
ITB – INSTITUTO TECNOLÓGICO DE BARUERI

LECO – LINGUAGEM ESTRUTURADA DE CONSULTA

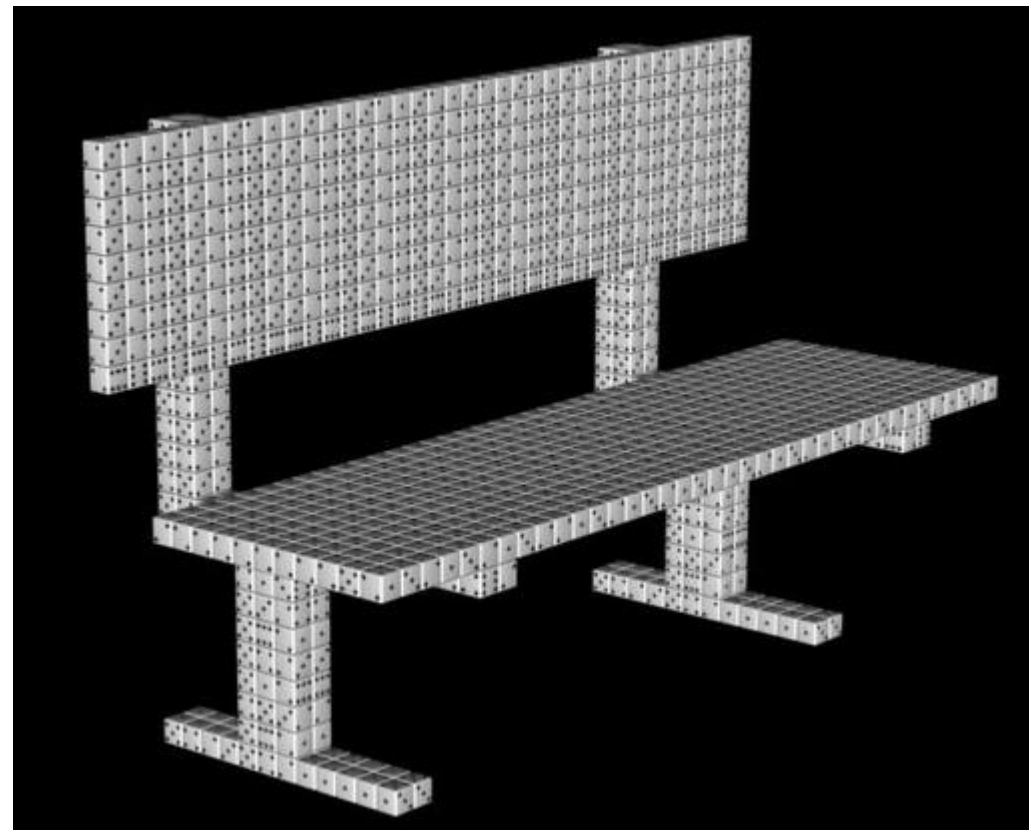
PROFESSORA VIVIANE



BANCOS DE DADOS

- Coleção de dados inter-relacionados representando informações sobre um domínio específico;
- Coleção de dados integrados que tem por objetivo atender as necessidades dos usuários;
- Conjunto de dados persistentes e manipuláveis que obedecem a um padrão de armazenamento;
- Conjunto de dados com uma estrutura regular que organizam uma informação;

Exemplos: dicionário, lista telefônica, controle do acervo de uma biblioteca, sistema de controle dos recursos humanos de uma empresa, dados pessoais de uma pessoa.



POR QUE UTILIZAR BANCOS DE DADOS?

- Compacto (elimina arquivos de papéis);
- Rapidez;
- Integrado (vários aplicativos utilizam o mesmo repositório de dados);
- Compartilhado (vários usuários podem acessar);
- Segurança (Confidencialidade, Integridade, Disponibilidade e Autenticidade);
- Padronizado;
- Suporte a transações.



SISTEMA GERENCIADOR DE BANCO DE DADOS (SGBD)

Conhecidos como SGBD, em inglês Data Base Management System - DBMS - são um conjunto de programas que permite aos usuários criar e manter um banco de dados, do qual sua principal meta é retirar da aplicação cliente a responsabilidade de gerenciar o acesso, a manipulação e a organização dos dados. O SGBD disponibiliza uma interface para que os clientes possam consultar, alterar/atualizar, incluir ou deletar os dados armazenados em um banco de dados.

Tem como característica guardar grandes informações de dados em massa, estruturando em registros e tabelas com funções para acesso e processamento das informações. Exemplos:

- Oracle;
- Microsoft SQL Server;
- DB2;
- Access;
- Dbase;
- MySQL;
- Greenplum;
- Teradata.



MODELAGEM DE DADOS

Modelagem de dados é o estudo das informações existentes em um contexto sob observação para a construção de um modelo de representação e entendimento de tal contexto.

Objetivos:

- Representar o ambiente observado
- Documentar e normalizar
- Fornecer processos de validação
- Observar processos de relacionamentos entre objetos
- Fornecer níveis de abstração de dados que omitem do usuário final detalhes sobre armazenamento dos dados (não há preocupação com um BD tecnicamente falando)



ABSTRAÇÃO

Abstração, em síntese, nada mais é do que a visão, sem conceituações técnicas, que obtemos em nossa mente de uma realidade qualquer do mundo real.

Esse primeiro modelo denominamos de minimundo, sem pensar nesse momento em automatizar ou em processar a informação desta realidade.



