

Aula 01

Banco de Dados II



**Departamento de
Computação**



ACADEMY



PLANO DE ENSINO

Objetivos

- **Gerais:**

- *Familiarizar o aluno com as tecnologias e metodologias para desenvolvimento de aplicações de maior grau de complexidade*

- **Específicos:**

- *Capacitar o aluno a implementar soluções utilizando bancos de dados apresentando as diversas tecnologias da área*

Plano de Ensino

- **Ementa:**

- *Fundamentos SQL; Introdução ao ambiente SQL Developer; a linguagem SQL; Linguagem de Definição de Dados; Linguagem de Manipulação de Dados; comandos para controle de transações; comandos e funções SQL; consultas avançadas; Linguagem para controle de dados; visões, sequências e índices.*



Plano de Ensino

- *Metodologia:*
 - *Aulas Teóricas*
 - *Laboratório de Informática (Prática)*
 - *Recursos audiovisual (Datashow)*



Plano de Ensino

- *Critérios de Avaliação:*
 - *Média Final (MF): NP1 + NP2 + Exercícios*
 - *Trabalhos + Exercícios → (0,5)*
 - *Aprovação: MF >= 6,0 (seis)*
 - *Frequência Mínima: 75%*



Conteúdo Programático

- 1. Recuperar dados utilizando a declaração SQL SELECT**
- 2. Restringir e Classificar os Dados**
- 3. Funções de Linha Simples**
- 4. Utilizar Funções de Conversão e Expressões Condicionais**
- 5. Exibir Dados a partir de Tabelas Múltiplas**

Conteúdo Programático

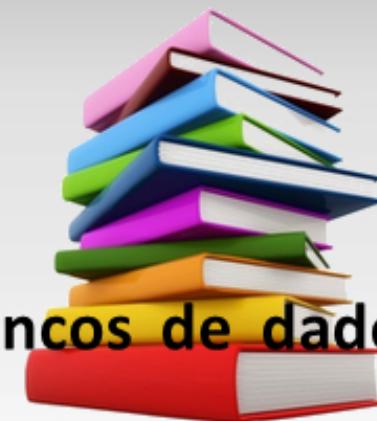
- 6. Utilizar Subconsultas para Resolver Problemas*
- 7. Utilizar Operadores de Conjunto*
- 8. Manipular Dados*
- 9. Utilizar Declarações DDL para Criar e Gerenciar Tabelas*
- 10. Criar outros Objetos de Esquema*

Plano de Ensino

- *Bibliografia*

- *Básica*

- DATE, C. **Introdução a sistemas de bancos de dados.** Rio de Janeiro: Campus, 2000.
 - HEUSER, C. A. **Projeto de bancos de dados.** Porto Alegre: Sagra-Luzzatto, 2000.
 - SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistema de Banco de Dados.** Campus, 2006.



- *Bibliografia*
 - *Complementar*
 - BATINI, C.; CERI, S.; NAVATHE, S. B. **Conceptual database design: an entity-relationship approach.** Redwood City, Calif.: Benjamin/Cummings Pub. Co., 1992.
 - ELMASRI, R.; NAVATHE, S. B. **Fundamentals of database systems.** Reading, Mass.: Addison-Wesley, 2002.

Plano de Ensino

- **Bibliografia**
 - **Complementar**
 - KORTH, Henry F.; SILBERSCHATZ, Abraham; S. Sudarshan; *Sistema de Banco de Dados*; 2006; Ed. Campus.
 - ELMASRI, Ramez; *Sistema de Banco de Dados - Fundamentos e Aplicações* - 4^a Edição, 2005; Pearson Education.
 - GUIMARÃES, Célio Cardoso; *Fundamentos de Bancos de Dados - Modelagem , Projeto e Linguagem Sql*; Unicamp

Exercícios

- *Entrega*
 - Plataforma AVA
 - **BD2 – Nome do Aluno – Exercício X**



- *Principais tarefas* de um *DBA*:
 - *Projeto e criação* de *BD*
 - *Ajuste e monitoramento de desempenho*
 - *Backup e Recuperação*
 - *Segurança* do *BD*
 - *Integridade* de *Dados (física, semântica e interna)*
 - *Supporte*
 - *Migração e Atualização* do *SGBD*

Certificação

- *Polêmicas:*
 - *Vale a pena investir em certificações?*
 - *Existe retorno ao investir tempo e dinheiro em certificações?*
 - A *experiência* é *mais relevante comparado* as *certificações?*



Certificação Oracle

Seja reconhecido como um
líder
do setor

Seja certificado pela Oracle e conquiste uma vantagem profissional.
Junte-se a mais de 2 milhões de pessoas certificadas que relataram
benefícios significativos em suas carreiras.

<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

O Programa de certificação da Oracle prepara você para o sucesso no setor de tecnologia, em mudança constante.

Para se destacar em um mercado de trabalho competitivo, você precisa provar que tem as habilidades e uma ampla base de conhecimentos para usar os produtos e as tecnologias mais recentes da Oracle, que incluem:

- Database
- Java
- MySQL
- Linux
- Oracle Cloud e aplicativos

<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

97% das empresas da Fortune 500 usam soluções da Oracle!

Com uma Certificação da Oracle, você adquire habilidades para usar essas tecnologias de forma eficaz e agregar valor à sua organização.

As pessoas que tomam decisões em uma empresa sabem o valor de um certificado de TI e o que ele significa para a organização. Investir no processo de Certificação da Oracle recompensa sua organização com funcionários motivados, produtivos e habilidosos, que são capazes de atender às futuras demandas de TI.

<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

Junte-se a 2 milhões de profissionais em todo o mundo que já são certificados pela Oracle.

A Certificação da Oracle valida suas habilidades de uma forma eficaz e ajuda você a resolver problemas reais do seu trabalho.

As credenciais são concedidas por meio da combinação das seguintes exigências, dependendo do nível de certificação:

- Passar nos exames necessários
- Fazer o treinamento necessário
- Concluir tarefas controladas por desempenho

<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

- ***Classificação:***
 - ***Oracle Certified Associate (OCA)***
 - *Habilidades e conhecimentos fundamentais, com 1 a 2 anos de experiência*
 - ***Oracle Certified Professional (OCP)***
 - *Esse nível mede habilidades mais profundas e avançadas. Tornar-se um OCP prova suas habilidades e seus amplos conhecimentos em uma área específica das tecnologias Oracle, com 2 a 3 anos de experiência*

Certificação Oracle

- ***Classificação:***

- ***Oracle Certified Master (OCM)***

- *É capaz de aplicar suas habilidades e seus conhecimentos na solução de problemas em um ambiente de avaliação prática que simula cenários reais*

- ***Oracle Certified Expert (OCE)***

- *Desenvolve habilidades comprovadas pelo Associado (OCA) ou Profissional certificado da Oracle (OCP)*

Certificação Oracle

Obtenha os seguintes benefícios com um certificado da Oracle:

- Compartilhe o selo digital seguro no seu perfil do LinkedIn, página do Facebook ou conta do Twitter para validar sua especialização profissional e técnica.
- Expanda suas habilidades de mercado e conquiste o reconhecimento necessário para atrair boas oportunidades de emprego.
- Prove que você tem experiência, motivação e comprometimento para ser o melhor na sua área.

Mais do que nunca, é importante obter uma vantagem competitiva no mercado e manter suas habilidades atualizadas.

Seu certificado da Oracle comprova sua especialização e valida seu conhecimento para ajudá-lo a alcançar novos objetivos e avançar em sua carreira.



<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

Gerentes responsáveis por contratações investem na certificação da equipe e recrutam pessoas certificadas pela Oracle

Pesquisas comprovam que uma certificação é muito valorizada pelos empregadores e gerentes responsáveis por contratações.

JOB GAINS AND CAREER BENEFITS

Certification can lead to more challenging and fulfilling roles that allow you to highlight your skills.



87%
Say that their Oracle certification enhances their professional credibility²



93%
Hiring managers who believe that certifications are beneficial¹—certification is often the deciding factor when decision-makers screen job candidates³



68%
Say that certification has increased their value in the job market⁴



6x
Increased LinkedIn profile views for people with certifications, boosting their visibility and career opportunities⁵

Investir em mão de obra certificada pela Oracle aumenta a confiança em uma equipe. O certificado dá à equipe condições de fornecer um serviço de melhor qualidade e encoraja as pessoas a trabalharem para ser as melhores em suas áreas.

O certificado da Oracle prepara sua empresa para ser líder do setor e comprova que seus funcionários estão atualizados com as habilidades e o conhecimento do setor.

Manter-se atualizado também significa investir no desenvolvimento profissional constante, o que mantém os funcionários motivados, produtivos e dedicados.

<https://www.oracle.com/br/corporate/features/oracle-certification.html>

Certificação Oracle

- ***Detalhes:***
 - ***Agendamento:***
<http://www.pearsonvue.com/oracle/>
 - ***Custo: U\$ 125,00 (cada exame)***
 - ***Presencial / OnLine (qualquer lugar com acesso à Internet (site da Person Vue)***

Certificação Oracle

- **OCA:**
 - *Atuar como DBA Oracle*
 - *Conhecer:*
 - *Arquitetura do BD Oracle*
 - *Instalação do Software Oracle Database*
 - *Criação de um BD Oracle*
 - *Gerenciamento de uma instância Oracle*
 - *Gerenciamento de estrutura de armazenamento de BD*
 - *Administração de segurança de usuários do BD*
 - *Gerenciamento de objetos de schemas*
 - *Gerenciamento de concorrência e acesso de dados*

Certificação Oracle

- **OCA (continuação):**
 - *Atuar como DBA Oracle*
 - *Conhecer:*
 - *Gerenciamento de dados de Undo*
 - *Implementação de Segurança no BD Oracle*
 - *Configuração do ambiente de rede Oracle*
 - *Manutenção proativa*
 - *Gerenciamento de performance*
 - *Conceitos e gerenciamento de backup e recuperação de dados*
 - *Recuperação de dados com Flashback*
 - *Exportação e importação de dados*

Certificação Oracle

- **OCA:**
 - ***Custos***

Tipo	Código	Valor	Quantidade de questões	Duração	Score para aprovação
Eletivo	1Z0-007	US\$ 125	52	120 min.	71%
Eletivo	1Z0-047	US\$ 125	70	120 min.	66%
Eletivo	1Z0-051	US\$ 125	70	120 min.	60%
Obrigatório	1Z0-052	US\$ 125	70	90 min.	66%

Certificação Oracle

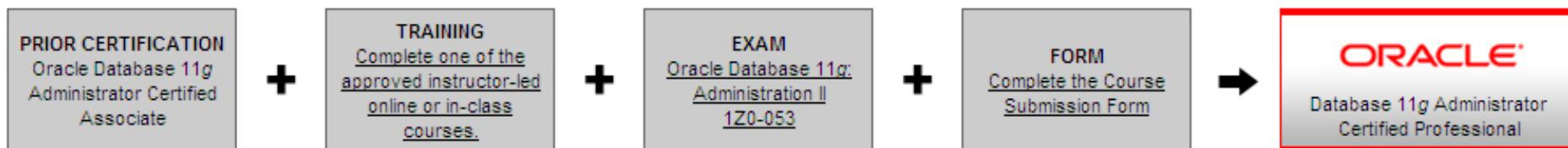
- *OCP:*
 - *Atuar como DBA Oracle*
 - *Conhecimento avançados em Administração de BD Oracle:*
 - *Uso de suporte à globalização*
 - *Recuperação e uso do RMAN (Recovery Manager)*
 - *Recuperação do BD*
 - *Recuperação de erros de usuários e perdas não críticas*
 - *Detecção e resolução de dados corrompidos*
 - *Gerenciamento automático do BD*
 - *Monitoramento e gerenciamento do armazenamento do BD*

Certificação Oracle

- *OCP (continuação):*
 - *Atuar como DBA Oracle*
 - *Conhecimento avançados em Administração de BD Oracle:*
 - *Monitoramento e gerenciamento de memória*
 - *Configuração e gerenciamento de Flashback Database*
 - *Gerenciamento de Recursos através do Database Resource Manager (DRM)*
 - *Automação de tarefas como o Scheduler*
 - *Flashback Data Archive*
 - *Automatic Storage Manager (ASM) Enhancements*

Certificação Oracle

- *OCP (continuação):*
 - *Obter OCA*
 - *Participar de um curso oficial da Oracle (presencial ou online)*
 - *Aprovação no exame 1Z0-053 (Oracle Database 11g: Administration II)*



Qual é o valor da hora técnica de um certificado?



ORACLE®

Certified Associate

INTRODUÇÃO

Introdução

- Há uma *demand*a sempre *crescente* por *pessoal* com *certificação/competência* na *indústria de TI*
- Os *benefícios* aos *colaboradores* são *significantes* – eles podem *estar certos* de que o *pessoal* tem *certo nível* de *competência* – e os *benefícios* às *pessoas*, em *termos* de *demand*a por seus *serviços*, são *igualmente grandes*. As *certificações Oracle* estão entre as *mais procuradas*.

- **Objetivos OCA (Parte 1):**
 - Recuperar dados utilizando a declaração SQL SELECT
 - *Especificar os recursos das declarações SQL SELECT*
 - *Executar uma Declaração SELECT básica*
 - Restringir e Classificar os Dados
 - *Limitar as linhas recuperadas por uma consulta*
 - *Utilizar substituição de E comercial para restringir e classificar resultado no tempo de execução*

- **Objetivos OCA (Parte 2):**
 - Funções de Linha Simples
 - *Descrever Vários Tipos de Funções* disponíveis em SQL
 - *Utilizar Funções* de *caractere*, *número* e *data* em *instruções SELECT*
 - Utilizar Funções de Conversão e Expressões Condicionais
 - *Descrever Vários Tipos de Funções de conversão* que estão *disponíveis* em SQL

- Objetivos OCA (Parte 3):
 - Utilizar Funções de Conversão e Expressões Condicionais
 - *Utilizar* as **funções de conversão** TO_CHAR, TO_NUMBER e TO_DATE
 - *Aplicar expressões condicionais* em uma **instrução SELECT**

- Objetivos OCA (Parte 4):
 - Informar Dados Agregados Utilizando as Funções de Grupo
 - *Identificar* as *Funções de Grupo Disponíveis*
 - *Descrever a utilização* de *funções de grupo*
 - *Dados de Grupo* utilizando a *cláusula GROUP BY*
 - *Incluir* ou *excluir linhas agregadas* utilizando a *cláusula HAVING*

- Objetivos OCA (Parte 5):
 - Exibir Dados a partir de Tabelas Múltiplas
 - *Gravar instruções SELECT para acessar dados de mais de uma tabela utilizando junções equivalentes e não equivalentes*
 - *Unir uma tabela a ela mesma utilizando autojunção*
 - *Visualizar Dados que geralmente não satisfazem uma condição de junção utilizando junções externas*
 - *Criar um produto cartesiano de todas as linhas de duas ou mais tabelas*

- **Objetivos OCA (Parte 6):**
 - Utilizar Subconsultas para Resolver Problemas
 - *Definir subconsultas*
 - *Descrever os tipos de problemas que as subconsultas podem resolver*
 - *Especificar os tipos de subconsultas*
 - *Gravar subconsultas de linhas únicas e múltiplas*

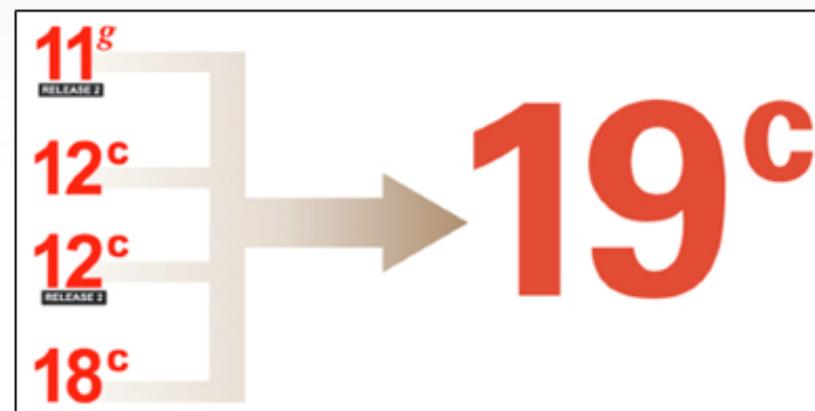
- **Objetivos OCA (Parte 7):**
 - Utilizar Operadores de Conjunto
 - *Descrever os operadores de conjunto*
 - *Utilizar um operador de conjunto para combinar consultas múltiplas em uma única consulta*
 - *Controlar a ordem das linhas retornadas*

- **Objetivos OCA (Parte 8):**
 - Manipular Dados
 - *Descrever* cada *instrução de DML*
 - *Incluir linhas em uma tabela*
 - *Modificar linhas em uma tabela*
 - *Excluir linhas de uma tabela*
 - *Transações de controle*

- Objetivos OCA (Parte 9):
 - Utilizar Declarações DDL para Criar e Gerenciar Tabelas
 - *Categorizar os principais objetos de BD*
 - *Revisar a estrutura da tabela*
 - *Especificar os tipos de dados que estão disponíveis para colunas*
 - *Criar uma tabela simples*
 - *Explicar como as restrições são criadas no momento da criação da tabela*

- Objetivos OCA (Parte 10):
 - Criar outros Objetos de Esquema
 - *Criar visualizações simples e complexas*
 - *Recuperar dados das visualizações*
 - *Criar, manter e utilizar sequências*
 - *Criar e manter índices*
 - *Criar sinônimos particulares e públicos*

