Oracle Schema – Criação e administração de um *schema*

AULA PL06

Hugo Peixoto

2018 – 2019 Universidade do Minho

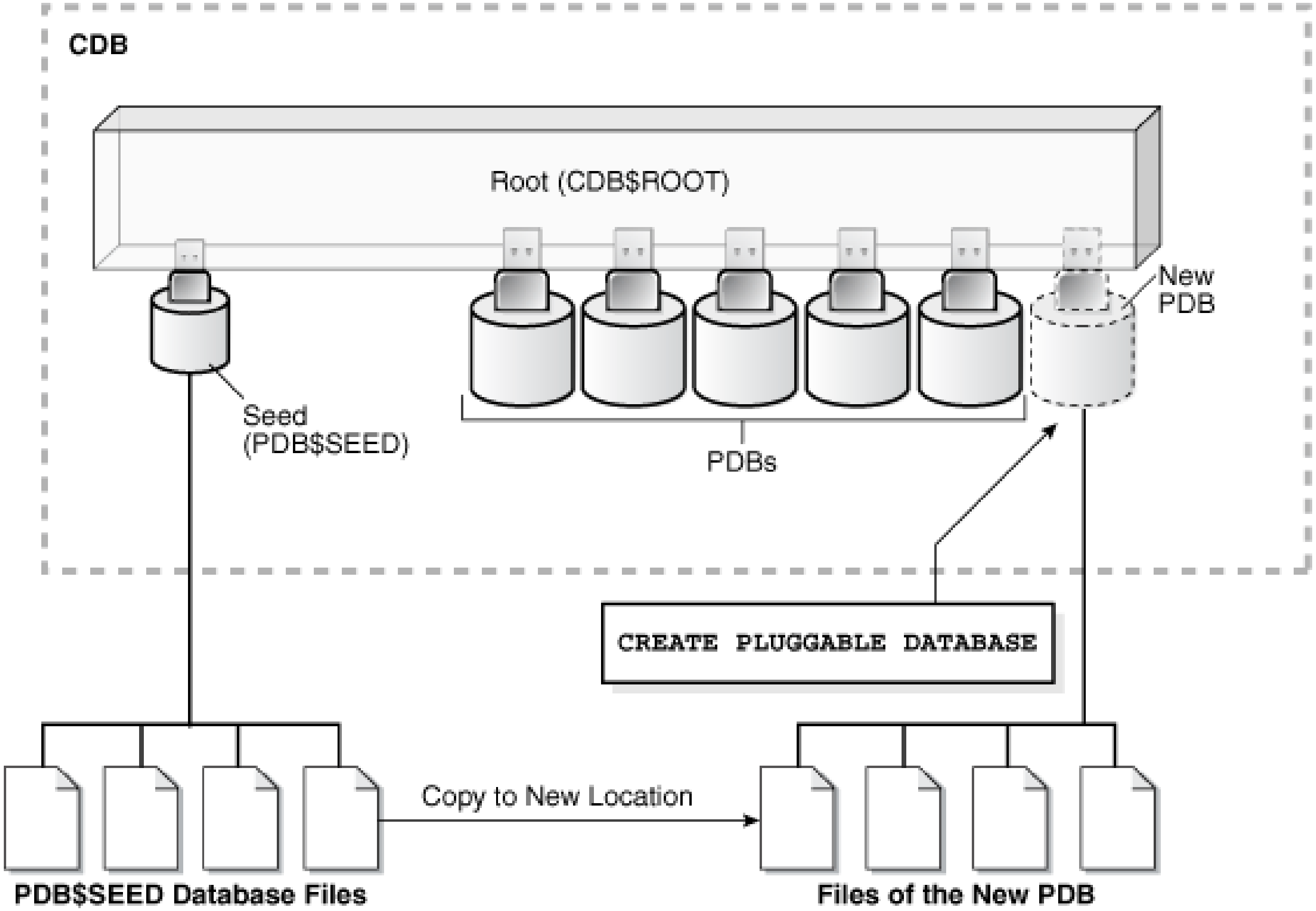
- Pluggable Databases
- Oracle schema
- Tablespaces | Datafiles | Objects
- Schema Objects
- Create schema / user
- FE05

http://hugoabpeixoto.github.io/aebd/

Bibliografia:

https://goo.gl/8HCSi8

Pluggable Databases



SID vs Service Name

SID - that is more like telling the listener "I want you to connect to this specific database, I know the 'address', here you go“.