ABSTRAÇÃO

Quando observamos a imagem, temos o resultado de um **Processo de Abstração** em que excluimos detalhes da estrutura de uma bicicleta como os pedais, os freios, os mecanismos de tração, etc.

MODELO DE DADOS

É um conjunto de conceitos que podem ser utilizados para descrever as estruturas lógicas e físicas de um banco de dados.

ELEMENTOS DE UM MODELO DE DADOS RELACIONAL

- **Entidade** pode ser definida como qualquer coisa do mundo real , abstrata ou concreta , na qual se deseja guardar informações. (*Tabela , File, etc..*). Exemplos de entidades : Cliente , Produto , Contrato , Vendas , etc.
- **Atributos** são características de uma entidade que a descrevem detalhadamente. (*coluna , campo , etc,..*). Exemplos de atributos : Código do Produto (Entidade Produto) , Nome do Cliente (Entidade Cliente).
- **Relacionamento** pode ser entendido como uma associação entre Entidades devido a regras de negócio. Normalmente ocorre entre duas ou mais Entidades, podendo ocorrer entre mesma a Entidade (auto-relacionamento).

MODELO DE DADOS

MODELO CONCEITUAL



MODELO LÓGICO



MODELO FÍSICO



MODELO CONCEITUAL



MODELO CONCEITUAL

Representa as regras de negócio sem limitações tecnológicas ou de implementação por isto é a etapa mais adequada para o envolvimento do usuário que não precisa ter conhecimentos técnicos. Neste modelo temos

- Visão Geral do negócio;
- Facilitação do entendimento entre usuários e desenvolvedores;
- Possui somente as entidades e atributos principais.



MODELO LÓGICO



MODELO LÓGICO

Descreve em formato as estruturas que estarão no banco de dados de acordo com a abordagem da tecnologia de SGBD utilizado (hierárquico, **relacional**, em rede e orientado a objeto). Suas características são :

- Deriva do modelo conceitual e da representação do negócio;
- Possui entidades associativas em lugar de relacionamentos n:n;
- Define as chaves primárias das entidades;
- Normalização até a 3a. forma normal;
- Adequação ao padrão de nomenclatura;
- Entidades e atributos documentados.



MODELO FÍSICO



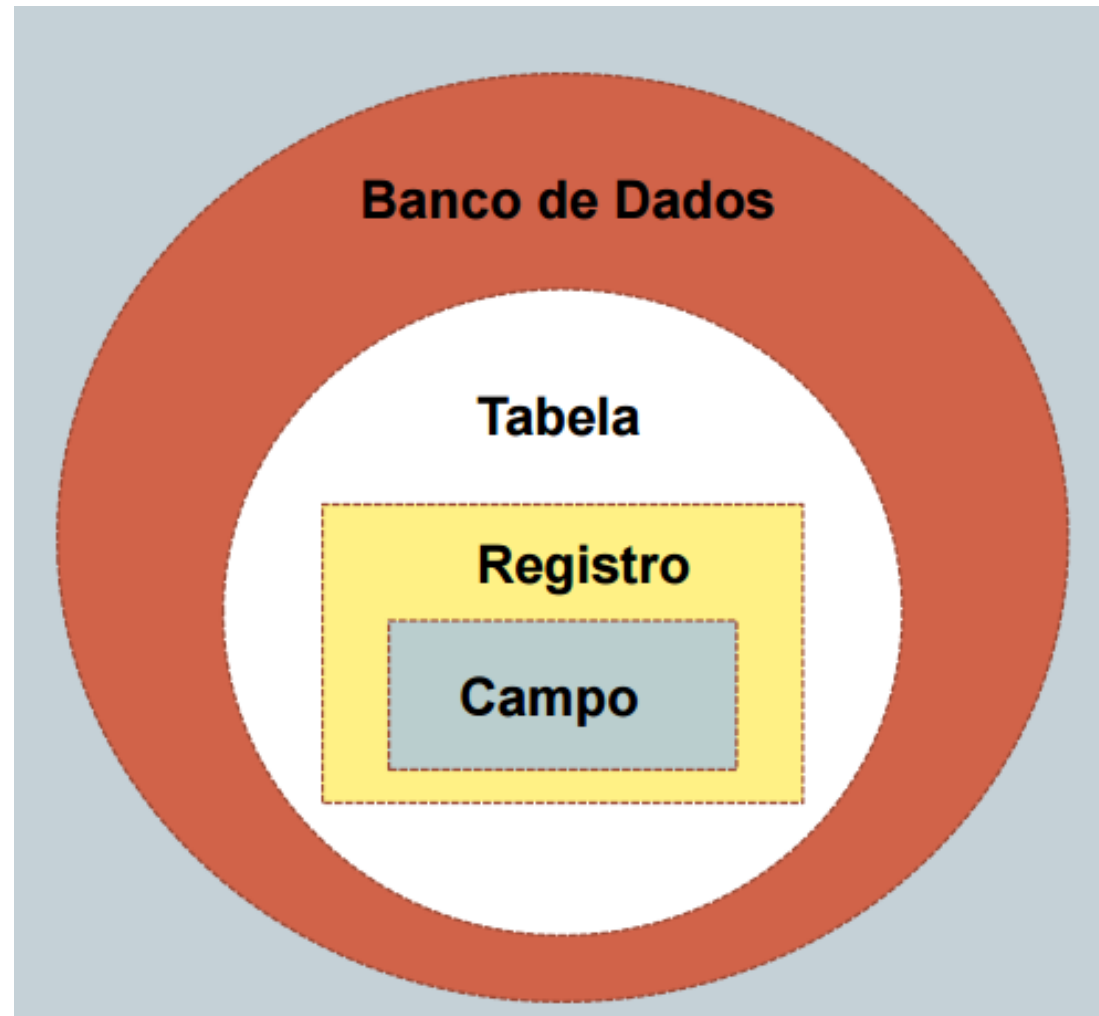
MODELO FÍSICO

Leva em consideração limites impostos pelo SGBD (Sistema Gerenciador de Banco de dados) e pelos requisitos não funcionais dos programas que acessam os dados.