19^c ORACLE® Database

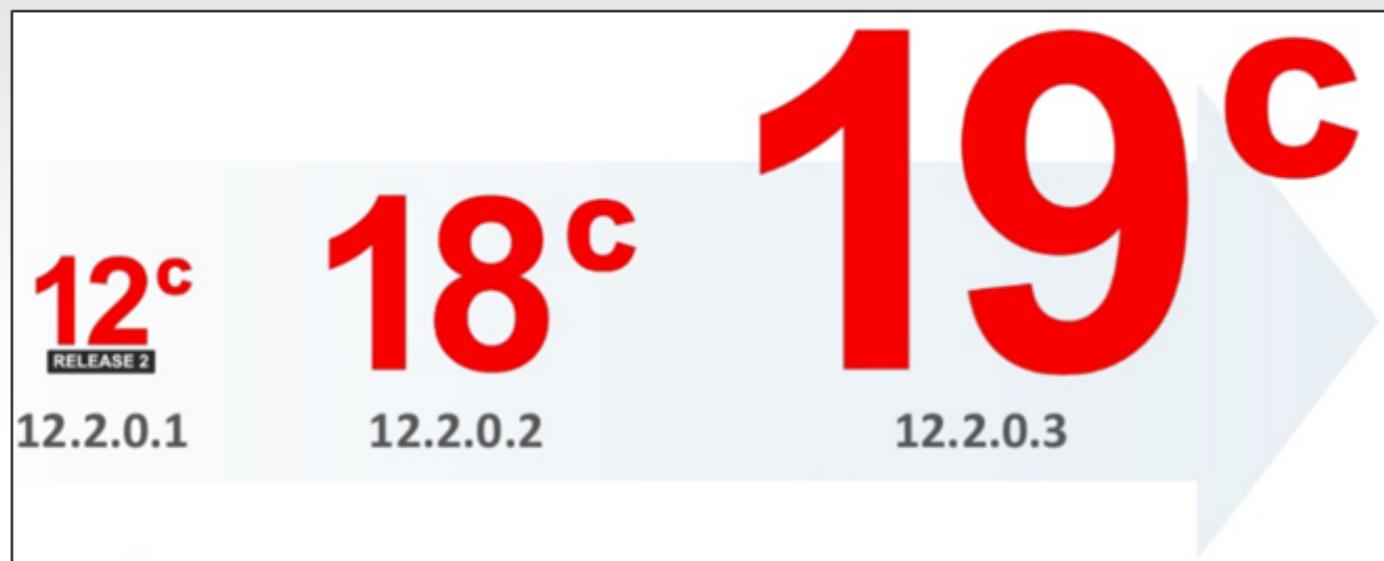


Oracle 19c

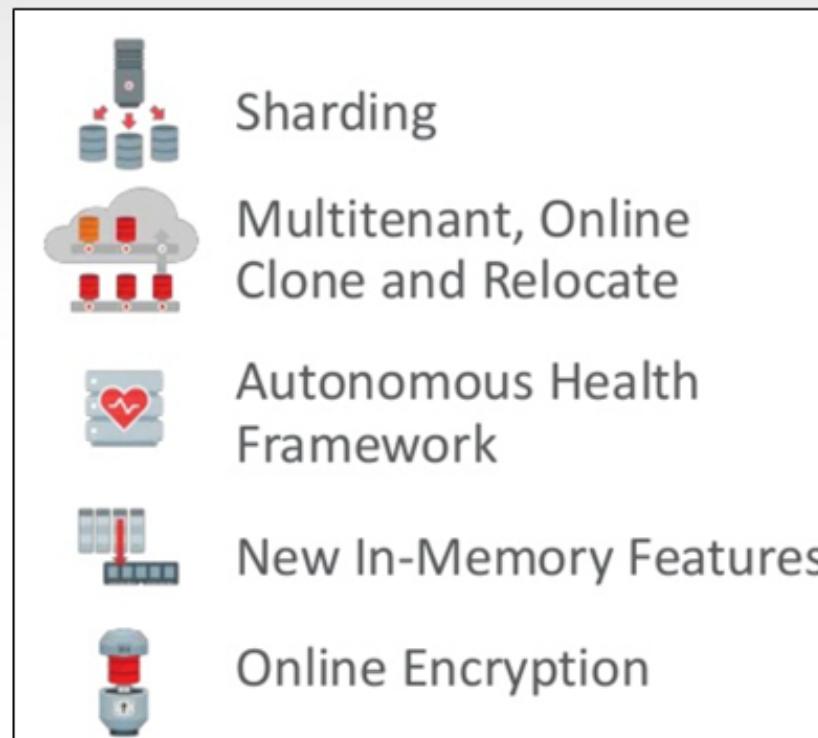
- *Database Release and Support Timelines up until 19c*



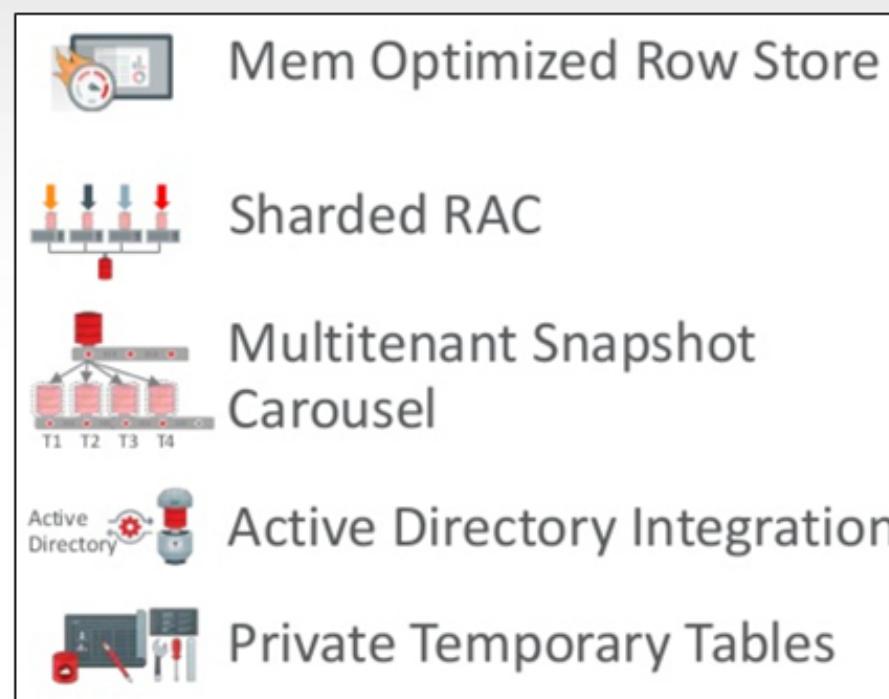
- *Last Release of the 12.2 family of products*



- *Continual improvements in each release...*
 - 12^c (release 2)



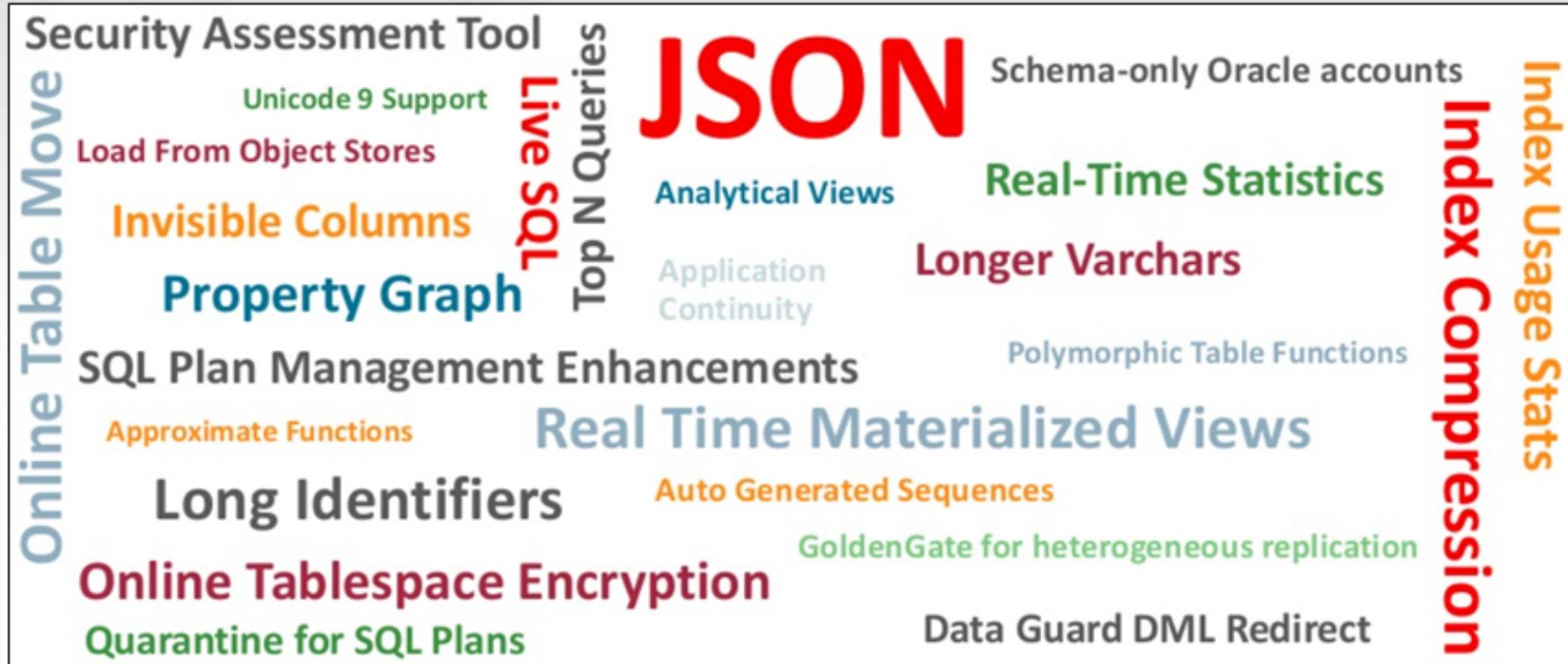
- *Continual improvements in each release...*
 - 18^c (12.2.0.2)



- *Continual improvements in each release...*
 - 19^c (12.2.0.3)



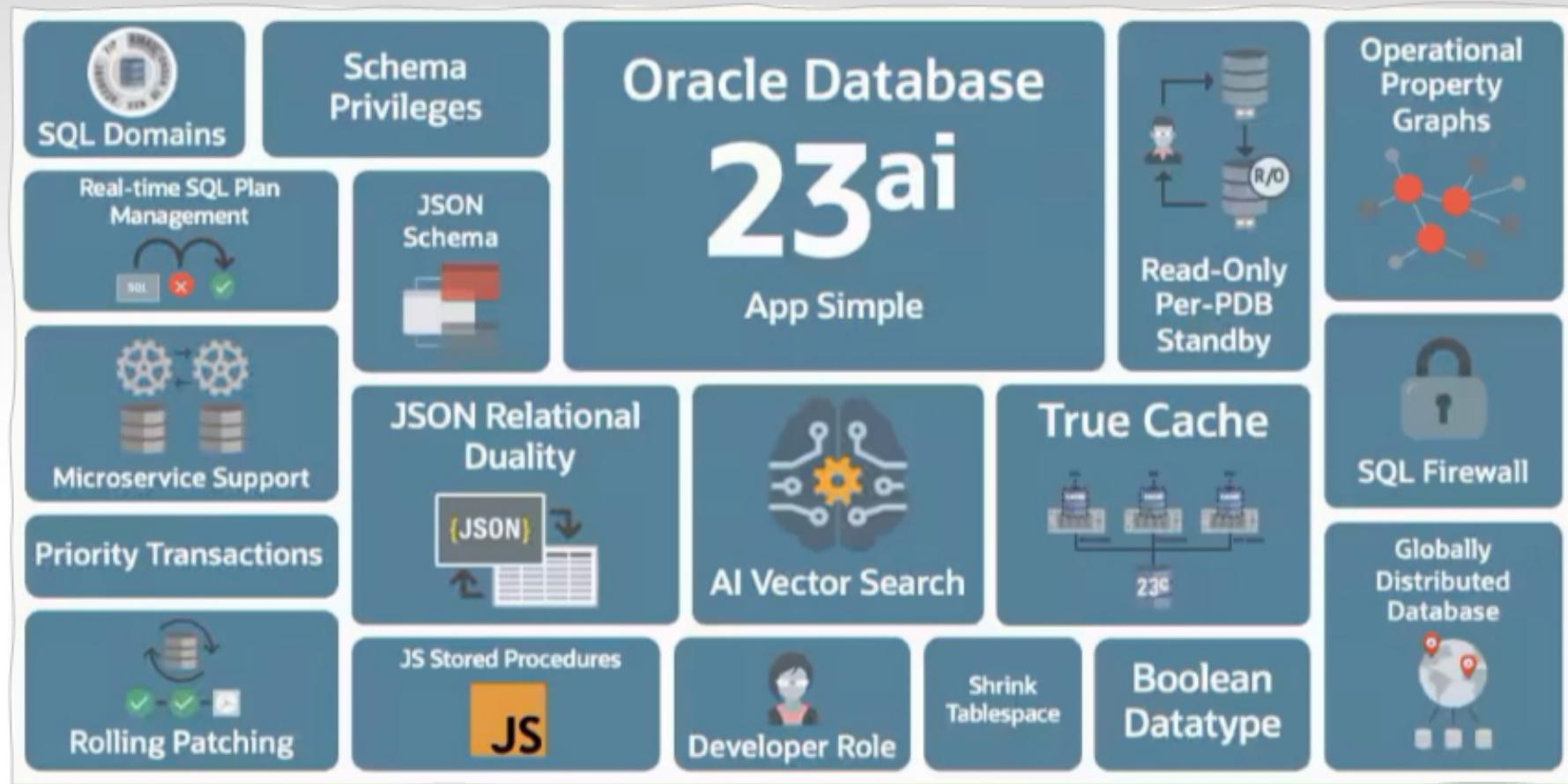
- *And Many Many More...*



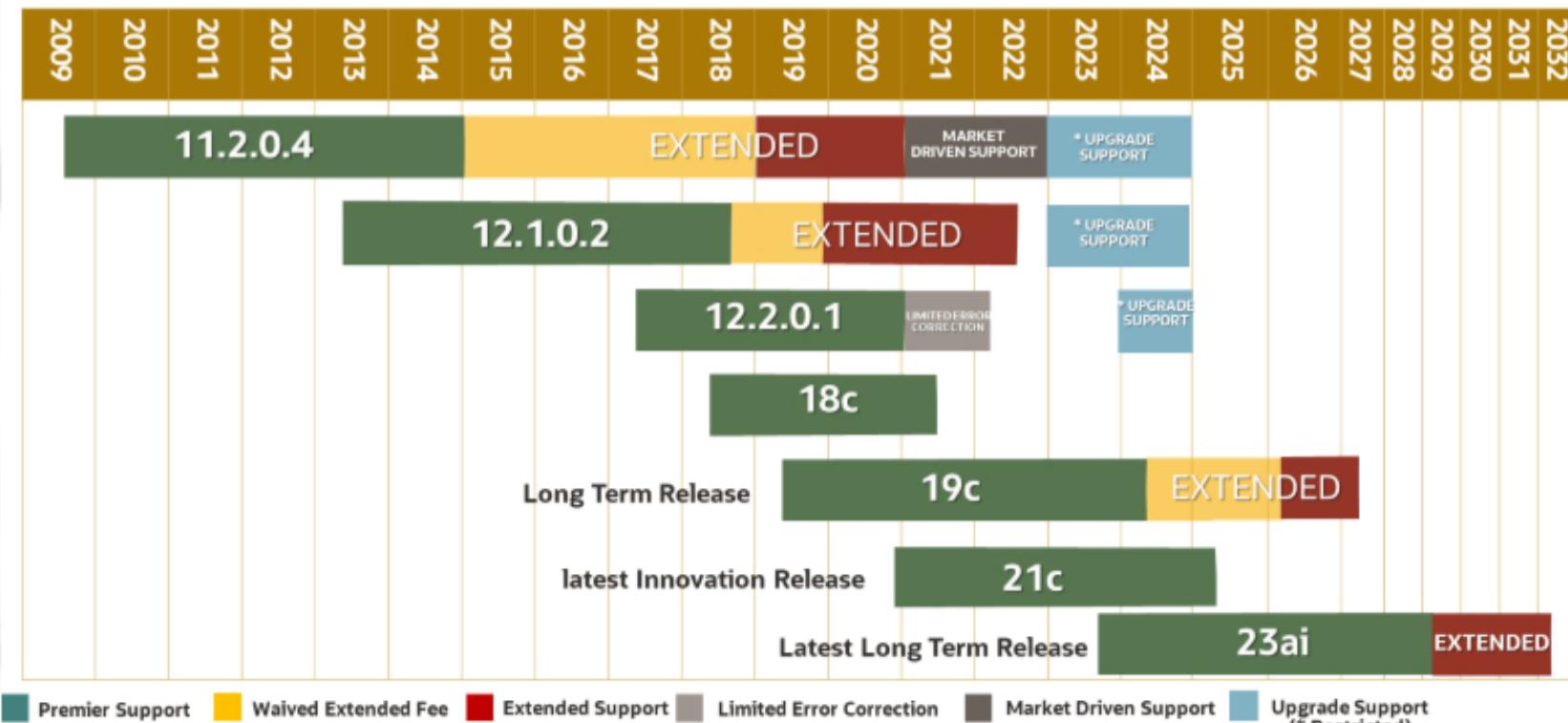
Oracle 23ai



Oracle 23ai

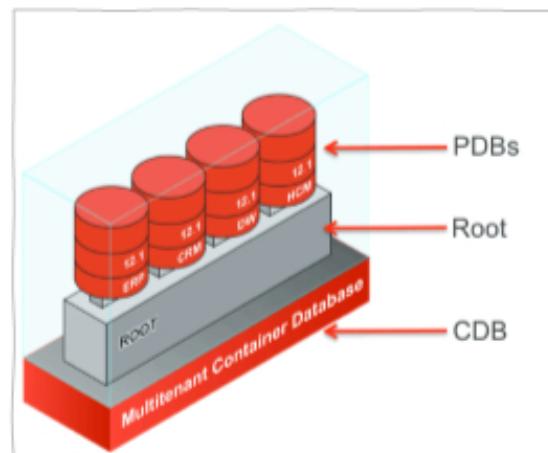


Database Releases and Support Timelines



Copyright © 2024, Oracle and/or its affiliates

- Os *BDs passam* a ser *administrados* em *uma única arquitetura (Multitenant Container Database)*
- No *CDB* os *recursos utilizados* pelo *Oracle* são *compartilhados* entre *vários BDs, inclusive* parte do *dicionário de dados e metadados*
- Os *dados de usuários* são *gerenciados separadamente*



