Institución Universitaria
Acreditada en Alta Calidad

ANDRES
MARTINEZ
GUTIERREZ

2018-2

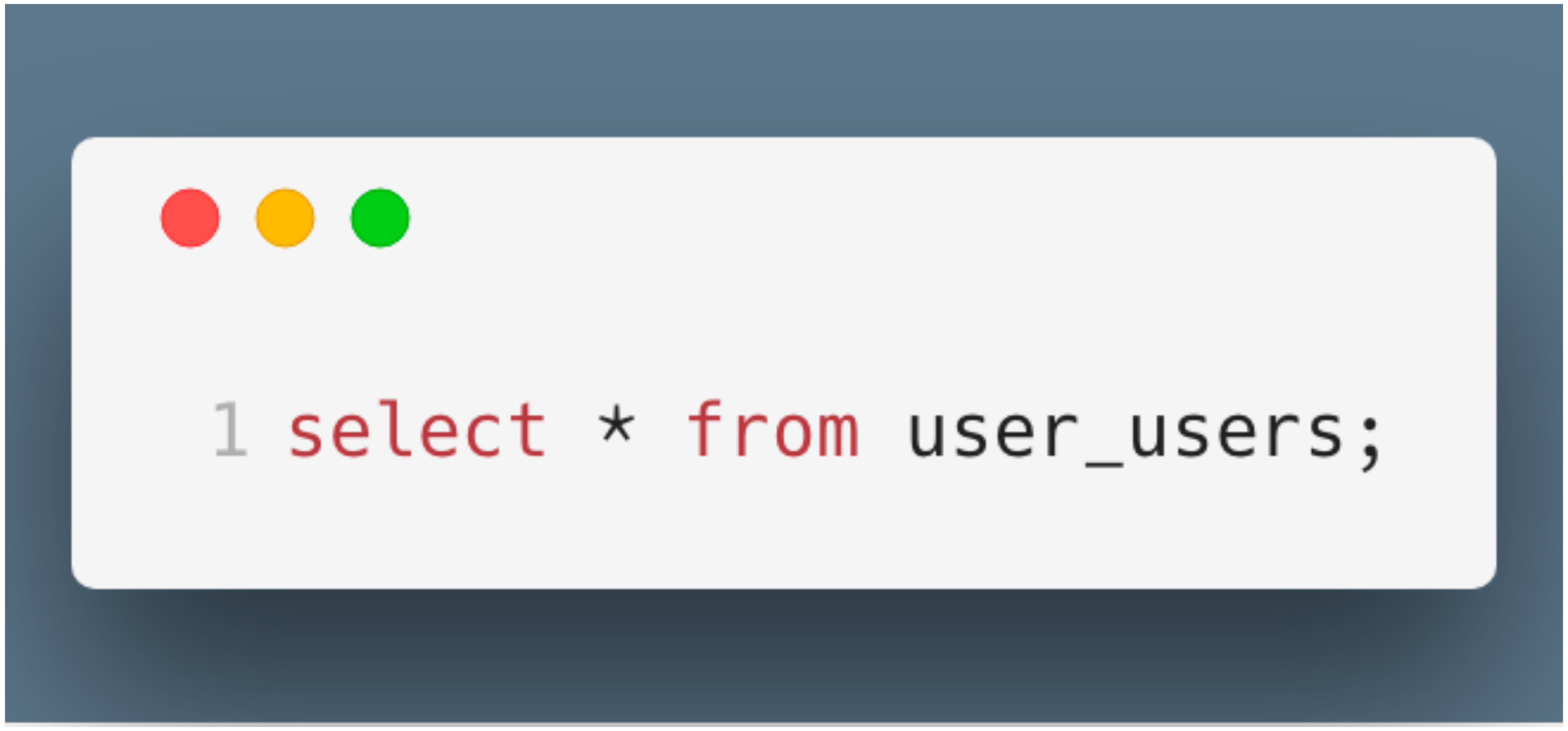
DATABASE ADMINISTRATION ADVANCED

AGENDA

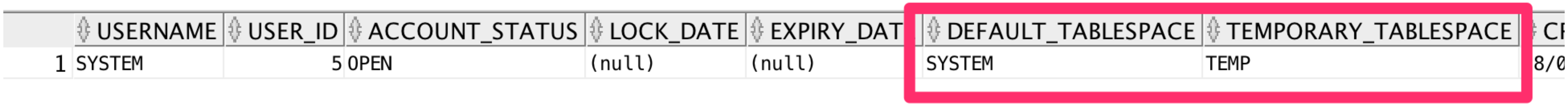
- Tablespaces
  - Datafiles - Control files
  - Types of tablespaces
  - Managing space
- Administration of users and privileges

Tablespaces

https://docs.oracle.com/cd/B19306_01/server.102/b14220/physical.htm
https://docs.oracle.com/cd/B19306_01/server.102/b14200/statements_7003.htm



```
select * from user_users;
```



	USERNAME	USER_ID	ACCOUNT_STATUS	LOCK_DATE	EXPIRY_DAT	DEFAULT_TABLESPACE	TEMPORARY_TABLESPACE	CR
1	SYSTEM	5	OPEN	(null)	(null)	SYSTEM	TEMP	8/0