- Tipos e tamanhos de campos;
- Índices;
- Nomenclaturas;
- Procedures e Triggers;
- Elaborado a partir do modelo lógico;
- Pode variar segundo o SGBD;
- Pode ter tabelas físicas (log; etc.);
- Pode ter colunas físicas (replicação).



HIERARQUIA DAS TABELAS, REGISTROS E CAMPOS



TABELA

Tabelas são objetos de banco de dados que contêm todos os dados em um banco de dados. Nas tabelas, os dados são organizados de maneira lógica em um formato de linha-e-coluna semelhante ao de uma planilha.

Exemplo Tabela Funcionário:

	CODIGO	NOME	CPF	SEXO	SALARIO
1	3005	JOSE FRANCISCO SOUZA FILHO	471663986-04	M	10000.50
2	3006	FABIO AGUIAR LIMA	329478954-87	M	2771.41
3	3007	ANTONIO CARLOS SILVA DE SOUZA	427744976-04	M	858.00
4	3008	MARIA LOPES	378093000-59	F	929.77
5	3009	ELISABETE COUTINHO	187530190-91	M	678.10

REGISTRO

Um registro é o conjunto de campos valorizados de uma tabela.

É a unidade básica para o armazenamento e recuperação de dados e que identifica a entrada de um único item de informação em particular numa tabela do banco de dados.

São chamados de tuplas ou n-uplas.

Também podemos chamar os registros de linhas da tabela.

Se uma tabela tem 20.000 linhas, tem 20.000 registros.

	CODIGO	NOME	CPF	SEXO	SALARIO
1	3005	JOSE FRANCISCO SOUZA FILHO	471663986-04	M	10000.50
2	3006	FABIO AGUIAR LIMA	329478954-87	M	2771.41
3	3007	ANTONIO CARLOS SILVA DE SOUZA	427744976-04	M	858.00
4	3008	MARIA LOPES	378093000-59	F	929.77
5	3009	ELISABETE COUTINHO	187530190-91	M	678.10

CAMPO

Campo ou coluna é a menor unidade destinada ao armazenamento de valores existentes em um arquivo ou tabela de um banco de dados.

Os dados armazenados são separados em pequenos fragmentos.

Cada campo contém um tipo de dado.

Exemplo de Campo na Tabela Funcionário: Nome

	CODIGO	NOME	CPF	SEXO	SALARIO
1	3005	JOSE FRANCISCO SOUZA FILHO	471663986-04	M	10000.50
2	3006	FABIO AGUIAR LIMA	329478954-87	M	2771.41
3	3007	ANTONIO CARLOS SILVA DE SOUZA	427744976-04	M	858.00
4	3008	MARIA LOPES	378093000-59	F	929.77
5	3009	ELISABETE COUTINHO	187530190-91	M	678.10

CONSTRAINTS

Constraints são objetos utilizados com a finalidade de estabelecer regras referentes à integridade e à consistência nas colunas das tabelas pertencentes a um sistema de banco de dados. Isso é importante para planejar e criar tabelas, pois devemos garantir a integridade dos dados presentes nas colunas e identificar os valores validos para tais dados.

Tipos de integridade	Tipos de constraint
Chave Primária	Constraint Primary Key
Chave Estrangeira	Constraint Foreign Key e Constraint References
Chave Primária Secundária ou Chave Única	Constraint Unique
Regras de Validação	Constraint Check

CONSTRAINTS

Regras de constraints

Cada tipo de constraint possui suas próprias regras. Vejamos, na tabela a seguir, quais são elas:

Tipos de constraints	Descrição
Constraint Primary Key (Chave Primária)	Uma coluna que é definida como chave primaria não pode aceitar valores nulos. Em cada tabela, pode haver somente uma constraint de chave primária.
Constraint Foreign Key (Chave Estrangeira)	Valores colunas podem ser definidas como chave estrangeira. No entanto, para que uma coluna seja definida dessa forma, é preciso que ela já tenha sido definida como chave primária em outra tabela. As colunas definidas como chave estrangeira podem aceitar valores nulos, e os datatypes das colunas relacionadas devem ser iguais.
Constraint Unique (Chave unica)	Várias colunas de uma tabela podem ser definidas como chave única e, ainda, aceitar valores nulos.
Constraint Check	Diversas colunas de uma tabela podem ser definidas como constraint check. Essas colunas podem aceitar valores nulos, mais isso depende das regras que são determinadas para elas.

SQL - STRUCTURED QUERY LANGUAGE

Structured Query Language, ou **Linguagem de Consulta Estruturada** ou **SQL**, é uma linguagem de pesquisa declarativa para banco de dados relacional (base de dados relacional).

Muitas das características originais do SQL foram inspiradas na álgebra relacional.

SQL - STRUCTURED QUERY LANGUAGE

Classificações do SQL:

- DDL → Data Definition Language - Linguagem de Definição de Dados
- DML → Data Manipulation Language - Linguagem de Manipulação de Dados
- DCL → Data Control Language - Linguagem de Controle de Dados
- DTL → Data Transaction Language - Linguagem de Transação de Dados
- DQL → Data Query Language - Linguagem de Consulta de Dados

DDL - LINGUAGEM DE DEFINIÇÃO DE DADOS

A DDL (Data Definition Language - Linguagem de Definição de Dados) permite ao utilizador definir tabelas novas e elementos associados.