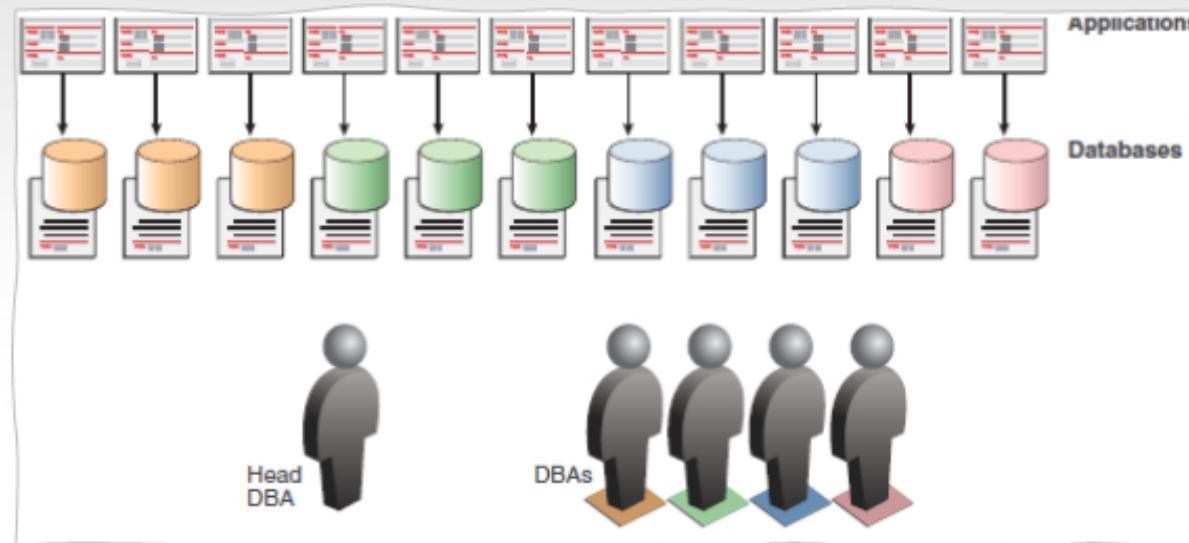
- *O compartilhamento é promovido por meio de uma estrutura denominada Container*
- Container: repositório onde “n” BDs que podem ser conectados ou desconectados, de forma transparente para as aplicações
- *Bancos de Dados Plugáveis (Pluggable Database)*

- Arquitetura Multitenant:

- *Uma única instância* de um *determinado software*
- *Executando em um servidor*
- *Disponibiliza* um *serviço* para *diversos clientes*
- *Cada cliente* é *chamado* de *Tenant (inquilino)*

- **Ambiente *antes* da consolidação dos BDs**
 - *100 servidores*
 - *Cada servidor* possui um **BD** onde o **consumo de hardware** é em **média 10% + 10%** de **tempo** para **administração**
 - Um **grupo** de **DBAs** deve **gerenciar** a **SGA** (*System Global Area*), **processos backgrounds**, **datafiles**, **archives**, **temp**, **undo**, **conta** de **usuários**, **segurança**, etc., para **cada** um dos **BDs individualmente**

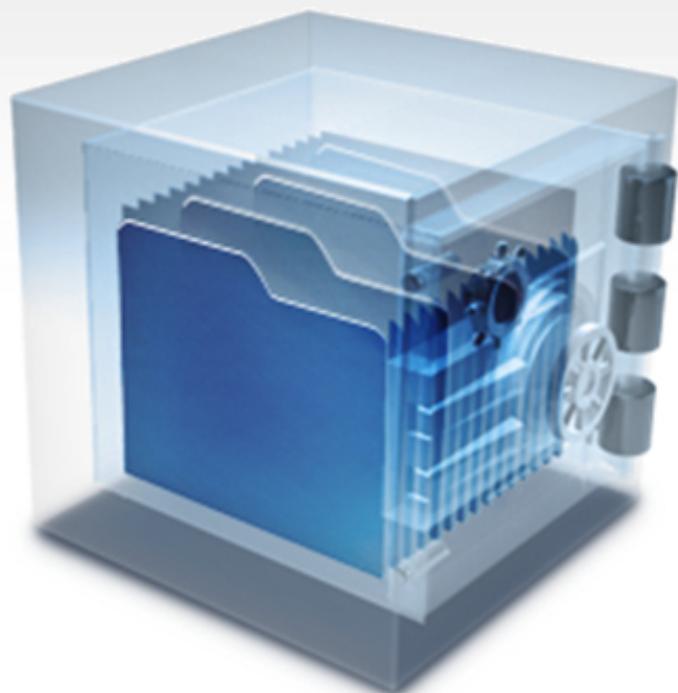
- *Ambiente **antes** da consolidação dos BDs*



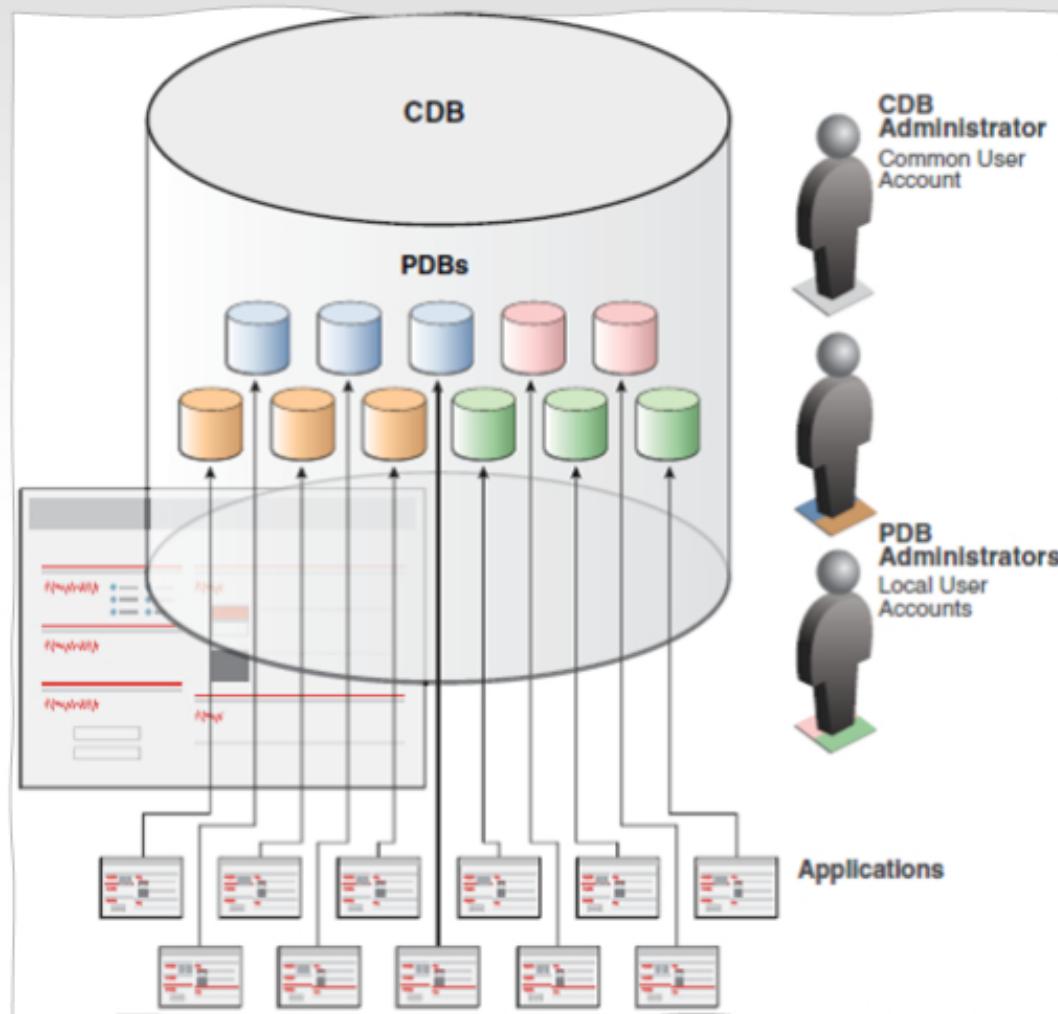
- **Ambiente com o CDB**

- *Criação de um container*
- *Inserir os 100 BDs dentro do container*
- *Gerenciar os BDs como se fosse apenas um*
- *Atualização é realizada no container e, automaticamente é aplicado em todos os BDs que se encontram conectados a ele*
- *Também é possível atualizar simplesmente desconectando um BD de um container e conectando em outro container que possua as atualizações necessárias*

- *Ambiente com o CDB (continuação)*
 - *Backups e recovery também* podem ser *realizados no nível* do CDB



- **Ambiente com o CDB (continuação)**





TECNOLOGIAS DE SERVIDOR

Tecnologias de Servidor

- *Existe uma família de produtos que compõe as tecnologias de servidor Oracle:*
 - O **Banco de Dados Oracle**;
 - O **Servidor de Aplicativos Oracle**;
 - **Oracle Enterprise Manager**;
 - **Várias ferramentas de desenvolvimento de aplicativos e linguagens.**

Tecnologias de Servidor

- *Banco de Dados:*
 - *Repositório para dados e mecanismo de gerenciamento do acesso* a eles
- *Servidor de Aplicativos Oracle:*
 - *Executa software em nome de usuários finais*
 - *Cria uma interface de usuário na forma de janelas exibidas em navegadores de usuários (frontend) e envia chamadas de recuperação de dados e modificação no BD para execução*

Tecnologias de Servidor

- *Oracle Enterprise Manager:*
 - *Ferramenta de administração abrangente para monitorar, gerenciar e ajustar processos Oracle*
 - *Além de processos Oracle, permite, por meio de plug-ins, monitorar, gerenciar e ajustar produtos de terceiros*

Tecnologias de Servidor

- *Ferramentas e Linguagens para desenvolver aplicativos:*
 - *Aplicativos executados nas máquinas de usuários finais* (modelo *cliente/servidor*)
 - *Aplicativos que são executados de maneira centralizada em servidores de aplicativo*

Tecnologias de Servidor

- A *mesclagem* das *tecnologias de servidor* e as *ferramentas de desenvolvimento* constituem uma *plataforma* para *desenvolvimento de aplicativos*, denominada de *Grade*;
- A *Grade* fornece *serviços* de *TI* que *proporciona* a *redução* de *custo* de *todo* o *ambiente fornecendo* poder *computacional* a *partir* de um *grupo* de *recursos disponíveis* para onde for *necessário*, sob *demandas*.

Arquitetura de Servidor Oracle

- Um *BD Oracle* é uma *série* de *arquivos* em *disco*
- Sua *existência* está *atrelada* até que *esses arquivos* sejam *efetivamente excluídos*
- **Não** há *limites práticos* para o *tamanho* e *número* desses *arquivos* (**não** existe *limites práticos* para o *tamanho* do *BD*)
- O *acesso* ao *BD* é *realizado através* da *instância* Oracle

Arquitetura de Servidor Oracle

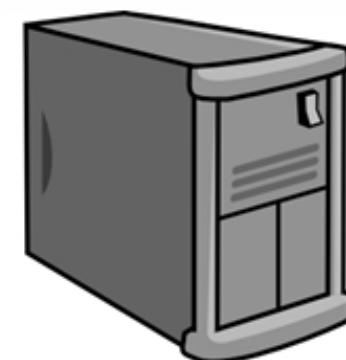
- Os *usuários* do *BD* *estabelecem sessões* contra a *instância*
- A *instância gerencia* todo o *acesso* ao *BD*
- É **impossível** em um *ambiente Oracle* qualquer, um *usuário* ter *contato direto* com o *BD*
- Uma *instância Oracle* com um *BD Oracle forma* um *servidor Oracle*

Arquitetura de Servidor Oracle

- O *Servidor Oracle* utiliza o *processamento cliente/servidor (duas camadas)*
- No *modelo cliente/servidor*, a *geração* da *interface de usuário* e *muito* da *lógica de aplicativos* são *segmentados* do *gerenciamento de dados*
- A *camada cliente* *cria* os *comandos SQL* e a *camada servidor* os *executa*
- Oracle Net: *protocolo* de *comunicação* de *rede* *utilizado* entre o *processo* de *usuário* e o *processo* de *servidor*

Arquitetura de Servidor Oracle

- ***Camada Cliente:***
 - *Constituída por dois componentes*
 - *Usuários*
 - *Processos de usuário*
- ***Camada Servidor:***
 - *Constituída por três componentes*
 - *Processos de servidor (executam SQL)*
 - *Instância*
 - *Próprio Banco de Dados*



Arquitetura de Servidor Oracle

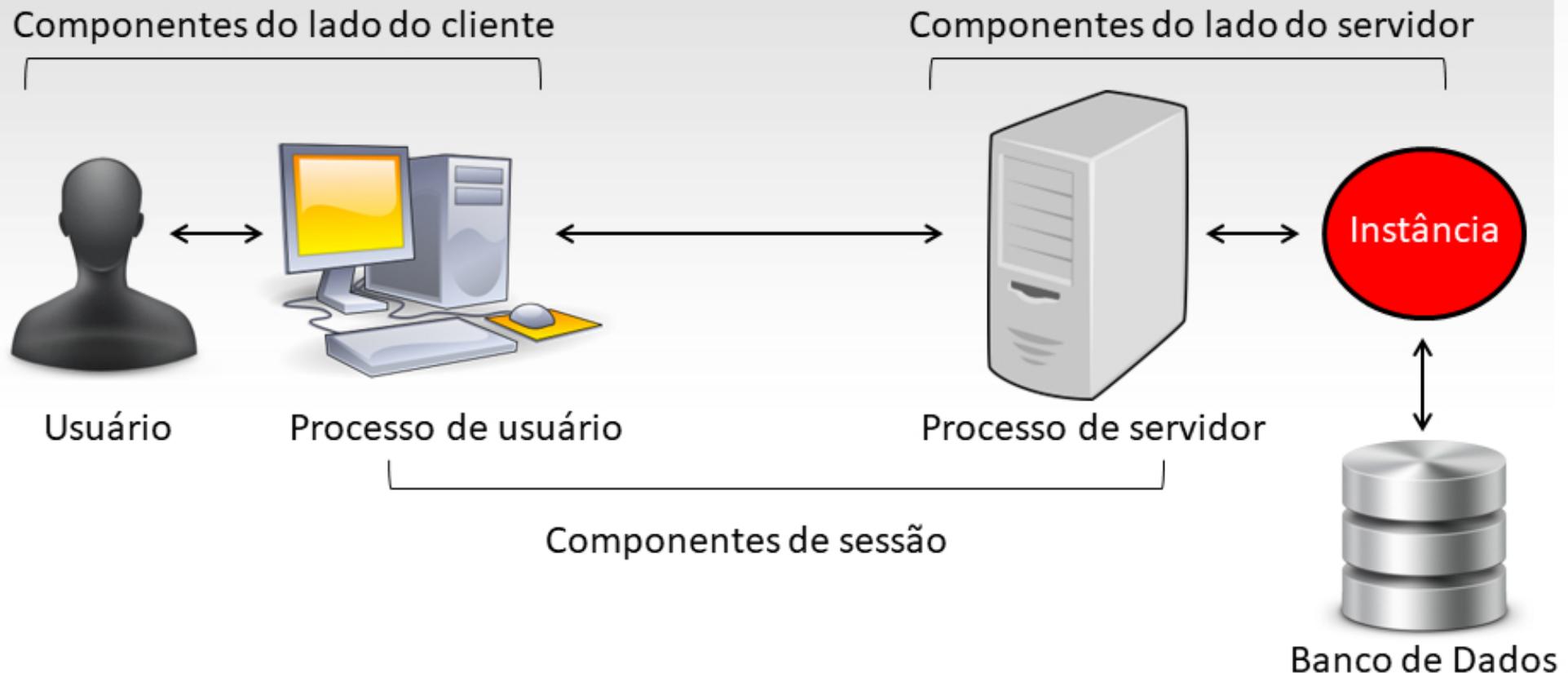
- **Sessão:**
 - *Processo de usuário* em *comunicação* com um *processo de servidor*
- *Processos de usuário e servidor* que *formam* as *sessões*, *ativados* sob *demandas* por *usuários* e *finalizados* quando *não* são mais *necessários* (*ciclo* de *logon* e *logoff*)
- **Processo de Usuário:**
 - *Qualquer software* do *lado* do *cliente* que seja *capaz* de se *conectar* a um *processo de servidor Oracle*