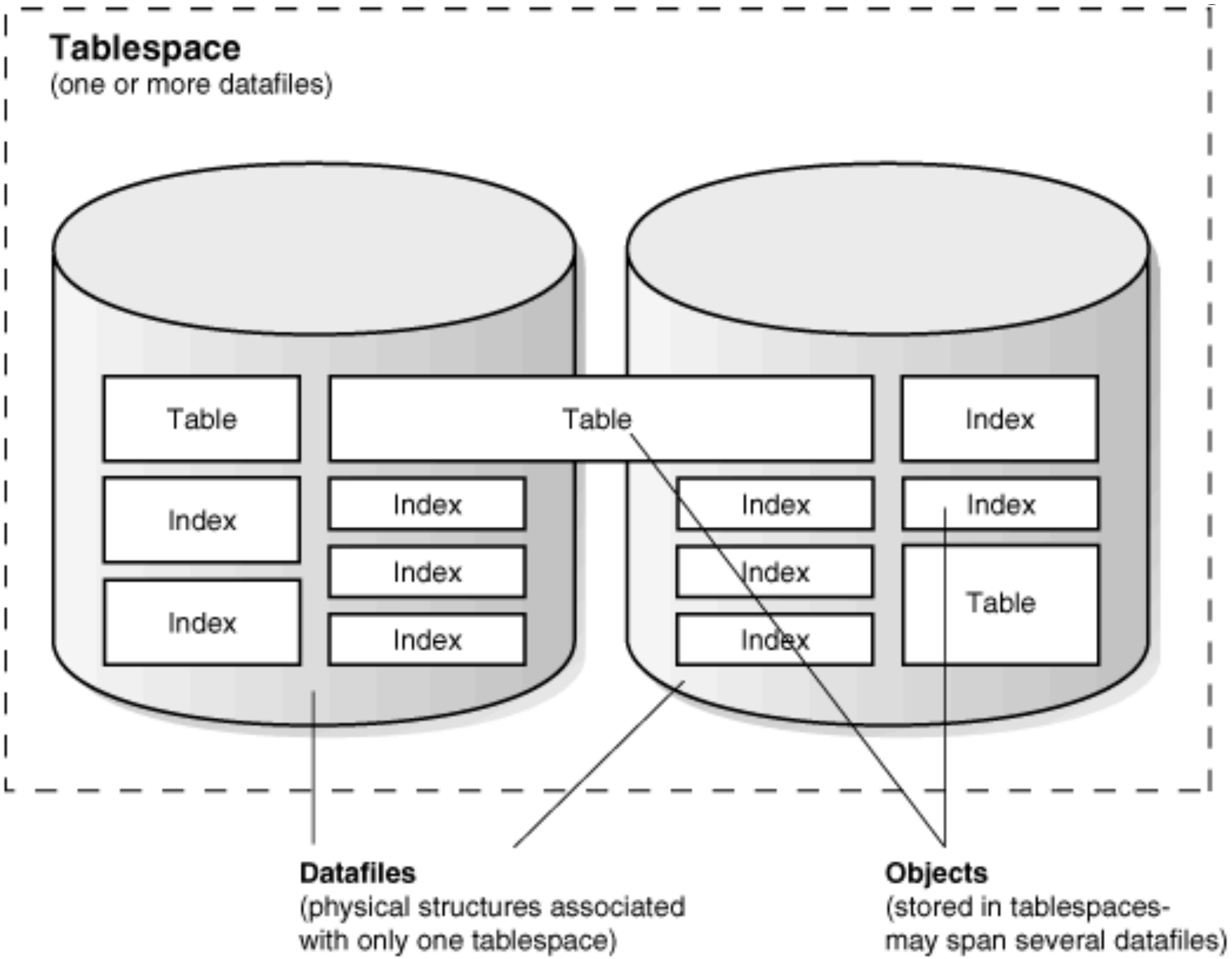
Tablespaces

- Primary logical database structure of any Oracle database.
  - **Tablespace -> Logical**
