Curso de Férias
SQL e PL/SQL Básico

MATERA
systems

1. Introdução a banco de dados.
2. Comandos DLL
3. Comandos DML.
4. Comandos DML II.
5. PLSQL.
6. Procedures e Functions.
7. SubQuerys.
8. Mais Objetos Oracle.
9. Packages.

MATERA systems

Introdução a banco de dados.

- O que é um banco de dados?
- Porque precisamos de um banco de dados?
- O que pode ser um banco de dados?

MATERA systems

Introdução a banco de dados.

- O que é um SGBD?
- Tipos de bancos de dados.
- O que é SQL (Structured Query Language).

MATERA systems

Comandos DDL

São comandos utilizados para definirem as estruturas de dados, como as tabelas que compõem um banco de dados, os cinco tipos básicos de instruções DDL são:

- CREATE: cria uma estrutura de banco de dados. Por exemplo, CREATE TABLE é usada para criar uma tabela; outro exemplo é CREATE USER, usada para criar um usuário do banco de dados.

- ALTER: modifica uma estrutura de banco de dados. Por exemplo, ALTER TABLE é usada para modificar uma tabela.

- DROP: remove uma estrutura de banco de dados. Por exemplo, DROP TABLE é usada para remover uma tabela.

Comandos DDL

Tipos de dados.

Cada valor manipulado pelo Oracle Database possui um tipo de dados.

Tipo	Descrição
VARCHAR2(comprimento_máximo)	Carácter de tamanho variável, podendo atingir o tamanho máximo de até 32767 bytes.
NUMBER [precisão, escala]	Tipo numérico fixo e de ponto flutuante.
DATE	Tipo para acomodar data e hora

Comandos DDL

- Criação de tabelas

Os dados são armazenados em estruturas chamadas tabelas, abaixo é apresentada a composição do comando create table.

```
Create table time
(
       id_time     number         not null,
       nome        varchar2(400)  not null
);
```

Comandos DDL

- Constraints

Constraints são objetos fundamentais para a escalabilidade, flexibilidade e integridade dos dados armazenados em um banco de dados. Elas aplicam regras específicas para os dados, garantem que os dados estejam em conformidade com os requisitos definidos. Existem alguns tipos de constraints no Oracle, a seguir elas são apresentadas:

Primary key: Cada tabela pode ter, no máximo, uma constraint de primary key (em português chave primária). A primary key pode ter mais que uma coluna da tabela. A constraint de primary key força que cada chave primária só pode ter um valor único, impondo em simultâneo a constraint unique e NOT NULL. Uma primary key vai criar um indice único, caso ainda não exista para a coluna em causa.

Foreign Key: A foreign key (em português chave estrangeira) é definida para uma tabela (conhecida como filha) que tem um relacionamento com outra tabela (conhecida como pai). O valor guardado na foreign key deverá ser o mesmo presente na primary key respectiva.

```
alter table TIME add constraint pk_time primary key (ID_TIME);
alter table jogador add constraint fk_time foreign key (id_time) references time(id_time);
```