Arquitetura de Servidor Oracle

- Em *nosso treinamento, dois processos de usuários* serão *utilizados*:
 - *SQL *Plus*
 - *SQL Developer*
- *Processos simples fornecidos* pela *Oracle* para *estabelecer sessões* contra um *servidor Oracle* e *emitir SQL ad hoc*

Arquitetura de Servidor Oracle

- **Conexão indireta** entre um **usuário** e um **BD**



Arquitetura de Servidor Oracle

- A *forma mais simples* do *servidor de BD* é uma *instância conectada* a um *BD*
- *Ambientes complexos*: um *BD* pode ser *aberto* por “n” *instâncias*
- RAC (Real Application Cluster):
 - *Escalabilidade*
 - *Desempenho*
 - *Indisponibilidade zero*

Servidor de Aplicativos Oracle

- *Substitui o software do lado do cliente (tradicionalmente instalado nos terminais do usuário final)*
- *Executa os aplicativos de maneira centralizada*
- *Apresenta o resultado ao usuário através de janelas exibidas localmente nos navegadores de Internet*
- Os *aplicativos* utilizam *dados armazenados de um ou mais servidores de BD*

Servidor de Aplicativos Oracle

- *Oracle Application Server:*
 - *Plataforma para desenvolver, implementar e gerenciar aplicativos Web*
 - *Aplicativo Web* pode ser *definido* como um *aplicativo* com o qual os *usuários* se *comunicam* através do *HTTP*
 - *Aplicativos executados no Oracle Application Server*, podem se *conectar* a *qualquer BD* para os *quais existem drivers compatíveis* com *Java* (*não* é *necessário utilizar* um *BD Oracle*)

Servidor de Aplicativos Oracle

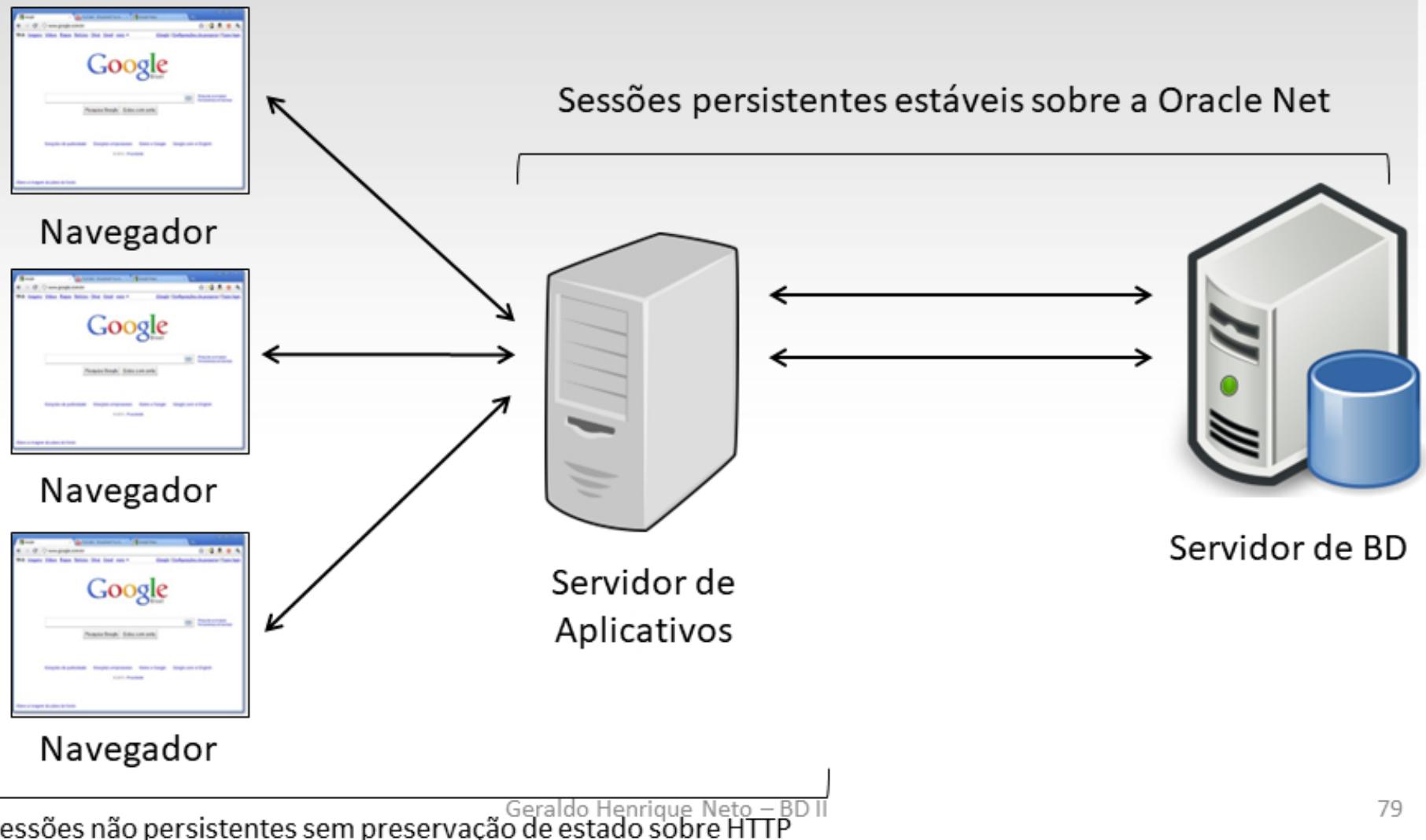
- **Modelo de Processamento (simples) de Aplicativos Web:**
 - *Camada de cliente* que **gerencia** a *interface* de **usuário**;
 - *Camada média* que **cria** a *interface* e **emite instruções SQL** para a *camada de dados*;
 - *Camada de dados* que **gerencia** os **próprios dados**.

Servidor de Aplicativos Oracle

- As *sessões* de **BD** são *sessões* de **cliente/servidor**
- *Consistem* em um *processo* de **usuário** e um *processo* de **servidor**
- Esse *modelo (cliente/servidor)* é **ineficiente** ao ser *comparado* ao *modelo* de **pool** de **conexões**
- **Pool de Conexões:**
 - O *servidor estabelece* um *número pequeno* de *sessões* de **BD persistentes** e as *disponibiliza* sob *demandas* para um *número grande* de *sessões* de **usuário final**

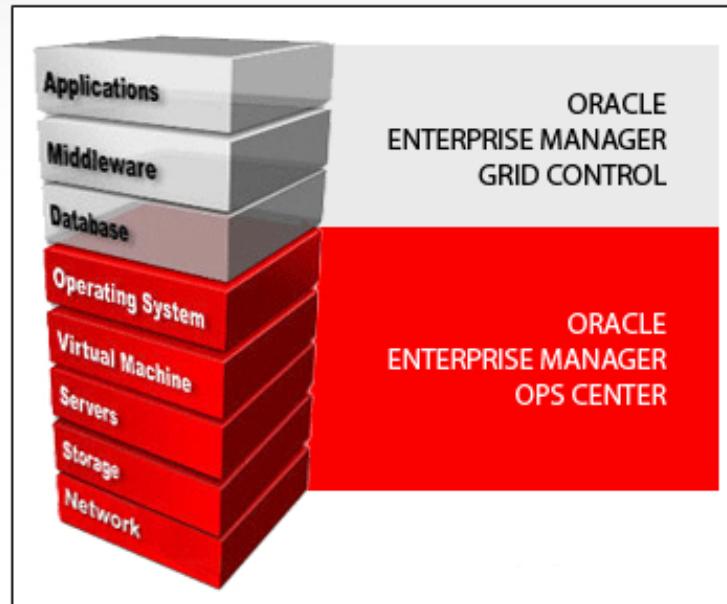
Servidor de Aplicativos Oracle

- **Pool de Conexões:**



Oracle Enterprise Manager

- O *Oracle Enterprise Manager* vem em *três formas*:
 - *Controle de BD*;
 - *Controle de Servidor de Aplicativo*;
 - *Controle de Grade*.



Oracle Enterprise Manager

- **Controle de Banco de Dados:**

- **Ferramenta gráfica** para **gerenciar** um **BD** (pode ser um **BD armazenado** em **cluster** por **RAC**)
- **Consiste** em um **processo Java** sendo **executado** na **máquina** de **servidor** do **BD**
- Os **administradores** se **conectam** ao **Controle** de **BD**, então, se **conecta** ao **servidor** de **BD**
- Possui **recursos** para **gerenciamento** e **monitoramento** em **tempo real**, para **execução** de **trabalhos agendados** e para **enviar condições** de **alerta interativamente** e por **e-mail**

Oracle Enterprise Manager

- **Controle de Servidor de Aplicativo:**
 - *Ferramenta gráfica* para *gerenciar* uma *instância* de *servidor* de *aplicativo* ou um *grupo* de *instâncias*
 - A *tecnologia* de *agrupamento* é *dependente* da *versão*
 - Até a *versão 2* do *Oracle Application Server 10g*, os *servidores* eram *gerenciados* com um *repositório* de *metadados* como um *ponto central* de *gerenciamento*
 - *Versão 3* em *diante*, a *tecnologia* baseou-se no *armazenamento* em *cluster J2EE* (**não** é de *propriedade* da *Oracle*)

Oracle Enterprise Manager

- ***Controle de Grade:***
 - ***Globaliza o ambiente de gerenciamento***
 - ***Repositório de gerenciamento e um ou mais servidores de gerenciamento controlam todo o ambiente (todos os BD e os servidores de aplicativo, onde quer que estejam)***
 - ***Gerencia nós, ou máquinas, nos quais os servidores trabalham, bem como (através de plug-ins) uma gama variada de produtos de terceiros***

Computação em Grade

- Um *item crítico* ao *conceito* de *computação em grade* é a **virtualização**
- *Usuários solicitam* um *serviço* de *aplicativo* e *deixam* a **Grade decidir** qual *servidor* de *aplicativo* pode fazer o *melhor fornecimento*
- *Servidores de Aplicativos solicitam* um *serviço* de **BD** e *permitem* que a **Grade decida** qual **nó RAC** *melhor atenderá* aos *dados*

Computação em Grade

- *Resultado:*
 - Usuários **não** têm *necessidade* nem *capacidade* de *saber* de *onde* seus *recursos computacionais* são *realmente fornecidos*
 - *Analogia:* a *eletricidade doméstica* é *fornecida* sob *demandas*, e o *proprietário* do *imóvel* **não** tem como dizer qual *estação* de *energia* está lhe *servindo* no *momento*

Linguagens

- *Dentro do BD é possível usar três linguagens:*
 - *1ª SQL:*
 - Utilizada para *acesso de dados*, mas **não** pode ser *utilizada* para *desenvolver aplicativos completos*
 - *Não* possui *recursos* para o *desenvolvimento* de *interfaces* de *usuário*
 - *Não* tem *estruturas procedurais* necessárias para *manipular linhas individualmente*
 - (as *outras duas linguagens disponíveis* dentro do *BD preenchem* essas *lacunas*)

Linguagens

- *Dentro do BD é possível usar três linguagens:*
 - **2ª PL/SQL:**
 - *Linguagem de terceira geração (3GL) pertencente à Oracle*
 - Possui os *conceitos procedurais normais* (como *if-then-else* e *looping*)
 - Possui *recursos* para *design de interface de usuário*
 - Permite *integrar chamadas SQL* para *recuperar uma ou mais linhas a partir do BD, processar* essas *linhas resultantes*, e na *sequência, gerar* mais *instruções SQL* para *escrever linhas* no *BD*

Linguagens

- *Dentro do BD é possível usar três linguagens:*
 - *3^a Java:*
 - *Oferece uma capacidade similar para integrar chamadas dentro do código Java*
 - *Qualquer desenvolvedor deve ser capaz de escrever códigos que funcionarão com um BD Oracle*



Linguagens

- *Outras linguagens* estão *disponíveis* para *desenvolver aplicativos* de *cliente/servidor* que sejam *executadas externamente* ao *BD*
- Para essas *linguagens*, a *Oracle Corporation* fornece *bibliotecas OCI (Oracle Call Interface)*
- *OCI permite* que *códigos escritos independentemente* da *linguagem* *estabeleçam sessões* a um *BD* e *invoquem comandos SQL*

Ferramentas

- *Diversas empresas não querem usar uma 3GL para o desenvolvimento de aplicativos de BD*
- A *Oracle Corporation* fornece *ferramentas rápidas de desenvolvimento de aplicativos* como *parte do Oracle Developer Suíte* (existe *vários produtos de terceiros*)
- Essas *ferramentas promovem uma produtividade significante comparado com uma 3GL*



INSTALAÇÃO – APPLIANCE DE MÁQUINA VIRTUAL COM ORACLE 19c/23AI

Objetivos

- *Constituir ambiente/cenário adequado para resolução dos laboratórios, permitindo incrementar o nível de abstração sobre Oracle Database, SQL Developer e outras ferramentas relacionadas*



Pré-Requisito

- *Instalação prévia do software Oracle **VM VirtualBox***



VirtualBox

Conteúdo (Appliance)

- *Oracle Technology Network Developer Day – Hands-on Database Application Development*
- *Aplicativos pré-configurados:*
 - *Oracle Database Single Instance & Oracle RAC 19c Enterprise/Standard Edition*
 - *Oracle Grid Infrastructure Release Update (19.3.1.0.0)*
 - *Oracle Linux 7 Update 6 with UEK Release 5 Update 2*
 - *Oracle SQL Developer*
 - *Oracle SQL Developer Data Modeler*

Requisitos de Hardware

- *Requisitos de hardware e SO:*
 - **2 Gb RAM**
 - **15 Gb de espaço livre em disco**
 - **Processador de 2GHz** (*mínimo recomendável*)
 - **Privilégios de administrador para usuário do SO**

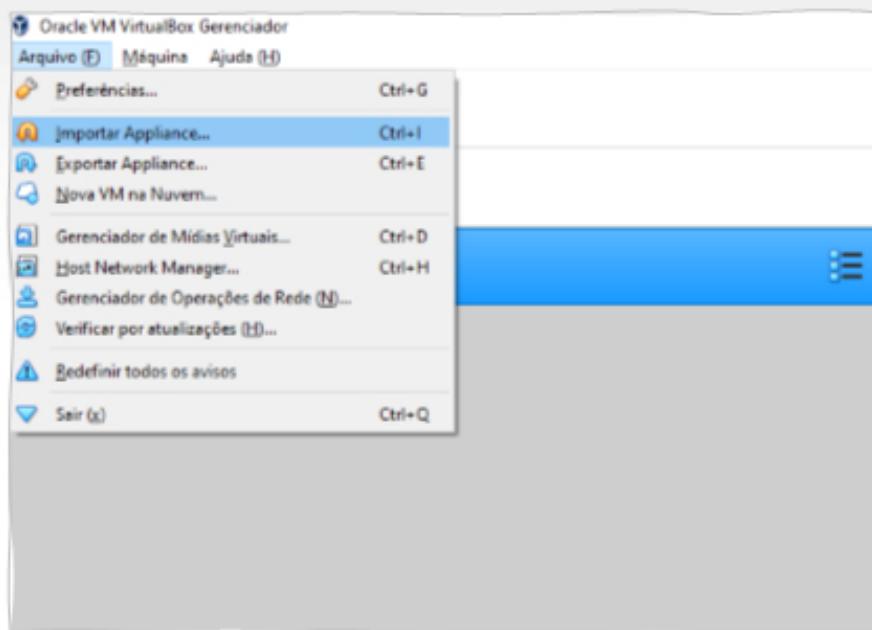
Importação do Appliance

1. Realize o *download* do arquivo Oracle Developer Days.ova

- ***Observação:***
 - Para *realizar* o *download* torna-se *necessário possuir* uma *conta “oracle”*

