

Banco de Dados - Oracle

Prof. Washington Almeida, MSC, ISF 27002



INTRODUÇÃO

- Toda organização tem informações que devem ser armazenadas e gerenciadas para atender aos seus requisitos.
- Uma organização deve **coletar e manter registros**.
- As informações devem estar disponíveis sempre que necessário.
- Um **banco de dados é uma coleção organizada de informações**.
- O **objetivo** de um banco de dados é **coletar, armazenar e recuperar informações** relacionadas para uso por **aplicativos de banco de dados**.

INTRODUÇÃO

- Um **Sistema de Gerenciamento de Banco de Dados (SGBD)** é um software que controla o armazenamento, a organização e a recuperação de dados.
- Um **aplicativo de banco de dados** é um programa que interage com um banco de dados para **acessar e manipular dados**.
- Um **banco de dados hierárquico** organiza dados em uma estrutura em árvore.
- Um **banco de dados de rede** é semelhante a um banco de dados hierárquico, exceto que os registros têm um relacionamento muitos-para-muitos em vez de um-para-muitos.

INTRODUÇÃO

- Um **banco de dados relacional** é um banco de dados em conformidade com o modelo relacional.
- O **modelo relacional** possui os seguintes **aspectos principais**:
 - **Estruturas**: objetos bem definidos armazenam ou acessam os dados de um banco de dados.
 - **Operações**: ações claramente definidas permitem que os aplicativos manipulem os dados e as estruturas de um banco de dados.
 - **Regras de integridade**: governam as operações nos dados e nas estruturas de um banco de dados.

INTRODUÇÃO

- Um **banco de dados relacional** armazena dados em um **conjunto de relações simples**.
- Uma **relação** é um **conjunto de tuplas**.
- Uma **tupla** é um **conjunto não ordenado de valores de atributos**.
- Uma **tabela** é uma representação bidimensional de uma relação na forma de **linhas (tuplas) e colunas (atributos)**.
- Cada linha de uma **tabela** possui o mesmo **conjunto de colunas**.
- Um **banco de dados relacional** é um banco de dados que armazena dados em **relações (tabelas)**.

INTRODUÇÃO

- O modelo relacional é a base para um **Sistema de Gerenciamento de Banco de Dados Relacional (SGBDR)**.
- Um **SGBDR** move dados para um banco de dados, armazena os dados e os recupera para que os aplicativos possam manipulá-los.
- Um **SGBDR** distingue entre os seguintes tipos de operações:
 - **Operações Lógicas:** um aplicativo especifica qual conteúdo é necessário.
 - **Operações Físicas:** o SGBDR determina como as coisas devem ser feitas e realiza a operação, armazena e recupera dados para que as operações físicas sejam transparentes para aplicativos de banco de dados.

INTRODUÇÃO

- O **Oracle Database** é um **SGBDR**.
- Um **SGBDR** que implementa recursos orientados a objetos, como tipos definidos pelo usuário, herança e polimorfismo, é chamado de **Sistema de Gerenciamento de Banco de Dados Objeto-Relacional (SGBDOR)**.
- O **Oracle Database** estendeu o modelo relacional a um **modelo objeto-relacional**, possibilitando o armazenamento de modelos de negócios complexos em um banco de dados relacional.

HISTÓRICO

- A versão atual do **Oracle Database** é o resultado de mais de 35 anos de desenvolvimento inovador.
- Em **1977**, ocorreu a fundação da **Oracle**. Larry Ellison, Bob Miner e Ed Oates iniciaram a consultoria Software Development Laboratories.
- Em **1979**, a RSI introduziu o **Oracle V2** como o primeiro SGBDR baseado em SQL disponível no mercado.
- O **Oracle V3**, lançado em **1983**, foi o primeiro banco de dados relacional a ser executado em mainframes, minicomputadores e PCs.
- O **Oracle V4** introduziu consistência de leitura multiversão.

HISTÓRIA

- O **Oracle V5**, lançada em **1985**, suportava computação em uma arquitetura cliente/servidor e sistemas de banco de dados distribuídos.
- O **Oracle V6** trouxe aprimoramentos para E/S de disco, bloqueio de linhas, escalabilidade e backup e recuperação.
- O **Oracle V6** introduziu a primeira versão da linguagem PL/SQL, uma extensão de procedimento proprietária para SQL.
- O **Oracle7**, lançado em **1992**, introduziu **gatilhos** e **procedimentos armazenados PL/SQL**.

HISTÓRIA

- O **Oracle8 i Database**, lançado em **1999**, forneceu suporte nativo para protocolos da Internet e suporte do servidor para Java, além de ter sido projetado para computação na Internet, permitindo que o banco de dados seja implantado em um ambiente de várias camadas.
- O **Oracle9 i Database** introduziu o **Oracle RAC** em 2001, permitindo que várias instâncias acessem um único banco de dados simultaneamente.
- O **Oracle Database 10g** introduziu a computação em grade em **2003**, permitindo que as organizações virtualizassem os recursos de computação criando uma infraestrutura de grade baseada em servidores de baixo custo.

HISTÓRIA

- O **Oracle ASM (Oracle Automatic Storage Management)** ajudou a tornar o banco de dados **auto gerenciado** e **auto ajustável**, virtualizando e simplificando o gerenciamento de armazenamento de banco de dados.
- O **Oracle Database 11g**, lançado em **2007**, introduziu uma série de novos recursos que permitiam que administradores e desenvolvedores se adaptassem rapidamente às mudanças nos requisitos de negócios.
- O **Oracle Database 12c**, lançado em **2013**, foi projetado para a nuvem, apresentando uma nova arquitetura **Multitenant**, armazenamento de colunas na memória e suporte para documentos JSON.

DIRETO DO CONCURSO

Questão 1 Ano: 2019 Banca: FCC Órgão: SEFAZ-BA Prova: FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Tecnologia da Informação - Prova II

Um Auditor Fiscal da área de Tecnologia da Informação ao usar o Oracle 11g *release 2* está enfrentando alguns problemas de disponibilidade causados por falhas em um servidor que executa uma aplicação personalizada. Para resolver o problema, sugeriu o uso de um recurso Oracle que permite executar essa aplicação em um conjunto de servidores em *cluster* de forma que se ocorrer uma falha em um deles, o Oracle continue a execução nos servidores restantes, garantindo assim os índices de disponibilidade esperados. O recurso Oracle correto sugerido pelo Auditor é conhecido como Oracle:

- a) RAC.
- b) BACULA.
- c) OAC.
- d) RMAN.
- e) OLTP.

DIRETO DO CONCURSO

Questão 1 Ano: 2019 Banca: FCC Órgão: SEFAZ-BA Prova: FCC - 2019 - SEFAZ-BA - Auditor Fiscal - Tecnologia da Informação - Prova II

Um Auditor Fiscal da área de Tecnologia da Informação ao usar o Oracle 11g *release 2* está enfrentando alguns problemas de disponibilidade causados por falhas em um servidor que executa uma aplicação personalizada. Para resolver o problema, sugeriu o uso de um recurso Oracle que permite executar essa aplicação em um conjunto de servidores em *cluster* de forma que se ocorrer uma falha em um deles, o Oracle continue a execução nos servidores restantes, garantindo assim os índices de disponibilidade esperados. O recurso Oracle correto sugerido pelo Auditor é conhecido como Oracle:

- a) RAC.
- b) BACULA.
- c) OAC.
- d) RMAN.
- e) OLTP.

LETRA A

SCHEMA

- Uma característica de um **SGBDR** é a **independência do armazenamento de dados físicos das estruturas de dados lógicas**.
- No **Oracle Database**, um schema de banco de dados é uma **coleção de estruturas de dados lógicas** ou objetos de esquema.
- Um **usuário** de banco de dados possui um schema de banco de dados, que tem o mesmo nome que o nome de usuário.
- **Objetos de schema** são **estruturas criadas pelo usuário** que se referem diretamente aos dados no banco de dados.

SCHEMA

- O banco de dados suporta muitos tipos de **objetos de schema**, dos quais os mais importantes são **tabelas** e **índices**.
- Um **objeto de schema** é um tipo de **objeto de banco de dados**.
- Alguns **objetos de banco de dados**, como perfis e funções, não residem em **schemas**.

TABELAS

- Uma **tabela** é um **conjunto de linhas**.
- Uma **coluna identifica um atributo da entidade** descrito pela tabela, enquanto uma **linha identifica uma instância da entidade**.
- Em geral, é atribuído a cada **coluna** um **nome**, um **tipo de dados** e uma **largura** ao criar a tabela.
- Opcionalmente, é possível especificar uma regra, chamada **restrição de integridade**, para uma coluna. Um exemplo é uma **NOT NULL restrição de integridade**. Essa restrição força a coluna a conter um valor em cada linha.

ÍNDICES

- Um **índice** é uma **estrutura de dados** opcional que você pode criar em uma ou mais colunas de uma tabela.
- Os índices podem aumentar o desempenho da recuperação de dados.
- Um banco de dados pode usar **índices** disponíveis para **localizar as linhas solicitadas com eficiência**.
- Os **índices** são lógica e fisicamente **independentes dos dados**.
- É possível **descartar e criar índices** sem afetar as tabelas ou outros índices.

LINGUAGEM SQL

- **SQL** é uma **linguagem declarativa** baseada em conjunto que fornece uma interface para um **SGBDR**, como o **Oracle Database**.
- **SQL** não é processual e **descreve o que deve ser feito**.
- **SQL** é a linguagem padrão ANSI para bancos de dados relacionais.
- Todas as operações nos dados em um **Oracle Database** são executadas usando instruções **SQL**.
- O **SQL** é usado para **criar tabelas, consultar e modificar dados** nas tabelas.

LINGUAGEM SQL

- Uma instrução **SQL** pode ser considerada um programa ou instrução de computador.
- Uma instrução **SQL** é uma **sequência de texto SQL**.

```
SELECT first_name, last_name FROM employees;
```

- As instruções **SQL** permitem executar as seguintes tarefas:
 - Consultar dados.
 - Inserir, atualizar e excluir linhas em uma tabela.
 - Crie, substitua, altere e solte objetos.
 - Controlar o acesso ao banco de dados e seus objetos.
 - Garantir consistência e integridade do banco de dados.

LINGUAGEM PL/SQL

- **Oracle SQL** é uma implementação do padrão ANSI.
- O **Oracle SQL** suporta vários recursos que vão além do SQL padrão.
- **PL/SQL** é uma **extensão procedural** para **Oracle SQL**.
- **PL/SQL** é integrado ao **Oracle Database**, permitindo que você use todas as **instruções, funções e tipos de dados SQL** do **Oracle Database**.
- É possível usar o **PL/SQL** para **controlar o fluxo** de um programa SQL, **usar variáveis** e **escrever procedimentos**.

LINGUAGEM PL/SQL

- Um **procedimento** ou **função PL/SQL** é um **objeto de schema** que consiste em um conjunto de instruções **SQL** e outras construções **PL/SQL**.
- As instruções são **agrupadas, armazenadas no banco de dados e executadas como uma unidade** para resolver um problema específico.
- O **Oracle Database** também pode armazenar unidades de programa escritas em **Java**.
- É possível chamar programas **PL/SQL** existentes a partir de **Java** e programas **Java** a partir de **PL/SQL**.

TRANSAÇÕES

- O **Oracle Database** foi projetado como um banco de dados **multiusuário**.
- O banco de dados deve **garantir que vários usuários possam trabalhar simultaneamente** sem danificar os dados uns dos outros.
- Uma **transação** é uma **unidade lógica de trabalho** atômica que **contém uma ou mais instruções SQL**.
- Um **SGBDR** deve poder agrupar **instruções SQL** para que sejam **todas confirmadas**, ou são aplicadas ou são revertidas.

TRANSAÇÕES

- O **Oracle Database** garante que uma transação seja bem-sucedida ou falhe com todas as instruções agrupadas.
- Uma transação move um **Oracle Database** de um estado consistente para outro.
- O princípio básico de uma transação é "tudo ou nada": uma operação atômica é bem-sucedida ou falha como um todo.

ACID

- **ACID** é um conceito que se refere às quatro propriedades de transação de um sistema de banco de dados: **A**tomicidade, **C**onsistência, **I**solamento e **D**urabilidade.
 - **Atomicidade:** Em uma transação envolvendo duas ou mais partes de informações discretas, ou a transação será executada totalmente ou não será executada, garantindo assim que as transações sejam atômicas.
 - **Consistência:** A transação cria um novo estado válido dos dados ou em caso de falha retorna todos os dados ao seu estado antes que a transação foi iniciada.
 - **Isolamento:** Uma transação em andamento deve permanecer isolada de qualquer outra operação, ou seja, garantimos que a transação não será interferida por nenhuma outra transação concorrente.
 - **Durabilidade:** Dados validados são registrados pelo sistema de tal forma que mesmo no caso de uma falha e/ou reinício do sistema, os dados estão disponíveis em seu estado correto.

CONCORRÊNCIA DE DADOS

- Um requisito de um **SGBDR multiusuário** é o controle da **simultaneidade de dados**, que é o acesso simultâneo dos mesmos dados por vários usuários.
- Sem **controles de simultaneidade**, os usuários podem alterar dados incorretamente, **comprometendo a integridade dos dados**.
- O objetivo de um SGBD é **reduzir o tempo de espera**.
- **Interações destrutivas**, que são interações que atualizam dados incorretamente ou alteram estruturas de dados subjacentes, devem ser evitadas.

CONCORRÊNCIA DE DADOS

- O **Oracle Database** usa **bloqueios** para controlar o acesso simultâneo aos dados.
- Um **bloqueio** é um **mecanismo que impede a interação destrutiva** entre transações que acessam um recurso compartilhado.
- Os **bloqueios** ajudam a **garantir a integridade dos dados**, permitindo o máximo acesso simultâneo aos dados.

CONSISTÊNCIA DOS DADOS

- No **Oracle Database**, cada usuário deve ter uma **visão consistente dos dados**, incluindo alterações visíveis feitas pelas próprias transações e transações confirmadas de outros usuários.
- O banco de dados deve **evitar o problema de atualização perdida**, que ocorre quando uma transação vê alterações não confirmadas feitas por outra transação simultânea.
- O **Oracle Database** sempre aplica a consistência de leitura no nível da instrução, **garantindo que os dados retornados por uma única consulta sejam confirmados e consistentes por um único momento**.

CONSISTÊNCIA DOS DADOS

- Dependendo do **nível de isolamento da transação**, este ponto é o horário em que a instrução foi aberta ou a hora em que a transação começou.
- O banco de dados pode fornecer **consistência de leitura para todas as consultas em uma transação**, conhecida como consistência de leitura no nível da transação.
- Cada instrução em uma transação vê **dados do mesmo ponto no tempo**, que é o horário em que a transação começou.

