

MBA em IA e Big Data

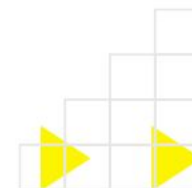


Curso 01 - Linguagens e Ferramentas para Inteligência Artificial e Big Data (Python e SQL)

Aula extra - outros conceitos de Bancos de Dados

Jose Fernando Rodrigues Junior
ICMC-USP São Carlos

Objetivo: prover uma visão geral de conceitos adicionais amplamente usados em bancos de dados





Tablespace

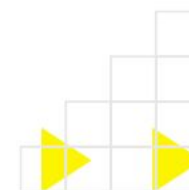
- O Oracle permite gerenciar o espaço em disco por meio da abstração Tablespace

- Exemplo:

```
CREATE TABLESPACE users  
    DATAFILE 'c:/temp/EspacoOracle.dbf'  
    SIZE 1000m;
```

- Documentação Oracle:

<https://www.oracletutorial.com/oracle-administration/oracle-create-tablespace/>





Database

- Assim como em outros Sistemas Gerenciadores de Bases de Dados, o Oracle possui a abstração "Database", a base propriamente dita

- Exemplo:

```
CREATE DATABASE meuNovoBancoDeDados           /*SID*/  
USER SYS IDENTIFIED BY sys_password  
USER SYSTEM IDENTIFIED BY system_password  
DEFAULT TEMPORARY TABLESPACE temp  
UNDO TABLESPACE undotbs1  
DEFAULT TABLESPACE users;
```

- Documentação Oracle: https://docs.oracle.com/cd/E18283_01/server.112/e17120/create003.htm





Role

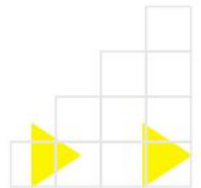
- Dentro de um banco de dados, há uma série de permissões
- É possível atribuir as permissões uma a uma a cada usuário, ou atribuir múltiplas permissões a um `role`

- Exemplo:

```
grant create session, create table, create view to usuarioMBA;
```

- Documentação Oracle:

<https://docs.oracle.com/javadb/10.8.3.0/ref/rrefsqlgrant.html>





User

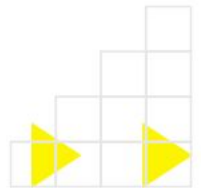
- Uma base sempre tem usuários que podem acessá-la

- Exemplo:

```
create user teste identified by uma_senha default tablespace USERS;  
grant unlimited tablespace to teste;  
grant usuarioMBA to teste;
```

- Documentação Oracle:

https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_8003.htm



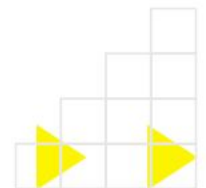


View

- Algumas consultas podem se tornar complexas

- Exemplo:

```
SELECT MATRICULA.ALUNO, AVG (NOTA) , COUNT (*)  
FROM ALUNO JOIN MATRICULA  
ON ALUNO.NUSP = MATRICULA.ALUNO  
GROUP BY MATRICULA.ALUNO  
HAVING AVG (NOTA) >= 6 AND COUNT (*) >= 5  
ORDER BY AVG (NOTA) ;
```





View

- Uma view pode simplificar o uso, e controlar o acesso
- Exemplo:

```
CREATE VIEW VALUNOS
```

```
AS
```

```
SELECT MATRICULA.ALUNO, AVG (NOTA) , COUNT (*)
```

```
FROM ALUNO JOIN MATRICULA
```

```
ON ALUNO.NUSP = MATRICULA.ALUNO
```

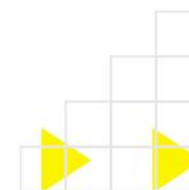
```
GROUP BY MATRICULA.ALUNO
```

```
HAVING AVG (NOTA) >= 6 AND COUNT (*) >= 5
```

```
ORDER BY AVG (NOTA) ;
```

- Para acessar os mesmos dados:

```
SELECT * FROM VALUNOS
```





View

- Pode-se alterar os nomes dos atributos

- Exemplo:

```
CREATE VIEW VALUNOS (NUSP, MEDIA, NDISCIPLINAS)
```

```
AS
```

```
SELECT MATRICULA.ALUNO, AVG (NOTA) , COUNT (*)
```

```
FROM ALUNO JOIN MATRICULA
```

```
ON ALUNO.NUSP = MATRICULA.ALUNO
```

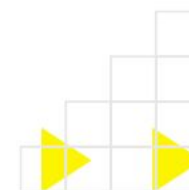
```
GROUP BY MATRICULA.ALUNO
```

```
HAVING AVG (NOTA) >= 6 AND COUNT (*) >= 5
```

```
ORDER BY AVG (NOTA) ;
```

- Para acessar os mesmos dados:

```
SELECT * FROM VALUNOS
```



View

- P
- E Selecionar e alterar os nomes dos atributos, não apenas simplifica o uso da consulta, mas também provê:
 - Segurança, os nomes originais não serão expostos
 - Independência, se os nomes originais mudarem, basta refazer a view

Documentação Oracle:

<https://www.oracletutorial.com/oracle-view/oracle-create-view/>

- P
- SELECT FROM VALORES



Índice

- É comum ter alguns atributos aparecendo de forma recorrente em consultas
- Se estes atributos são chaves, eles são automaticamente indexados, portanto, a pesquisa ocorre rapidamente
- Em outros casos é preciso criar o índice manualmente

- Exemplo:

```
CREATE INDEX nome_aluno_ix  
ON ALUNO (NOME) ;
```

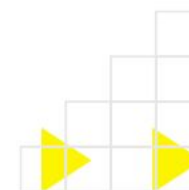
- Assim, consultas como

```
SELECT * FROM ALUNO WHERE ALUNO.NOME = 'Joao Pedro'
```

podem ser resolvidas mais rapidamente

- Documentação Oracle:

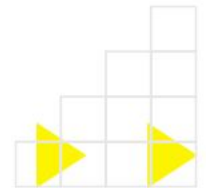
<https://www.oracletutorial.com/oracle-index/oracle-create-index/>





Trigger

- Algumas ações no banco de dados podem muito bem serem automatizadas
- Exemplo: atributos derivados, atributos cujos valores dependem do conteúdo de outros atributos/tabelas
- Por exemplo: o atributo TURMA.NAlunos depende do número de alunos matriculados, uma informação que fica na tabela MATRICULA
- Pode-se criar um trigger tal que após cada insert em MATRICULA, faz-se uma atualização em TURMA



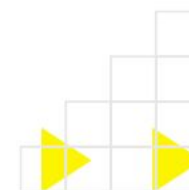


Trigger

- Pode-se criar um trigger tal que após cada insert em MATRICULA, faz-se uma atualização em TURMA

```
CREATE OR REPLACE TRIGGER NroDeAlunos
AFTER INSERT ON Matricula
FOR EACH ROW
DECLARE
    NroAlunos NUMBER;
BEGIN
    SELECT count(*) INTO NroAlunos
    FROM Matricula
    WHERE Sigla = :new.Sigla and Numero = :new.Numero;

    UPDATE Turma set NAlunos = NroAlunos
    WHERE Sigla = :new.Sigla and Numero = :new.Numero;
END NroDeAlunos;
```





Trigger

- Este é apenas um exemplo de uso de triggers
- Outras aplicações incluem
 - Auditoria
 - Segurança
 - Automação de inclusão de dados
 - Validação de dados
 - Consistência de dados mais avançada
- Documentação Oracle:
<https://www.oracletutorial.com/plsql-tutorial/oracle-trigger/>