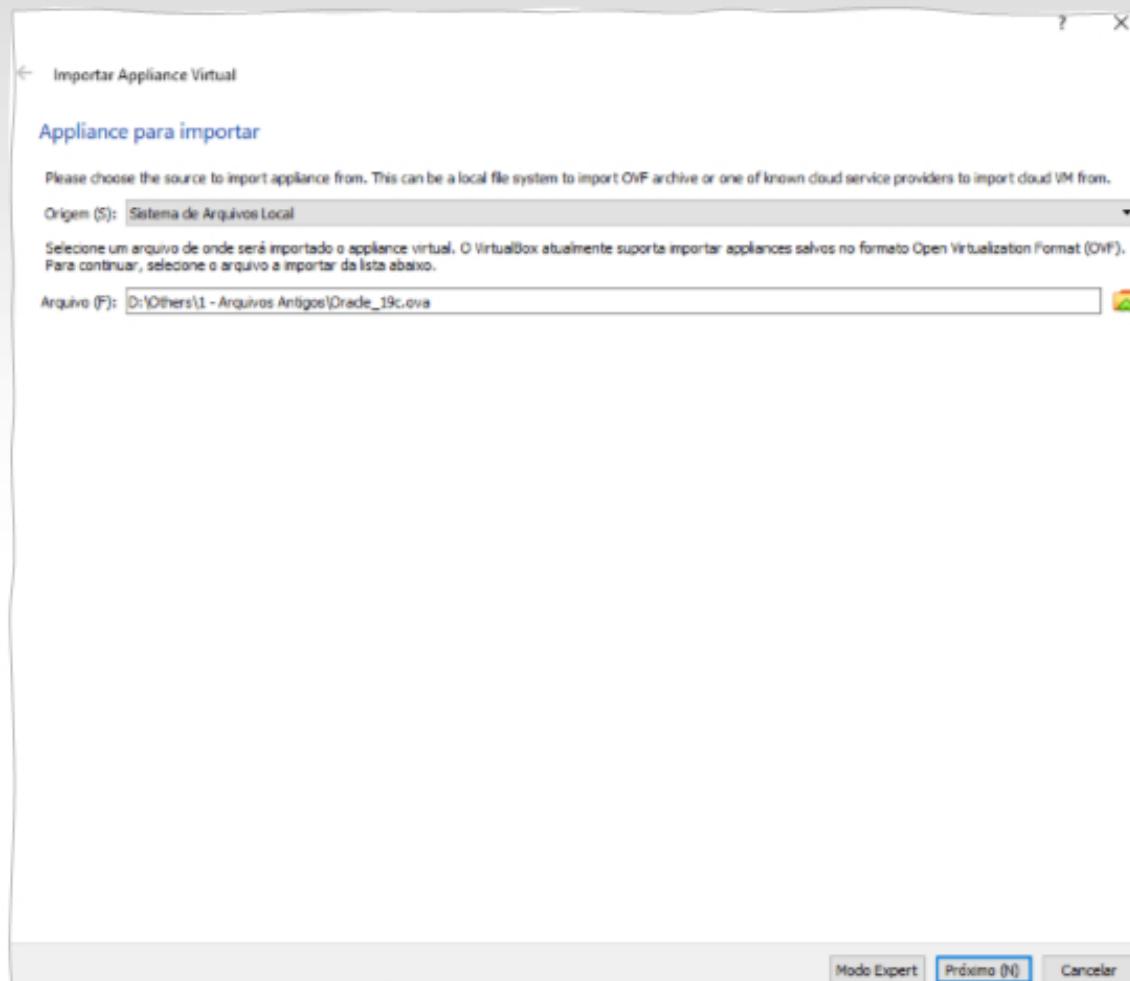
Importação do Appliance

2. Execute o Oracle VM VirtualBox e selecione “Arquivo” → “Importar Appliance”, para abrir o *Appliance Import Wizard*



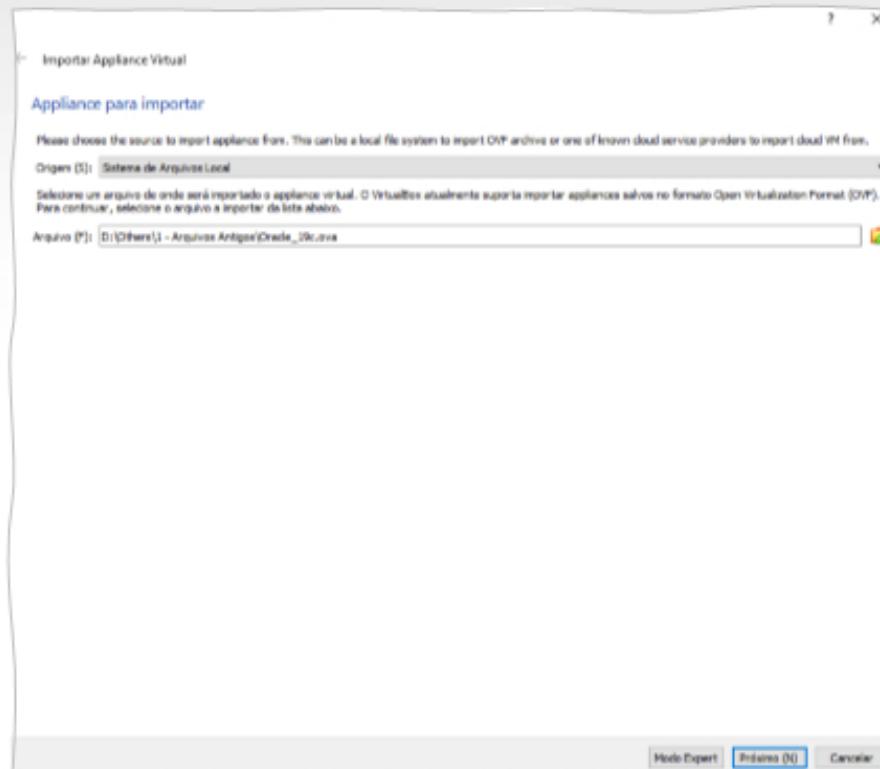
Importação do Appliance

3. Clique no botão Escolher



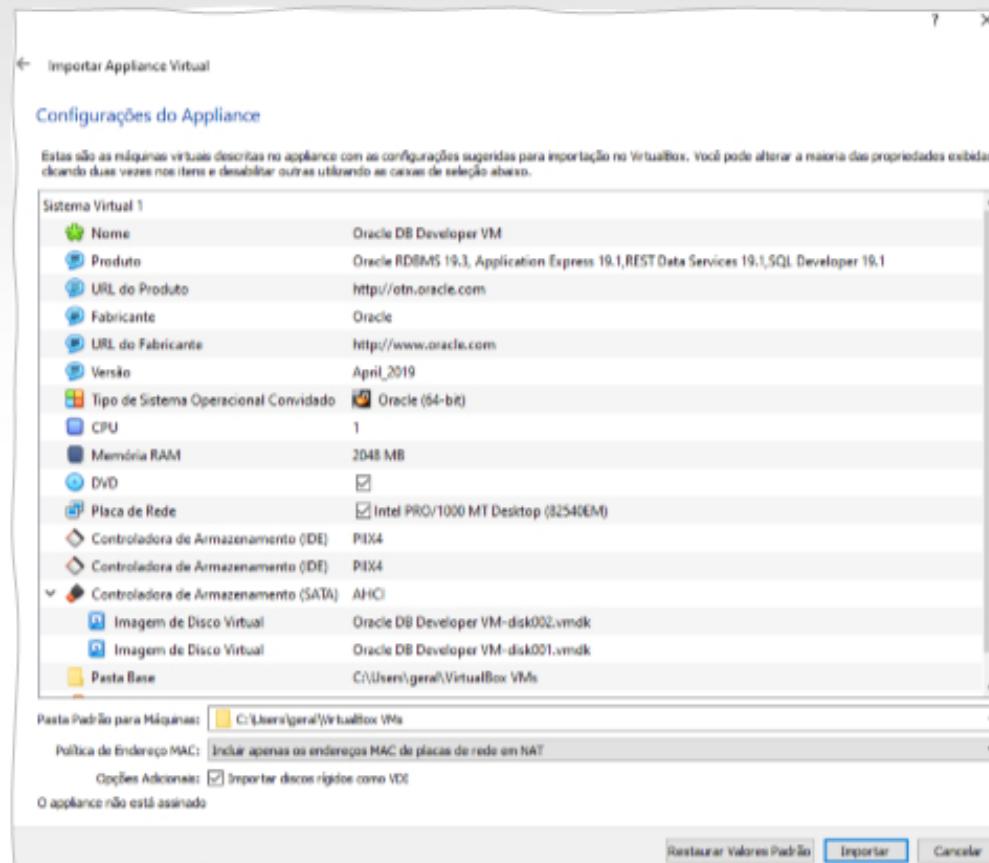
Importação do Appliance

4. *Localize e selecione o arquivo “Oracle Developer Days.ova” baixado no item 1 e clique no botão “Próximo”*



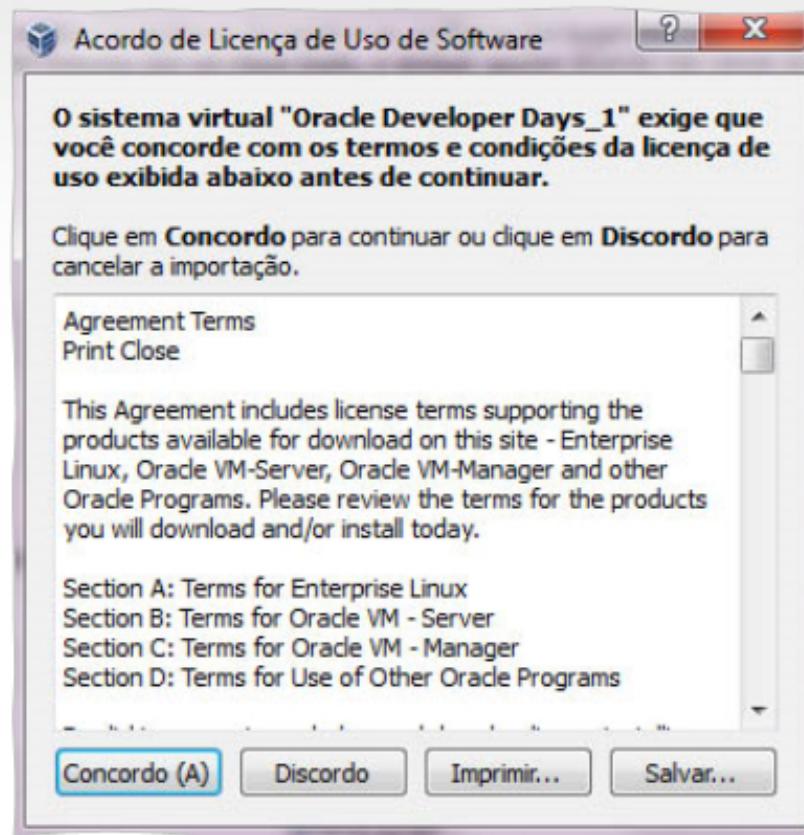
Importação do Appliance

5. Será apresentado as configurações de importação de Appliance. Clique no botão “Importar”



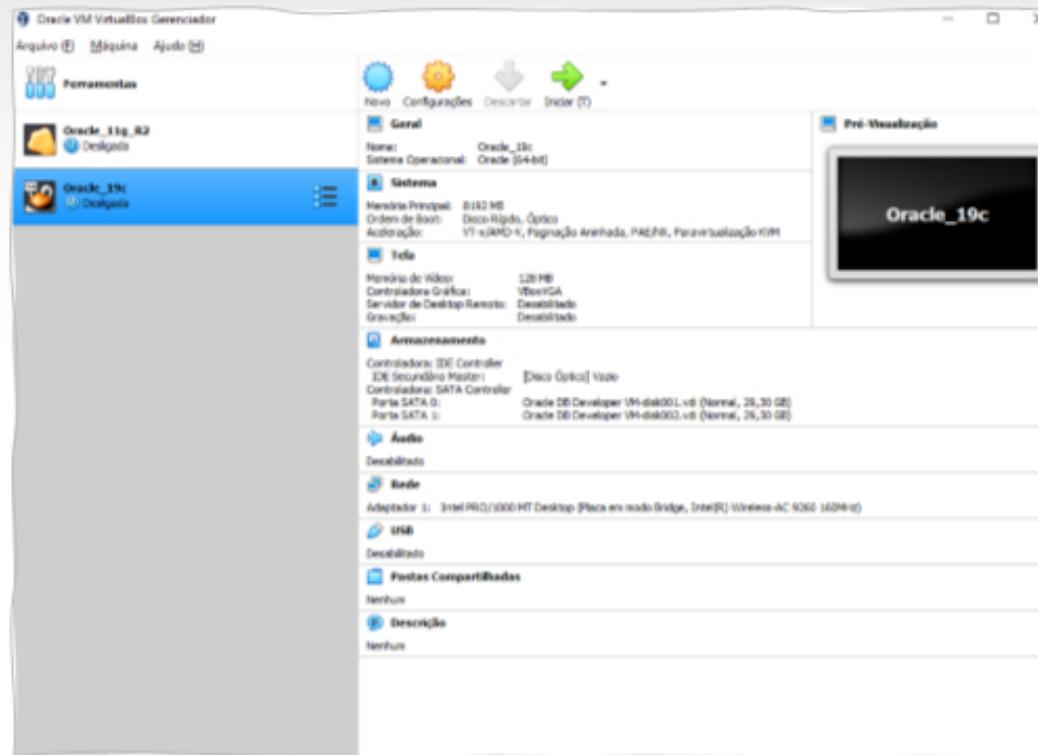
Importação do Appliance

6. Leia os *termos* e *condições* da *licença* de *uso* e clique no *botão* “*Concordo*”. Na *sequência*, o *processo* de *importação* será *realizado*



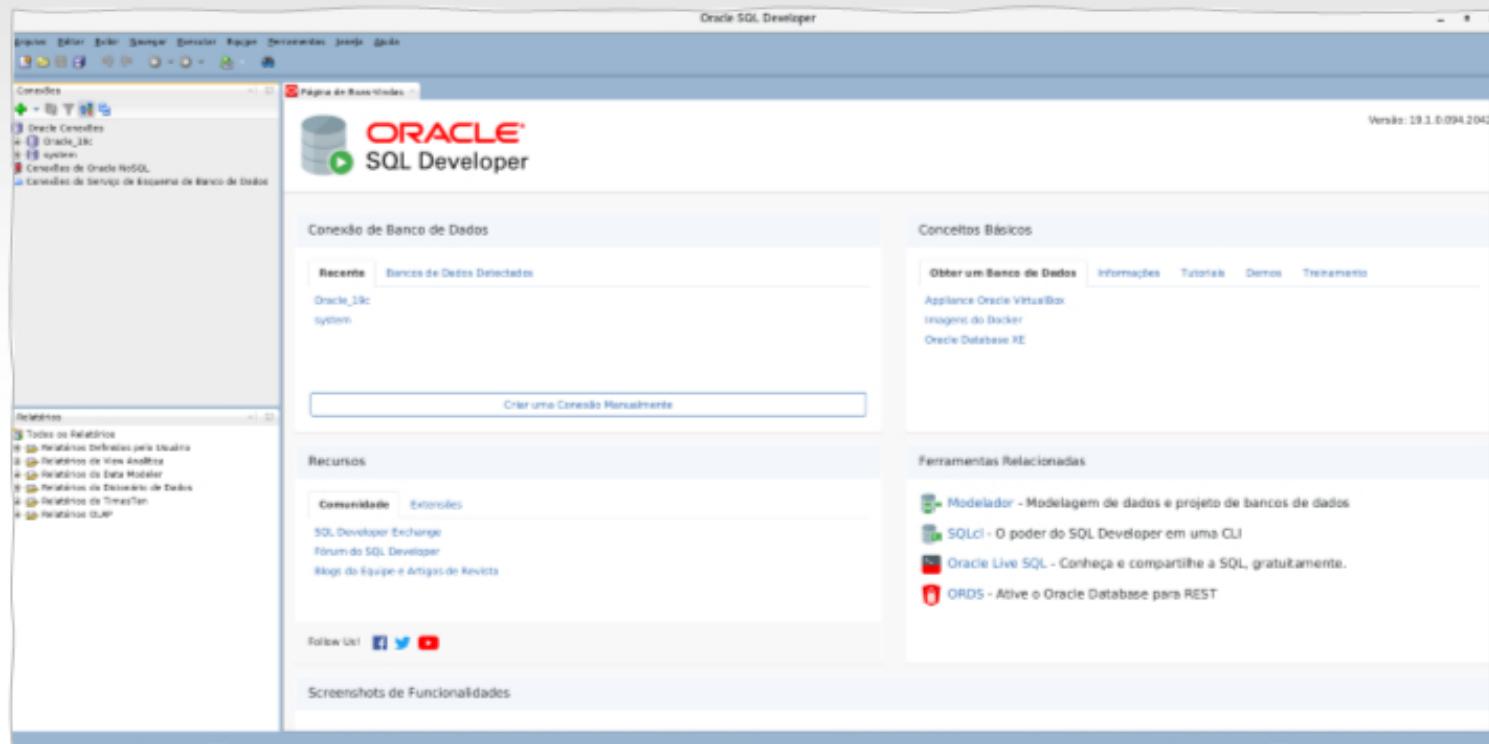
Importação do Appliance

7. Finalizando o processo de importação, você já pode iniciar a máquina virtual. Dê um duplo clique no item Oracle Developer Days