  - **Datafile -> Physical**

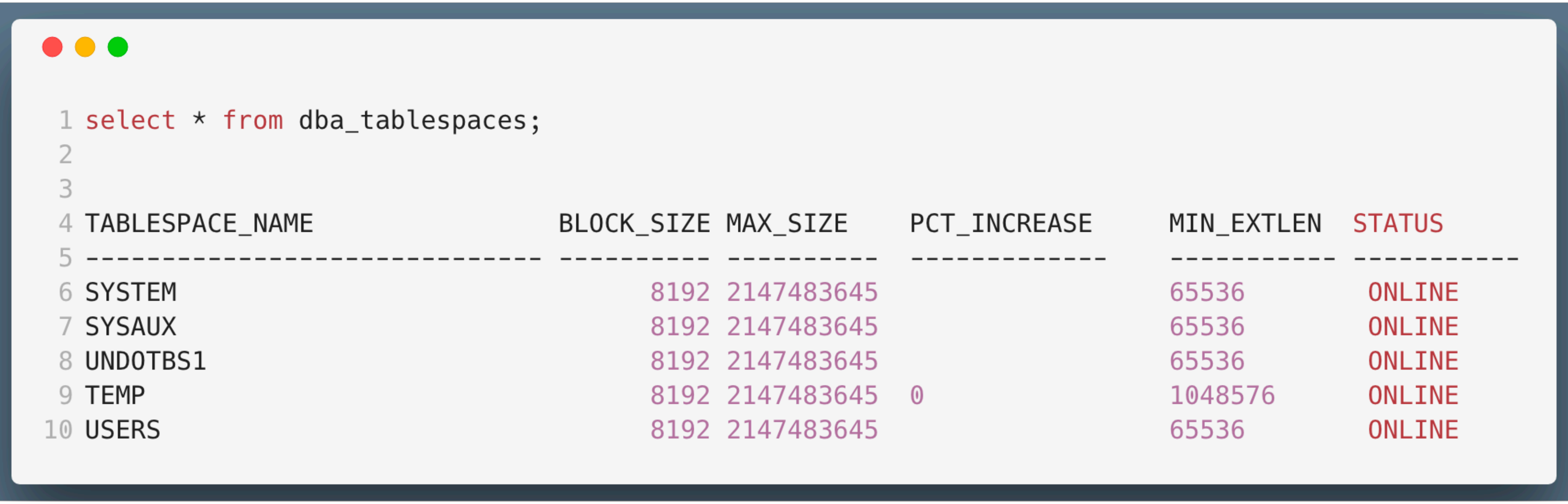
- Oracle database = 1 or more **Tablespaces**