CONSISTÊNCIA DOS DADOS

- CONNECT / DISCONNECT
- Abre uma conexão com o Oracle CONN[ECT] usuário/senha;
- DISC[ONNECT];
- Finaliza a conexão com o Oracle
- Torna persistente (i.e., commit) as alterações pendentes no banco de dados.

COMMIT

- Finaliza uma transação com sucesso
 - Atualiza os dados do banco de dados – torna persistente as alterações da transação
 - Apaga os savepoints da transação
 - Libera todas as travas utilizadas pela transação
-
- DECLARE ...
 - BEGIN ...
 - UPDATE cursa SET nT2 = 0.0 WHERE NUSP = 2;
 - UPDATE cursa SET nT2 = 0.0 WHERE NUSP = 5;
 - COMMIT;
 - END;

ROLLBACK

- Finaliza uma transação sem sucesso
- Desfaz os efeitos das operações da transação
- Apaga os savepoints da transação
- Libera todas as travas utilizadas pela transação

- DECLARE ...
- BEGIN ...
- INSERT INTO aluno VALUES (1, 'Alexandra', 'f');
- INSERT INTO cursa VALUES (1, 1, 7.0, 8.0, 7.5, 9.0, NULL);
- ...
- EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
- ROLLBACK;
- ...
- END;

ROLLBACK

- DECLARE aluno_id aluno.NUSP%Type;
- BEGIN
- UPDATE aluno SET ... WHERE NUSP = aluno_id;
- DELETE FROM aluno WHERE SAVEPOINT do_insert;
- INSERT INTO aluno VALUES (...);
- EXCEPTION WHEN DUP_VAL_ON_INDEX THEN
- ROLLBACK TO do_insert;
- END;

SAVEPOINT

- Podem existir vários savepoints em uma transação
- – commit ou rollback: remove todos savepoints
- – rollback to: remove savepoints marcados depois do savepoint de retorno
- Nomes de savepoints
- – não precisam ser declarados
- – podem ser usados novamente em uma transação

ARQUITETURA

- Um **servidor de banco de dados** é a chave para o gerenciamento de informações.
- Um **servidor** gerencia confiavelmente uma grande quantidade de dados em um ambiente multiusuário, para que **os usuários possam acessar simultaneamente os mesmos dados**.
- Um **servidor de banco de dados** também **impede o acesso não autorizado** e fornece soluções eficientes para recuperação de falhas.

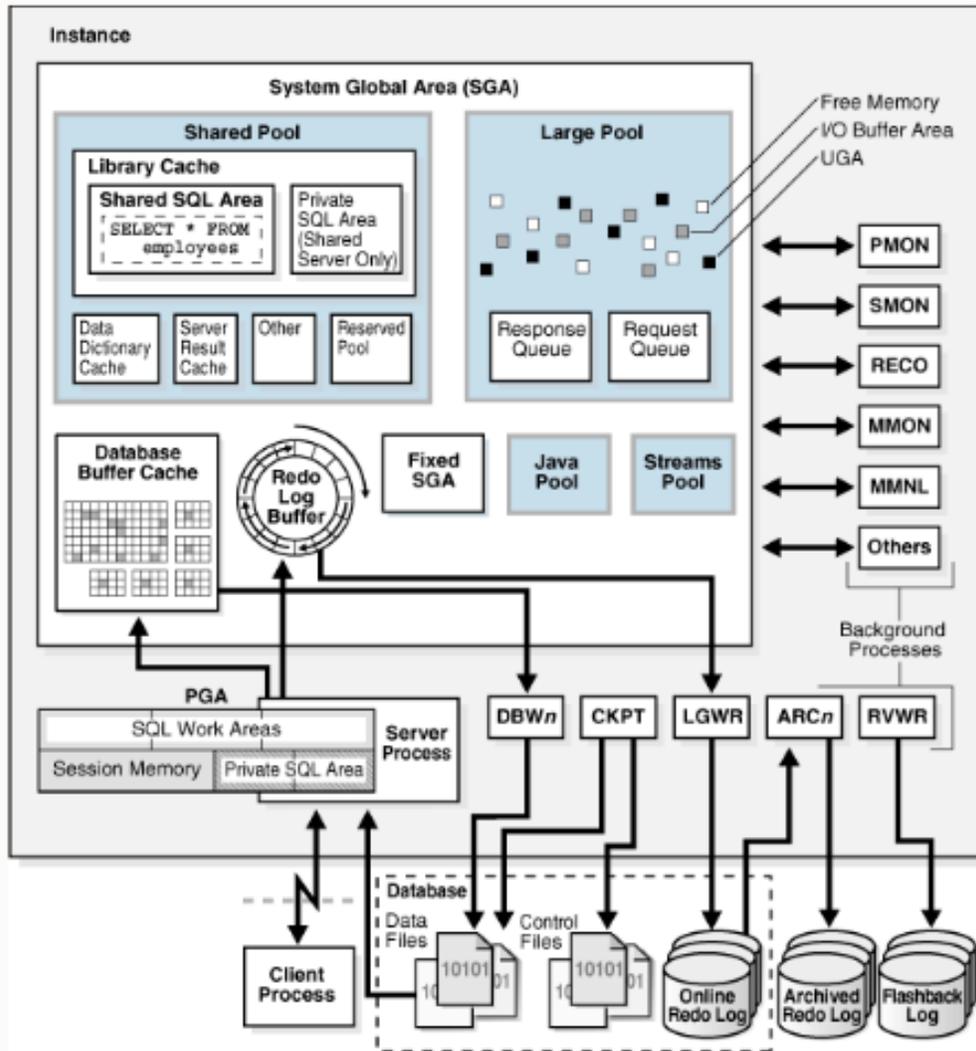
ARQUITETURA

- Um servidor de **Oracle Database** consiste em **um banco de dados** e em pelo menos **uma instância** de banco de dados.
- Como uma instância e um banco de dados estão tão intimamente conectados, o termo **Oracle Database** às vezes é usado para se **referir à instância e ao banco de dados**.
- **Banco de dados:**
 - É um **conjunto de arquivos**, localizados no disco, que armazenam dados.
 - Os arquivos do **banco de dados** podem existir independentemente de uma instância do banco de dados.

ARQUITETURA

- **Instância de banco de dados:**
 - É um **conjunto de estruturas de memória** que gerenciam arquivos de banco de dados.
 - A **instância** consiste em uma **área de memória compartilhada**, denominada **área global do sistema (SGA, System Global Area)**, e um **conjunto de processos em segundo plano (Background Processes)**.
 - Uma **instância** pode existir independentemente dos arquivos do banco de dados.

ARQUITETURA



- Para cada conexão de usuário com a instância, um processo de cliente (**client process**) executa o aplicativo.
- Cada processo de cliente está associado ao seu próprio processo de servidor (**server process**).
- O processo de servidor possui sua própria **memória de sessão privada**, a **área global do programa (PGA, Program Global Area)**.

ESTRUTURAS DE ARMAZENAMENTO

- Um banco de dados pode ser considerado da **perspectiva física e lógica**.
- **Dados físicos** são dados visíveis no nível do sistema operacional.
- **Dados lógicos**, como uma tabela, são significativos apenas para o banco de dados.
- O banco de dados possui **estruturas físicas e estruturas lógicas**.
- É possível **gerenciar o armazenamento físico de dados** sem afetar o acesso às **estruturas de armazenamento lógico**.

ESTRUTURAS DE ARMAZENAMENTO

- **Estruturas de Armazenamento Físicas (Physical Storage Structures):**
 - São os arquivos que armazenam os dados.
 - Ao executar uma instrução **CREATE DATABASE** alguns arquivos são criados.
 - **Arquivos de dados (data files):** arquivos de dados físicos, que contêm todos os dados do banco de dados.
 - **Arquivos de controle (control files):** contém metadados que especificam a estrutura física do banco de dados, incluindo o nome do banco de dados e os nomes e locais dos arquivos de banco de dados.
 - **Arquivos de log (online redo log files):** registram todas as alterações feitas nos dados.

ESTRUTURAS DE ARMAZENAMENTO

- **Estruturas de Armazenamento Lógicas (Logical Storage Structures):**
 - Permitem que o **Oracle Database** tenha um controle refinado do uso do espaço em disco.
 - **Blocos de dados (data blocks):** os dados do **Oracle Database** são armazenados em blocos de dados. Corresponde a um número específico de bytes no disco.
 - **Extensões (extents):** é um número específico de blocos de dados logicamente contíguos, obtidos em uma única alocação, usado para armazenar um tipo específico de informação.

ESTRUTURAS DE ARMAZENAMENTO

- **Segmentos (segments):** é um conjunto de extensões alocadas para um objeto de usuário (tabela ou índice), desfazer dados ou dados temporários.
- **Espaços de tabela (tablespaces):** um banco de dados é dividido em unidades de armazenamento lógico chamadas de espaços de tabela. Um espaço de tabela é o contêiner lógico para segmentos. Cada espaço de tabela consiste em pelo menos um arquivo de dados.

ESTRUTURAS DE INSTÂNCIA

- Um **Oracle Database** usa **estruturas e processos de memória** para gerenciar e acessar o banco de dados.
- Quando os aplicativos se conectam a um **Oracle Database**, eles se conectam a uma **instância de banco de dados**.
- A instância atende aos aplicativos alocando outras áreas de memória além da SGA (**system global area**) e iniciando outros processos além dos processos em segundo plano (**background processes**).

ESTRUTURAS DE INSTÂNCIA

- **Processos de Banco de Dados Oracle (Oracle Database Processes)**
 - Um **processo** é um mecanismo em um sistema operacional que pode executar uma série de etapas.
 - Uma **instância de banco de dados Oracle** possui alguns tipos de processos.
 - **Processos de cliente (client processes)**: criados e mantidos para executar o código de um programa ou aplicativo ou de uma ferramenta Oracle.
 - **Processos em segundo plano (background processes)**: consolidam funções que, de outra forma, seriam tratadas por vários programas do **Oracle Database** em execução para cada processo do cliente.

ESTRUTURAS DE INSTÂNCIA

- **Processos em segundo plano (background processes):** executam E/S de forma assíncrona e monitoram outros processos do **Oracle Database** para fornecer maior paralelismo para melhor desempenho e confiabilidade.
- **Processos de servidor (server processes):** se comunicam com os processos do cliente e interagem com o **Oracle Database** com o objetivo de atender às solicitações.
- Os processos **Oracle** incluem processos do servidor e processos em segundo plano.
- Na maioria dos ambientes, os processos **Oracle** e clientes são executados em computadores separados.

ESTRUTURAS DE INSTÂNCIA

- **Estruturas de Instância de Memória (Instance Memory Structures)**
 - O **Oracle Database** cria e usa estruturas de memória para código de programa, dados compartilhados entre usuários e áreas de dados privadas para cada usuário conectado.
 - **Área global do sistema (system global area):** é um grupo de estruturas de memória compartilhada que contêm dados e informações de controle para uma instância de banco de dados.
 - **Área global do programa (program global area):** é uma região de memória que contém dados e informações de controle para um servidor ou processo em segundo plano. O acesso ao PGA é exclusivo para o processo. Cada processo de servidor e processo de segundo plano tem seu próprio PGA.

ARQUITETURA DE APLICAÇÃO

- Uma arquitetura de aplicação é o ambiente no qual **um aplicativo de banco de dados se conecta a um Oracle Database**.
- As duas arquiteturas de banco de dados mais comuns são **cliente/servidor** e **multicamadas**.
- Em uma **arquitetura cliente/servidor**, o aplicativo cliente **inicia uma solicitação** para que uma **operação seja executada no servidor** de banco de dados.
- O servidor executa o **Oracle Database** e **gerencia o acesso simultâneo a dados compartilhados**.
- O servidor recebe e processa solicitações originadas de clientes.

ARQUITETURA DE APLICAÇÃO

- Em uma **arquitetura multicamada tradicional**, um ou mais servidores de aplicativos **executam partes da operação**.
- Um **servidor de aplicação contém grande parte da lógica do aplicativo**, fornece acesso aos dados para o cliente e executa algum processamento de consulta.
- **A carga no banco de dados diminui**.
- O servidor de aplicação pode servir como uma **interface entre clientes e os bancos de dados**, além de fornecer um nível adicional de segurança.

ARQUITETURA DE APLICAÇÃO

- Uma **arquitetura orientada a serviços (SOA)** é uma arquitetura multicamada na qual a funcionalidade do aplicativo é encapsulada em serviços.
- Os serviços SOA são geralmente implementados como **serviços da Web**.
- O **Oracle Database** pode atuar como um **provedor de serviços da Web** em um ambiente tradicional de várias camadas ou SOA.

ARQUITETURA DE REDE

- O **Oracle Net Services** é a interface entre o banco de dados e os protocolos de comunicação de rede.
- Processamento distribuído e bancos de dados distribuídos.
- Os protocolos de comunicação definem a maneira como os dados são transmitidos e recebidos em uma rede.
- O **Oracle Net**, um componente do **Oracle Net Services**, estabelece e mantém uma sessão de rede de um aplicativo cliente com um servidor de banco de dados.

ARQUITETURA DE REDE

- O **Oracle Net** atua como um transportador de dados entre o aplicativo cliente e o servidor de banco de dados, **trocando mensagens entre eles**.
- Outro componente importante do **Net Services** é o **Oracle Net Listener**, que é um processo executado no banco de dados ou em outro local da rede.
- Os aplicativos clientes enviam solicitações de conexão ao **ouvinte (listener)**, que **gerencia o tráfego dessas solicitações para o banco de dados**.
- Quando uma conexão é estabelecida, o cliente e o banco de dados se comunicam diretamente.

ARQUITETURA DE REDE

- **Arquitetura de servidor dedicado:**

- Cada processo do cliente se conecta a um processo do servidor dedicado.
- O processo do servidor não é compartilhado com nenhum outro cliente durante a sessão do cliente.
- A cada nova sessão é atribuído um processo de servidor dedicado.

- **Arquitetura de servidor compartilhado:**

- O banco de dados usa um conjunto de processos do servidor compartilhado para várias sessões.
- Um processo do cliente se comunica com um **expedidor (dispatcher)**, que permite que muitos clientes se conectem à mesma instância de banco de dados, sem a necessidade de um processo de servidor dedicado para cada cliente.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** CESPE / CEBRASPE **Órgão:** TJ-RJ **Prova:** CESPE / CEBRASPE - 2021
- TJ-RJ - Analista Judiciário - Analista de Sistemas

Assinale a opção que apresenta o nome do arquivo utilizado no sistema gerenciador de banco de dados Oracle para armazenar os nomes de serviço de rede, de modo que a aplicação (cliente) consiga se conectar ao servidor Oracle, e permitir também a adição de strings de conexão, manualmente, pelo cliente.