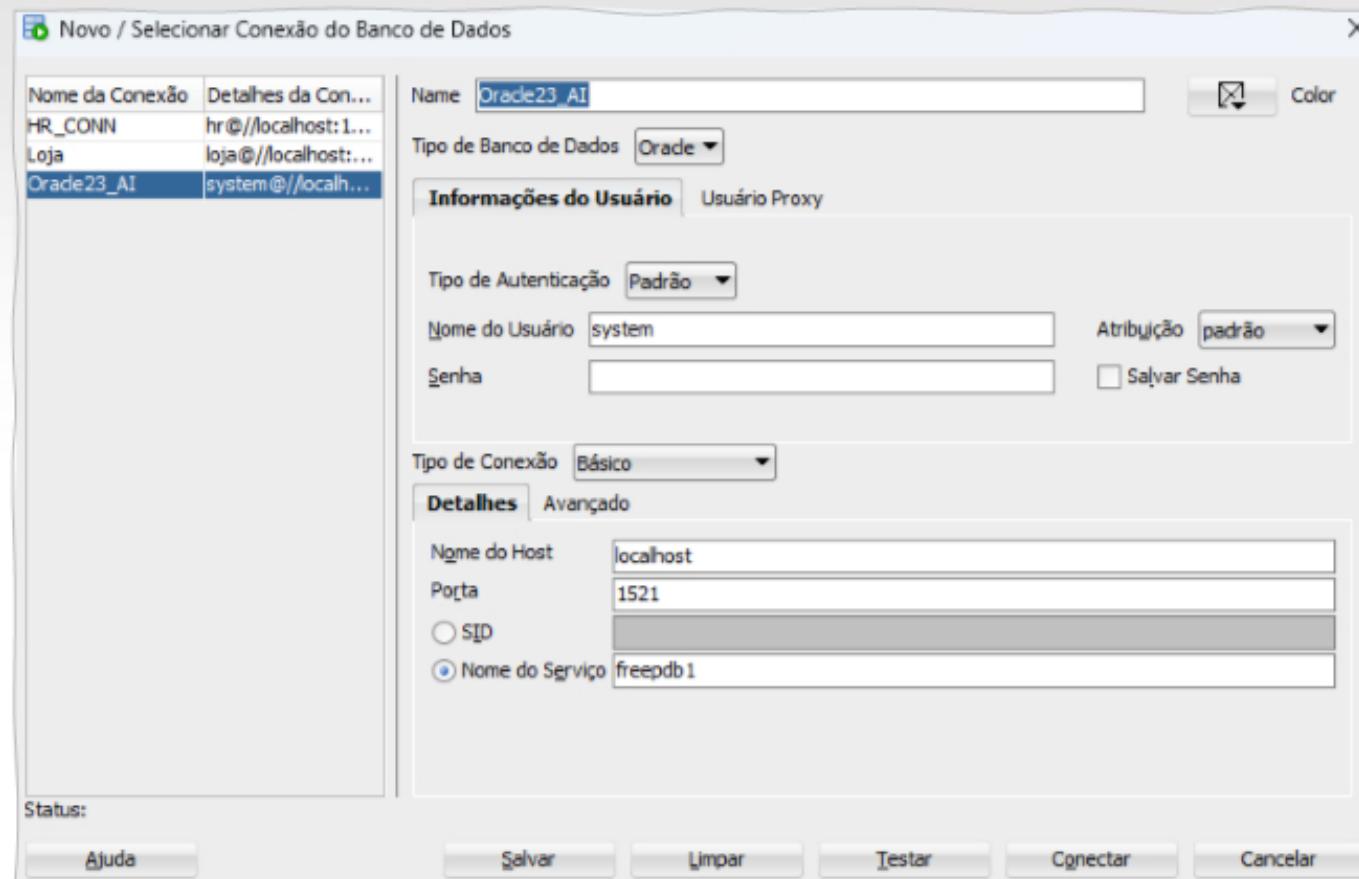
Importação do Appliance

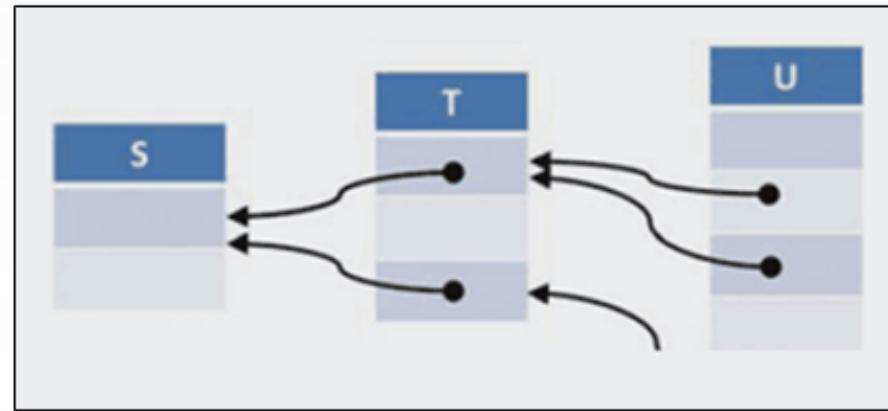
8. Para *realizar* a *autenticação*, utilize *username* (*system*) / *password* (*oracle*)



Importação do Appliance

8. Para *realizar* a *autenticação*, utilize *username* (*system*) / *password* (*oracle*)





BANCO DE DADOS RELACIONAL

Banco de Dados Relacional

- *O que é um Banco de Dados Relacional?*
 - *Originalmente desenvolvido* em **1970**
 - Dr. E. F. **Codd**
 - *Paper (A Relational Model of Data for Large Shared Data Banks)*
 - É *uma coleção de dados relacionados, organizados* em **tabelas**

- *Software* utilizado para *acessar* as *informações armazenadas* em um *Banco de Dados*
- *Exemplos:*
 - *Oracle*
 - *PostgreSQL*
 - *MySQL*
 - *SQL Server*
 - *DB2*
 - Etc.



Linguagem SQL

- *Linguagem comum*
- *Estendida para a maioria dos Bancos de Dados*
- *Permite:*
 - *Recuperar*
 - *Adicionar*
 - *Modificar*
 - *Excluir*



Informação

Structured Query Language

- *Linguagem padrão* projetada para *acessar BD Relacional*
- *Primeira implementação:*
 - IBM – 1970
 - Projeto denominado *System-R*
- *Atualmente* é uma *linguagem padronizada* e *reconhecida* pela American National Standards Institute (**ANSI**)

Structured Query Language

- *Existem 05 (cinco) tipos de instruções SQL:*
 1. *Instruções de consultas: recuperam linhas armazenadas nas tabelas do BD (SELECT)*
 2. *Instruções DML (Data Manipulation Language)*
 - *Modificam o conteúdo das tabelas*
 - *INSERT: adiciona* linhas
 - *UPDATE: altera* linhas
 - *DELETE: remove* linhas

Structured Query Language

- *Existem 05 (cinco) tipos de instruções SQL:*

3. Instruções DDL (Data Definition Language)

- *Definem as estruturas de dados (objetos)*
- *CREATE: cria uma estrutura de banco de dados:*
 - ***CREATE TABLE***
 - ***CREATE USER***
 - ***CREATE DATABASE***
 - Etc.
- *ALTER: modifica uma estrutura de banco de dados:*
 - ***ALTER TABLE***
 - ***ALTER DATABASE***
 - Etc.

Structured Query Language

- Existem 05 (cinco) tipos de *instruções SQL*:

3. Instruções DDL (Data Definition Language)

- **DROP:** remove uma *estrutura* de *banco de dados*:
 - **DROP TABLE**
 - **DROP USER**
 - Etc.
- **RENAME:** altera o *nome* de um *objeto*
- **TRUNCATE:** exclui *todas* as *linhas* de uma *tabela*

Structured Query Language

- *Existem 05 (cinco) tipos de instruções SQL:*

4. Instruções TC (Transaction Control)

- *Registram permanentemente as alterações feitas nos dados ou, desfazem essas alterações*
 - *COMMIT: registra permanentemente as alterações realizadas nos dados*
 - *ROLLBACK: desfaz as alterações feitas nos dados*
 - *SAVEPOINT: define um “ponto de salvamento” no qual você pode reverter às alterações*

Structured Query Language

- *Existem 05 (cinco) tipos de instruções SQL:*

5. Instruções DCL (Data Control Language)

- *Alteram as permissões nas estrutura de banco de dados*
- *GRANT: concede a outro usuário acesso às estruturas do banco de dados*
- *REVOKE: impede que outro usuário acesse as estruturas do banco de dados*

Structured Query Language

- *Existem outras maneiras de acessar os registros oriundos do banco de dados:*
 - *Oracle Forms e Reports*
 - *SQL Embutidas:*
 - C++
 - Java
 - C#
 - PHP
 - Etc.



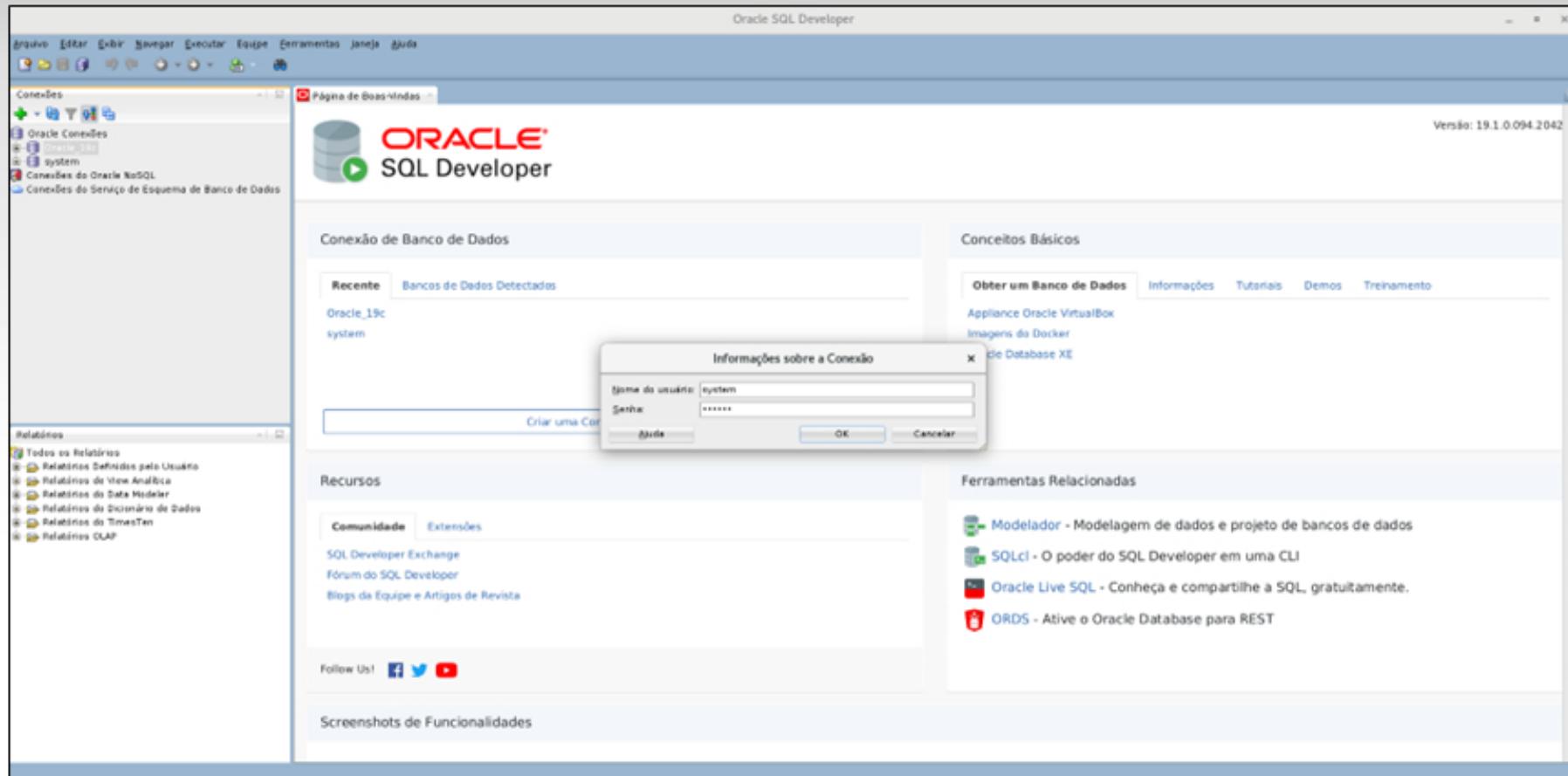
*Através de um
drive de conexão*

SQL Developer

- *Interface gráfica* a qual **permite**:
 - *Inserir instruções SQL*
 - *Examinar tabelas*
 - *Executar scripts*
 - *Editar e depurar código PL/SQL*
 - Etc.
- *Executado nas plataformas:*
 - *Windows*
 - *Linux*
 - *Mac OS X*



SQL Developer



Tipos de Dados (Oracle)

- ***char(comprimento)***
 - armazena *string* de *comprimento fixo*
 - parâmetro *comprimento* especifica o *comprimento* do *string*
 - *string* de *comprimento menor* será *preenchido* com *espaços no final*
- ***varchar2(comprimento)***
 - armazena *string* de *comprimento variável*
 - parâmetro *comprimento* especifica o *comprimento máximo* do *string*

Tipos de Dados (Oracle)

- *date*
 - *Armazena data e hora*
- *integer*
 - *Armazena valores inteiros*
 - *Valor inteiro não contém um ponto flutuante (trata-se de número inteiro, como 1, 10 e 115)*

Tipos de Dados (Oracle)

- *number(precisão, escala)*
 - armazena números de *ponto flutuante*, todavia, também pode ser *usado* para *armazenar valores inteiros*
 - precisão: número máximo de *dígitos* (à *esquerda* e à *direita* de um *ponto decimal*)
 - escala: número máximo de *dígitos* à *direita* de um *ponto decimal*

Tipos de Dados (Oracle)

- ***binary_float***
 - *lançado no Oracle Database 10g*
 - *armazena um número de ponto flutuante de 32 bits e precisão simples*
- ***binary_double***
 - *introduzido no Oracle Database 10g*
 - *armazena um número de ponto flutuante de 64 bits e precisão dupla*

Criando o Esquema

- *Loja imaginária*
- *Venda de livros, vídeos, DVDs e CDs*
- *Contém informações sobre clientes, funcionários, produtos e vendas*



Data Definition Language

- *Utilizada para criar objetos e estruturas*
- *Criando um usuário do BD (loja imaginária)*
 - Para *criar* um *usuário* no *BD*, utilize a *instrução CREATE USER*
 - *Sintaxe:*
CREATE USER nome_usuário IDENTIFIED BY senha;
 - *Onde:*
 - *nome_usuário*: corresponde a *identificação* do *usuário*
 - *senha*: é a *senha correspondente* ao *usuário*

Data Definition Language

- ***Exemplo:***

- *Criar um usuário identificado de “loja” com a senha correspondente a “loja_senha”*

```
CREATE USER loja IDENTIFIED BY loja_senha;
```

Data Control Language

- **Exemplo:**

- Atribuindo permissões para o usuário *loja* (**GRANT**)
- O **usuário** deve ser *capaz* de **conectar** no *BD* (**permissão connect**)
- Criar *objetos* como *tabelas* (**permissão resource**)
- *Observação:* as **permissões** são *concedidas* por um **usuário privilegiado** (**usuário system / sys**)

GRANT *connect, resource* **TO** *loja*;

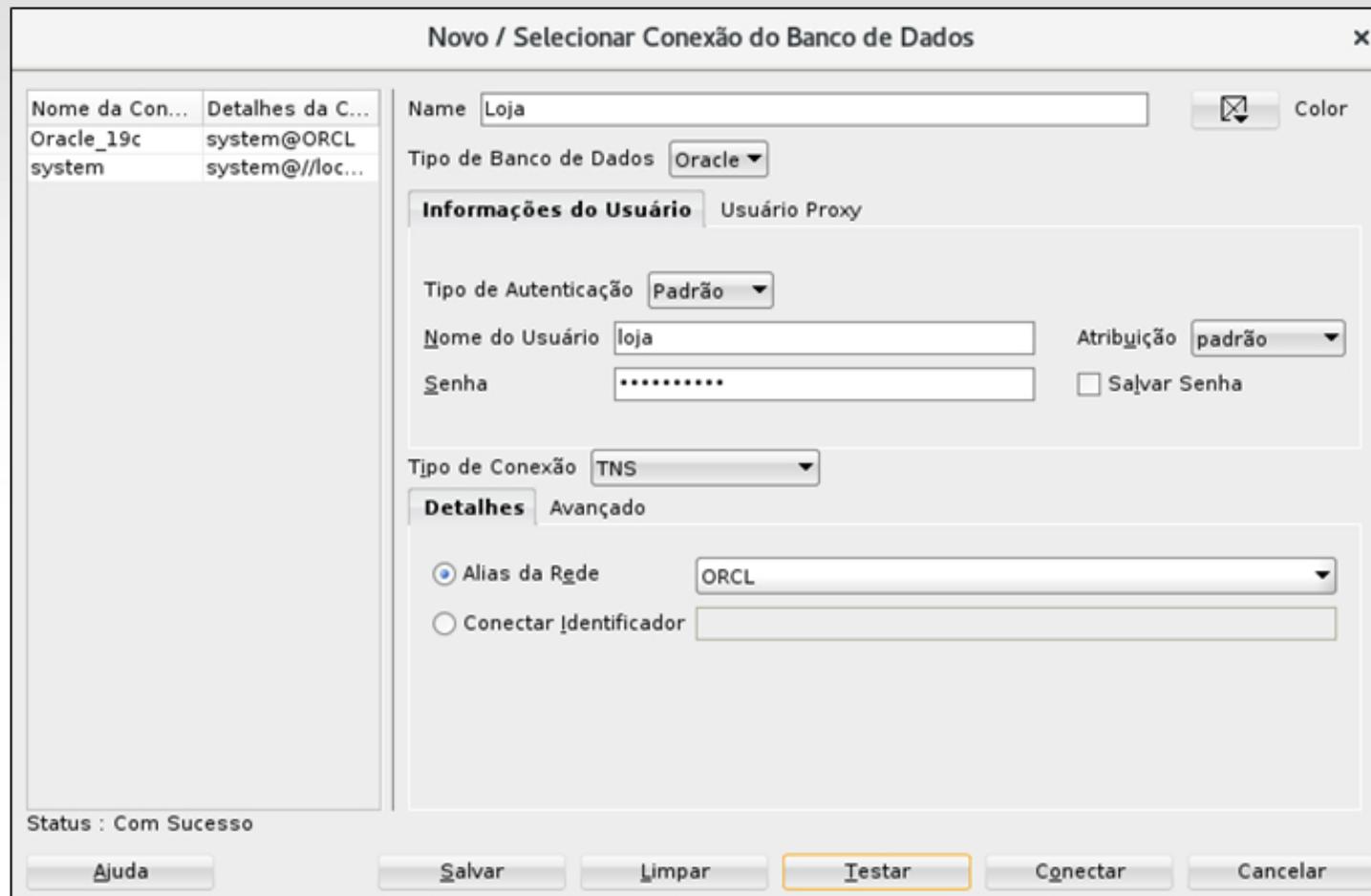
-- Only Oracle 19c/23ai

ALTER USER *loja* **QUOTA** *100M* **ON** **USERS**;