Os comandos básicos da DDL são:

- CREATE cria um objeto (uma Tabela, por exemplo) dentro da base de dados.
- DROP apaga um objeto do banco de dados.
- ALTER realiza alterações. pode ser utilizada com diferentes finalidades: adicionar uma coluna a uma tabela existente, modificar as propriedades de uma coluna que já esteja criada, excluir uma coluna, renomear uma tabela ou renomear colunas.

Outros comandos:

- RENAME
- TRUNCATE
- COMMENT

DML - LINGUAGEM DE MANIPULAÇÃO DE DADOS

A DML é um conjunto da linguagem usada para inserir, atualizar e apagar dados.

- INSERT é usada para inserir um registro a uma tabela existente.
- UPDATE para mudar os valores de dados em uma ou mais linhas da tabela existente.
- DELETE permite remover linhas existentes de uma tabela.
- MERGE é utilizado para selecionar linhas de uma ou mais tabelas (origem) para inserções ou alterações em uma ou mais tabelas (destino).

DCL - LINGUAGEM DE CONTROLE DE DADOS

DCL controla os aspectos de autorização de dados e licenças de usuários para controlar quem tem acesso para ver ou manipular dados dentro do banco de dados.

Duas palavras-chaves da DCL:

- GRANT - autoriza ao usuário executar operações.
- REVOKE - remove ou restringe a capacidade de um usuário de executar operações.

DTL - LINGUAGEM DE TRANSAÇÃO DE DADOS

DTL gerencia as alterações feitas por instruções DML.

As alterações dos dados podem ser agrupados em transações lógicas.

- COMMIT envia todos os dados das mudanças permanentemente.
- ROLLBACK faz com que as mudanças nos dados existentes desde que o último COMMIT ou ROLLBACK sejam descartadas.
- SAVEPOINT defini uma marca na transação corrente.

DQL - LINGUAGEM DE CONSULTA DE DADOS

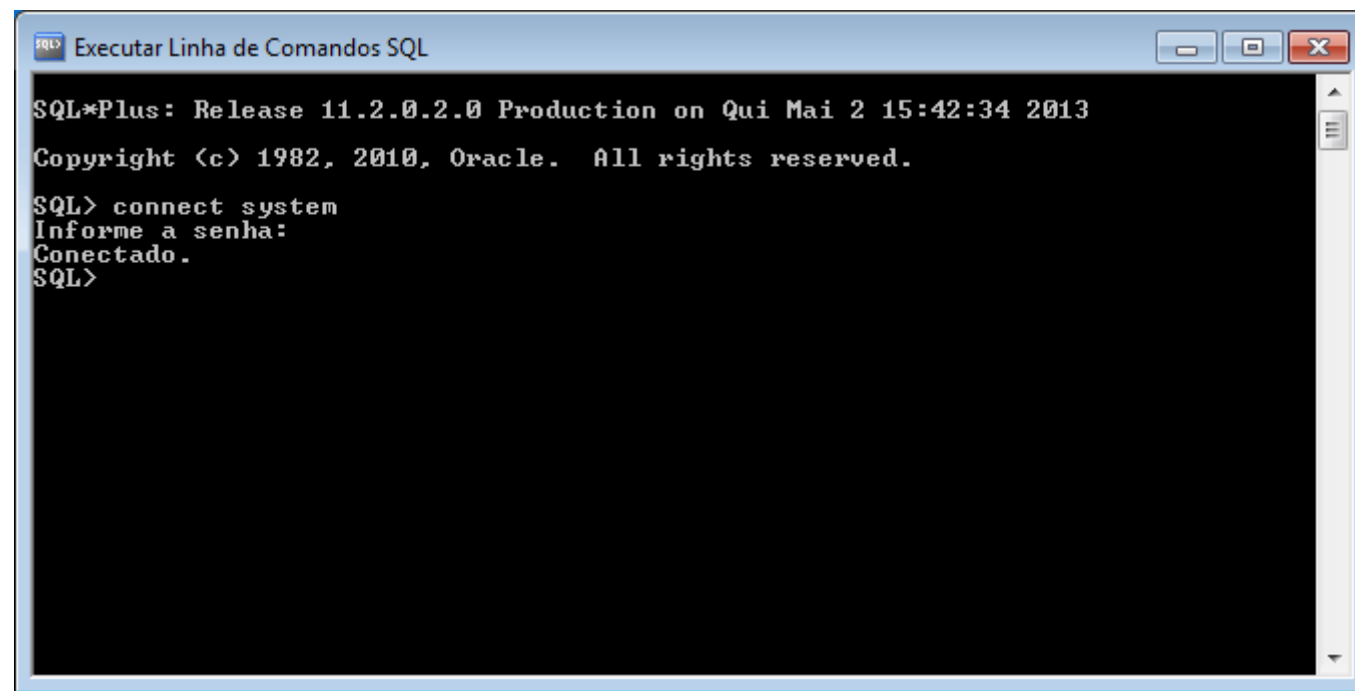
Embora tenha apenas um comando, a DQL é a parte da SQL mais utilizada.

O comando `SELECT` permite ao usuário especificar uma consulta ("query") como uma descrição do resultado desejado. Esse comando é composto de várias cláusulas e opções, possibilitando elaborar consultas das mais simples às mais elaboradas.