Alternativas

- a) oratab
- b) spfile.ora
- c) listener.ora
- d) .bashprofile
- e) tnsnames.ora

DIRETO DO CONCURSO

Ano: 2021 **Banca:** CESPE / CEBRASPE **Órgão:** TJ-RJ **Prova:** CESPE / CEBRASPE - 2021
- TJ-RJ - Analista Judiciário - Analista de Sistemas

Assinale a opção que apresenta o nome do arquivo utilizado no sistema gerenciador de banco de dados Oracle para armazenar os nomes de serviço de rede, de modo que a aplicação (cliente) consiga se conectar ao servidor Oracle, e permitir também a adição de strings de conexão, manualmente, pelo cliente.

Alternativas

- a) oratab
- b) spfile.ora
- c) listener.ora
- d) .bashprofile
- e) **tnsnames.ora**

ARQUITETURA MULTITENANT

- A **arquitetura multitenant** permite que um banco de dados Oracle seja um **CDB (Multitenant Container Database)**.
- Um **CDB** é um **único banco de dados físico** que contém zero, um ou muitos **bancos de dados conectáveis** criados pelo usuário.
- Um **banco de dados conectável (pluggable database)** é uma coleção portátil de schemas, objetos de schema e objetos de não schema que aparece para um cliente **Oracle Net** como um **não-CDB**.
- Um **não-CDB** é um banco de dados Oracle **tradicional** que não pode conter **PDBs**.

ARQUITETURA MULTITENANT

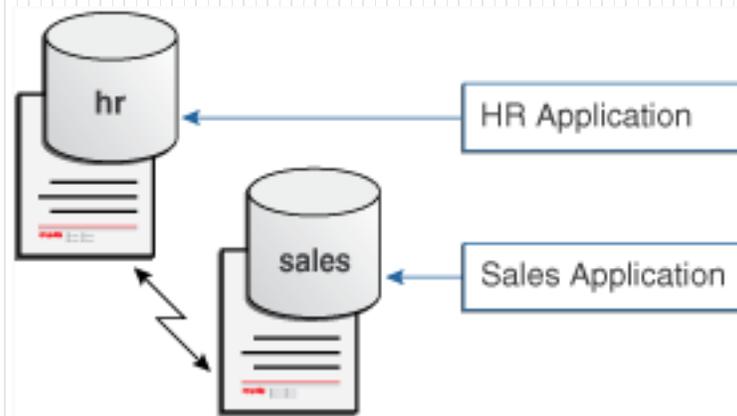
- Ao **consolidar** vários bancos de dados físicos de servidores separados **em um único banco de dados em um único servidor**, a **arquitetura multitenant** oferece os seguintes benefícios:
 - Redução de custos de hardware.
 - Movimentação mais fácil e rápida de dados e códigos.
 - Gerenciamento e monitoramento mais fáceis do banco de dados físico.
 - Separação de dados e código.
 - Separação de funções entre um **administrador do PDB**, que gerencia apenas os PDBs aos quais ele ou ela recebe privilégios, e o **administrador do CDB**, que gerencia todo o CDB.

ARQUITETURA MULTITENANT

- Os benefícios da **capacidade de gerenciamento** incluem:
 - Atualização mais fácil de dados e códigos desconectando e conectando **PDBs**.
 - Testes mais fáceis usando **PDBs** para desenvolvimento antes de conectá-los ao **CDB** de produção.
 - Capacidade de definir limites de desempenho para memória e E/S no nível do **PDB**.
 - Capacidade de instalar, atualizar e gerenciar uma definição de aplicativo principal em um contêiner de aplicativo, que é um conjunto de **PDBs** conectados a uma raiz de aplicativo comum.

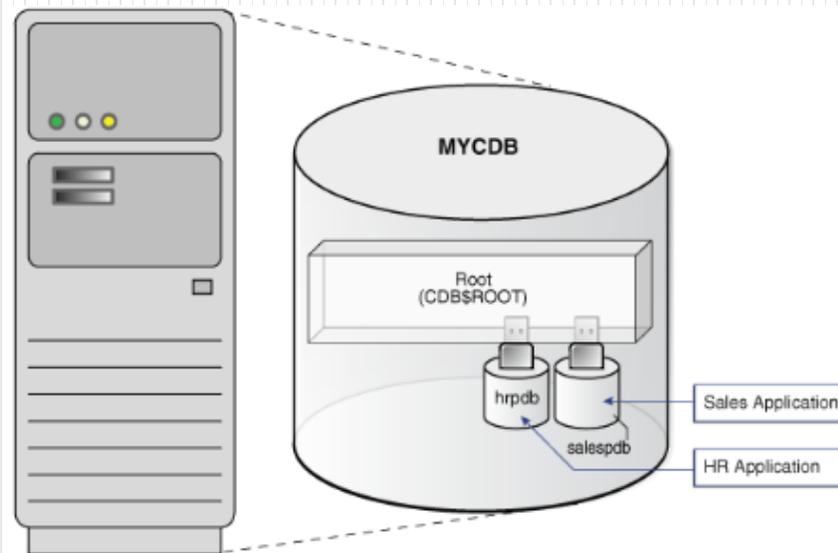
ARQUITETURA MULTITENANT

- Arquitetura **não-CDB**:
 - Cada **não-CDB** possui sua própria memória e conjunto de arquivos de banco de dados e reside em seu próprio computador.
 - Cada **não-CDB** possui seu próprio aplicativo de usuário dedicado.



ARQUITETURA MULTITENANT

- Arquitetura **Multitenant**:
 - **MYCDB** contém dois **PDBs**: **hrpdb** e **salespdb**.
 - Para administrar o próprio **CDB** ou qualquer **PDB** dentro dele, um administrador do **CDB** pode se conectar à raiz do **CDB**



DIRETO DO CONCURSO

Questão 2 Ano: 2019 Banca: IBFC Órgão: IDAM Prova: IBFC - 2019 - IDAM - Técnico de Nível Superior - Analista de Redes

Quanto aos Sistemas de Gerenciamento de Bancos de Dados (SGBD), analise as afirmativas abaixo e assinale a alternativa correta.

- I. o MySQL, mesmo depois que foi adquirido pela Oracle, utiliza a linguagem SQL.
 - II. uma das linguagens que o PostgreSQL utiliza é o PL/ pgSQL, assim como PL/Python.
 - III. para o processamento de transações o Oracle utiliza da linguagem de programação PL/SQL.
- a) Apenas as afirmativas I e II são tecnicamente verdadeiras
 - b) Apenas as afirmativas II e III são tecnicamente verdadeiras
 - c) Apenas as afirmativas I e III são tecnicamente verdadeiras
 - d) As afirmativas I, II e III são tecnicamente verdadeiras

DIRETO DO CONCURSO

Questão 2 Ano: 2019 Banca: IBFC Órgão: IDAM Prova: IBFC - 2019 - IDAM - Técnico de Nível Superior - Analista de Redes

Quanto aos Sistemas de Gerenciamento de Bancos de Dados (SGBD), analise as afirmativas abaixo e assinale a alternativa correta.

- I. o MySQL, mesmo depois que foi adquirido pela Oracle, utiliza a linguagem SQL.
 - II. uma das linguagens que o PostgreSQL utiliza é o PL/ pgSQL, assim como PL/Python.
 - III. para o processamento de transações o Oracle utiliza da linguagem de programação PL/SQL.
- a) Apenas as afirmativas I e II são tecnicamente verdadeiras
 - b) Apenas as afirmativas II e III são tecnicamente verdadeiras
 - c) Apenas as afirmativas I e III são tecnicamente verdadeiras
 - d) As afirmativas I, II e III são tecnicamente verdadeiras

LETRA D

DIRETO DO CONCURSO

Questão 3 Ano: 2018 Banca: UFPR Órgão: UFPR Prova: UFPR - 2018 - UFPR
- Técnico de Tecnologia da Informação

Em relação à TABLESPACE no Oracle e no PostgreSQL, é correto afirmar:

- a) As “tablespace” são arquivos de índice que organizam os dados das tabelas segundo uma chave e ordem definidas pelo usuário.
- b) No PostgreSQL, as tablespace são compostas por arquivos físicos indicados na cláusula DATAFILE do comando CREATE TABLESPACE.
- c) Em PostgreSQL, os objetos são armazenados em um único tablespace.
- d) No Oracle, a tablespace temporária padrão do usuário é utilizada para armazenar os segmentos undo.
- e) As “tablespace” dividem os bancos de dados fisicamente.

DIRETO DO CONCURSO

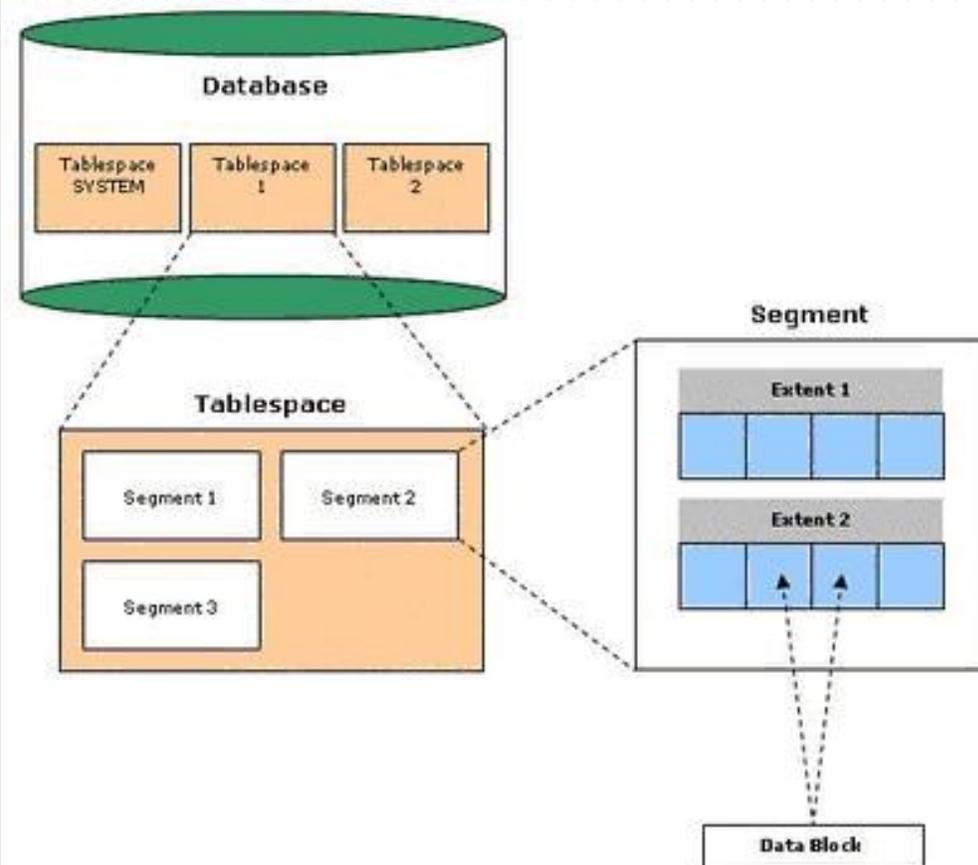
Resposta: LETRA E

Enquanto os **esquemas** dividem um banco de dados em **particionamentos lógicos**, o **tablespace** divide um banco de dados **fisicamente**.

Tablespace no ORACLE é uma pasta no sistema de arquivos utilizada para armazenamento físico dos dados e objetos de um banco de dados. O Oracle armazena dados logicamente em tablespaces e fisicamente em arquivos de dados (datafiles).

A melhor analogia para se explicar banco de dados, tablespace, arquivo de dados, tabelas e dados é a imagem de um fichário. Imagine um banco de dados como um fichário: as gavetas dentro do fichário são os tablespaces; as pastas nessas gavetas são os arquivos de dados; os papéis em cada pasta são as tabelas; a informação escrita no papel de cada pasta são os dados. Em resumo, o tablespace é um modo de agrupar arquivos de dados.

Referência: <https://www.oracle.com/technetwork/pt/articles/database-performance/introducao-conceito-de-tablespaces-495850-ptb.html>



TABLESPACE

O que é o tablespace **USERS**?

Tablespace padrão para os usuários. Se um usuário criar um objeto, tal como uma tabela ou um índice, sem especificar o tablespace, o Oracle o cria no tablespace padrão do usuário, isso se o tablespace padrão do usuário foi definido para utilizar o tablespace **USERS**.

O que é o tablespace **SYSTEM**?

O tablespace **SYSTEM** (tablespace de sistema) é uma parte obrigatória de todo banco de dados Oracle. É onde o Oracle armazena todas as informações necessárias para o seu próprio gerenciamento. Em resumo, **SYSTEM** é o tablespace mais crítico do banco de dados porque ele contém o dicionário de dados. Se por algum motivo ele se tornar indisponível, a instância do Oracle abortará.

Por esse motivo, o tablespace **SYSTEM** nunca pode ser colocado offline, ao contrário de um tablespace comum como, por exemplo, o tablespace **USERS**.

O que é o tablespace **TEMP**?

O tablespace **TEMP** (tablespace temporário) é onde o Oracle armazena todas as suas tabelas temporárias. É o quadro branco ou papel de rascunho do banco de dados. Assim como às vezes precisamos de um lugar para anotar alguns números para poder somá-los, o Oracle também precisa de algum espaço em disco temporário. O Oracle geralmente utiliza o tablespace temporário para armazenar objetos transitórios durante as classificações e agrupamentos de dados durante a execução de uma SQL contendo as cláusulas **ORDER BY** e **GROUP BY**, entre outras.

É importante dizer também que os dados de sessão das tabelas temporárias globais (Global Temporary Tables) também ficam no tablespace **TEMP**. Assim como o tablespace **SYSTEM** é o tablespace mais crítico do banco de dados, o tablespace **TEMP** é o **menos crítico do banco de dados exatamente porque armazena apenas os segmentos temporários** durante as operações de classificação de dados e, como tal, no caso de uma falha, ele pode simplesmente ser dropado e recriado, em vez de ser restaurado e recuperado.

TABLESPACE

O que é o tablespace UNDO?