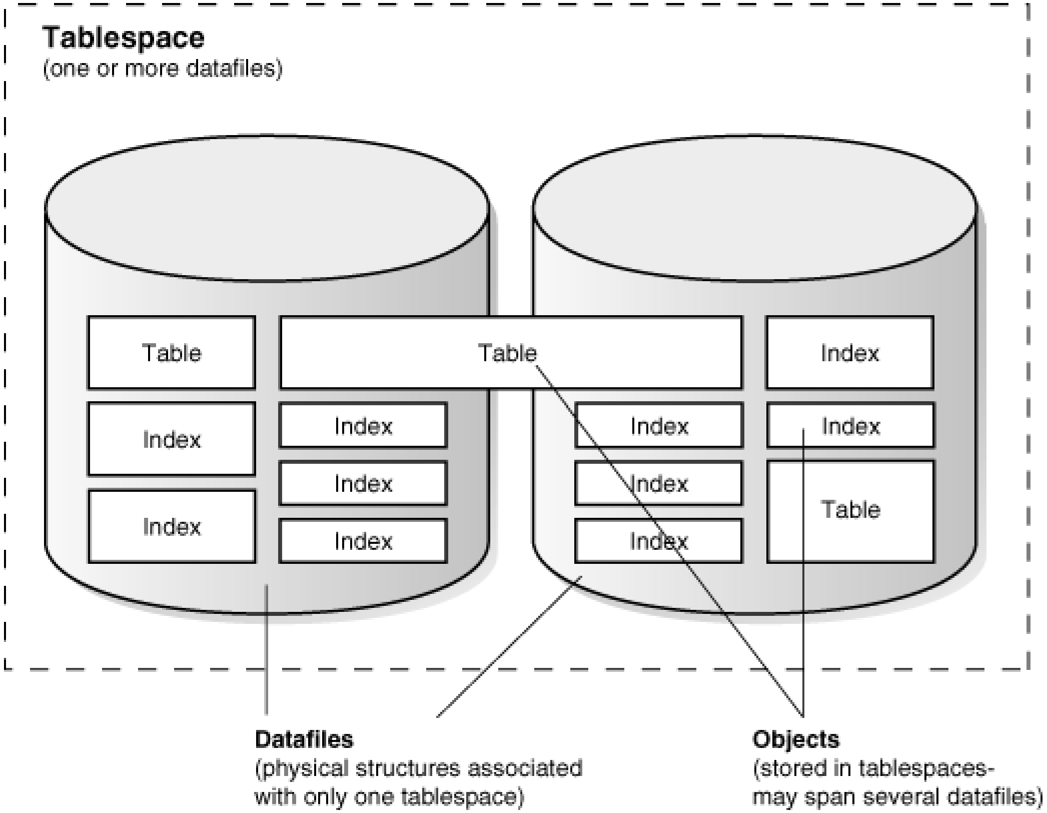
SERVICE NAME - you are asking the listener to put you in touch with a database that can service your request, a database that registers using that service.

Oracle Schema

A schema is a way to logically group objects in a single collection and provide a unique namespace for objects

User account + collection of all objects

Tablespaces | Datafiles | Objects



Tablespaces | Datafiles | Objects

- An Oracle database consists of one or more logical storage units called tablespaces, which collectively store all of the database's data;

- Each tablespace in an Oracle database consists of one or more files called datafiles, which are physical structures that conform to the operating system in which Oracle is running.

- A database's data is collectively stored in the datafiles that constitute each tablespace of the database.

Tablespaces | Datafiles | Objects

For example, the simplest Oracle database would have one tablespace and one datafile.

Another database can have three tablespaces, each consisting of two datafiles (for a total of six datafiles).

More Space for a Database! How?!