ORACLE DATABASE



ORACLE DATABASE



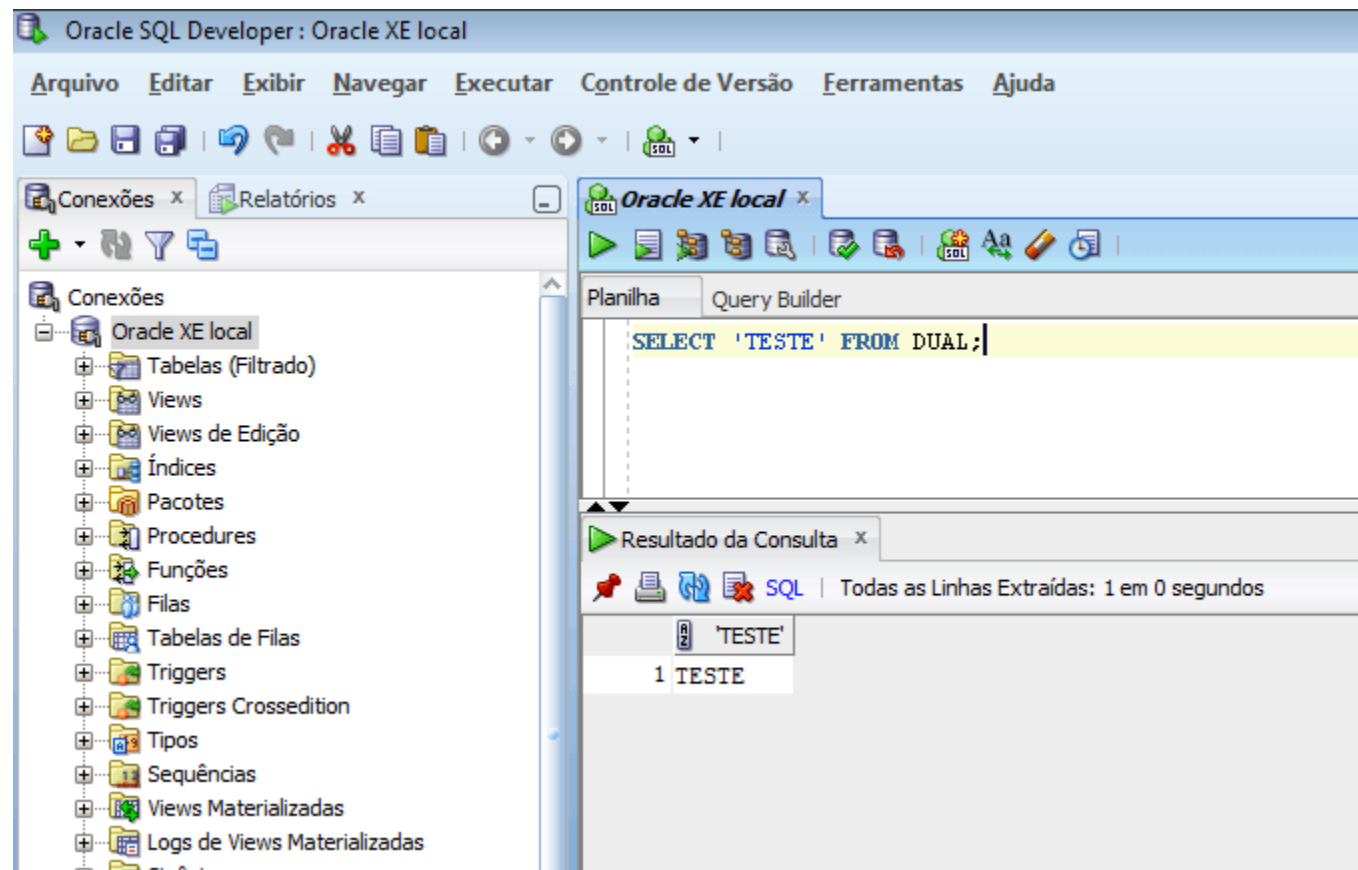
```
SQL> Executar Linha de Comandos SQL

SQL*Plus: Release 11.2.0.2.0 Production on Qui Mai 2 15:42:34 2013
Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect system
Informe a senha:
Conectado.
SQL>
```