Todos os bancos de dados Oracle precisam de um local para armazenar informações a desfazer. O que isso significa? Esse tablespace que contém seus segmentos de reconstrução em versões anteriores ao Oracle 9i chamado de RBS (tablespace de rollback), possui a capacidade de recuperar transações incompletas ou abortadas.

Um segmento de undo é usado para salvar o valor antigo quando um processo altera dados de um banco de dados. Ele armazena a localização dos dados e também os dados da forma como se encontravam antes da modificação.

O que é o tablespace SYSAUX?

Este tablespace auxiliar não existe nas versões anteriores ao Oracle 10g e foi criado especialmente para aliviar o tablespace **SYSTEM** de segmentos associados a algumas aplicações do próprio banco de dados.

Como resultado da criação desse tablespace, alguns gargalos de I/O frequentemente associados ao tablespace **SYSTEM** foram reduzidos ou eliminados. Vale a pena salientar que não é bom que o tablespace **SYSAUX** seja colocado no modo offline, pelo fato de correr o risco do banco de dados não funcionar corretamente.

DIRETO DO CONCURSO

Questão 4 Ano: 2018 Banca: FAURGS Órgão: BANRISUL Prova: FAURGS - 2018 - BANRISUL -
Administração de Bancos de Dados

Sobre as *tablespaces* criadas automaticamente durante o processo de criação de um banco de dados no Oracle 11G, é correto afirmar que:

- a) UNDO contém as tabelas utilizadas por ferramentas do próprio banco de dados.
- b) USERS contém informações de recuperação de transações.
- c) SYSAUX contém dados temporários, criados durante a execução de instruções SQL.
- d) SYSTEM contém o Dicionário de Dados e todas as informações para o gerenciamento do banco de dados.
- e) TEMP é a tabela padrão que contém todos os objetos criados pelo usuário.

DIRETO DO CONCURSO

Resposta: LETRA D

O tablespace **SYSTEM** é responsável por armazenar o dicionário de dados e seus respectivos objetos (metadados). Quanto mais objetos forem criados nos bancos de dados de uma instância Oracle, maior será este tablespace.

SEQUÊNCIAS

- Uma **sequência** é um objeto de esquema a partir do qual vários usuários podem gerar números inteiros exclusivos.
- Um **gerador de sequência** fornece um método altamente escalonável e de bom desempenho para gerar chaves substitutas para um tipo de dados numérico.
- O exemplo a seguir cria a sequência **customers_seq**.

```
CREATE SEQUENCE customers_seq
START WITH      1000
INCREMENT BY    1
NOCACHE
NOCYCLE;
```

SEQUÊNCIAS

- O mesmo **gerador de sequência** pode gerar números para várias tabelas.
- O gerador pode criar **chaves primárias** automaticamente e coordenar chaves em várias linhas ou tabelas.
- O **gerador de sequência** é útil em ambientes multiusuário para gerar números exclusivos sem a sobrecarga de E/S do disco ou bloqueio de transação.
- A **sequência** gera automaticamente os valores corretos para cada usuário.

SCRIPT

```
SQL> create sequence pubs1;
```

Sequence created.

```
SQL> select pubs1.nextval from dual;
```

```
  NEXTVAL  
-----  
         1
```

```
SQL> select pubs1.nextval from dual;
```

```
  NEXTVAL  
-----  
         2
```

When creating a sequence, there is a lot of flexibility in how the sequence generates the next number using the Oracle NEXTVAL function:

```
SQL> create sequence pubs2
```

```
  2 start with 8  
  3 increment by 2  
  4 maxvalue 10000  
  5 cycle  
  6 cache 5;
```

Sequence created.

```
SQL> select pubs2.nextval from dual;
```

```
  NEXTVAL  
-----  
         8
```

In the sequence example, the Oracle NEXTVAL function would return 10 as the next number in the sequence.

https://docs.oracle.com/cd/B12037_01/server.101/b10759/statements_6014.htm

SINÔNIMOS

- Um **sinônimo** é um alias para um objeto de esquema.
- Não requer armazenamento além de sua definição no dicionário de dados.
- Os **sinônimos** podem simplificar as instruções SQL para usuários do banco de dados.
- É possível criar sinônimos **privados** e **públicos**.
- Um **sinônimo privado** está no esquema de um usuário específico que tem controle sobre sua disponibilidade para outros.
- Um **sinônimo público** é acessível a todos os usuários do banco de dados.
- Sinônimos Privados são mais recomendados por motivo de segurança.

SINÔNIMOS

- Um sinônimo pode ser criado para uma **table**, **view**, **sequence**, **procedure**, **function** ou **package** no Banco de Dados Local.
- Podemos também criar sinônimo via Database Link para um objeto em um outro Banco de Dados.
- Sinônimos Públicos - somente usuários com privilegio DBA podem criar.

SCRIPT

```
CREATE OR REPLACE SYNONYM <synonym_name> FOR  
<object_name>;  
DROP SYNONYM <synonym_name>;
```

```
SQL> SELECT * FROM USER.CLIENTE;
```

```
ID NAME
```

```
-----
```

```
1 JOSE  
2 MARIA  
3 JOANA  
4 JOAO
```

SCRIPT

```
SQL> CREATE OR REPLACE SYNONYM CLIENTE FOR USER.CLIENTE;
```

Synonym created.

```
SQL> SELECT * FROM CLIENTE;
```

ID NAME

1 JOSE

2 MARIA

3 JOANA

4 JOAO

INTEGRIDADE DOS DADOS

- É importante que os dados mantenham a **integridade dos dados**, que é a **aderência às regras de negócios**.
- As **regras de negócios** especificam condições e relacionamentos que sempre devem ser verdadeiros ou sempre falsos.
- Uma **restrição de integridade** é um objeto de esquema criado e descartado usando SQL.
- **Facilidade declarativa**: as restrições de integridade são definidas usando instruções SQL.
- **Regras centralizadas**: definidas para tabelas e armazenadas no dicionário de dados .

INTEGRIDADE DOS DADOS

- O **Oracle Database** permite aplicar restrições no nível da **tabela** e da **coluna**.
- Uma **chave** é a coluna ou o conjunto de colunas incluídas na definição de certos tipos de restrições de integridade.
- Valores individuais em uma chave são chamados de **valores-chave**.
- As chaves **descrevem os relacionamentos entre as tabelas e colunas** de um banco de dados relacional.

INTEGRIDADE DOS DADOS

- Restrições de integridade:

NOT NULL	Permite ou não permite inserções ou atualizações de linhas contendo um nulo em uma coluna especificada.
Chave única	Proíbe que várias linhas tenham o mesmo valor na mesma coluna ou combinação de colunas, mas permite que alguns valores sejam nulos.
Chave primária	Combina uma NOT NULL restrição e uma restrição exclusiva. Proíbe várias linhas de terem o mesmo valor na mesma coluna ou combinação de colunas e proíbe valores de serem nulos.

Chave estrangeira	Designa uma coluna como chave estrangeira e estabelece um relacionamento entre a chave estrangeira e uma chave primária ou exclusiva, chamada chave referenciada .
Verifica	Requer um valor de banco de dados para obedecer a uma condição especificada.

INTEGRIDADE DOS DADOS

- É possível especificar como e quando o **Oracle Database** deve impor a restrição, determinando o estado da restrição.
- O banco de dados permite especificar se uma restrição **se aplica a dados existentes ou futuros**.
- É possível definir restrições para validar (**VALIDATE**) ou não validar (**NOVALIDATE**) os dados existentes.

INTEGRIDADE DOS DADOS

ENABLE	VALIDATE	Os dados existentes e futuros devem obedecer à restrição. Uma tentativa de aplicar uma nova restrição a uma tabela preenchida resulta em um erro se as linhas existentes violarem a restrição.
ENABLE	NOVALIDATE	O banco de dados verifica a restrição, mas não precisa ser verdadeiro para todas as linhas. Portanto, as linhas existentes podem violar a restrição, mas as linhas novas ou modificadas devem estar em conformidade com as regras.
DISABLE	VALIDATE	O banco de dados desabilita a restrição, elimina seu índice e evita a modificação das colunas restritas.
DISABLE	NOVALIDATE	A restrição não está marcada e não é necessariamente verdadeira.

TIPOS DE DADOS

- **Tipos Numéricos** — NUMBER.
- **Tipos de Data** — DATE, TIMESTAMP.
- **Tipos de String** — CHAR, VARCHAR2 limite de 4000 Bytes.

https://docs.oracle.com/database/121/SQLRF/sql_elements001.htm#SQLRF0021

```
CREATE TABLE employees
( employee_id  NUMBER(6)
, first_name  VARCHAR2(20)
, last_name   VARCHAR2(25)
  CONSTRAINT emp_last_name_nn NOT NULL
, email       VARCHAR2(25)
  CONSTRAINT emp_email_nn NOT NULL
, phone_number VARCHAR2(20)
, hire_date   DATE
  CONSTRAINT emp_hire_date_nn NOT NULL
, job_id      VARCHAR2(10)
  CONSTRAINT emp_job_nn NOT NULL
, salary      NUMBER(8,2)
, commission_pct NUMBER(2,2)
, manager_id  NUMBER(6)
, department_id NUMBER(4)
, CONSTRAINT emp_salary_min
  CHECK (salary > 0)
, CONSTRAINT emp_email_uk
  UNIQUE (email)
);
```

DIRETO DO CONCURSO

Ano: 2021 **Banca:** SELECON **Órgão:** EMGEPRON **Prova:** SELECON - 2021 - EMGEPRON - Analista de Sistemas (Auditoria)

No que diz respeito aos tipos de dados suportados pelo banco de dados Oracle, para armazenar caracteres de tamanho variável e números inteiros, as variáveis devem ser declaradas, respectivamente, dos seguintes tipos:

Alternativas

- a) VARCHAR e NUMBER
- b) VARCHAR e INTEGER
- c) STRING e NUMBER
- d) STRING e INTEGER

DIRETO DO CONCURSO

Ano: 2021 **Banca:** SELECON **Órgão:** EMGEPRON **Prova:** SELECON - 2021 - EMGEPRON - Analista de Sistemas (Auditoria)

No que diz respeito aos tipos de dados suportados pelo banco de dados Oracle, para armazenar caracteres de tamanho variável e números inteiros, as variáveis devem ser declaradas, respectivamente, dos seguintes tipos:

Alternativas

- a) **VARCHAR e NUMBER**
- b) VARCHAR e INTEGER
- c) STRING e NUMBER
- d) STRING e INTEGER

DICIONÁRIO DOS DADOS

- É um **conjunto de tabelas somente leitura** que fornece metadados administrativos sobre o banco de dados.
- Um **dicionário de dados** contém informações como as seguintes:
 - As definições de cada objeto de esquema no banco de dados, incluindo valores padrão para colunas e informações de restrição de integridade.
 - A quantidade de espaço alocado e atualmente usado pelos objetos de esquema.
 - Os nomes dos usuários do **Oracle Database**, privilégios e funções concedidos aos usuários e informações de auditoria relacionadas aos usuários.

DICIONÁRIO DOS DADOS

- O **dicionário de dados** é uma parte central do gerenciamento de dados para todos os bancos de dados Oracle.
- O **Oracle Database** armazena dados do dicionário de dados em tabelas, assim como outros dados, os **usuários podem consultar os dados com SQL**.
- O **dicionário de dados** consiste em tabelas e visualizações básicas.

DICIONÁRIO DOS DADOS

- Conjuntos de exibição do **dicionário de dados**:

Prefixo	Acesso do Usuário	Conteúdo	Notas
DBA_	Administradores de banco de dados	Todos os objetos	Algumas DBA_ visualizações têm colunas adicionais contendo informações úteis para o administrador.
ALL_	Todos os usuários	Objetos para os quais o usuário tem privilégios	Inclui objetos de propriedade do usuário. Essas visualizações obedecem ao conjunto atual de funções ativadas.
USER_	Todos os usuários	Objetos pertencentes ao usuário	As visualizações com o prefixo USER_ geralmente excluem a coluna OWNER. Essa coluna está implícita nas USER_ visualizações para o usuário que está emitindo a consulta.

DIRETO DO CONCURSO

Ano: 2022 Banca: FAURGS Órgão: SES-RS Prova: FAURGS - 2022 - SES-RS - Administrador de Banco de Dados - Edital nº 15

Considere as seguintes afirmações sobre o Dicionário de Dados (DD) do Sistema de Gerência de Banco de Dados Oracle (19c ou superior).

I - O DD contém informações sobre usuários, objetos de esquema e estruturas de armazenamento.

II - O DD é modificado toda vez que um comando DDL é executado com sucesso.

III - O Oracle fornece visões sobre porções de interesse do DD, prefixadas por DBA_ e USER_. As visões com prefixo DBA_ são voltadas ao administrador da base de dados e contêm todos os objetos da base. As visões com prefixo USER_ permitem a um usuário específico ver seus objetos e os objetos sobre os quais tem privilégios.

Quais estão corretas?

Alternativas

- a) Apenas I.
- b) Apenas II.
- c) Apenas III.
- d) Apenas I e II.
- e) I, II e III.

DIRETO DO CONCURSO

Ano: 2022 Banca: FAURGS Órgão: SES-RS Prova: FAURGS - 2022 - SES-RS - Administrador de Banco de Dados - Edital nº 15

Considere as seguintes afirmações sobre o Dicionário de Dados (DD) do Sistema de Gerência de Banco de Dados Oracle (19c ou superior).

I - O DD contém informações sobre usuários, objetos de esquema e estruturas de armazenamento.

II - O DD é modificado toda vez que um comando DDL é executado com sucesso.

III - O Oracle fornece visões sobre porções de interesse do DD, prefixadas por DBA_ e USER_. As visões com prefixo DBA_ são voltadas ao administrador da base de dados e contêm todos os objetos da base. As visões com prefixo USER_ permitem a um usuário específico ver seus objetos e os objetos sobre os quais tem privilégios.

Quais estão corretas?

Alternativas

- a) Apenas I.
- b) Apenas II.
- c) Apenas III.
- d) **Apenas I e II.**
- e) I, II e III.

SQL

- **SQL** é uma **linguagem declarativa** de alto nível baseada em conjunto com a qual todos os programas e usuários acessam dados em um banco de dados Oracle.
- O **SQL** fornece uma interface para o **Oracle Database**.

```
SELECT    last_name, first_name
FROM      hr.employees
WHERE     last_name LIKE 'K%'
ORDER BY  last_name, first_name;
```

- O banco de dados recupera todas as linhas que satisfazem a condição **WHERE**, também chamada de **predicado**.