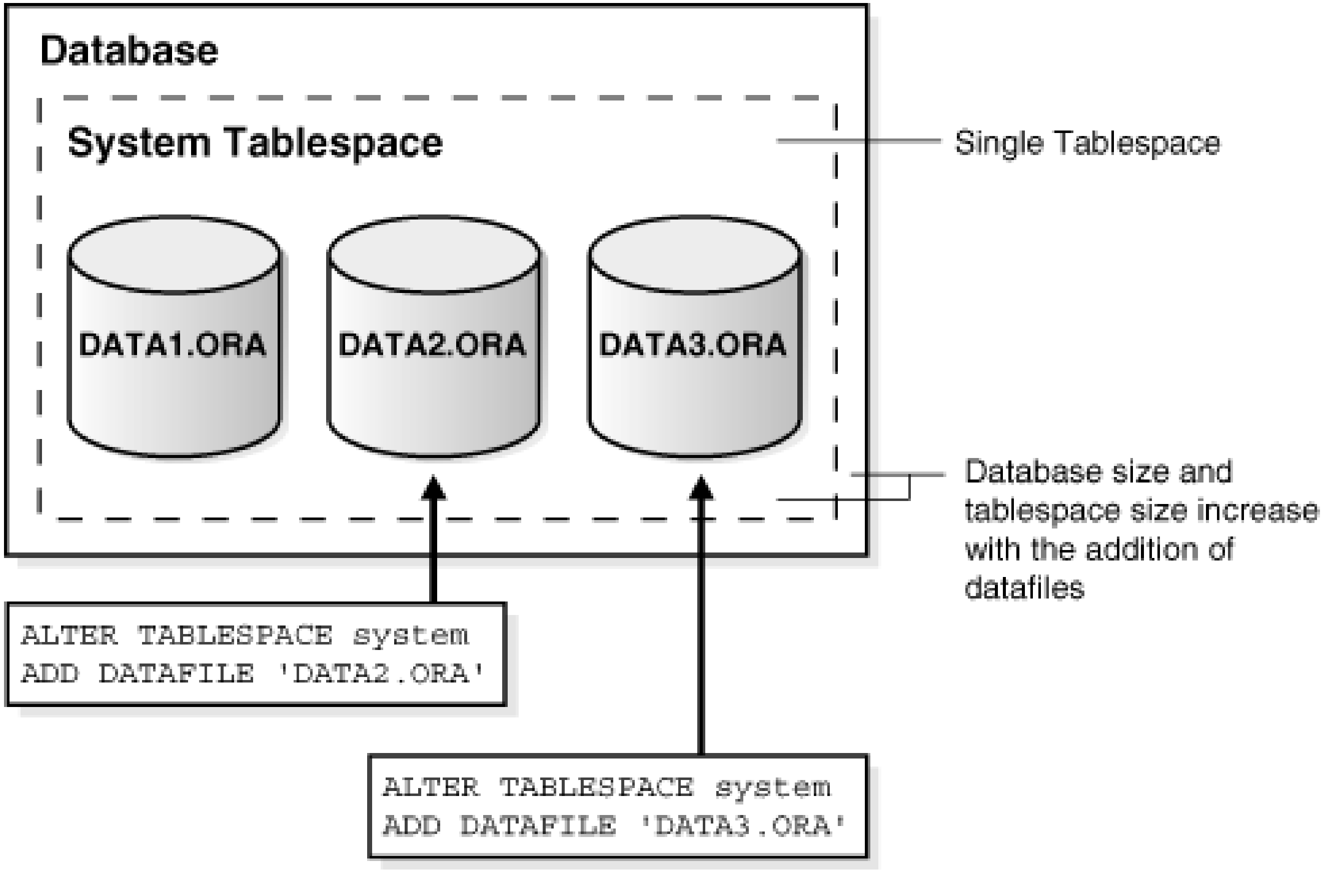
You can enlarge a database in three ways:

1) Add a datafile to a tablespace.

2) Add a new tablespace.

3) Increase the size of a datafile.