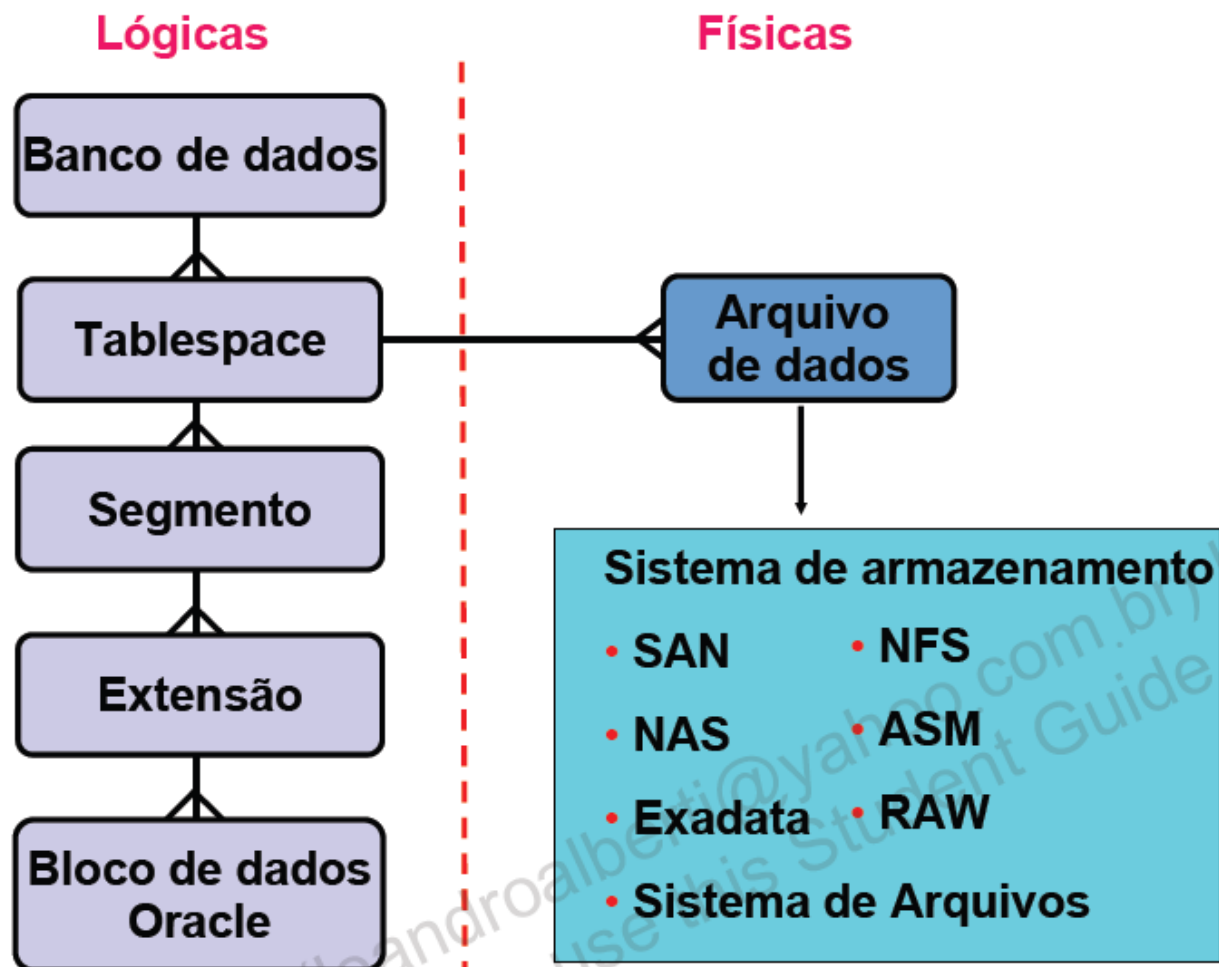
Executar Linha de Comandos SQL

ORACLE DATABASE



Oracle SQL Developer

ARQUITETURA DE ARMAZENAMENTO- ORACLE



ARQUITETURA DE ARMAZENAMENTO-ORACLE

Há quatro grandes componentes que compõe essa arquitetura:

- tablespaces,
- segmentos,
- extensões e
- bloco de dados.

Este é um dos principais fundamentos para administração de banco de dados Oracle, saber seu funcionamento e conceito é uma grande vantagem na construção de modelo físico de banco de dados.

BLOCO DE DADOS (DATA BLOCK)

O bloco de dados corresponde a um número específico de bytes do banco de dados para armazenado em disco. Sendo que esse valor é especificado durante a criação do banco de dados, pelo parâmetro `db_block_size`.

Então, o Oracle gerencia todo o espaço de armazenamento nos arquivos de dados (datafiles) através dessas pequenas unidades chamado blocos de dados, que carregam informações importantes como cabeçalho, diretório da tabela, diretório de linha, dados de tabelas ou índices e espaço livre para inserções ou atualizações de dados.

EXTENSÕES (EXTENT)

As extensões são unidades lógicas de armazenamento composta por um conjunto de blocos de dados, um ou mais extensões formam um segmento.

As extensões são muito utilizadas para definir a característica de armazenamento de algum objeto, como um tabela ou índice. Eles podem influenciar muito na fragmentação de espaço ou ajudar a definir um bom plano de crescimento da base de dados.

EXTENSÕES (EXTENT)

Os segmentos é um conjunto de extensões que possui todos os dados necessários para uma estrutura de armazenamento lógico, como as tablespaces.

Para cada tabela criada, o Oracle aloca um ou mais extensões para formar um segmento de dados, assim, também podemos dizer que o Oracle aloca um ou mais extensões para formar um segmento de índice.

SEGMENTOS

Na arquitetura do Oracle, podemos encontrar quatro tipos de **segmentos**:

Dados	<i>Data</i>	Tabelas
Índices	<i>Index</i>	Índices
Desfazer	<i>Rollback</i>	Rollbacks Segments e UNO
Temporários	<i>Temporary</i>	Consultados de Agrupamentos, Sumarização, Ordenação e etc.

TABLESPACE

O Banco de dados Oracle é composto por uma ou mais unidades lógicas de armazenamento. Estas unidades lógicas são chamadas de TABLESPACES, basicamente são subdivisões do Banco de Dados, onde os segmentos (estruturas para armazenamento de dados) semelhantes são agrupados, facilitando o gerenciamento destes segmentos dentro do Banco de Dados. Pode-se dizer que as TABLESPACES são grupos lógicos de armazenamento.

Com o objetivo de organizar o Banco de Dados, criam-se TABLESPACES para os objetos do Banco de Dados (Segmentos se referem aos objetos do banco de dados), ou seja, cria-se uma TABLESPACE para armazenar Tabelas, outra para armazenar índices e assim por diante.