SQL

- Todas as instruções **SQL** usam o **otimizador**, um componente do banco de dados que determina os meios mais eficientes de acessar os dados solicitados.
- O **Oracle SQL** inclui muitas extensões para a linguagem SQL padrão ANSI/ISO, e as ferramentas e aplicativos do **Oracle Database** fornecem instruções adicionais.
- Uma **instrução SQL** é um programa ou instrução de computador que consiste em identificadores, parâmetros, variáveis, nomes, tipos de dados e **palavras reservadas** em SQL.

SQL

- As instruções **Oracle SQL** são divididas nas seguintes categorias:
 - Instruções DDL (Data Definition Language)
 - Instruções DML (Data Manipulation Language)
 - Declarações de Controle de Transação
 - Instruções de Controle de Sessão
 - Declaração de Controle do Sistema
 - Instruções SQL Incorporadas

SQL

- **Instruções DDL (Data Definition Language).**
- O **DDL** permite alterar os atributos de um objeto sem alterar os aplicativos que acessam o objeto.
- As instruções **DDL** permitem:
 - Criar, alterar e descartar objetos de esquema e outras estruturas de banco de dados, incluindo o próprio banco de dados e os usuários do banco de dados (**CREATE**, **ALTER** ou **DROP**).
 - Excluir todos os dados nos objetos de esquema sem remover a estrutura desses objetos (**TRUNCATE**).
 - Conceder e revogar privilégios e funções (**GRANT**, **REVOKE**).
 - Ativar e desativar as opções de auditoria (**AUDIT**, **NOAUDIT**).
 - Adicionar um comentário ao dicionário de dados (**COMMENT**).

SQL

- **Instruções DDL (Data Definition Language).**

```
CREATE TABLE plants
  ( plant_id    NUMBER PRIMARY KEY,
    common_name VARCHAR2(15) );

INSERT INTO plants VALUES (1, 'African Violet'); # DML statement

INSERT INTO plants VALUES (2, 'Amaryllis'); # DML statement

ALTER TABLE plants ADD
  ( latin_name VARCHAR2(40) );

GRANT READ ON plants TO scott;

REVOKE READ ON plants FROM scott;

DROP TABLE plants;
```

SQL

- **Instruções DDL (Data Definition Language).**
- Consultam ou manipulam dados em objetos de esquema existentes.
- As instruções **DML** permitem:
 - Recuperar ou buscar dados de uma ou mais tabelas ou visões (**SELECT**).
 - Adicionar novas linhas de dados a uma tabela ou visão (**INSERT**) especificando uma lista de valores de coluna ou usando uma subconsulta para selecionar e manipular dados existentes.
 - Alterar os valores da coluna nas linhas existentes de uma tabela ou visão (**UPDATE**).
 - Atualizar ou inserir linhas condicionalmente em uma tabela ou visualização (**MERGE**).
 - Remover linhas de tabelas ou visualizações (**DELETE**).

SQL

- **Instruções DML (Data Manipulation Language).**

```
SELECT * FROM employees;
```

```
INSERT INTO employees (employee_id, last_name, email, job_id, hire_date, salary)  
VALUES (1234, 'Mascis', 'JMASCIS', 'IT_PROG', '14-FEB-2008', 9000);
```

```
UPDATE employees SET salary=9100 WHERE employee_id=1234;
```

```
DELETE FROM employees WHERE employee_id=1234;
```

SELECT

- Uma **consulta** é uma operação que recupera dados de uma tabela ou exibição.
- **SELECT** é a única instrução SQL que você pode usar para consultar dados.
- O conjunto de dados recuperados da execução de uma instrução **SELECT** é conhecido como **conjunto de resultados**.
- A instrução **SELECT** especifica quais colunas devem ser mostradas no resultado. A **projeção** produz um subconjunto das colunas na tabela.

SELECT

- Uma **expressão** é uma combinação de um ou mais valores, operadores e funções SQL que resolvem para um valor.
- A **lista de expressões** que aparece após a palavra-chave **SELECT** e antes da cláusula **FROM** é chamada de **lista de seleção**.
- A cláusula **FROM** especifica as tabelas ou visões das quais os dados devem ser recuperados.
- A cláusula **WHERE** especifica uma condição para filtrar linhas, produzindo um subconjunto das linhas na tabela.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** INSTITUTO AOCP **Órgão:** Câmara de Teresina - PI **Prova:** INSTITUTO AOCP - 2021 - Câmara de Teresina - PI - Analista de Informática

Larissa está criando um comando SQL para seu banco de dados ORACLE 12c e precisa limitar as linhas de resultados. Como ela pode obter esse resultado?

Alternativas

- a) Usando a cláusula FETCH.
- b) Usando a cláusula TOP.
- c) Usando a cláusula LIMIT.
- d) Usando o comando DELETE.
- e) Usando a cláusula SKIP.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** INSTITUTO AOCP **Órgão:** Câmara de Teresina - PI **Prova:** INSTITUTO AOCP - 2021 - Câmara de Teresina - PI - Analista de Informática

Larissa está criando um comando SQL para seu banco de dados ORACLE 12c e precisa limitar as linhas de resultados. Como ela pode obter esse resultado?

Alternativas

- a) Usando a cláusula **FETCH**.
- b) Usando a cláusula TOP.
- c) Usando a cláusula LIMIT.
- d) Usando o comando DELETE.
- e) Usando a cláusula SKIP.

```
Oracle: select * from demo fetch first 10 rows only;  
SQL Server: select top 10 * from demo;  
MySQL: select * from demo limit 10;
```

JOINS

- Uma **junção** é uma consulta que combina linhas de duas ou mais tabelas, visualizações ou visualizações materializadas.

```
SELECT email, department_name
FROM   employees
JOIN   departments
ON     employees.department_id = departments.department_id
WHERE  employee_id IN (100,103)
ORDER BY email;
```

EMAIL	DEPARTMENT_NAME
-----	-----
AHUNOLD	IT
SKING	Executive

JOINS

- A maioria das **junções** possui pelo menos uma **condição de junção**, na cláusula **FROM** ou na cláusula **WHERE**, que compara duas colunas, cada uma de uma tabela diferente.
- O banco de dados combina pares de linhas, cada uma contendo uma linha de cada tabela, para a qual a condição de associação é avaliada **TRUE**.
- O **otimizador** determina a ordem na qual o banco de dados associa tabelas com base nas condições de **junção**, **índices** e estatísticas disponíveis para as tabelas.

JOINS

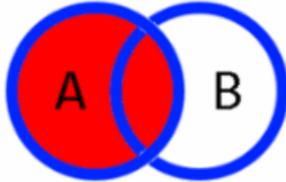
- Os **tipos de junção** incluem o seguinte:
- **INNER JOIN** ou **Junção Interna**
 - É uma junção de duas ou mais tabelas que retorna apenas linhas que satisfazem a condição de junção.
- **OUTER JOIN** ou **Junção Externa**
 - Retorna todas as linhas que atendem à condição de junção e também retorna linhas de uma tabela para as quais nenhuma linha da outra tabela satisfaz a condição.

JOINS

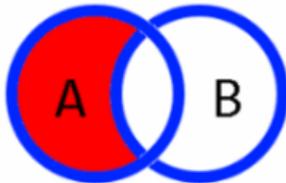
- **LEFT OUTER JOIN** ou **Junção Externa Esquerda**
 - O resultado para duas tabelas, A e B, sempre contém todos os registros da tabela esquerda A, mesmo que a condição de junção não corresponda a um registro na tabela B direita.
- **RIGHT OUTER JOIN** ou **Junção Externa Direita**
 - O resultado para duas tabelas, A e B, contém todos os registros da tabela da direita B, mesmo que a condição de junção não corresponde a uma linha na tabela esquerda A.

SQL JOINS

LEFT OUTER JOIN

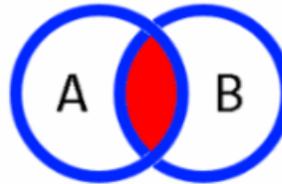


```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY
```



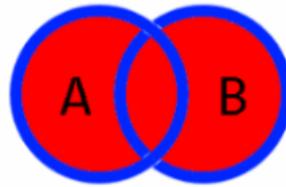
```
SELECT *  
FROM TableA a  
LEFT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE b.KEY IS NULL
```

INNER JOIN

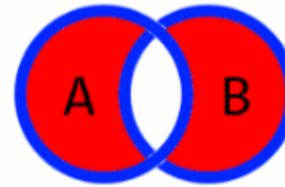


```
SELECT *  
FROM TableA a  
INNER JOIN TableB b  
ON a.KEY = b.KEY
```

FULL OUTER JOIN

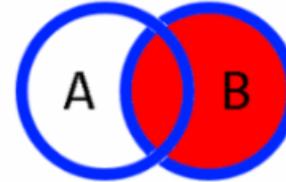


```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY
```

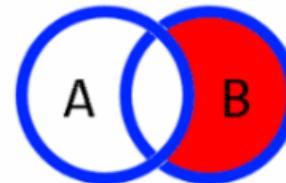


```
SELECT *  
FROM TableA a  
FULL OUTER JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL  
OR b.KEY IS NULL
```

RIGHT OUTER JOIN



```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY
```



```
SELECT *  
FROM TableA a  
RIGHT JOIN TableB b  
ON a.KEY = b.KEY  
WHERE a.KEY IS NULL
```

DIMENSÕES

- Um **Data Warehouse** típico possui dois componentes importantes: **dimensões** e **fatos**.
- Uma **dimensão** é qualquer categoria usada para especificar perguntas de negócios.
- Um **fato** é um evento ou entidade associado a um conjunto específico de valores de dimensão.
- Uma **tabela de dimensões** é uma estrutura lógica que define **relacionamentos hierárquicos** (pai/filho) entre pares de colunas ou conjuntos de colunas.

DIMENSÕES

- Você cria **dimensões** com a instrução SQL **CREATE DIMENSION**.
- As colunas em uma **dimensão** podem vir da mesma tabela (desnormalizada) ou de várias tabelas (total ou parcialmente normalizadas).
- Normalizadas ou desnormalizadas, os relacionamentos hierárquicos entre as colunas devem ser especificados na instrução **CREATE DIMENSION**.

SUBQUERY

- Uma **subconsulta** é uma instrução **SELECT** aninhada dentro de outra instrução SQL.
- Cada parte da consulta de uma instrução é chamada de **bloco de consulta**.

```
SELECT first_name, last_name
FROM employees
WHERE department_id
IN ( SELECT department_id
      FROM departments
      WHERE location_id = 1800 );
```

SUBQUERY

- A estrutura da instrução SQL não força o banco de dados a executar a consulta interna primeiro.
- O **otimizador** determina a melhor sequência de etapas para recuperar as linhas solicitadas.

INSTRUÇÕES DE CONTROLE DE TRANSAÇÃO

- As instruções de controle de transação gerenciam as alterações feitas pelas instruções **DML** e agrupam as instruções em **transações**.
- Essas instruções permitem:
 - Fazer alterações em uma transação permanente (**COMMIT**).
 - Desfazer as alterações em uma transação, desde que a transação foi iniciada (**ROLLBACK**) ou desde um ponto de salvamento (**ROLLBACK TO SAVEPOINT**).
 - Definir um ponto para o qual é possível reverter (**SAVEPOINT**).
 - Estabelecer propriedades para uma transação (**SET TRANSACTION**).
 - Especificar se uma restrição de integridade adiada é verificada após cada instrução DML ou quando a transação é confirmada (**SET CONSTRAINT**).

INSTRUÇÕES DE CONTROLE DE TRANSAÇÃO

```
SET TRANSACTION NAME 'Update salaries';
```

```
SAVEPOINT before_salary_update;
```

```
UPDATE employees SET salary=9100 WHERE employee_id=1234 # DML
```

```
ROLLBACK TO SAVEPOINT before_salary_update;
```

```
UPDATE employees SET salary=9200 WHERE employee_id=1234 # DML
```

```
COMMIT COMMENT 'Updated salaries';
```

DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** SAAE de Barretos - SP **Prova:** VUNESP - 2018 - SAAE de Barretos - SP - Assistente de Tecnologia da Informação

No sistema gerenciador de bancos de dados Oracle 12c, o comando para terminar uma transação e desfazer quaisquer alterações feitas durante tal transação é o

- a) FETCH.
- b) CLOSE.
- c) TRIGGER.
- d) ROLLBACK.
- e) INSTEAD OF.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** SAAE de Barretos - SP **Prova:** VUNESP - 2018 - SAAE de Barretos - SP - Assistente de Tecnologia da Informação

No sistema gerenciador de bancos de dados Oracle 12c, o comando para terminar uma transação e desfazer quaisquer alterações feitas durante tal transação é o

- a) FETCH.
- b) CLOSE.
- c) TRIGGER.
- d) **ROLLBACK.**
- e) INSTEAD OF.

INSTRUÇÕES DE CONTROLE DE SESSÃO

- As instruções de controle de sessão gerenciam dinamicamente as propriedades de uma **sessão do usuário**.
- Uma **sessão** é uma entidade lógica na memória da instância do banco de dados que representa o estado de um logon de usuário atual em um banco de dados.

INSTRUÇÕES DE CONTROLE DE SESSÃO

- As instruções de controle de sessão permitem:
 - Altere a sessão atual executando uma função especializada, como definir o formato de data padrão (**ALTER SESSION**).
 - Habilite e desabilite as funções, que são grupos de privilégios, para a sessão atual (**SET ROLE**).

```
ALTER SESSION  
  SET NLS_DATE_FORMAT = 'YYYY MM DD HH24:MI:SS';
```

INSTRUÇÕES DE CONTROLE DE SISTEMA

- Uma instrução de controle do sistema altera as propriedades da **instância do banco de dados**.
- A única declaração de controle do sistema é **ALTER SYSTEM**.

```
ALTER SYSTEM SWITCH LOGFILE;
```

```
ALTER SYSTEM KILL SESSION '39, 23';
```

- A instrução **ALTER SYSTEM** não confirma implicitamente a transação atual.

INSTRUÇÕES SQL INCORPORADAS

- **Instruções SQL Incorporadas.**
- As instruções SQL incorporadas incorporam DDL, DML e instruções de controle de transação em um **programa de linguagem procedural**.
- As instruções incorporadas são usadas com os pré-compiladores Oracle.
- O SQL incorporado é uma abordagem para incorporar o SQL em seus aplicativos de linguagem procedural.