Tablespaces

- Each table space has at least 1 datafile:
  - 1 Tablespace - 1 datafile (Minimum)
  - 3 Tablespaces - each 2 datafiles (6 datafiles)

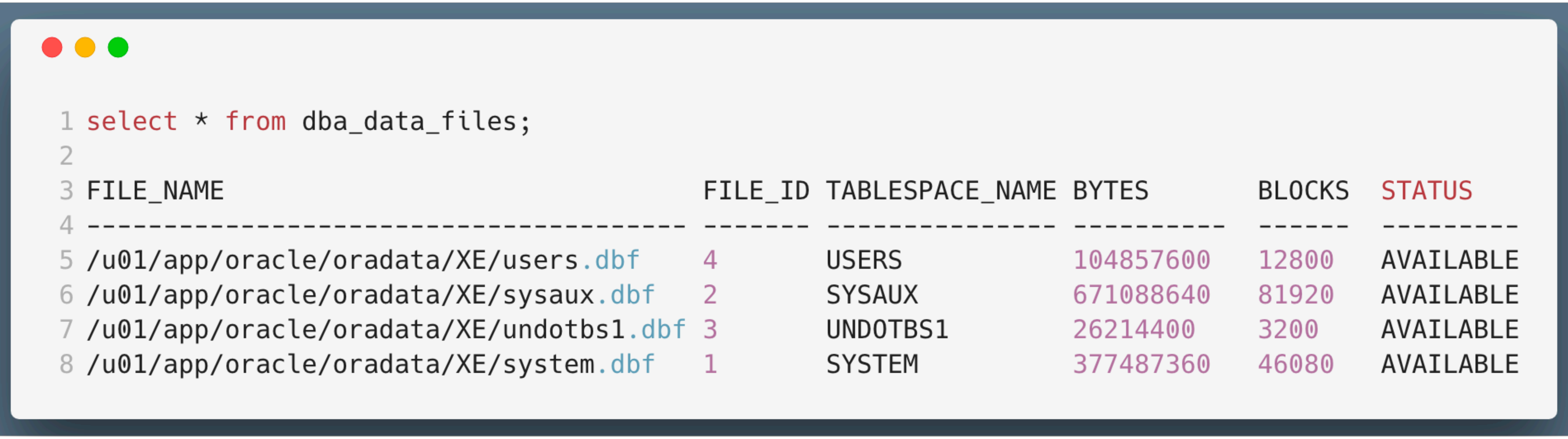

Tablespace
(one or more datafiles)
Table
Table
Index
Index
Index
Index
Index
Index
Index
Index
Index
Index
Index
Table
Datafiles
(physical structures associated
with only one tablespace)
Objects
(stored in tablespaces-
may span several datafiles)

How do I list all tablespaces?



```
select * from dba_tablespaces;


TABLESPACE_NAME                BLOCK_SIZE MAX_SIZE    PCT_INCREASE     MIN_EXTLEN  STATUS
------------------------------ ---------- ----------  -------------    ----------- -----------
SYSTEM                               8192 2147483645                   65536        ONLINE
SYSAUX                               8192 2147483645                   65536        ONLINE
UNDOTBS1                             8192 2147483645                   65536        ONLINE
TEMP                                 8192 2147483645  0                1048576      ONLINE
USERS                                8192 2147483645                   65536        ONLINE
```