TABLESPACE

```
CREATE TABLESPACE TBS_MIF2AN  
DATAFILE 'MIF2AN.dbf' SIZE 40M AUTOEXTEND ON  
ONLINE  
PERMANENT  
EXTENT MANAGEMENT LOCAL AUTOALLOCATE  
SEGMENT SPACE MANAGEMENT AUTO;
```

TABLESPACE

Onde:

CREATE - Comando de definição de dados (DDL) para criação de objetos em SQL.

TABLESPACE - Indica o tipo de objeto a ser criado.

TBS_NOME_TABLESPACE - Nome que deseja atribuir a sua TABLESPACE, sugere-se usar o prefixo "TBS_".

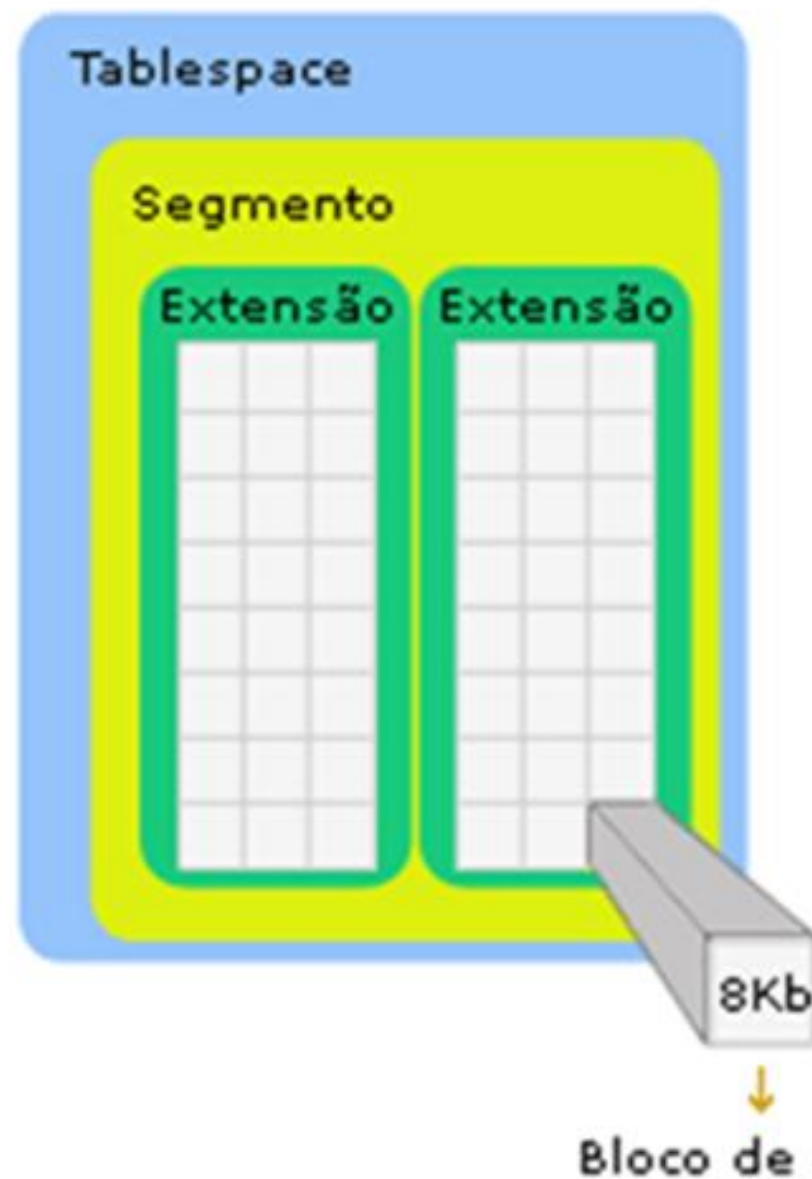
DATAFILE - Indica que será criado um DATAFILE junto com sua TABLESPACE (é necessário criar ao menos um DATAFILE).

NOME_DATAFILE.dbf - Nome escolhido para o DATAFILE, contido entre aspas simples seguido da extensão .dbf.

ONLINE - Torna a TABLESPACE disponível imediatamente após sua criação.

SIZE - Foi definido 40 Megas como tamanho inicial, é um tamanho estimado. A propriedade não declarada **AUTOEXTEND** permite um tamanho máximo ilimitado. O tamanho inicial é aumentado conforme houver necessidade, ou seja, assim que o arquivo atingir o tamanho inicialmente definido.

Arquitetura de Armazenamento Lógico



USUÁRIO

Criação de usuário

```
CREATE USER nome_de_usuario IDENTIFIED BY sua_senha [DEFAULT TABLESPACE nome_da_tablespace] [TEMPORARY TABLESPACE tablespace_temporaria];
```

nome_de_usuario – É nome do usuário que será criado;

sua_senha – É a senha para o usuário que está sendo criado;

nome_da_tablespace – É a tablespace padrão onde os objetos do banco de dados são armazenados. Se essa opção for omitida, o banco assume a tablespace SYSTEM padrão;

tablespace_temporaria – É a tablespace padrão onde são armazenados os objetos temporários, como tabelas temporárias por exemplo. Se essa opção for omitida um tablespace temporário TEMP é assumida.