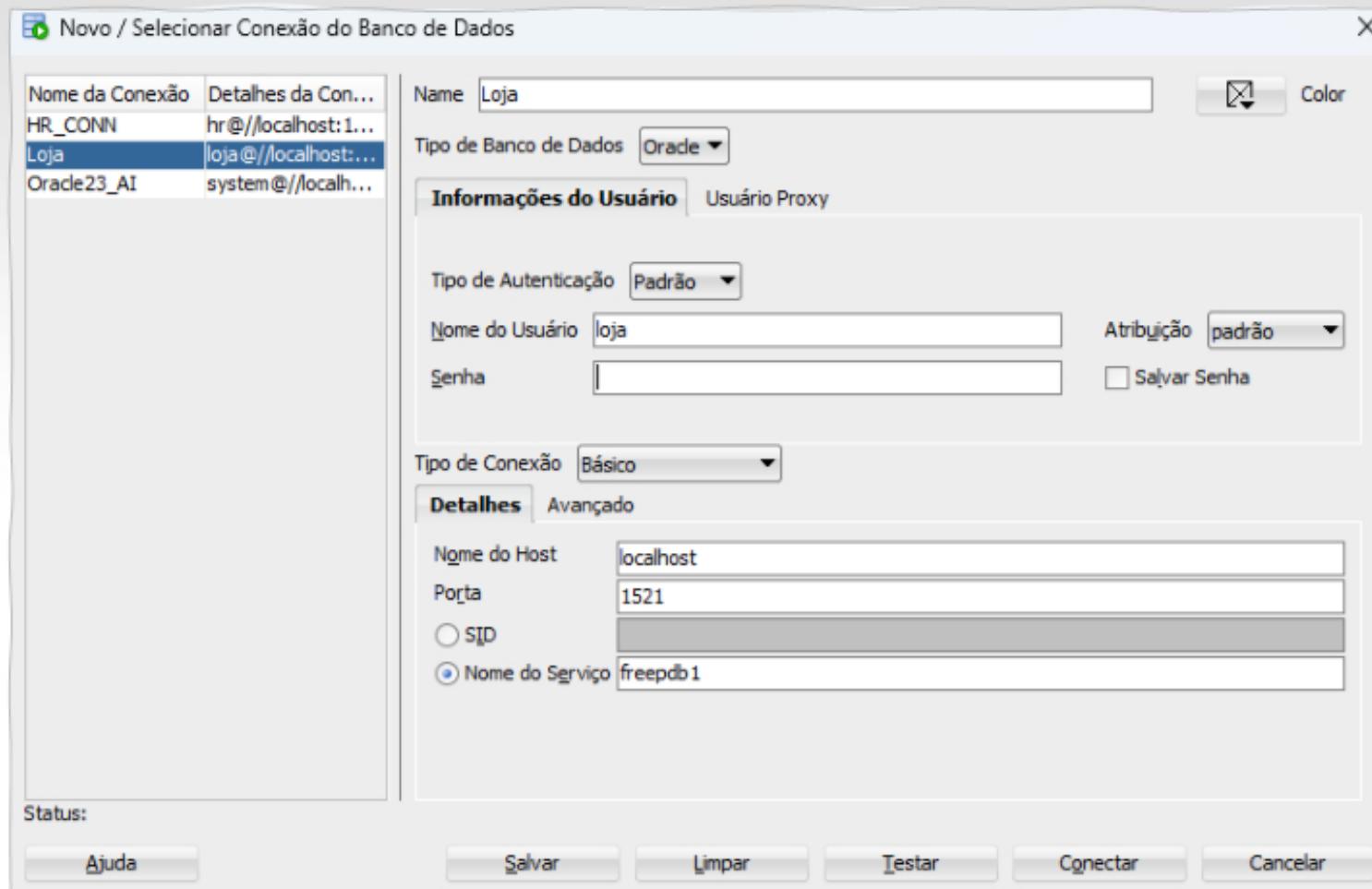
Data Control Language

- **New Connection (Oracle 19c):**



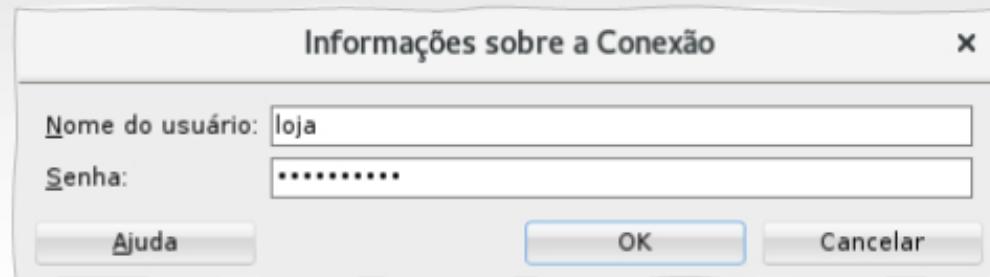
Data Control Language

- **New Connection (Oracle 23ai):**



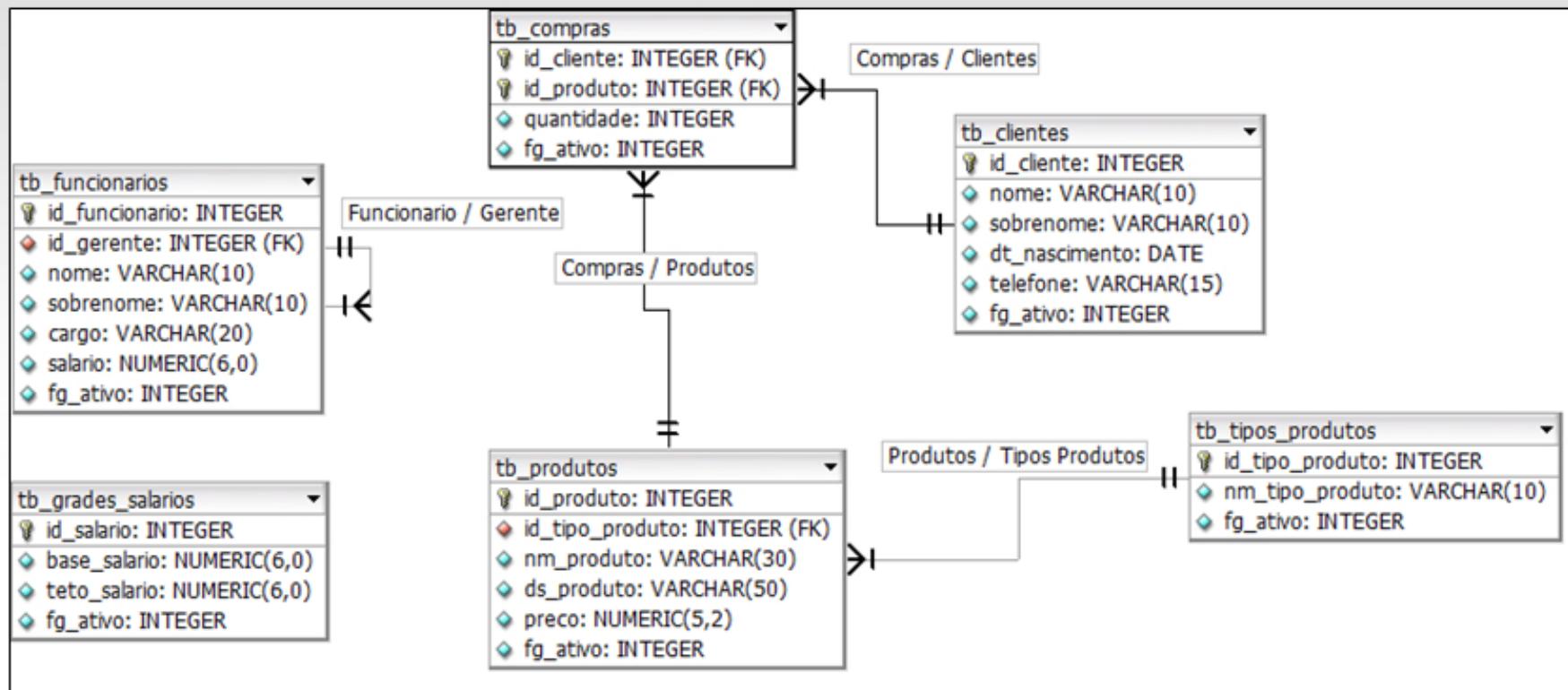
Data Control Language

- ***Login / New Connection:***



Criando o Esquema (Loja)

- *Esquema (Loja):*



Criando o Esquema (Loja)

- *Examinando os objetos (table) do esquema Loja*
 - *Informações mantidas no esquema “loja” incluem:*
 - *Detalhes do cliente*
 - *Tipos de produtos vendidos*
 - *Detalhes do produto*
 - Um *histórico* dos *produtos adquiridos* pelos *clientes*
 - *Funcionários da loja*
 - *Níveis salariais*



Criando o Esquema (Loja)

- *Informações armazenadas:*
 - *Clientes:* contém detalhes dos *clientes*
 - *Tipos de Produtos:* contém os *tipos de produtos vendidos*
 - *Produtos:* contém detalhes dos *produtos*
 - *Compras:* produtos *adquiridos* pelos *clientes*
 - *Funcionários:* contém *detalhes* dos *funcionários*
 - *Grades Salariais:* contém *detalhes* dos *níveis salariais*



DDL (DATA DEFINITION LANGUAGE)

Criando os Objetos

- ***Exemplo:***

- ***Tabela tb_clientes***

```
CREATE TABLE tb_clientes(  
    id_cliente      INTEGER,  
    nome            VARCHAR2(10) NOT NULL,  
    sobrenome       VARCHAR2(10) NOT NULL,  
    dt_nascimento   DATE,  
    telefone        VARCHAR2(12),  
    fg_ativo        INTEGER,  
    CONSTRAINT pk_tb_clientes_id_cliente PRIMARY KEY(id_cliente)  
);
```

Criando os Objetos

- *Exemplo:*

- *Tabela tb_tipos_produtos*

```
CREATE TABLE tb_tipos_produtos(  
    id_tipo_produto      INTEGER ,  
    nm_tipo_produto      VARCHAR2(10) NOT NULL,  
    fg_ativo              INTEGER,  
    CONSTRAINT pk_tb_tipos_produtos_id_tipo_produto  
        PRIMARY KEY(id_tipo_produto)  
);
```

Criando os Objetos

- *Exemplo:*
 - *Tabela tb_produtos*

```
CREATE TABLE tb_produtos(  
    id_produto          INTEGER,  
    id_tipo_produto     INTEGER,  
    nm_produto          VARCHAR2(30) NOT NULL,  
    ds_produto          VARCHAR2(50),  
    preco                NUMBER(5,2),  
    fg_ativo             INTEGER,  
    CONSTRAINT pk_tb_produtos_id_produto PRIMARY KEY(id_produto),  
    CONSTRAINT fk_tb_produtos_id_tipo_produto  
        FOREIGN KEY(id_tipo_produto)  
        REFERENCES tb_tipos_produtos(id_tipo_produto));
```

Criando os Objetos

- *Exemplo:*

- *Tabela tb_compras*

```
CREATE TABLE tb_compras(  
    id_produto      INTEGER,  
    id_cliente      INTEGER,  
    quantidade      INTEGER,  
    fg_ativo        INTEGER,  
    CONSTRAINT fk_tb_compras_id_produto FOREIGN KEY(id_produto)  
        REFERENCES tb_produtos(id_produto),  
    CONSTRAINT fk_tb_compras_id_cliente FOREIGN KEY(id_cliente)  
        REFERENCES tb_clientes(id_cliente),  
    CONSTRAINT pk_tb_compras_id_produto_id_cliente  
        PRIMARY KEY(id_produto, id_cliente));
```

Criando os Objetos

- *Exemplo:*

- *Tabela tb_funcionarios*

```
CREATE TABLE tb_funcionarios(  
    id_funcionario      INTEGER,  
    id_gerente          INTEGER,  
    nome                VARCHAR2(10) NOT NULL,  
    sobrenome           VARCHAR2(10) NOT NULL,  
    cargo               VARCHAR2(20),  
    salario             NUMBER(8,2),  
    fg_ativo            INTEGER,  
    CONSTRAINT pk_tb_funcionarios_id_func PRIMARY KEY(id_funcionario),  
    CONSTRAINT fk_tb_funcionarios_id_gerente FOREIGN KEY(id_gerente)  
        REFERENCES tb_funcionarios(id_funcionario));
```

Criando os Objetos

- *Exemplo:*
 - *Tabela tb_grades_salarios*

```
CREATE TABLE tb_grades_salarios(  
    id_salario      INTEGER,  
    base_salario    NUMBER(8,2),  
    teto_salario    NUMBER(8,2),  
    fg_ativo        INTEGER,  
    CONSTRAINT pk_tb_grades_salarios_id_salario  
        PRIMARY KEY(id_salario)  
);
```



DML (DATA MANIPULATION LANGUAGE)

Adicionando linhas/tuplas

- Instrução ***INSERT***:
 - ***Utilizada*** para ***adicionar linhas*** em ***tabelas*** do ***BD***
 - Podemos ***tornar*** as ***operações permanentes*** usando a ***instrução COMMIT*** e ou ***desfazê-las*** com a ***instrução ROLLBACK***
 - Em ***uma instrução INSERT***, podemos ***especificar*** as ***seguintes informações***:
 - A ***tabela*** na qual a ***linha*** será ***inserida***
 - Uma ***lista*** de ***colunas*** para as quais iremos ***especificar valores***
 - Uma ***lista*** de ***valores*** a serem ***armazenados*** nas ***colunas especificadas***

Adicionando linhas/tuplas

- Comando **DESCRIBE**:
 - Permite *identificar* as *colunas definidas* como **NOT NULL**

```
DESCRIBE tb_clientes;
```

Name	Null	Type
ID_CLIENTE	NOT NULL	NUMBER(38)
NOME	NOT NULL	VARCHAR2(10)
SOBRENOME	NOT NULL	VARCHAR2(10)
DT_NASCIMENTO		DATE
TELEFONE		VARCHAR2(12)
FG_ATIVO		NUMBER(38)

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Adiciona uma linha na tabela *tb_clientes*
 - A *ordem* dos *valores* na lista **VALUES** corresponde à *ordem* na qual as *colunas* são *especificadas* na lista de *colunas*
- **Sintaxe:**

```
INSERT INTO tabela [ (campo1, campo2, ..., campo n) ]  
VALUES  
(valor_campo1, valor_campo2, ..., valor_campo n)
```

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Adiciona linhas na tabela *tb_clientes*

```
INSERT INTO tb_clientes (id_cliente, nome, sobrenome, dt_nascimento,  
                        telefone, fg_ativo)  
VALUES (1, 'John', 'Brown', '01/Jan/1965', '800-555-1211', 1);
```

```
INSERT INTO tb_clientes (id_cliente, nome, sobrenome, dt_nascimento,  
                        telefone, fg_ativo)  
VALUES (2, 'Cynthia', 'Green', '05/Feb/1968', '800-555-1212', 1);
```

```
INSERT INTO tb_clientes  
VALUES (3, 'Steve', 'White', '16/Mar/1971', '800-555-1213', 1);
```

Adicionando linhas/tuplas

- Instrução ***INSERT***:
 - Adiciona linhas na *tabela tb_clientes*

```
INSERT INTO tb_clientes (id_cliente, nome, sobrenome, dt_nascimento,  
                        telefone, fg_ativo)  
VALUES (4, 'Gail', 'Black', '', '800-555-1214', 1);
```

```
INSERT INTO tb_clientes (id_cliente, nome, sobrenome, dt_nascimento,  
                        telefone, fg_ativo)  
VALUES (5, 'Doreen', 'Blue', '20/May/1970', NULL, 1);
```

```
INSERT INTO tb_clientes (id_cliente, nome, sobrenome, dt_nascimento,  
                        telefone, fg_ativo)  
VALUES (6, 'Fred', 'Brown', '01/Jan/1970', '800-555-1215', 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Verificando as *linhas inseridas* na *tb_clientes*

```
SELECT *  
FROM tb_clientes;
```

ID_CLIENTE	NOME	SOBRENOME	DT_NASCIMENTO	TELEFONE	FG_ATIVO
1	John	Brown	01-JAN-65	800-555-1211	1
2	Cynthia	Green	05-FEB-68	800-555-1212	1
3	Steve	White	16-MAR-71	800-555-1213	1
4	Gail	Black	(null)	800-555-1214	1
5	Doreen	Blue	20-MAY-70	(null)	1
6	Fred	Brown	01-JAN-70	800-555-1215	1

Adicionando linhas/tuplas

- Instrução **INSERT**:

– *Adiciona linhas na tabela tb_tipos_produtos*

DESCRIBE tb_tipos_produtos;

```
INSERT INTO tb_tipos_produtos(id_tipo_produto, nm_tipo_produto,  
                                fg_ativo)
```

```
VALUES (1, 'Book', 1);
```

```
INSERT INTO tb_tipos_produtos(id_tipo_produto, nm_tipo_produto,  
                                fg_ativo)
```

```
VALUES (2, 'Video', 1);
```

```
INSERT INTO tb_tipos_produtos(id_tipo_produto, nm_tipo_produto,  
                                fg_ativo)
```