Comandos DDL

- Comentários

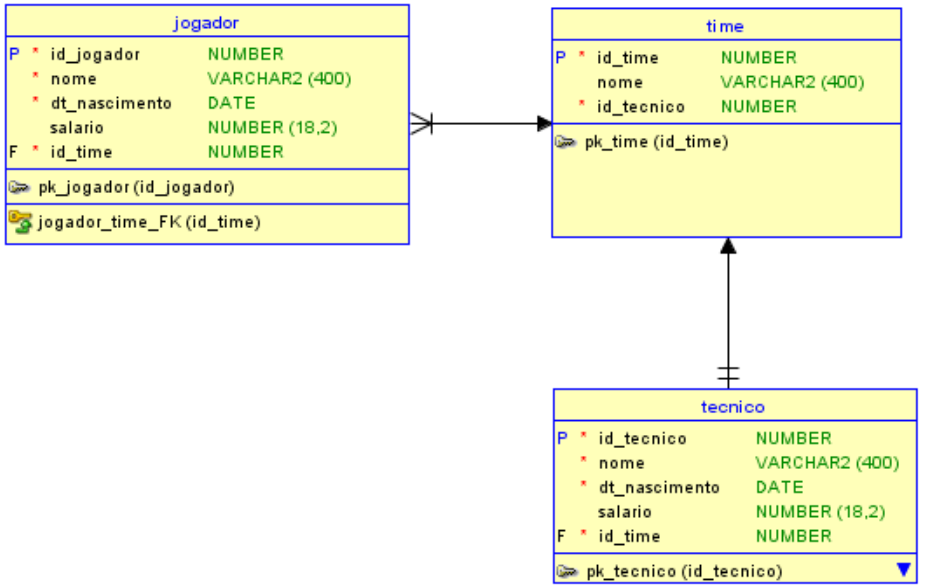
Ao criar uma tabela, é possível definir comentários para a tabela e colunas, isso auxilia no entendimento do objetivo da tabela e colunas.

```
comment on table TIME  is '[Cadastro] Tabela para armazenamento de times.';

comment on column TIME.id_time  is 'Código Identificador do time.';
comment on column TIME.nome  is 'Nome do Time.';
```

Comandos DDL

- Exercícios:
  - Criar as tabelas time, técnico e jogador;
  - Definir constraints para as tabelas;
  - Criar comentários para as tabelas e as colunas;



MATERA systems

Comandos DML

Data Manipulation Language (DML) são utilizados para o gerenciamento de dados dentro de objetos do banco.

- A instrução SELECT é utilizada para recuperar os dados do banco de dados.

```
Select * from time where nome = 'BARCELONA'  order by nome
```

- O COMMIT é um comando utilizado no controle transacional, faz com quem o dado inserido, alterado ou removido seja realmente persistido(salvo) no banco de dados.

- O ROLLBACK é um comando utilizado também no controle transacional, ele desfaz as alterações de dados realizadas desde o início da Rotina, Checkpoint(savepoint) ou último COMMIT.

Comandos DML

A instrução INSERT é utilizada para inserir dados no banco de dados.

```
insert into time (id_time, nome) values (1,'BARCELONA')
```

SEQUENCE.

nextval: retorna o próximo valor da sequence.
Curval: retorna o último número gerado pela sequence.

```
create sequence seq_jogador
minvalue 1    maxvalue 999999999999999999999999999
start with 1    increment by 1
nocache;

/*exemplo de utilização de sequences*/
select seq_jogador.nextval from dual
```

Comandos DML

- Exercícios:
  - Inserir 2 times.
  - Inserir 2 técnicos.
  - Inserir 11 jogadores em um time.
  - Listar todos jogadores de um determinado time.
  - Listar todos times.
  - Listar técnicos com mais de 40 anos.
  - Inserir os jogadores existentes para o outro time (select insert com sequence).

Comandos DML

A instrução UPDATE é utilizada para alterar dados já existentes no banco de dados.

```
update time set nome = 'BARCELONA FUTEBOL' where id_time = 1
```

```
update time set nome = 'BARCELONA FUTEBOL ALTERADO' where nome = 'BARCELONA FUTEBOL'
```

```
update time set nome = 'BARCELONA FUTEBOL ALTERADO' where nome = 'BARCELONA FUTEBOL  '
```

```
update time set nome = 'BARCELONA' || 'FUTEBOL'
where id_time = 1
```

```
update time set nome = nome || ' COMPLEMENTO'
where id_time = 1
```

```
update time set nome = 'BARCELONA FUTEBOL', SEGUNDO_NOME = 'SEM SEGUNDO NOME'
where id_time = 1
```

Comandos DML

- Exercícios:
  - Inserir um time novo.
  - Alterar todos jogadores de um time para o novo time.
  - Aumentar em 10% o salário de todos jogadores do novo time.
  - Aumentar o salário de todos técnicos em 20%.

Comandos DML

A instrução DELETE é utilizada para remover dados no banco de dados.

```
delete times where id_time = 1

delete jogador where salario = 10000
```

Comandos DML

- Exercícios:
  - Inserir um time novo.
  - Inserir 3 jogadores extras no time novo.
  - Alterar o salário de 3 jogadores para valores acima de R$ 100.000,00.
  - Remover jogadores do novo time com salários superiores R$ 100.000,00.
  - Remover times que estejam sem jogadores e técnicos.