USUÁRIO

Criação de usuário

```
CREATE USER nome_de_usuario IDENTIFIED BY sua_senha [DEFAULT TABLESPACE  
nome_da_tablespace] [TEMPORARY TABLESPACE tablespace_temporaria];
```

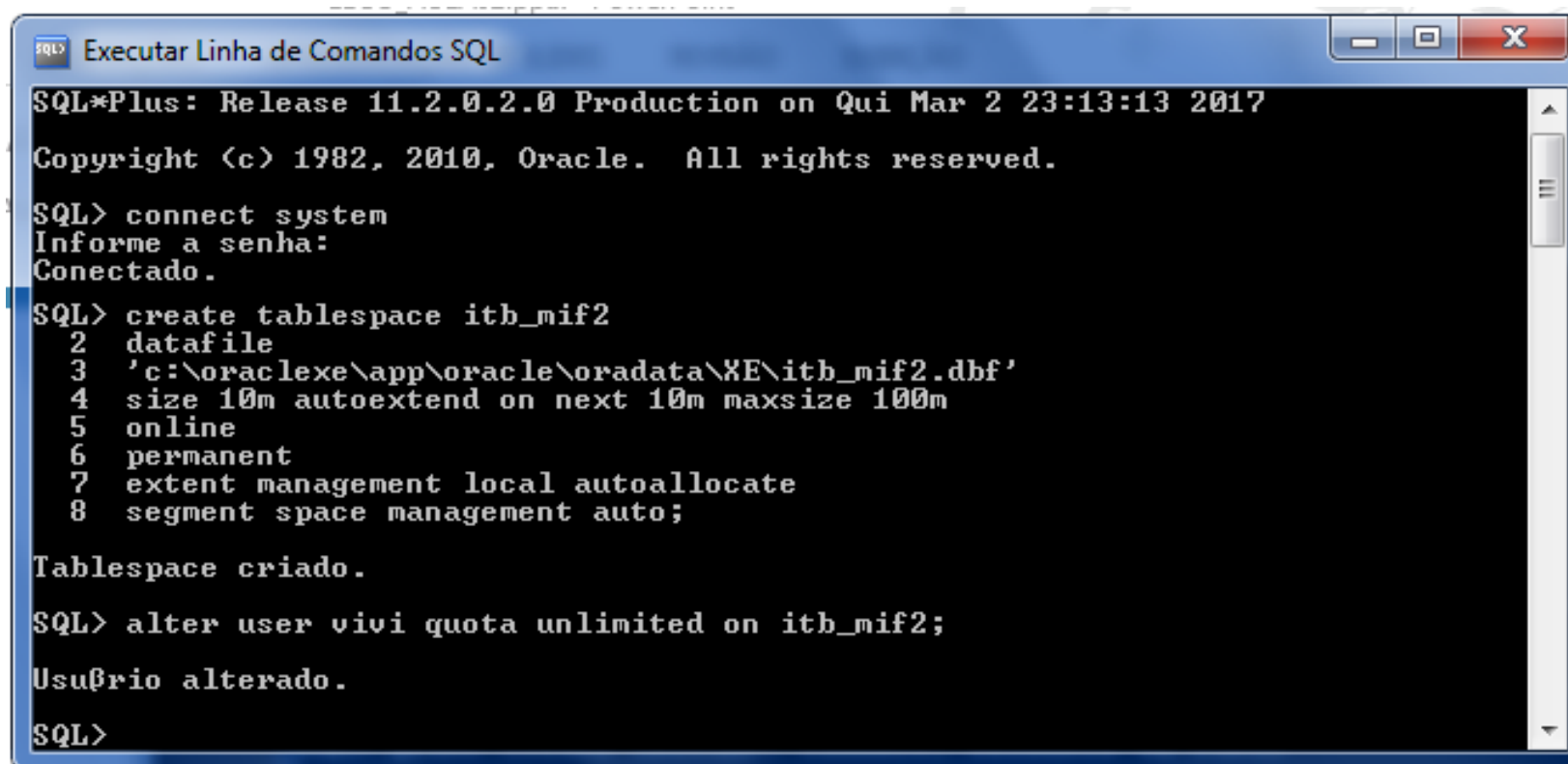
nome_de_usuario – É nome do usuário que será criado;

sua_senha – É a senha para o usuário que está sendo criado;

nome_da_tablespace – É a tablespace padrão onde os objetos do banco de dados são armazenados. Se essa opção for omitida, o banco assume a tablespace SYSTEM padrão;

tablespace_temporaria – É a tablespace padrão onde são armazenados os objetos temporários, como tabelas temporárias por exemplo. Se essa opção for omitida um tablespace temporário TEMP é assumida.

TABLESPACE



```
SQL> Executar Linha de Comandos SQL
SQL*Plus: Release 11.2.0.2.0 Production on Qui Mar 2 23:13:13 2017
Copyright (c) 1982, 2010, Oracle. All rights reserved.

SQL> connect system
Informe a senha:
Conectado.

SQL> create tablespace itb_mif2
2  datafile
3  'c:\oracle\exe\app\oracle\oradata\XE\itb_mif2.dbf'
4  size 10m autoextend on next 10m maxsize 100m
5  online
6  permanent
7  extent management local autoallocate
8  segment space management auto;

Tablespace criado.

SQL> alter user vivi quota unlimited on itb_mif2;

Usuário alterado.

SQL>
```

TABLESPACE

```
SQL > create tablespace NOVA_TABLESPACE
2      datafile
3      'c:\oracle\product\XE\oradata\nova_tablespace.dbf'
        size 100m autoextend on next 50m maxsize 500m
4      online
5      permanent
6      extent management local autoallocate
7      segment space management auto;
```

Tablespace Created.

```
SQL > create user JEAN
2      identified by JEAN
3      default tablespace NOVA_TABLESPACE
4      temporary tablespace TEMP;
```

User Created.

```
SQL > grant create session, connect, resource to JEAN;
Grant succeeded.
```

```
SQL > alter user JEAN quota unlimited on NOVA_TABLESPACE;
User Altered.
```