INSTRUÇÕES SQL INCORPORADAS

- As instruções SQL incorporadas permitem:
 - Definir, alocar e liberar um cursor (**DECLARE CURSOR, OPEN, CLOSE**).
 - Especificar um banco de dados e conectar a ele (**DECLARE DATABASE, CONNECT**).
 - Atribuir nomes de variáveis (**DECLARE STATEMENT**).
 - Inicializar os descritores (**DESCRIBE**).
 - Especificar como as condições de erro e aviso são tratadas (**WHENEVER**).
 - Analisar e executar instruções SQL (**PREPARE, EXECUTE, EXECUTE IMMEDIATE**).
 - Recuperar dados do banco de dados (**FETCH**).

OPTIMIZER

- O **otimizador** gera planos de execução descrevendo possíveis métodos de execução.
- O **otimizador** determina qual plano de execução é mais eficiente considerando várias fontes de informação.
- O **otimizador** gera a maioria das formas possíveis de processamento de uma consulta e atribui um custo a cada etapa do plano de execução gerado.
- O plano com o menor custo é escolhido como o **plano de consulta** a ser executado.

OPTIMIZER

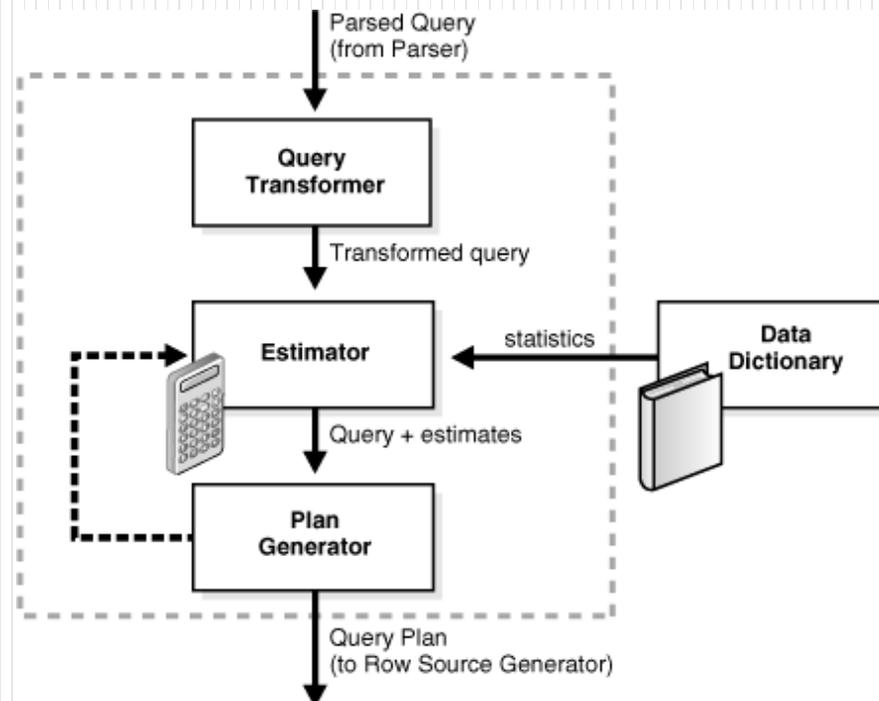
- Para determinar o melhor plano de execução para uma instrução SQL, o **otimizador** executa as seguintes operações:
 - Avaliação de expressões e condições.
 - Inspeção de restrições de integridade para aprender mais sobre os dados e otimizar com base nesses metadados.
 - Transformação de declaração.
 - Escolha das metas do otimizador.
 - Escolha dos caminhos de acesso.
 - Escolha de pedidos de junção.

OPTIMIZER

- É possível influenciar as opções do **otimizador** definindo a meta do **otimizador** e reunindo estatísticas representativas para o **otimizador**.
- É possível definir a meta do otimizador para um dos seguintes:
 - **Rendimento total:** A dica **ALL_ROWS** instrui o **otimizador** a obter a última linha do resultado para o aplicativo cliente o mais rápido possível.
 - **Tempo de resposta inicial:** A dica **FIRST_ROWS** instrui o **otimizador** a obter a primeira linha para o cliente o mais rápido possível.

OPTIMIZER

- O otimizador contém três componentes principais: transformador, estimador e gerador de plano.



OPTIMIZER

- O **transformador** de consulta determina se é útil alterar a forma da consulta para que o otimizador possa gerar um melhor plano de execução.
- O **estimador** determina o custo total de um determinado plano de execução.
- O **gerador de plano** tenta planos diferentes para uma consulta enviada. O otimizador escolhe o plano com o menor custo.

OPTIMIZER

- O **otimizador** executa as seguintes operações:
 - Recebe a consulta analisada e gera um conjunto de possíveis planos para a instrução SQL com base nos caminhos e dicas de acesso disponíveis.
 - Estima o custo de cada plano com base nas estatísticas no dicionário de dados.
 - Compara os custos dos planos e escolhe o plano de menor custo, conhecido como plano de consulta, para passar ao gerador de origem de linhas.

OPTIMIZER

- Um caminho de acesso é a técnica que uma consulta usa para recuperar linhas.
- O banco de dados pode usar vários caminhos de acesso diferentes para recuperar dados de uma tabela.
- **Verificações de tabela completa:** lê todas as linhas de uma tabela e filtra aquelas que não atendem aos critérios de seleção.
- **Digitalizações Rowid:** o rowid de uma linha especifica o arquivo de dados e o bloco de dados que contém a linha e o local da linha nesse bloco.

OPTIMIZER

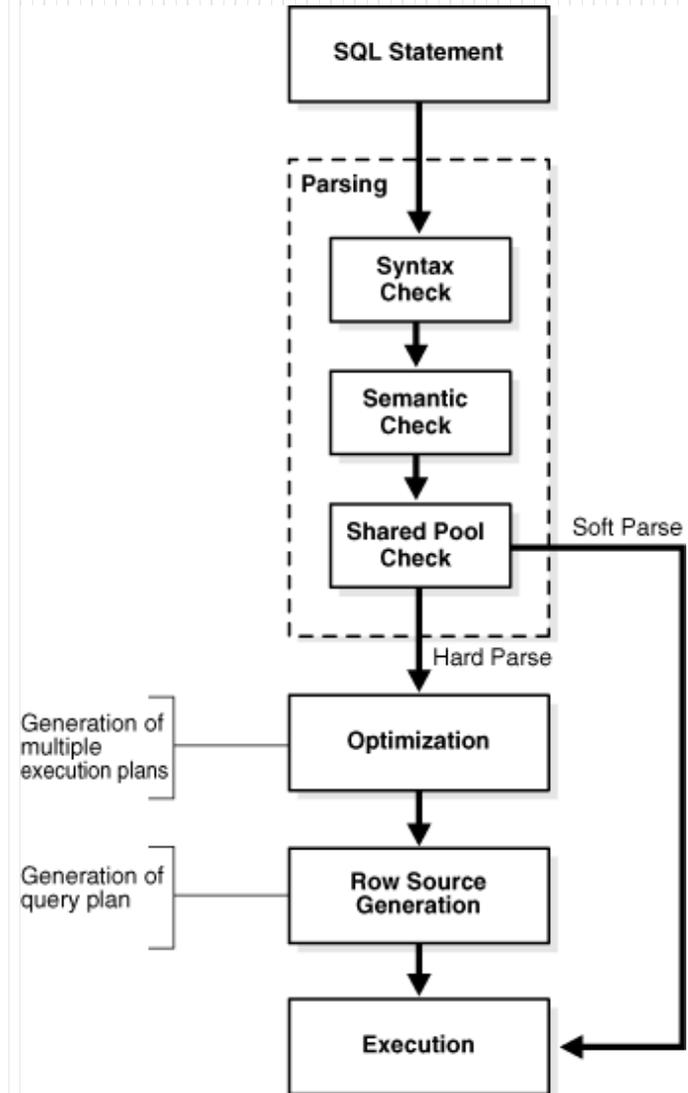
- **Varreduras de índice:** pesquisa em um índice os valores da coluna indexada acessados pela instrução SQL.
- **Verificações de cluster:** recupera dados de uma tabela armazenada em um cluster de tabelas indexadas
- **Varreduras de hash:** localiza linhas em um cluster de hash, onde todas as linhas com o mesmo valor de hash são armazenadas no mesmo bloco de dados

OPTIMIZER

- As **estatísticas do otimizador** são uma **coleção de dados que descrevem detalhes** sobre o banco de dados e os objetos no banco de dados.
- As **estatísticas do otimizador** incluem o seguinte:
 - **Estatísticas da tabela:** inclui o número de linhas, o número de blocos e o comprimento médio das linhas.
 - **Estatísticas da coluna:** inclui o número de valores e nulos distintos em uma coluna e a distribuição de dados.
 - **Estatísticas do índice:** inclui o número de blocos foliares e os níveis de índice.
 - **Estatísticas do sistema:** inclui desempenho e utilização de CPU e E/S.

SQL

- **Processamento SQL.**
- Os estágios gerais do processamento SQL são **análise, otimização, geração de origem de linha e execução.**



SQL

- **Análise SQL.**
- Envolve a **separação das partes de uma instrução SQL** em uma estrutura de dados que pode ser processada por outras rotinas.
- Quando uma instrução SQL é emitida, uma **chamada de análise** ao banco de dados para preparar a instrução para execução é feita.
- A chamada de análise abre ou cria um **cursor**, que é um identificador para a **área SQL privada** específica da sessão que contém uma instrução SQL analisada e outras informações de processamento.

SQL

- **Análise SQL.**
- O cursor e a área SQL privada estão no **PGA**.
- Durante a **chamada de análise**, o banco de dados executa as seguintes verificações:
 - Verificação de sintaxe.
 - Verificação semântica.
 - Verificação de pool compartilhado.

SQL

- **Otimização SQL.**
- A **otimização de consulta** é o processo de escolha dos meios mais eficientes de executar uma instrução SQL.
- O **otimizador** usa o número de linhas, o tamanho do conjunto de dados e outros fatores para gerar possíveis planos de execução.
- O banco de dados usa o **plano com o menor custo**.
- O **DDL** nunca é otimizado, a menos que inclua um componente **DML**, como uma subconsulta que requer otimização.

SQL

- **Geração de origem de linha SQL.**
- O **gerador de origem de linha** é um software que recebe o plano de execução ideal do otimizador e produz um plano iterativo, chamado de **plano de consulta**.
- O **plano de consulta** assume a forma de uma **combinação de etapas**.
- Cada etapa retorna um **conjunto de linhas**.
- Uma **origem de linha** é um conjunto de linhas retornado por uma etapa do plano de execução junto com uma estrutura de controle que pode processar iterativamente as linhas.

SQL

- **Execução SQL.**
- Cada origem de linha na árvore produzida pelo gerador de origem de linha é executada.
- Durante a execução, se os dados não estiverem na memória, **o banco de dados lê os dados do disco na memória.**
- O banco de dados **remove todos os bloqueios e travas necessárias para garantir a integridade dos dados** e registra as alterações feitas durante a execução do SQL.
- O estágio final do processamento de uma instrução SQL é fechar o cursor.

BACKUP

- O Recovery Manager (RMAN) é um cliente Oracle Database que executa tarefas de backup e recuperação em seus bancos de dados e automatiza a administração de suas estratégias de backup. Ele simplifica muito o backup, a restauração e a recuperação de arquivos de banco de dados.
- O ambiente RMAN consiste nos utilitários e bancos de dados que desempenham um papel no backup de seus dados. No mínimo, o ambiente do RMAN deve incluir os seguintes componentes:

BACKUP

- **Um banco de dados de destino**

Um banco de dados Oracle ao qual o RMAN está conectado com a palavra-chave TARGET. Um banco de dados de destino é um banco de dados no qual o RMAN está executando operações de backup e recuperação. O RMAN sempre mantém metadados sobre suas operações em um banco de dados no arquivo de controle do banco de dados. Os metadados do RMAN são conhecidos como repositório RMAN.

- **O cliente RMAN**

Um executável do Oracle Database que interpreta comandos, direciona as sessões do servidor para executar esses comandos e registra sua atividade no arquivo de controle do banco de dados de destino. O executável RMAN é instalado automaticamente com o banco de dados e normalmente está localizado no mesmo diretório que os outros executáveis do banco de dados. Por exemplo, o cliente RMAN no Linux está localizado em `$ORACLE_HOME/bin`

BACKUP

- Alguns ambientes usam os seguintes componentes opcionais:

- **Uma área de recuperação rápida**

Um local de disco no qual o banco de dados pode armazenar e gerenciar arquivos relacionados a backup e recuperação. Você define o local e o tamanho da área de recuperação rápida com os parâmetros de inicialização `DB_RECOVERY_FILE_DEST` e `DB_RECOVERY_FILE_DEST_SIZE`.

- **Um gerenciador de mídia**

Um aplicativo necessário para que o RMAN interaja com dispositivos de mídia sequencial, como bibliotecas de fitas. Um gerenciador de mídia controla esses dispositivos durante o backup e a recuperação, gerenciando o carregamento, rotulagem e descarregamento de mídia. Os dispositivos de gerenciamento de mídia às vezes são chamados de dispositivos SBT (backup do sistema em fita).

BACKUP

- **Um catálogo de recuperação**

Um esquema de banco de dados separado usado para registrar a atividade do RMAN em um ou mais bancos de dados de destino. Um catálogo de recuperação preserva os metadados do repositório RMAN se o arquivo de controle for perdido, tornando muito mais fácil restaurar e recuperar após a perda do arquivo de controle.

O banco de dados pode substituir os registros mais antigos no arquivo de controle, mas o RMAN mantém os registros para sempre no catálogo, a menos que os registros sejam excluídos pelo usuário.