Comandos DML II

- **Junções de Dados e Apelidos**
  - **INNER JOIN**
  - **LEFT JOIN**
  - **RIGHT JOIN**

Comandos DML II | Inner Join

Quando queremos juntar duas ou mais tabelas, que internamente, tenham valores correspondentes (parte amarela).

/ PLSQL JOIN */*

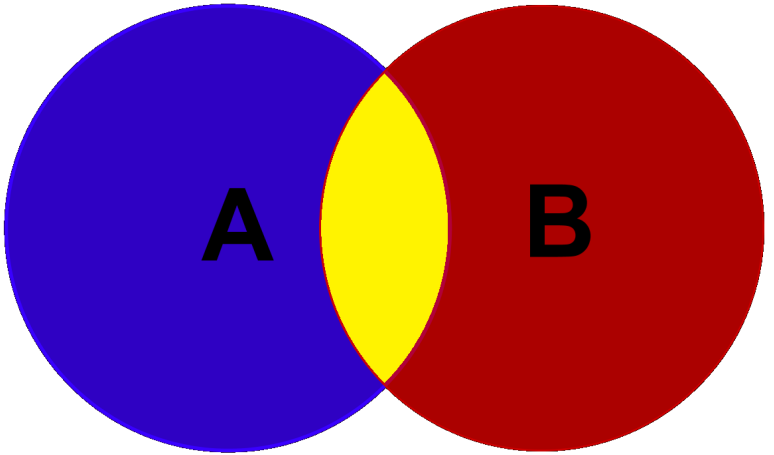
```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG, EQUIPE    EQU
WHERE  JOG.ID_EQUIPE = EQU.ID_EQUIPE;
```

/ SQL JOIN */*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG JOIN EQUIPE EQU ON JOG.ID_EQUIPE = EQU.ID_EQUIPE;
```

/ SQL JOIN (ANSI)*/*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG INNER JOIN EQUIPE EQU ON JOG.ID_EQUIPE = EQU.ID_EQUIPE;
```



MATERA systems

Comandos DML II | Left Join

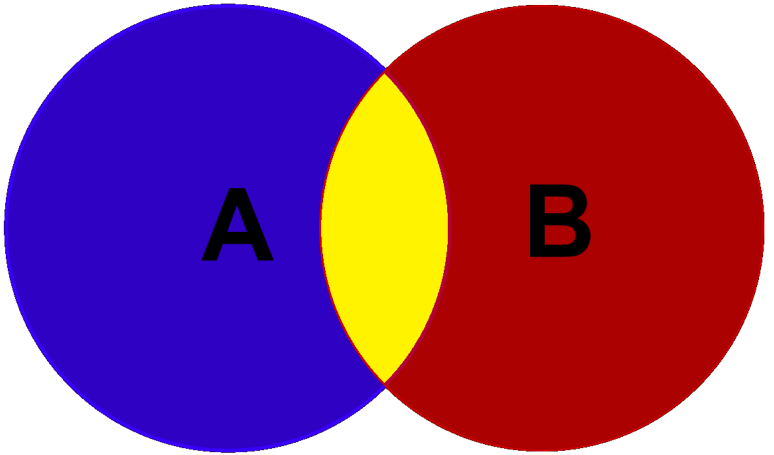
Serve para selecionar todos os itens de uma tabela A com uma tabela B mesmo que A não esteja relacionado com a tabela B (Parte Azul).