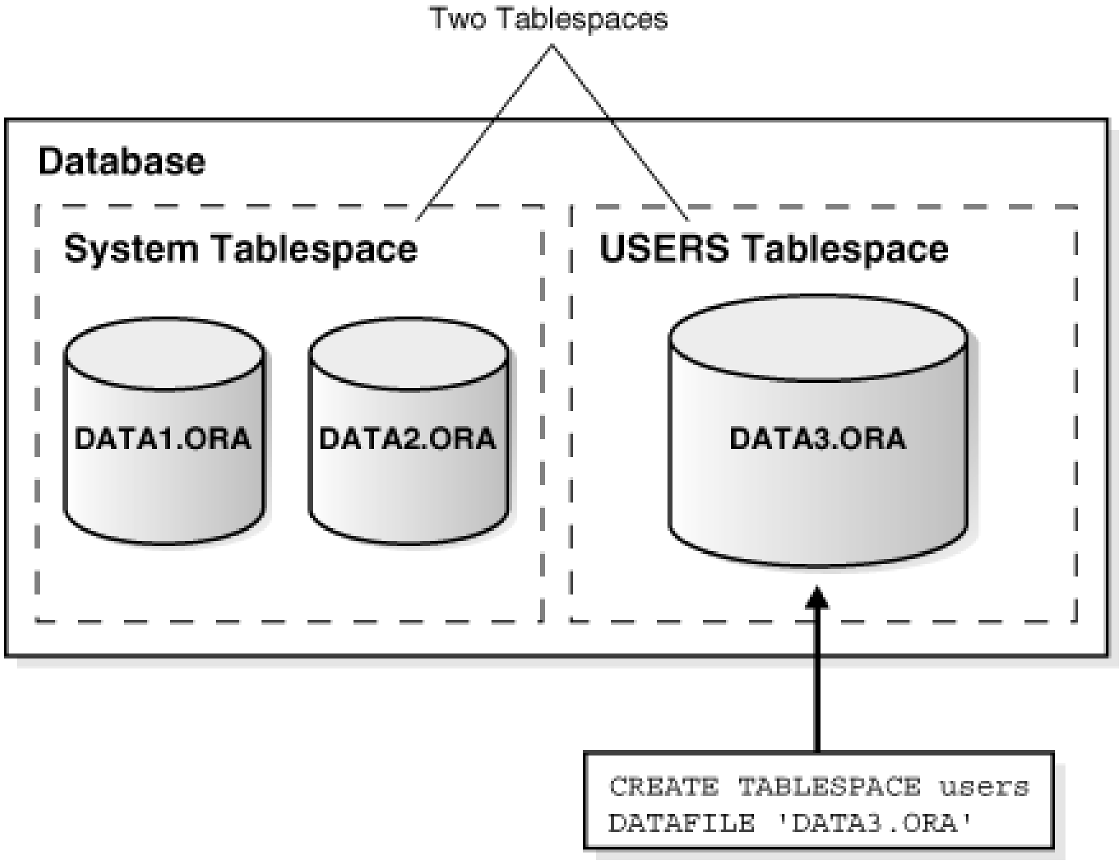
1) Add datafile to Tablespace


Database
System Tablespace
DATA1.ORA
DATA2.ORA
DATA3.ORA
Single Tablespace
Database size and tablespace size increase with the addition of datafiles

```
ALTER TABLESPACE system
ADD DATAFILE 'DATA2.ORA'
```

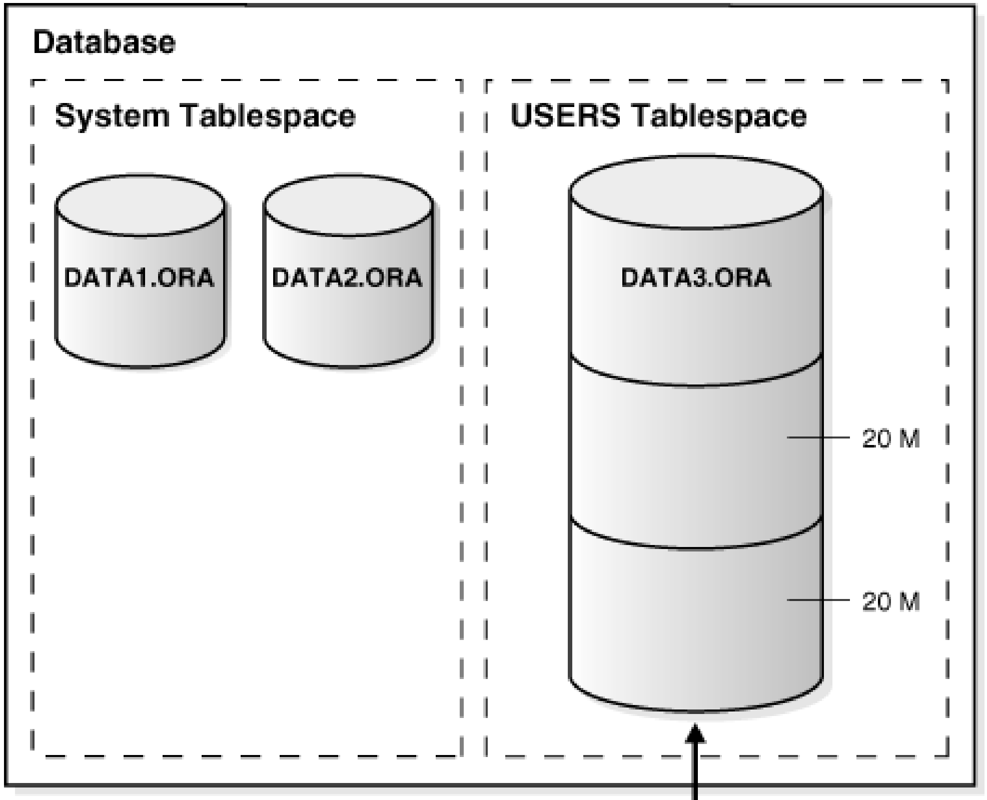
```
ALTER TABLESPACE system
ADD DATAFILE 'DATA3.ORA'
```

2) New Tablespace



```
CREATE TABLESPACE users
DATAFILE 'DATA3.ORA'
```

3) Increase Datafile size



```
ALTER DATABASE
DATAFILE 'DATA3.ORA'
   AUTOEXTEND ON NEXT 20M
      MAXSIZE 1000M;
```

Types of Tablespaces

Permanent:

These tablespaces store objects in segments that are permanent – that persist beyond the duration of a session or transaction.

Undo:

These tablespaces store segments that may be retained beyond a transaction, but are basically used to:

Provide read consistency for SELECT statements that access tables that have rows that are in the process of being modified.