```
VALUES (3, 'DVD', 1);
```

Adicionando linhas/tuplas

- Instrução ***INSERT***:

– *Adiciona linhas na tabela tb_tipos_produtos*

```
INSERT INTO tb_tipos_produtos(id_tipo_produto, nm_tipo_produto,  
                               fg_ativo)
```

```
VALUES (4, 'CD', 1);
```

```
INSERT INTO tb_tipos_produtos(id_tipo_produto, nm_tipo_produto,  
                               fg_ativo)
```

```
VALUES (5, 'Magazine', 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - *Verificando as linhas inseridas na tb_tipos_produtos*

```
SELECT *  
FROM tb_tipos_produtos;
```

ID_TIPO_PRODUTO	NM_TIPO_PRODUTO	FG_ATIVO
1	Book	1
2	Video	1
3	DVD	1
4	CD	1
5	Magazine	1

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Adiciona linhas na tabela tb_produtos

```
DESCRIBE tb_produtos;
```

```
INSERT INTO tb_produtos(id_produto, id_tipo_produto, nm_produto,  
ds_produto, preco, fg_ativo)
```

```
VALUES (1, 1, 'Modern Science', 'A description of modern science',  
19.95, 1);
```

```
INSERT INTO tb_produtos(id_produto, id_tipo_produto, nm_produto,  
ds_produto, preco, fg_ativo)
```

```
VALUES (2, 1, 'Chemistry', 'Introduction to Chemistry', 30, 1);
```

Adicionando linhas/tuplas

- Instrução ***INSERT***:
 - Adiciona linhas na *tabela tb_produtos*

```
INSERT INTO tb_produtos(id_produto, id_tipo_produto, nm_produto,  
                      ds_produto, preco, fg_ativo)  
VALUES (3, 2, 'Supernova', 'A star explodes', 25.99, 1);
```

```
INSERT INTO tb_produtos(id_produto, id_tipo_produto, nm_produto,  
                      ds_produto, preco, fg_ativo)  
VALUES (4, 2, 'Tank War', 'Action movie about a future war', 13.95, 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Verificando as *linhas inseridas* na *tb_produtos*

```
SELECT *  
FROM tb_produtos;
```

ID_PRODUTO	ID_TIPO_PRODUTO	NM_PRODUTO	DS_PRODUTO	PRECO	FG_ATIVO
1	1	Modern Science	A description of modern science	19.95	1
2	1	Chemistry	Introduction to Chemistry	30	1
3	2	Supernova	A star explodes	25.99	1
4	2	Tank War	Action movie about a future war	13.95	1

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Adiciona linhas na tabela *tb_compras*

DESCRIBE tb_compras;

```
INSERT INTO tb_compras(id_cliente, id_produto, quantidade, fg_ativo)
VALUES (1, 1, 1, 1);
```

```
INSERT INTO tb_compras(id_cliente, id_produto, quantidade, fg_ativo)
VALUES (2, 1, 3, 1);
```

Adicionando linhas/tuplas

- Instrução ***INSERT***:

– *Adiciona linhas na tabela tb_compras*

```
INSERT INTO tb_compras(id_cliente, id_produto, quantidade, fg_ativo)  
VALUES (1, 4, 1, 1);
```

```
INSERT INTO tb_compras(id_cliente, id_produto, quantidade, fg_ativo)  
VALUES (2, 2, 1, 1);
```

```
INSERT INTO tb_compras(id_cliente, id_produto, quantidade, fg_ativo)  
VALUES (1, 3, 1, 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução ***INSERT***:
 - *Verificando as linhas inseridas na tb_compras*

```
SELECT *  
FROM tb_compras;
```

ID_PRODUTO	ID_CLIENTE	QUANTIDADE	FG_ATIVO
1	1	1	1
1	2	3	1
4	1	1	1
2	2	1	1
3	1	1	1

Adicionando linhas/tuplas

- Instrução **INSERT**:

– *Adiciona linhas na tabela tb_funcionarios*

DESCRIBE tb_funcionarios;

```
INSERT INTO tb_funcionarios (id_funcionario, id_gerente, nome,
sobrenome, cargo, salario, fg_ativo)
VALUES (1, NULL, 'James', 'Smith', 'CEO', 8000.00, 1);
```

```
INSERT INTO tb_funcionarios (id_funcionario, id_gerente, nome,
sobrenome, cargo, salario, fg_ativo)
VALUES (2, 1, 'Ron', 'Johnson', 'Sales Manager', 6000.00, 1);
```

Adicionando linhas/tuplas

- Instrução **INSERT**:

– *Adiciona linhas na tabela tb_funcionarios*

```
INSERT INTO tb_funcionarios (id_funcionario, id_gerente, nome,  
                           sobrenome, cargo, salario, fg_ativo)
```

```
VALUES (3, 2, 'Fred', 'Hobbs', 'Salesperson', 1500.00, 1);
```

```
INSERT INTO tb_funcionarios (id_funcionario, id_gerente, nome,  
                           sobrenome, cargo, salario, fg_ativo)
```

```
VALUES (4, 2, 'Susan', 'Jones', 'Salesperson', 5000.00, 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - Verificando as *linhas inseridas* na *tb_funcionarios*

```
SELECT *  
FROM tb_funcionarios;
```

ID_FUNCIONARIO	ID_GERENTE	NOME	SOBRENOME	CARGO	SALARIO	FG_ATIVO
1	(null)	James	Smith	CEO	8000	1
2	1	Ron	Johnson	Sales Manager	6000	1
3	2	Fred	Hobbs	Salesperson	1500	1
4	2	Susan	Jones	Salesperson	5000	1

Adicionando linhas/tuplas

- Instrução **INSERT**:

- *Adiciona linhas na tabela tb_grades_salarios*

```
DESCRIBE tb_grades_salarios;
```

```
INSERT INTO tb_grades_salarios (id_salario, base_salario, teto_salario,  
                                fg_ativo)
```

```
VALUES (1, 1, 2500.00, 1);
```

```
INSERT INTO tb_grades_salarios (id_salario, base_salario, teto_salario,  
                                fg_ativo)
```

```
VALUES (2, 2500.01, 5000.00, 1);
```

Adicionando linhas/tuplas

- Instrução ***INSERT***:

– *Adiciona linhas na tabela tb_grades_salarios*

```
INSERT INTO tb_grades_salarios (id_salario, base_salario, teto_salario,  
                                fg_ativo)
```

```
VALUES (3, 5000.01, 7500.00, 1);
```

```
INSERT INTO tb_grades_salarios (id_salario, base_salario, teto_salario,  
                                fg_ativo)
```

```
VALUES (4, 7500.01, 9999.99, 1);
```

COMMIT;

Adicionando linhas/tuplas

- Instrução **INSERT**:
 - *Verificando as linhas inseridas na tb_grades_salarios*

```
SELECT *  
FROM tb_grades_salarios;
```

ID_SALARIO	BASE_SALARIO	TETO_SALARIO	FG_ATIVO
1	1	2500	1
2	2500,01	5000	1
3	5000,01	7500	1
4	7500,01	9999,99	1

Alterando linhas/tuplas

- Instrução **UPDATE**:
 - Altera *linhas* em *tabelas* do *BD*
 - Em uma *instrução UPDATE*, podemos *especificar* as seguintes *informações*:
 - A *tabela* que *contém* as *linhas* a *serem alteradas*
 - Uma *cláusula WHERE* especificando as *linhas* a *serem alteradas*
 - Uma *lista de nomes* de *colunas*, junto com seus *novos valores*, *especificado* com a *cláusula SET*

Alterando linhas/tuplas

- Instrução ***UPDATE***:
 - **Sintaxe:**

```
UPDATE tabela  
    SET campo1 = valor_campo1, campo2 = valor_campo2,  
        ...  
    [ WHERE <condição> ]
```

Alterando linhas/tuplas

- **Exemplo:**

- *Atualizar o sobrenome para “Orange” do cliente cujo ID corresponda ao número 2*

- **Cuidado:**

- *Esquecer de adicionar a cláusula WHERE, todas as linhas serão atualizadas*

- **Observação:**

- **ROLLBACK** desfaz as alterações feitas nas linhas

Alterando linhas/tuplas

- **Exemplo:**

- *Atualizar o sobrenome* para “*Orange*” do *cliente* cujo *ID* corresponda ao *número 2*

```
UPDATE tb_clientes
SET sobrenome = 'Orange'
WHERE id_cliente = 2;
```

```
SELECT *
FROM tb_clientes;
```

```
ROLLBACK;
```

ID_CLIENTE	NOME	SOBRENOME	DT_NASCIMENTO	TELEFONE	FG_ATIVO
1	John	Brown	01-JAN-65	800-555-1211	1
2	Cynthia	Green	05-FEB-68	800-555-1212	1
3	Steve	White	16-MAR-71	800-555-1213	1
4	Gail	Black	(null)	800-555-1214	1
5	Doreen	Blue	20-MAY-70	(null)	1
6	Fred	Brown	01-JAN-70	800-555-1215	1

Removendo linhas/tuplas

- Instrução ***DELETE***:
 - Remove *linhas* em *tabelas* do *BD*
 - A ***cláusula WHERE*** limita as *linhas* que se *deseja excluir*, caso ***contrário***, todas as *linhas* serão *excluídas* da *tabela*

- ***Sintaxe:***

```
DELETE  
FROM {tabela}  
[ WHERE <condição> ]
```

Removendo linhas/tuplas

- *Exemplo:*
 - *Remover o cliente que possui o ID equivalente a 2*

ID_CLIENTE	NOME	SOBRENOME	DT_NASCIMENTO	TELEFONE	FG_ATIVO
1	John	Brown	01-JAN-65	800-555-1211	1
2	Cynthia	Green	05-FEB-68	800-555-1212	1
3	Steve	White	16-MAR-71	800-555-1213	1
4	Gail	Black	(null)	800-555-1214	1
5	Doreen	Blue	20-MAY-70	(null)	1
6	Fred	Brown	01-JAN-70	800-555-1215	1

```
DELETE  
FROM tb_clientes  
WHERE id_cliente = 2;
```

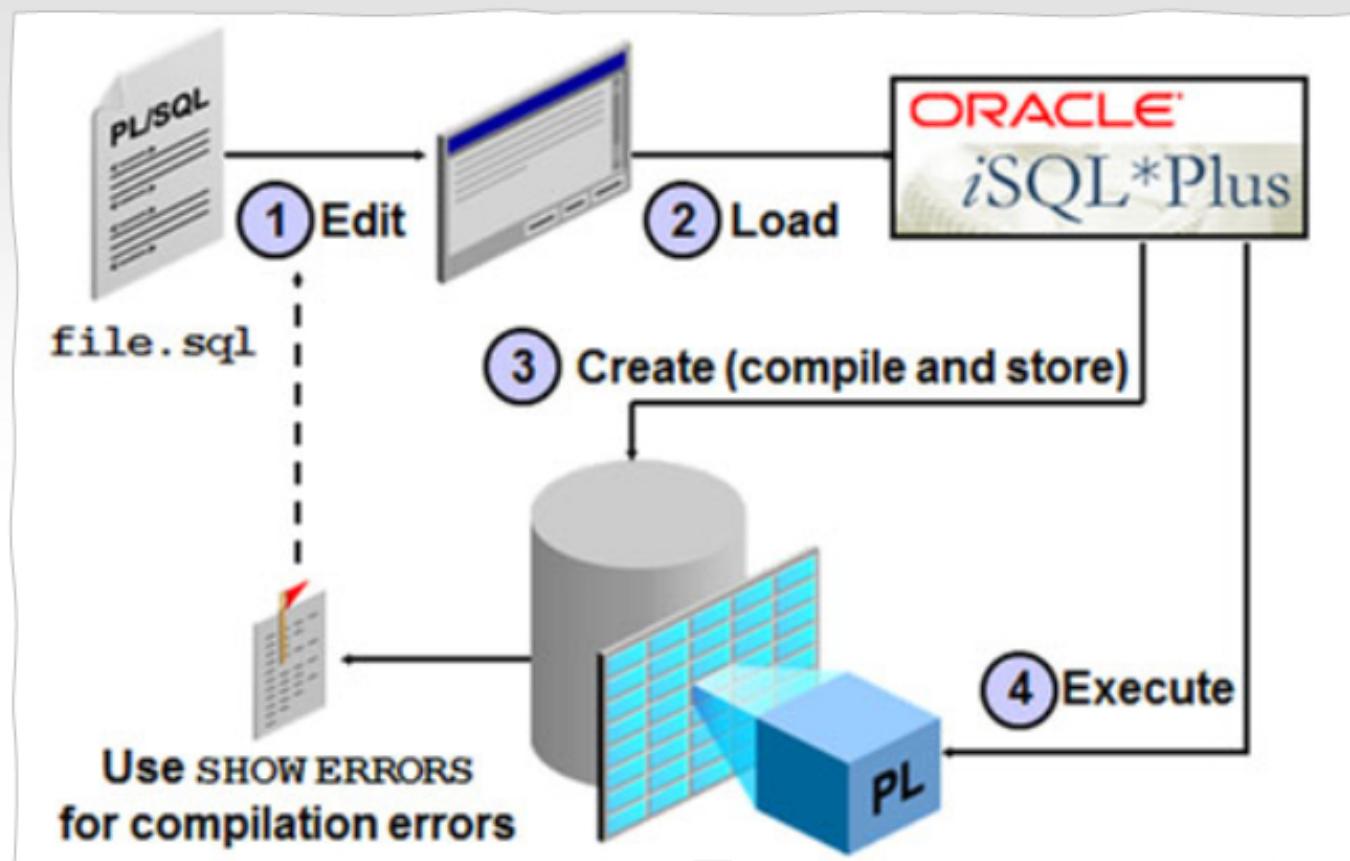
COMMIT;

Introdução ao PL/SQL

- **PL/SQL:**
 - *Linguagem procedural* da *Oracle*
 - *Permite adicionar programação* em *torno* de *instruções SQL*
 - *Usado principalmente* para *criação* de *procedures* e *functions* em um *BD*
- **Código PL/SQL:**
 - *Declaração* de *variáveis*
 - *Lógica condicional* (*if-then-else*, etc.)
 - *Loops*
 - *Procedures* e *functions*

Introdução ao PL/SQL

- **Benefícios:**



Introdução ao PL/SQL

- *Exemplo (01):*
 - Criar um *procedure nomeada* de “*get_cliente*”
 - O *ID* do *cliente* deverá ser *passado* como *parâmetro*
 - Caso o *cliente* **não** seja *encontrado*, o *procedure* deverá *exibir* uma *mensagem informativa*

Introdução ao PL/SQL

- *Exemplo (01):*

```
CREATE OR REPLACE PROCEDURE get_cliente(
    p_id_cliente IN tb_clientes.id_cliente%TYPE)
AS
    v_nome        tb_clientes.nome%TYPE;
    v_sobrenome   tb_clientes.sobrenome%TYPE;
    v_controle    INTEGER;

BEGIN
    SELECT COUNT(*) INTO v_controle
    FROM tb_clientes
    WHERE id_cliente = p_id_cliente;
```

continua....

Introdução ao PL/SQL

- ***Exemplo (01):***

```
IF (v_controle = 1) THEN  
    SELECT nome, sobrenome INTO v_nome, v_sobrenome  
    FROM tb_clientes  
    WHERE id_cliente = p_id_cliente;
```

```
DBMS_OUTPUT.PUT_LINE('Cliente localizado: ' || v_nome || ''  
                     || v_sobrenome);
```

```
ELSE
```

```
    DBMS_OUTPUT.PUT_LINE('Cliente NÃO localizado');
```

```
END IF;
```

continua....

Introdução ao PL/SQL

- *Exemplo (01):*

EXCEPTION

WHEN OTHERS THEN

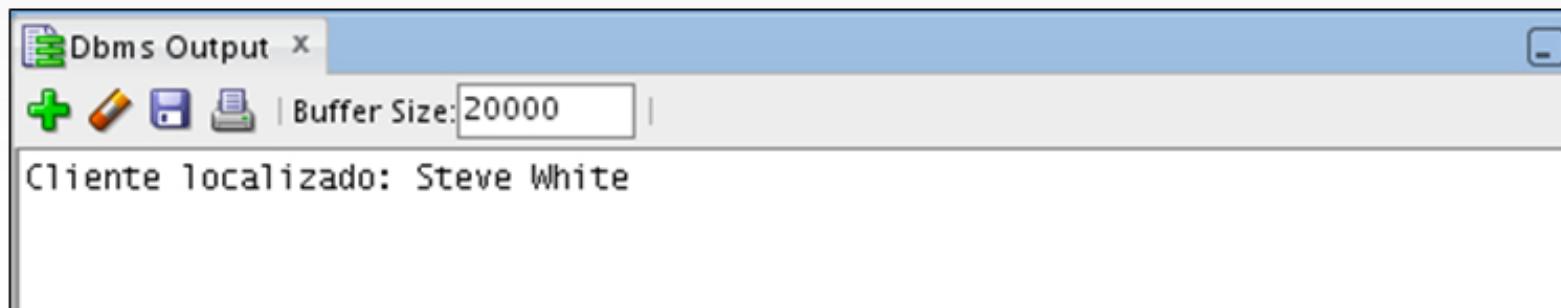
DBMS_OUTPUT.PUT_LINE('Erro');

END get_cliente;

Introdução ao PL/SQL

- *Exemplo (01):*
 - *Invocando o procedure get_cliente*

```
CALL get_cliente(3);
```



Exercícios

