normalmente. A parada é realizada apenas depois de todas as sessões terminarem normalmente. Esta é a Parada Esperta (Smart Shutdown).  
**-SIGINT:** o servidor não aceita novas conexões e envia para todos os processos servidores existentes o sinal SIGTERM, fazendo com que estes interrompam suas transações correntes e terminem imediatamente. Depois aguarda os processos servidores saírem e, finalmente, para. Esta é a Parada Rápida (Fast Shutdown).  
**-SIGQUIT: e**sta é a Parada Imediata (Immediate Shutdown), que faz o processo postmaster enviar um sinal SIGQUIT para todos os processos descendentes e sair imediatamente, sem parar de forma apropriada. Da mesma maneira, os processos descendentes saem imediatamente após receber o sinal SIGQUIT. Provoca uma recuperação (refaz o log do WAL) na próxima inicialização. Somente é recomendado em caso de emergência.

O  **pg\_hba.conf:** é o responsável pelo controle da autenticação de usuário.

Triggers são usados para diferentes propósitos, como os seguintes:

1. Impor regras de negócios complexas que não podem ser definidas usando as restrições de integridade.  
2. Manter regras de segurança complexas.  
3. Gerar automaticamente valores para colunas derivadas.  
4. Coletar informações estatísticas sobre acessos a uma tabela.  
5. Previr transações inválidas.  
6. Fornecer valores para auditoria.

Uma **visão** é uma tabela virtual é não uma tabela física. Uma visão (ou view) é uma única tabela que é derivada de outras tabelas. Por não necessariamente existir em forma física é considerada uma tabela virtual. É importante destacar que certos sistemas de bancos de dados permitem que as visões sejam armazenadas. Essas visões armazenadas são chamadas de visões materializadas

**Resumo View:**

* É derivada de uma ou mais relações e armazena os dados em uma **tabela virtual**do banco de dados.
* Torna as consultas mais ágeis.
* Uma view é realmente derivada de uma ou mais relações.
* Permite implementar segurança em um banco de dados omitindo dados irrelevantes para algum grupo de usuário.
* Permitido criar uma *view* com base na definição de outra *view.*

**Dicionário de dados:**  dados sobre os dados - **dicionário de dados** (do inglês *data dictionary*) é uma coleção de metadados que contêm definições e representações de elementos de dados. Dentro do contexto de SGBD, um dicionário de dados é um grupo de tabelas, habilitadas apenas para leitura ou consulta, ou seja, é uma base de dados, propriamente dita, que entre outras coisas, mantém as seguintes informações:

* Definição precisa sobre elementos de dados
* Perfis de usuários, papéis e privilégios.
* Descrição de objetos
* Restrições de integridade
* Stored procedures (pequeno trecho de programa de computador, armazenado em um SGBD, que pode ser chamado frequentemente por um programa principal) e gatilhos
* Estrutura geral da base de dados
* Informação de verificação
* Alocações de espaço
* Índices

Em uma relação, os nomes das colunas são únicos, as linhas são distintas entre si, e a ordem da disposição das linhas e colunas é **irrelevante** para o banco de dados.

* **Em uma relação, os nomes das colunas são únicos (Certa).** Os nomes das colunas são usados para ajudar na interpretação do significado dos valores em cada linha. Não faz muito sentido termos várias colunas com o mesmo nome, pois seriam interpretados do mesmo jeito, e teríamos informações redundantes. Por isso, os nomes das colunas são únicos.
* **Em uma relação, as linhas são distintas entre si**. **(Certa)**. Cada linha representa um registro diferente, contendo todos os mesmos atributos dentro de uma tabela. Cada linha da tabela descreve uma única ocorrência de entidade no interior do conjunto de entidades.
* **Em uma relação, a ordem da disposição***das linhas e colunas é irrelevante***para o banco de dados**. **(Errada)**.

Guarde isso: **a ordem das linhas não é relevante(é irrelevante)**. **A ordem das colunas é relevante**, a não ser que a correspondência entre elas e seus valores seja mantida.