Provide the ability to rollback a transaction that fails to commit.

Temporary:

This tablespace stores segments that are transient and only exist for the duration of a session or a transaction. Mostly, a temporary tablespace stores rows for sort and join operations.

Datafile

A tablespace in an Oracle database consists of one or more physical datafiles.

A datafile can be associated with **only one** tablespace and **only one database**.

When a datafile is first created, the allocated disk space is formatted but does not contain any user data. However, Oracle reserves the space to hold the data for future segments of the associated tablespace — it is used exclusively by Oracle. As the data grows in a tablespace, Oracle uses the free space in the associated datafiles to allocate extents for the segment.

Datafile

Locally managed **temporary tablespaces** have temporary datafiles (tempfiles), which are similar to ordinary datafiles, with the following exceptions:

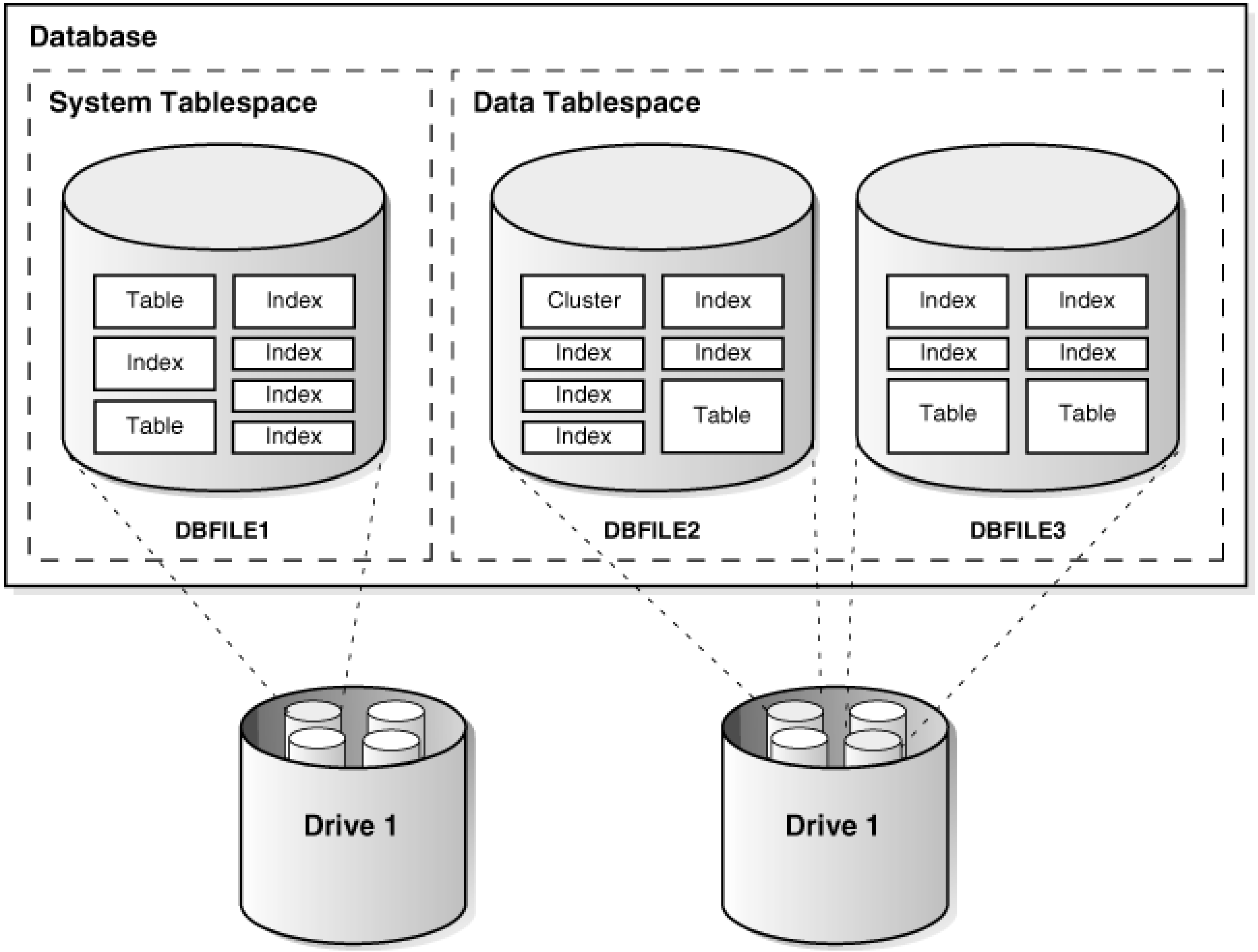
Tempfiles are always set to **NOLOGGING** mode;