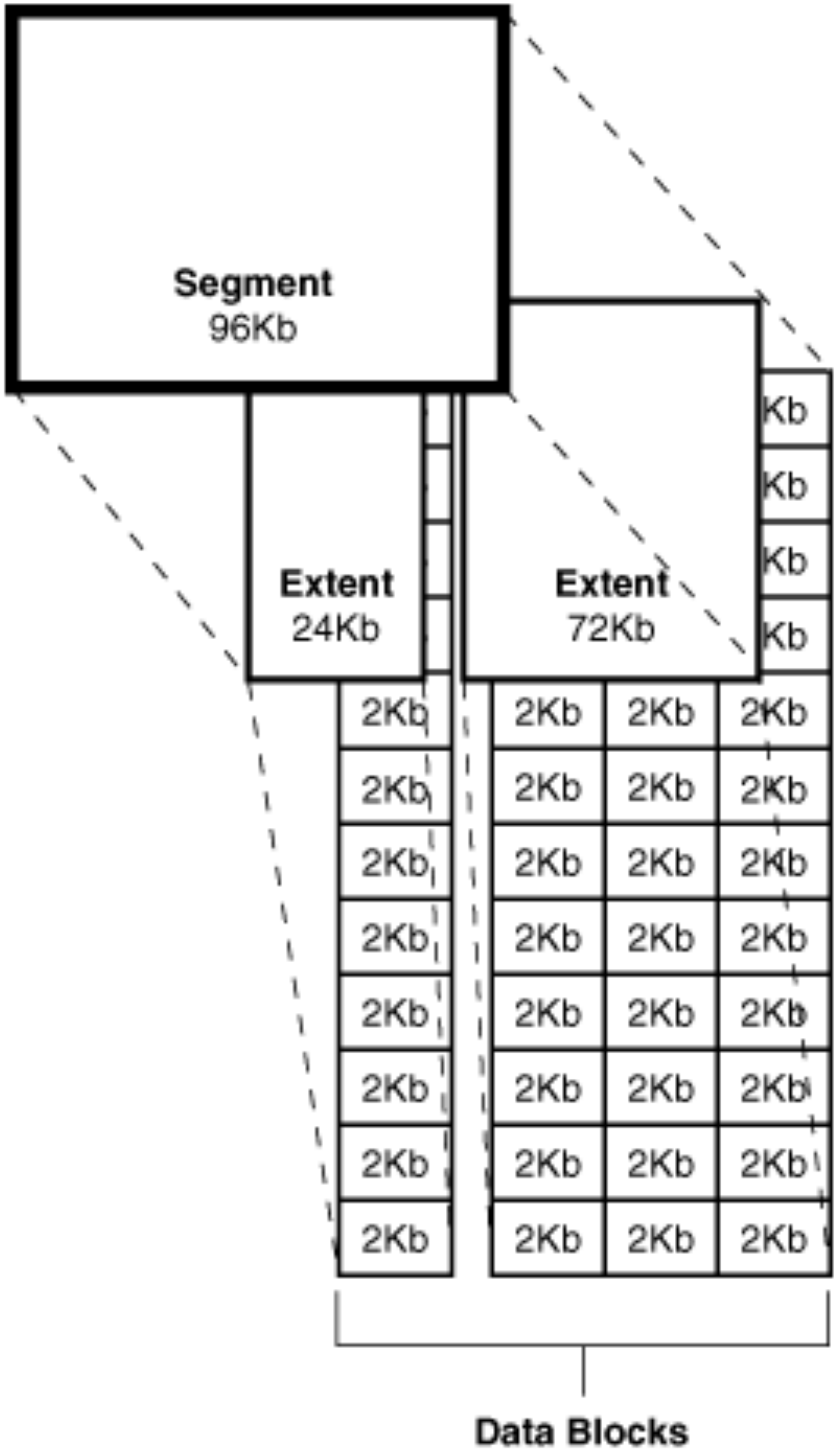
How do I list all datafiles?



```
select * from dba_data_files;

FILE_NAME                               FILE_ID TABLESPACE_NAME BYTES       BLOCKS  STATUS
--------------------------------------- ------- --------------- ----------  ------  ---------
/u01/app/oracle/oradata/XE/users.dbf    4       USERS           104857600   12800   AVAILABLE
/u01/app/oracle/oradata/XE/sysaux.dbf   2       SYSAUX          671088640   81920   AVAILABLE
/u01/app/oracle/oradata/XE/undotbs1.dbf 3       UNDOTBS1        26214400    3200    AVAILABLE
/u01/app/oracle/oradata/XE/system.dbf   1       SYSTEM          377487360   46080   AVAILABLE
```

Tablespaces’ Overview

- **Database:** One or more logical storage units (**Tablespaces**)
- Tablespaces are divided into logical units of storage (**segments**)
- Segments are divided into (**extents**)
- Extents are a collection of contiguos (**blocks**)



Segment
96Kb
Extent
24Kb
Extent
72Kb
2Kb
Data Blocks

The size of a tablespace is the size of the datafiles.