/ PLSQL LEFT JOIN */*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG, EQUIPE    EQU
WHERE  JOG.ID_EQUIPE = EQU.ID_EQUIPE(+);
```

/ SQL JOIN (ANSI)*/*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG LEFT JOIN EQUIPE EQU ON JOG.ID_EQUIPE = EQU.ID_EQUIPE;
```



MATERA systems

Comandos DML II | Right Join

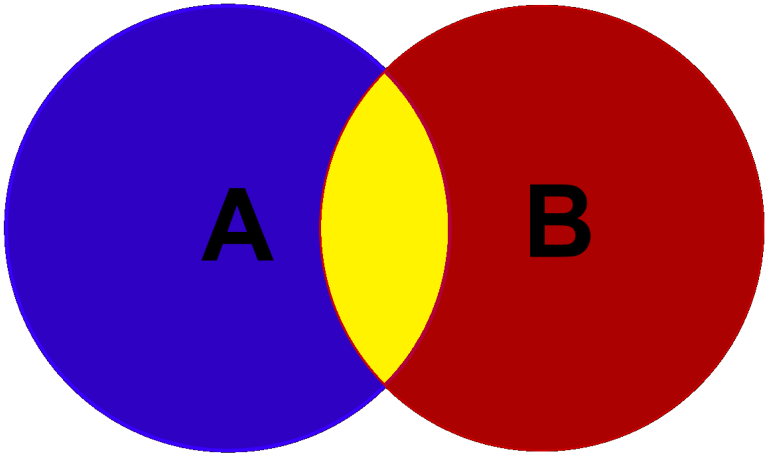
Funciona como o left outer join, mas ao contrário. (Parte Vermelha).

/ PLSQL LEFT JOIN */*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG, EQUIPE    EQU
WHERE  JOG.ID_EQUIPE(+) = EQU.ID_EQUIPE;
```

/ SQL JOIN (ANSI)*/*

```
SELECT JOG.NOME     NOME,
       EQU.NOME AS NOME_DA_EQUIPE
FROM   JOGADOR JOG RIGHT JOIN EQUIPE EQU ON JOG.ID_EQUIPE = EQU.ID_EQUIPE;
```



MATERA systems

Comandos DML II | Ordenações

Utilizamos as ordenações para ordenar os resultados de uma consulta.
Podemos ordenar por ordem crescente(*asc*) ou decrescente(***desc***).

```
select nome, data_nascimento as data_nasc from jogador order by nome asc

select nome, data_nascimento as data_nasc from jogador order by nome, data_nascimento asc

select nome, data_nascimento as data_nasc from jogador order by 2, 1 desc

select nome, data_nascimento as data_nasc from jogador order by data_nasc desc
```

Comandos DML II | Ordenações

- Exercícios:
  - Selecione os Times em ordem crescente.
  - Selecione os nomes de jogadores e seus respectivos nomes dos times ordenado(asc) pela data de nascimento dos jogadores.

Comandos DML II | Agrupamentos de Dados

Utilizamos agrupamento para juntar os dados equivalentes com a palavra group by.

- Com a utilização e grupos podemos utilizas as funções de agregação, que permitem realizar cálculos sobre o resultado da consulta retornada.
- Todas colunas selecionadas que não estão sendo utilizadas em algum tipo de função de agregação deverão estar declaradas no “group by”.

```
Select t.nome
from   jogador as j,
       time    as t
where  t.id_time = j.time_id_time
group  by t.nome
```

Comandos DML II | Funções de Agregação

As Funções de Agregação são utilizadas para manipular os dados agrupados.

COUNT → Conta o número de linhas afetadas pelo comando.
SUM → Calcula o somatório do valor das colunas especificadas.
AVG → Calcula a média aritmética dos valores das colunas.
MIN → Seleciona o menor valor da coluna de um grupo de linhas.
MAX → Seleciona o maior valor da coluna de um grupo de linhas.

```
Select Count(*), t.nome
from   jogador as j,
       time    as t
where  t.id_time = j.time_id_time
group  by t.nome
```

Comandos DML II | Funções de Agregação

- Exercícios:
  - Gere uma consulta retornando a folha de pagamento de cada equipe.
  - Gere uma consulta retornando a média salarial de cada equipe.
  - Gere uma consulta que retorne o menor salário de cada equipe.
  - Gere uma consulta que retorne o maior salário de cada equipe.

Contato

www.matera.com

Altieres de Matos
altieres.matos@matera.com
altitdb@gmail.com
Linkedin: https://goo.gl/kb3fyq

Junior Miqueletti
junior.miqueletti@matera.com
juniormiqueletti@gmail.com
Linkedin: https://goo.gl/ryjTG1

Thank you :D

MATERA systems