You cannot make a tempfile **read only;**

You cannot create a tempfile with the **ALTER DATABASE** statement;

Media recovery does not recognize **tempfiles.**

Schema Objects



Schema Objects

Tables;

Views;

Materialized Views;

Dimensions;

Sequences;

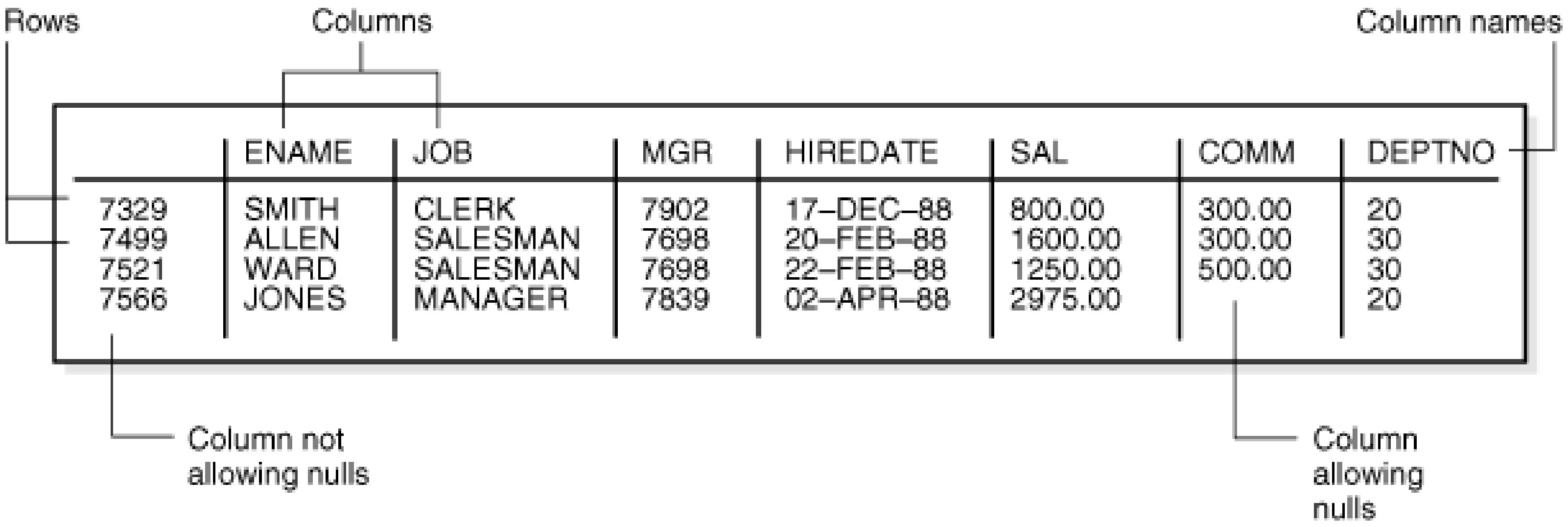
Synonyms;

Indexes;

Database links;