The size of the database is the collective size of the tablasepaces

Types of table spaces

Types of tablespaces

- Bigfile Tablespaces
- SYSTEM tablespace
- SYSAUX tablespace
- Undo tablespace
- Default temporary tablespace
- Read-only tablespace

Bigfile tablespaces

- Ultralarge files - 64 bits
- Up to 8 exabytes
- ***<u>One datafile</u>***
- Simplifies data management
- Smallfile tablespaces can contain up to 1024 files (datafiles)
- A Bigfile tablespace can contain only one file that can be 1024 times larger than a small file tablespace.

System tablespace

- Oracle creates automatically when the database is created

- Always online

- Contains the data dictionary tables for the entire database (Stored in “datafile 1”)

- PL/SQL (Procedures, functions, packages, triggers)

SYSAUX tablespace

- Auxiliary tablespace to the SYSTEM tablespace
- Always created during database creation / upgrade
- Centralized location for database metadata that does not reside in the SYSTEM tablespace
- Not removable nor droppable

UNDO tablespace

- Stores undo information (rollback, transactions, recover database, recover from logical corruptions)

- It is not possible to create tables or indexes

- DML operations within transactions

- Each Oracle instance is assigned one (and only one) undo tablespace

Read-Only tablespace

- Eliminates the need to perform backup and recovery of
- large, static portions of a database

- Cannot be modified

- If you recover a database, is not necessary to recover
- read-only tablespaces

Temporary tablespace

- Data persists only for the duration of the session

- Improve the concurrency of multiple sort operations

- Avoid space management operations

Temporary tablespace

- All operations that use sorts, including joins, index builds, ordering, computing aggregates (GROUP BY), and collecting optimizer statistics, benefit from temporary tablespaces

- The performance gains are significant with Real Application Clusters.

Recommendations

Oracle recommends that you create ***at least one additional*** *tablespace to store user data separate from data dictionary information*

Create few tablespaces as possible ***1 or 2 with*** *auto-extent enable datafiles rather than many small datafiles.*

Managing Space in tablespaces

Managing space

- Tablespaces allocate space in extents

- There are two methods to **keep track of their free and used space:**

  1. Locally managed

  2. Dictionary managed

Locally managed

- Extents management by the tablespace

- Maintains a bitmap in each datafile to keep track of the free or used status of blocks.

- Oracle changes the bitmap when a new extent is allocated

Locally managed

- There are two options to specify how free and used space within a segment is to be managed:
  - **Auto:** Use bitmaps (Enables Oracle to manage free space more automatically) ***Default.***
  - **Manual:** Use free lists for managing free space within a segment (Lists of data blocks that have space available for inserting rows)