BACKUP

- Para fazer um backup de banco de dados consistente:
- Inicie o RMAN e conecte-se a um banco de dados de destino.
- Encerre o banco de dados de forma consistente e, em seguida, monte-o.
- Por exemplo, digite os comandos a seguir para garantir que o banco de dados esteja em um estado consistente para um backup:
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP FORCE DBA;
RMAN> SHUTDOWN IMMEDIATE;
RMAN> STARTUP MOUNT;
- Execute o comando `BACKUP DATABASE`.
- Por exemplo, digite o seguinte comando no prompt do RMAN para fazer backup do banco de dados no dispositivo de backup padrão:
RMAN> BACKUP BANCO DE DADOS;
- A seguinte variação do comando cria backups de cópia de imagem de todos os arquivos de dados no banco de dados:
RMAN> BACKUP AS COPY DATABASE;
- Abra o banco de dados e retome as operações normais.
- O comando a seguir abre o banco de dados:
RMAN> ALTER DATABASE OPEN;

DIRETO DO CONCURSO

Ano: 2021 **Banca:** FGV **Órgão:** Banestes **Prova:** FGV - 2021 - Banestes - Analista em Tecnologia da Informação - Segurança da Informação

Num ambiente Oracle, considere o acionamento de um utilitário por meio do prompt de comando, como exibido a seguir.

```
% rman
```

```
RMAN> CONNECT TARGET SYS@prod
```

É correto afirmar que o comando disparado refere-se ao(s):

Alternativas

- a) acesso aos manuais de utilização do Oracle;
- b) monitoramento da performance em tempo real;
- c) critérios para criação de índices para bancos de dados;
- d) mecanismos para criação de backups e recuperação de bancos de dados;
- e) procedimentos e facilidades para uso/acesso remoto.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** FGV **Órgão:** Banestes **Prova:** FGV - 2021 - Banestes - Analista em Tecnologia da Informação - Segurança da Informação

Num ambiente Oracle, considere o acionamento de um utilitário por meio do prompt de comando, como exibido a seguir.

```
% rman
```

```
RMAN> CONNECT TARGET SYS@prod
```

É correto afirmar que o comando disparado refere-se ao(s):

Alternativas

- a) acesso aos manuais de utilização do Oracle;
- b) monitoramento da performance em tempo real;
- c) critérios para criação de índices para bancos de dados;
- d) mecanismos para criação de backups e recuperação de bancos de dados;**
- e) procedimentos e facilidades para uso/acesso remoto.

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:** Quadrix - 2021 - CFT - Analista de Tecnologia da Informação Júnior

Uma ferramenta que pode ser usada para se realizar o backup de um banco de dados Oracle é o RMAN.

Certo

Errado

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:** Quadrix - 2021 - CFT - Analista de Tecnologia da Informação Júnior

Uma ferramenta que pode ser usada para se realizar o backup de um banco de dados Oracle é o RMAN.

Certo

Errado

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:** Quadrix - 2021 - CFT - Analista de Tecnologia da Informação Júnior

Julgue o item, relativos aos elementos de interconexão de redes e às ferramentas de backup do banco de dados Oracle.

Para realizar o backup de um banco de dados Oracle, independentemente de qual ferramenta esteja usando, o administrador de banco de dados (DBA) deverá desligar o banco de dados com o comando shutdown immediate antes de iniciar o backup.

Certo

Errado

DIRETO DO CONCURSO

Ano: 2021 **Banca:** Quadrix **Órgão:** CFT **Prova:** Quadrix - 2021 - CFT - Analista de Tecnologia da Informação Júnior

Julgue o item, relativos aos elementos de interconexão de redes e às ferramentas de backup do banco de dados Oracle.

Para realizar o backup de um banco de dados Oracle, independentemente de qual ferramenta esteja usando, o administrador de banco de dados (DBA) deverá desligar o banco de dados com o comando shutdown immediate antes de iniciar o backup.

Certo

Errado

TRIGGERS

- O uso correto de **gatilhos** permite criar e implantar aplicativos mais robustos e que usam o banco de dados com mais eficiência.
- **Gatilhos** podem ser usados para:
 - Gerar automaticamente valores de coluna derivados.
 - Impedir transações inválidas.
 - Fornecer auditoria e log de eventos.
 - Registrar informações sobre o acesso à tabela.
- É possível usar **gatilhos** para impor regras de negócios de baixo nível comuns a todos os aplicativos clientes.

TRIGGERS

- **Gatilhos** e as **Restrições de Integridade** para definir e aplicar qualquer tipo de regra de integridade.
- A **Oracle** recomenda enfaticamente que você **gatilhos** sejam usados para impor regras de negócios complexas não definíveis usando uma restrição de integridade.
- O uso excessivo de **gatilhos** pode resultar em interdependências complexas.

TRIGGERS

- O **Oracle Database** suporta os seguintes **tipos de gatilhos**:
- **ROW TRIGGER** ou **Gatilho de Linha**: é acionado toda vez que a tabela é afetada pela instrução de acionamento.
- **STATEMENT TRIGGER** ou **Gatilho de Instrução**: é acionado uma vez em nome da instrução de acionamento, independentemente do número de linhas afetadas pela instrução de acionamento.
- **INSTEAD OF TRIGGER** ou **Gatilho “Em Vez De”**: é acionado pelo banco de dados Oracle em vez de executar a instrução de acionamento.

TRIGGERS

- **EVENT TRIGGER** ou **Gatilho de Evento**: pode ser usado para publicar informações sobre eventos do banco de dados.
- Um **gatilho de eventos do sistema (SYSTEM EVENT TRIGGER)** pode ser causado por eventos como inicialização e encerramento da instância de banco de dados ou mensagens de erro.
- Um **gatilho de eventos do usuário (USER EVENT TRIGGER)** é acionado devido a eventos relacionados ao logon e logoff do usuário, instruções DDL e instruções DML.

TRIGGERS

- Um **gatilho simples** é um gatilho único em uma tabela que permite especificar ações para exatamente um dos seguintes pontos de tempo:
 - Antes da declaração de demissão.
 - Antes de cada linha afetada pela instrução de disparo.
 - Após cada linha afetada pela instrução de disparo.
 - Após a declaração de demissão.
- Um **gatilho composto** pode disparar em vários pontos de tempo.

TRIGGERS

- A instrução **CREATE TRIGGER** cria ou substitui um gatilho de banco de dados.

```
CREATE TRIGGER trigger_name
  triggering_statement
  [trigger_restriction]
BEGIN
  triggered_action;
END;
```

- Um gatilho **PL/SQL** possui os seguintes componentes básicos: **nome do acionador**; **o evento ou instrução de gatilho**; **restrição de gatilho**; **ação acionada**.

TRIGGERS

- Este exemplo cria um **gatilho** que dispara quando uma instrução INSERT, UPDATE ou DELETE é executada sobre uma tabela de itens de linha.

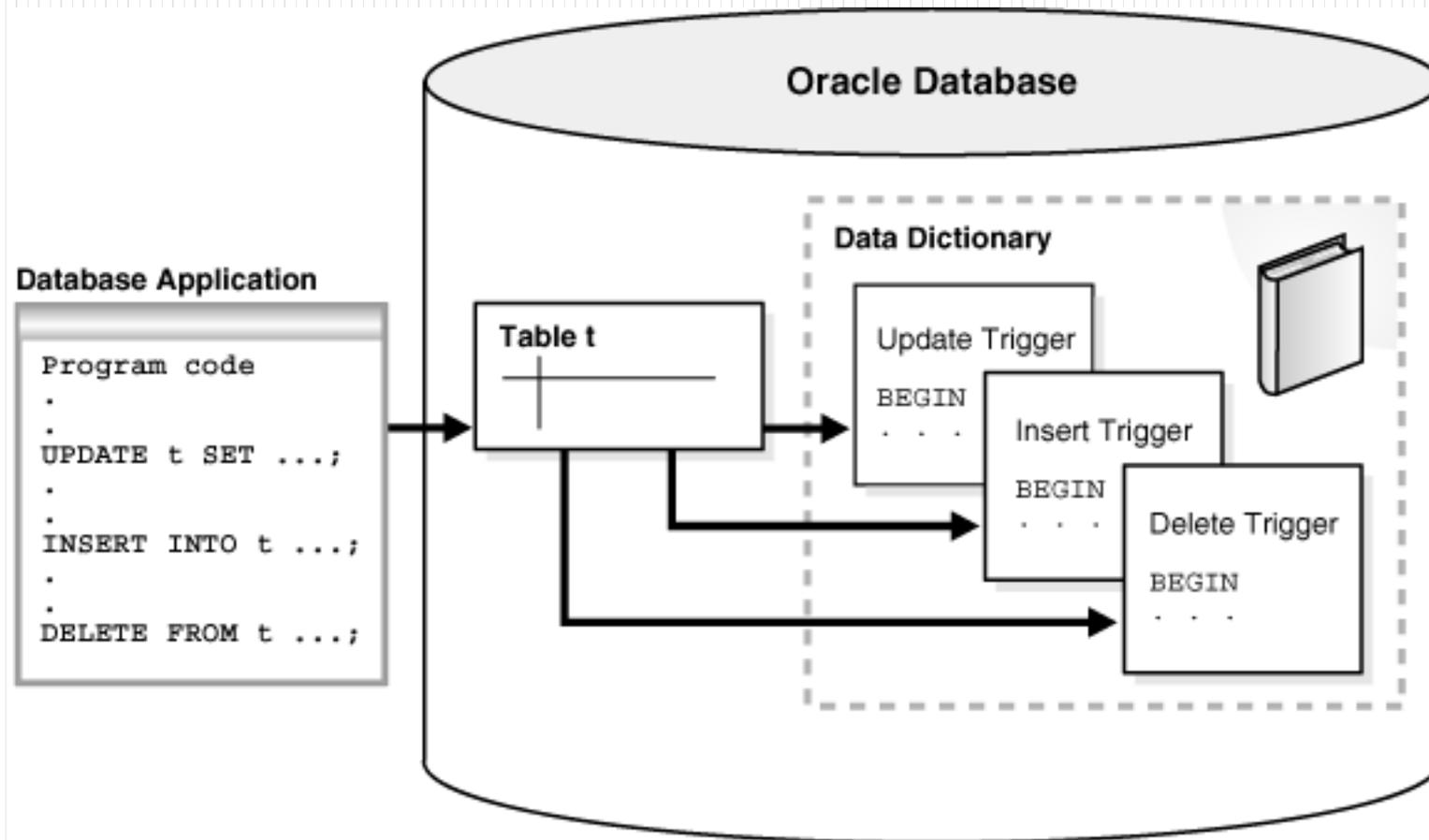
```
CREATE TABLE orders
( order_id NUMBER PRIMARY KEY,
  /* other attributes */
  line_items_count NUMBER DEFAULT 0 );
```

```
CREATE TABLE lineitems
( order_id REFERENCES orders,
  seq_no    NUMBER,
  /* other attributes */
  CONSTRAINT lineitems PRIMARY KEY(order_id,seq_no) );
```

TRIGGERS

- O **Oracle Database** executa um **gatilho** internamente usando as mesmas etapas da execução do subprograma.
- O banco de dados Oracle armazena **gatilhos PL/SQL** em forma compilada em um esquema de banco de dados, assim como procedimentos armazenados **PL/SQL**.
- Quando uma instrução **CREATE TRIGGER** é confirmada, o código **PL/SQL** compilado é armazenado no banco de dados. O pool compartilhado remove o código-fonte do gatilho **PL/SQL**.

TRIGGERS



DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** Prefeitura de Registro - SP
Prova: VUNESP - 2018 - Prefeitura de Registro - SP - Técnico em Informática

O sistema gerenciador de bancos de dados Oracle 12C possui, dentre seus tipos de *triggers* (gatilhos), além de BEFORE e AFTER, o tipo:

- a) VARCHAR
- b) INSTEAD OF
- c) NULL
- d) LOCK
- e) EXIT

DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** Prefeitura de Registro - SP
Prova: VUNESP - 2018 - Prefeitura de Registro - SP - Técnico em Informática

O sistema gerenciador de bancos de dados Oracle 12C possui, dentre seus tipos de *triggers* (gatilhos), além de BEFORE e AFTER, o tipo:

- a) VARCHAR
- b) **INSTEAD OF**
- c) NULL
- d) LOCK
- e) EXIT

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** SEFAZ-SC **Prova:** FCC - 2018 - SEFAZ-SC - Auditor-Fiscal da Receita Estadual - Tecnologia da Informação (Prova 3)

Em um Oracle Database, um profissional de TI deseja criar uma trigger chamada `Atualiza_salario_func`, com o objetivo de exibir as diferenças entre o novo (`NEW.salarioFunc`) e o antigo salário (`OLD.salarioFunc`) constantes da tabela de funcionários. Para tanto, ele escreveu o seguinte código:

```
CREATE TRIGGER Atualiza_salario_func
  BEFORE I ON II
  FOR EACH ROW
  WHEN (NEW.CODFUNC > 0)
  DECLARE
    diferenca_salario number;
  BEGIN
    diferenca_salario := :NEW.salarioFunc - :OLD.salarioFunc;
    dbms_output.put('Salário antigo:' || :OLD.salarioFunc);
    dbms_output.put('Salário novo:' || :NEW.salarioFunc);
    dbms_output.put_line(' A diferença de salário foi de: ' || diferenca_salario);
  END;
/
```

As lacunas **I** e **II** devem especificar, correta e respectivamente, os elementos genéricos

- a) condition e database_name.
- b) triggering_event e column_name.
- c) column_event e table_name.
- d) row_event e column_name.
- e) triggering_event e table_name.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** SEFAZ-SC **Prova:** FCC - 2018 - SEFAZ-SC - Auditor-Fiscal da Receita Estadual - Tecnologia da Informação (Prova 3)

Em um Oracle Database, um profissional de TI deseja criar uma trigger chamada `Atualiza_salario_func`, com o objetivo de exibir as diferenças entre o novo (`NEW.salarioFunc`) e o antigo salário (`OLD.salarioFunc`) constantes da tabela de funcionários. Para tanto, ele escreveu o seguinte código:

As lacunas **I** e **II** devem especificar, correta e respectivamente, os elementos genéricos

- a) `condition` e `database_name`.
- b) `triggering_event` e `column_name`.
- c) `column_event` e `table_name`.
- d) `row_event` e `column_name`.
- e) **`triggering_event` e `table_name`.**

Justificativa:

Exemplo:

```
CREATE OR REPLACE TRIGGER [NOME DA TRIGGER]
BEFORE INSERT OR DELETE OR UPDATE
ON [NOME DA TABELA]
```

DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** PauliPrev - SP **Prova:** VUNESP - 2018 - PauliPrev - SP - Técnico de Tecnologia da Informação

Considere os seguintes comandos do sistema gerenciador de bancos de dados Oracle 12c: CEIL (27.7) e FLOOR (12.7)

Os resultados da execução de ambos os comandos são, respectivamente:

- a) 27 e 12.7
- b) 27.7 e 12.7
- c) 27.7 e 13
- d) 28 e 12
- e) 28 e 13

DIRETO DO CONCURSO

Ano: 2018 **Banca:** VUNESP **Órgão:** PauliPrev - SP **Prova:** VUNESP - 2018 - PauliPrev - SP - Técnico de Tecnologia da Informação

Considere os seguintes comandos do sistema gerenciador de bancos de dados Oracle 12c: CEIL (27.7) e FLOOR (12.7)

Os resultados da execução de ambos os comandos são, respectivamente:

- a) 27 e 12.7
- b) 27.7 e 12.7
- c) 27.7 e 13
- d) **28 e 12**
- e) 28 e 13

DIRETO DO CONCUSO

Ano: 2018 **Banca:** FCC **Órgão:** SEFAZ-SC **Prova:** FCC - 2018 - SEFAZ-SC - Auditor-Fiscal da Receita Estadual - Tecnologia da Informação (Prova 3)

Um profissional de TI deseja utilizar algumas funções do Oracle Database 12c para:

- I. Remover todos os caracteres especificados do começo ou do final de uma *string*.
- II. Obter a localização de uma *substring* em uma *string*.