Stored Procedures…

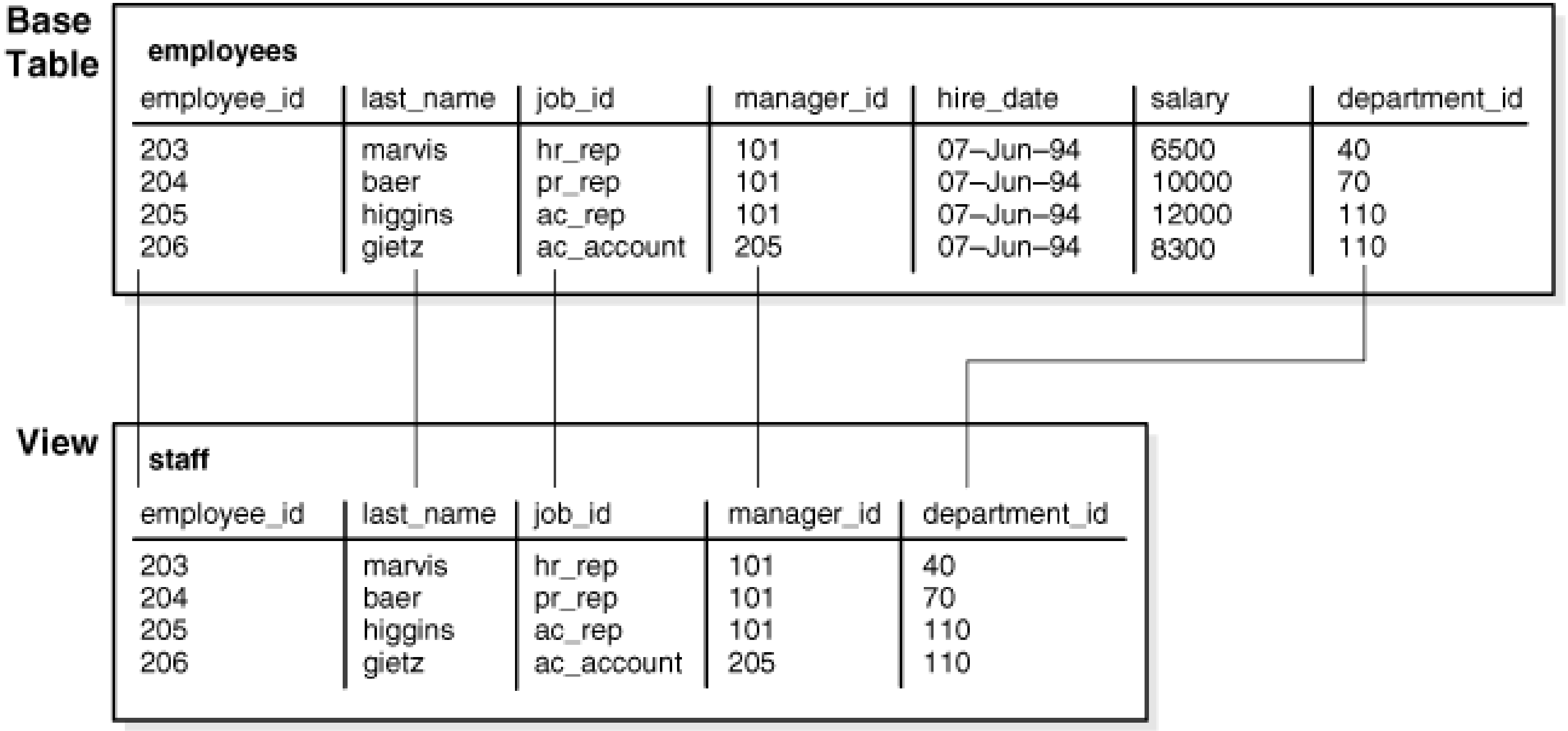
Tables



	ENAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7329	SMITH	CLERK	7902	17–DEC–88	800.00	300.00	20
7499	ALLEN	SALESMAN	7698	20–FEB–88	1600.00	300.00	30
7521	WARD	SALESMAN	7698	22–FEB–88	1250.00	500.00	30
7566	JONES	MANAGER	7839	02–APR–88	2975.00		20

Rows
Columns
Column names
Column not allowing nulls
Column allowing nulls

Views



Base Table

employees

employee_id	last_name	job_id	manager_id	hire_date	salary	department_id
203	marvis	hr_rep	101	07–Jun–94	6500	40
204	baer	pr_rep	101	07–Jun–94	10000	70
205	higgins	ac_rep	101	07–Jun–94	12000	110
206	gietz	ac_account	205	07–Jun–94	8300	110

View

staff

employee_id	last_name	job_id	manager_id	department_id
203	marvis	hr_rep	101	40
204	baer	pr_rep	101	70
205	higgins	ac_rep	101	110
206	gietz	ac_account	205	110

Sequences

Sequence numbers are Oracle integers of up to 38 digits defined in the database. A sequence definition indicates general information, such as the following:

- The name of the sequence
- Whether the sequence ascends or descends
- The interval between numbers
- Whether Oracle should cache sets of generated sequence numbers in memory

Synonyms

A synonym is an alias for any table, view, materialized view, sequence, procedure, function, package, type, Java class schema object, user-defined object type, or another synonym.

Synonyms are often used for security and convenience. For example, they can do the following:

- Mask the name and owner of an object
- Provide location transparency for remote objects of a distributed database
- Simplify SQL statements for database users
- Enable restricted access similar to specialized views when exercising fine-grained access control

Procedures and Functions

- PL/SQL subprograms are named PL/SQL blocks that can be invoked with a set of parameters. PL/SQL provides two kinds of subprograms:

- **Procedures** – These subprograms do not return a value directly; mainly used to perform an action.

- **Functions** – These subprograms return a single value; mainly used to compute and return a value.

Procedures and Functions

Parts & Description
Declarative Part It is an optional part. However, the declarative part for a subprogram does not start with the DECLARE keyword. It contains declarations of types, cursors, constants, variables, exceptions, and nested subprograms. These items are local to the subprogram and cease to exist when the subprogram completes execution.
Executable Part This is a mandatory part and contains statements that perform the designated action.
Exception-handling This is again an optional part. It contains the code that handles run-time errors.

Create schema / user

1. Create Tablespace
2. Create Datafile
3. Create user
4. Grant Privileges
5. Create objects

Create tablespace command

To create a tablespace you must have the CREATE TABLESPACE privilege.

```
CREATE TABLESPACE tablespace
  [DATAFILE clause]
  [MINIMUM EXTENT integer[K|M]]
  [SIZE integer [K]]
  [LOGGING|NOLOGGING]
  [DEFAULT storage_clause ]
  [ONLINE|OFFLINE]
  [PERMANENT|TEMPORARY]
```

Create tablespace command

TABLESPACE: This clause specifies the tablespace name.

DATAFILE: This clause names the one or more datafiles that will comprise the tablespace and includes the full path, example:

```
DATAFILE '/u01/student/dbockstd/oradata/USER350data01.dbf' SIZE 10M
```

Create tablespace command

MINIMUM EXTENT: Every used extent for the tablespace will be a multiple of this integer value. Use either T, G, M or K to specify terabytes, gigabytes, megabytes, or kilobytes.

SIZE: This specifies the size of the tablespace - Use either T, G, M or K to specify terabytes, gigabytes, megabytes, or kilobytes.

LOGGING: This is the default – all tables, indexes, and partitions within a tablespace have modifications written to Online Redo Logs.

NOLOGGING: This option is the opposite of LOGGING and is used most often when large direct loads of clean data are done during database creation for systems that are being ported from another file system or DBMS to Oracle.

DEFAULT storage_clause: This specifies default parameters for objects created inside the tablespace. Individual storage clauses can be used when objects are created to override the specified DEFAULT.

Create tablespace command

OFFLINE: This parameter causes a tablespace to be unavailable after creation.

PERMANENT: A permanent tablespace can hold permanent database objects.

TEMPORARY: A temporary tablespace can hold temporary database objects, e.g., segments created during sorts as a result of ORDER BY clauses or JOIN views of multiple tables.

Create user command

```
CREATE USER Scott
    IDENTIFIED BY New_Pa$$w0rd
    DEFAULT TABLESPACE Users
    TEMPORARY TABLESPACE Temp
    QUOTA 10M ON Users
    QUOTA 5M ON Data01
    ACCOUNT UNLOCK
    PASSWORD EXPIRE;
```

Create user command

- Scott has two tablespaces identified, one for DEFAULT storage of objects and one for TEMPORARY objects. If not identified the default tablespace will be the SYSTEM tablespace

- Scott has a quota set on 2 tablespaces.

- Scott has the resource limitations allocated by the PROFILE named accountant. The account is unlocked (the default – alternatively the account could be created initially with the LOCK specification)

- The PASSWORD EXPIRE clause requires Scott to change the password prior to connecting to the database. After the password is set, when the user logs on using SQLPlus or any other software product that connects to the database, the user receives the following message at logon, and is prompted to enter a new password.

Alter user command

```
ALTER USER Scott
   IDENTIFIED by New_Pa$$w0rd
   DEFAULT TABLESPACE Data01
   TEMPORARY TABLESPACE Temp
   QUOTA 100M ON Data01
   QUOTA 0 ON Inventory_TBS;
```

Drop user command

```
DROP USER User105;
DROP USER Scott CASCADE;
```

Dropping a user causes the user and the user schema to be immediately deleted from the database.

If the user has created objects within their schema, it is necessary to use the CASCADE option in order to drop a user.

If you fail to specify CASCADE when user objects exist, an error message is generated and the user is not dropped.

In order for a DBA to drop a user, the DBA must have the DROP USER system privilege.

GRANT CONNECT USER command

```
GRANT CONNECT TO USER;
```

It is necessary to add a grant to the user connect to the database.

GRANT roles

```
GRANT RESOURCE TO USER;

*GRANT DBA TO USER;
```

Data Dictionary Tables for User Accounts

The only data dictionary table used by a DBA for user account information is DBA_USERS.

```
SELECT username, account_status, default_tablespace FROM dba_users;
```

```
USERNAME  ACCOUNT_STATUS                DEFAULT_TABLESPACE
---------------               --------------------  -------------------
OUTLN                         OPEN                            SYSTEM
USER350                       OPEN                            USERS
DBOCK                         OPEN                            DATA01
SYS                           OPEN                            SYSTEM
SYSTEM                        OPEN                            SYSTEM
USER349                       EXPIRED                         SYSTEM
SCOTT                         EXPIRED                         USERS
TSMSYS                        EXPIRED & LOCKED                SYSTEM
DIP                           EXPIRED & LOCKED                SYSTEM
DBSNMP                        EXPIRED & LOCKED                SYSAUX
ORACLE_OCM                    EXPIRED & LOCKED                SYSTEM
```

Oracle Schema – Criação e administração de um *schema*

AULA PL06

Hugo Peixoto

2018 – 2019 Universidade do Minho