Para tanto, ele deve usar, respectivamente, as funções

- a) Lpad e Initcap.
- b) Trim e Instr.
- c) Chr e Instr.
- d) Chr e Initcap.
- e) Lpad e Instr.

DIRETO DO CONCUSO

Ano: 2018 **Banca:** FCC **Órgão:** SEFAZ-SC **Prova:** FCC - 2018 - SEFAZ-SC - Auditor-Fiscal da Receita Estadual - Tecnologia da Informação (Prova 3)

Um profissional de TI deseja utilizar algumas funções do Oracle Database 12c para:

- I. Remover todos os caracteres especificados do começo ou do final de uma *string*.
- II. Obter a localização de uma *substring* em uma *string*.

Para tanto, ele deve usar, respectivamente, as funções

- a) Lpad e Initcap.
- b) **Trim e Instr.**
- c) Chr e Instr.
- d) Chr e Initcap.
- e) Lpad e Instr.

Justificativa:

Trim - Remove caracteres especificados à esquerda e à direita de uma string de caracteres.

INSTR (INSTR, INSTRB, INSTRC, INSTR2 e INSTR4) - pesquisam uma sequência de caracteres por uma substring usando caracteres e retornam a posição na string que é o primeiro caractere de uma ocorrência especificada da substring.

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** MPE-PE **Prova:** FCC - 2018 - MPE-PE - Analista Ministerial - Informática

O sistema gerenciador de bancos de dados Oracle 12c possui a função GREATEST, inserida no seguinte comando do PLSQL:

```
SELECT GREATEST ('Olavo', 'Pedro', 'Silvio', 'Armando');
```

Tal comando tem como resultado:

- a) Armando
Armando
Silvio
- b) Pedro
Olavo
- c) NULL
Armando
Olavo
- d) Pedro
Silvio
- e) Silvio

DIRETO DO CONCURSO

Ano: 2018 **Banca:** FCC **Órgão:** MPE-PE **Prova:** FCC - 2018 - MPE-PE - Analista Ministerial - Informática

O sistema gerenciador de bancos de dados Oracle 12c possui a função GREATEST, inserida no seguinte comando do PLSQL:

```
SELECT GREATEST ('Olavo', 'Pedro', 'Silvio', 'Armando');
```

Tal comando tem como resultado:

- a) Armando
Armando
Silvio
- b) Pedro
Olavo
- c) NULL
Armando
Olavo
- d) Pedro
Silvio
- e) **Silvio**

Justificativa:

A função GREATEST retorna a maior expressão em uma lista de expressões. Todas as expressões após a primeira são implicitamente convertidas no tipo de dados da primeira expressão antes da comparação. Para recuperar a menor expressão em uma lista de expressões, use LEAST.

Exemplo: A instrução a seguir seleciona a sequência que é a última na sequência alfabética.

```
SHOW GREATEST ('Harry', 'Harriot', 'Harold')
```

```
RESULTADO - > Harry
```

```
SHOW GREATEST (5, 3, 18)
```

```
RESULTADO - > 18
```

Funções de Agregação

Funções de agregação executam operações sobre diversos valores para criar resultados de resumo.

Função	Exemplo	Descrição
Avg	Avg(Sales)	Calcula a média de um conjunto numérico de valores.
Bin	Bin(UnitPrice BY ProductName)	Seleciona qualquer atributo numérico de uma dimensão, tabela de fatos ou medida contendo valores de dados e os coloca em um número específico de bins. Essa função é tratada como um novo atributo de dimensão para propósitos como agregação, filtragem e detalhamento.
Count	Count(Products)	Determina o número de itens com um valor não nulo.
First	First(Sales)	Seleciona o primeiro valor retornado não nulo do argumento de expressão. A função First opera no nível mais detalhado especificado em sua dimensão definida explicitamente.
Last	Last(Sales)	Seleciona o último valor retornado não nulo da expressão.
Max	Max(Revenue)	Calcula o valor máximo (valor numérico mais alto) das linhas que satisfazem ao argumento de expressão numérica.
Median	Median(Sales)	Calcula o valor mediano (médio) das linhas que satisfazem ao argumento de expressão numérica. Quando houver um número par de linhas, o mediano é o meio de duas linhas. Esta função sempre retorna o dobro.
Min	Min(Revenue)	Calcula o valor mínimo (valor numérico mais baixo) das linhas que satisfazem ao argumento de expressão numérica.
StdDev	StdDev(Sales) StdDev(DISTINCT Sales)	Retorna o desvio padrão de um conjunto de valores. O tipo de retorno é sempre duplo.
StdDev_Pop	StdDev_Pop(Sales) StdDev_Pop(DISTINCT Sales)	Retorna o desvio padrão de um conjunto de valores que usa a fórmula computacional para variação de população e desvio padrão.
Sum	Sum(Revenue)	Calcula a soma obtida com a soma de todos os valores que satisfazem ao argumento de expressão numérica.

Funções de Strings

As funções de string executam diversas manipulações de caractere. Elas operam em strings de caracteres.

Função	Exemplo	Descrição
Ascii	Ascii('a')	Converte uma string de caractere único em seu código ASCII correspondente, entre 0 e 255. Se a expressão do caractere avaliar vários caracteres, é retornado o código ASCII correspondente ao primeiro caractere na expressão.
Bit_Length	Bit_Length('abcdef')	Retorna o tamanho, em bits, de uma string especificada. Cada caractere Unicode tem 2 bytes de comprimento (igual a 16 bits).
Char	Char(35)	Converte um valor numérico entre 0 e 255 no valor de caractere correspondente ao código ASCII.
Char_Length	Char_Length(Customer_Name)	Retorna o tamanho, em número de caracteres, de uma string especificada. Espaços em branco iniciais e finais não são contados no tamanho da string.
Concat	SELECT DISTINCT Concat ('abc', 'def') FROM employee	Concatena duas strings de caracteres.
Insert	SELECT Insert('123456', 2, 3, 'abcd') FROM table	Insere uma string de caracteres especificada, em um local especificado de outra string de caracteres.
Left	SELECT Left('123456', 3) FROM table	Retorna um número especificado de caracteres a partir do lado esquerdo de uma string.
Tamanho	Length(Customer_Name)	Retorna o tamanho, em número de caracteres, de uma string especificada. O comprimento informado não leva em consideração os caracteres em branco à esquerda.
Locate	Locate('d' 'abcdef')	Retorna a posição numérica de uma string de caracteres em outra string de caracteres. Se a string de caracteres não for encontrada na string que está sendo pesquisada, a função retornará o valor 0.
LocateN	Locate('d' 'abcdef', 3)	Como Locate, retorna a posição numérica de uma string de caracteres em outra string de caracteres. LocateN inclui um argumento de número inteiro que permite a você especificar uma posição inicial para começar a pesquisa.

Função	Exemplo	Descrição
Lower	Lower(Customer_Name)	Converte uma string de caracteres em letras minúsculas.
Octet_Length	Octet_Length('abcdef')	Retorna o número de bytes de uma string especificada.
Position	Position('d', 'abcdef')	Retorna a posição numérica de <i>strExpr1</i> em uma expressão de caractere. Se <i>strExpr1</i> não for encontrado, a função retornará 0.
Repeat	Repeat('abc', 4)	Repete uma expressão especificada <i>n</i> vezes.
Replace	Replace('abcd1234', '123', 'zz')	Substitui um ou mais caracteres de uma expressão de caractere especificada por um ou mais caracteres.
Right	SELECT Right('123456', 3) FROM table	Retorna um número especificado de caracteres da direita de uma string.
Space	Space(2)	Insere espaços em branco.
Substring	Substring('abcdef' FROM 2)	Cria uma nova string começando por um número fixo de caracteres na string original.
SubstringN	Substring('abcdef' FROM 2 FOR 3)	Como Substring, cria uma nova string começando por um número fixo de caracteres na string original. SubstringN inclui um argumento de número inteiro que permite a você especificar o tamanho da nova string, em número de caracteres.
TrimBoth	Trim(BOTH '_' FROM '_abcdef_')	Remove caracteres especificados à esquerda e à direita de uma string de caracteres.
TrimLeading	Trim(LEADING '_' FROM '_abcdef')	Remove caracteres especificados à esquerda de uma string de caracteres.
TrimTrailing	Trim(TRAILING '_' FROM 'abcdef_')	Remove caracteres especificados à direita de uma string de caracteres.
Upper	Upper(Customer_Name)	Converte uma string de caracteres em letras maiúsculas.

Funções de Conversão

Funções de conversão convertem um valor de uma forma para outra.

Função	Exemplo	Descrição
Cast	Cast(hiredate AS CHAR(40)) FROM employee	Altera o tipo de dados de uma expressão ou um literal nulo para outro tipo de dados. Por exemplo, você pode converter um nome_de_cliente (um tipo de dados Char ou Varchar) ou uma data de nascimento (um literal de data/horário). Use Cast para alterar para um tipo de dados de Data. Não use ToDate.
IfNull	IfNull(Sales, 0)	Testa se uma expressão é avaliada como valor nulo; se for, designará o valor especificado à expressão.
IndexCol	SELECT IndexCol(VALUEOF (NQ_SESSION.GEOGRAPHY_LEVEL), Country, State, City), Revenue FROM Sales	Usa informações externas para retornar a coluna apropriada para que o usuário com sign-in efetuado veja.
NullIf	SELECT e.last_name, NULLIF(e.job_id, j.job_id) "Old Job ID" FROM employees e, job_history j WHERE e.employee_id = j.employee_id ORDER BY last_name, "Old Job ID";	Compara duas expressões. Se forem iguais, a função retornará nulo. Se forem diferentes, a função retornará a primeira expressão. Você não pode especificar o literal NULL para a primeira expressão.
To_DateTime	SELECT To_DateTime ('2009-03-0301:01:00', 'yyyy-mm-dd hh:mi:ss') FROM sales	Converte strings literais do formato dateTime em um tipo de dados DateTime.

Funções Matemáticas

As funções matemáticas descritas nesta seção executam operações matemáticas.

Função	Exemplo	Descrição
Abs	Abs(Profit)	Calcula o valor absoluto de uma expressão numérica.
Acos	Acos(1)	Calcula o cosseno inverso de uma expressão numérica.
Asin	Asin(1)	Calcula o seno inverso de uma expressão numérica.
Atan	Atan(1)	Calcula a tangente inversa de uma expressão numérica.
Atan2	Atan2(1, 2)	Calcula o arco tangente de y/x , onde y é a primeira expressão numérica e x é a segunda.
Ceiling	Ceiling(Profit)	Arredonda uma expressão numérica não inteira para o próximo número inteiro mais alto. Se a expressão numérica resultar em um número inteiro, a função CEILING retornará esse número.
Cos	Cos(1)	Calcula o cosseno de uma expressão numérica.
Cot	Cot(1)	Calcula a cotangente de uma expressão numérica.
Degrees	Degrees(1)	Converte uma expressão de radianos em graus.
Exp	Exp(4)	Envia o valor para a potência especificada. Calcula e elevado à enésima potência, em que e é a base do logaritmo natural.
ExtractBit	Int ExtractBit(1, 5)	Recupera um bit de uma determinada posição em um número inteiro. Ela retorna um inteiro de 0 ou 1 correspondente à posição do bit.
Floor	Floor(Profit)	Arredonda uma expressão numérica não inteira para o próximo número inteiro mais baixo. Se a expressão numérica resultar em um número inteiro, a função FLOOR retornará esse número.
Log	Log(1)	Calcula o logaritmo natural de uma expressão.
Log10	Log10(1)	Calcula o logaritmo de base 10 de uma expressão.
Mod	Mod(10, 3)	Divide a primeira expressão numérica pela segunda e retorna a parte restante do quociente.

Função	Exemplo	Descrição
Pi	Pi()	Retorna o valor constante de pi.
Power	Power(Profit, 2)	Utiliza a primeira expressão numérica e a eleva à potência especificada na segunda expressão numérica.
Radians	Radians(30)	Converte uma expressão de graus em radianos.
Rand	Rand()	Retorna um número pseudoaleatório entre 0 e 1.
RandFromSeed	Rand(2)	Retorna um número pseudoaleatório baseado em um valor pré-implantado. Para um valor pré-implantado fornecido, será gerado o mesmo conjunto de números aleatórios.
Round	Round(2.166000, 2)	Arredonda uma expressão numérica para n dígitos de precisão.
Sign	Sign(Profit)	Esta função retorna o seguinte: <ul style="list-style-type: none">•1 se a expressão numérica for avaliada como número positivo•-1 se a expressão numérica for avaliada como número negativo•0 se a expressão numérica for avaliada como zero
Sin	Sin(1)	Calcula o seno de uma expressão numérica.
Sqrt	Sqrt(7)	Calcula a raiz quadrada do argumento de expressão numérica. A expressão numérica deve ser avaliada como um número não-negativo.
Tan	Tan(1)	Calcula a tangente de uma expressão numérica.
Truncate	Truncate(45.12345, 2)	Trunca um número decimal para retornar um número especificado de casas após o ponto decimal.



STUDY **HARD**

Referências

- <https://br.ccm.net/contents/872-introducao-ao-sgbd-oracle>
- <http://www.criarweb.com/artigos/230.php>
- <https://www.portalgsti.com.br/oracle-database/sobre/>
- <https://www.devmedia.com.br/estruturas-de-memoria-logicas-do-oracle-revista-sql-magazine-95/23322>
- <http://www.linhadecodigo.com.br/artigo/99/a-arquitetura-do-oracle.aspx>
- <https://www.oracle.com/technetwork/pt/indexes/products/index.html>
- https://docs.oracle.com/cloud/help/pt_BR/reportingcs_use/BILPD/GUID-4CBCE8D4-CF17-43BD-AAEF-C5D614A8040A.htm#BILUG686
- https://docs.oracle.com/cd/B28359_01/olap.111/b28126/dml_functions_1090.htm#OLADM547
- <https://www.profissionaloracle.com.br/2009/06/22/criando-e-gerenciando-sinonimos-oracle/>
- <https://cooperati.com.br/2014/03/oracle-comandos-basicos-sqlplus/>