Dictionary managed

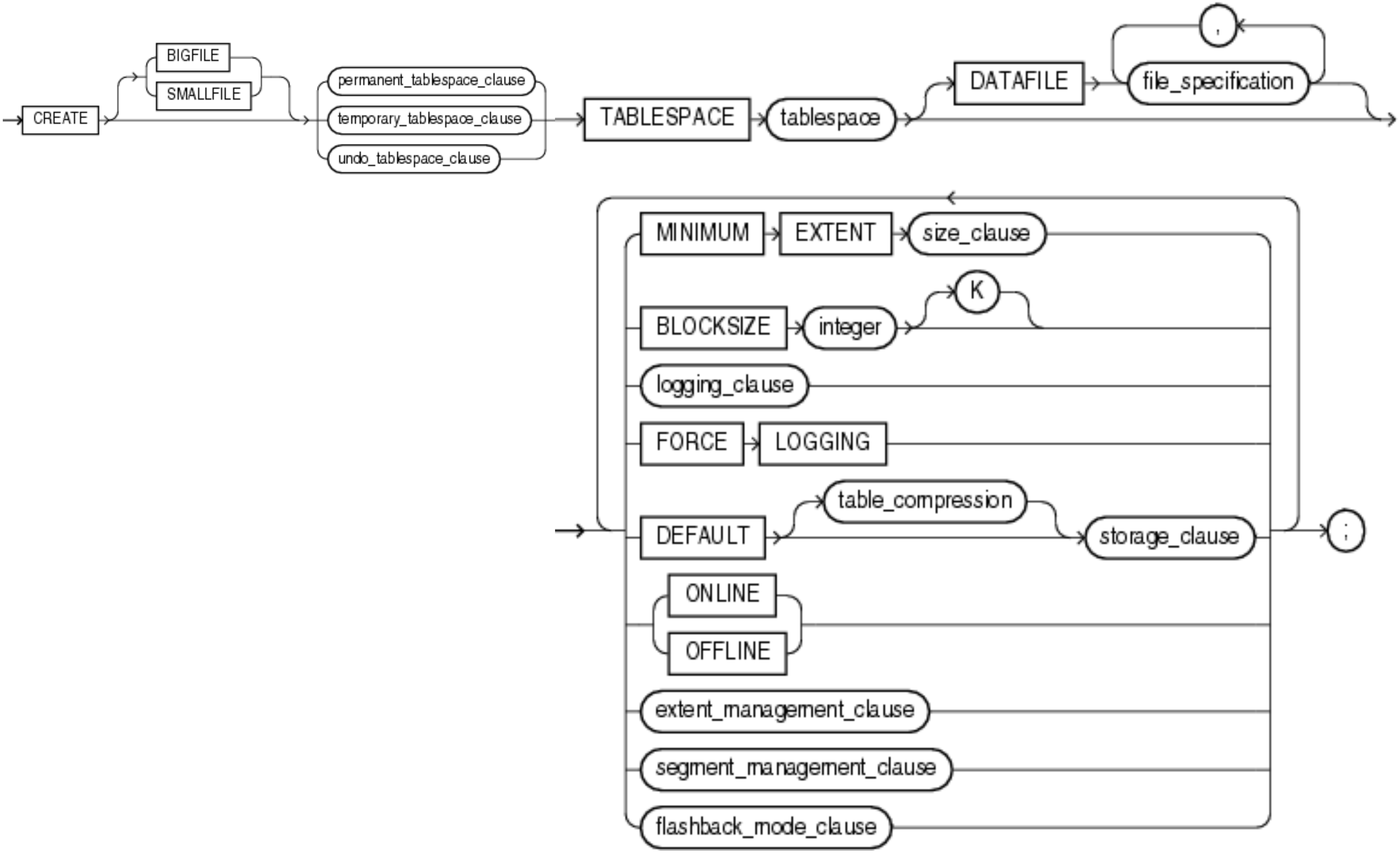
- Used in earlier versions of Oracle

- Oracle updates the appropriate tables in the data dictionary whenever an extent is allocated or free for reuse

- Oracle stores rollback information about each update of the dictionary tables

How to create tablespaces

https://gist.github.com/amartinezg/e8dc391e3bcfb565da9ea707cba628aa


CREATE
BIGFILE
SMALLFILE
permanent_tablespace_clause
temporary_tablespace_clause
undo_tablespace_clause
TABLESPACE
tablespace
DATAFILE
,
file_specification
MINIMUM
EXTENT
size_clause
BLOCKSIZE
integer
K
logging_clause
FORCE
LOGGING
DEFAULT
table_compression
storage_clause
ONLINE
OFFLINE
extent_management_clause
segment_management_clause
flashback_mode_clause
;

Creating tablespaces

- You use CREATE TABLESPACE command

- You can not rename or drop the SYSTEM tablespace or take it to offline.

- You must have CREATE TABLESPACE (ALTER TABLESPACE) system privilege

Considerations

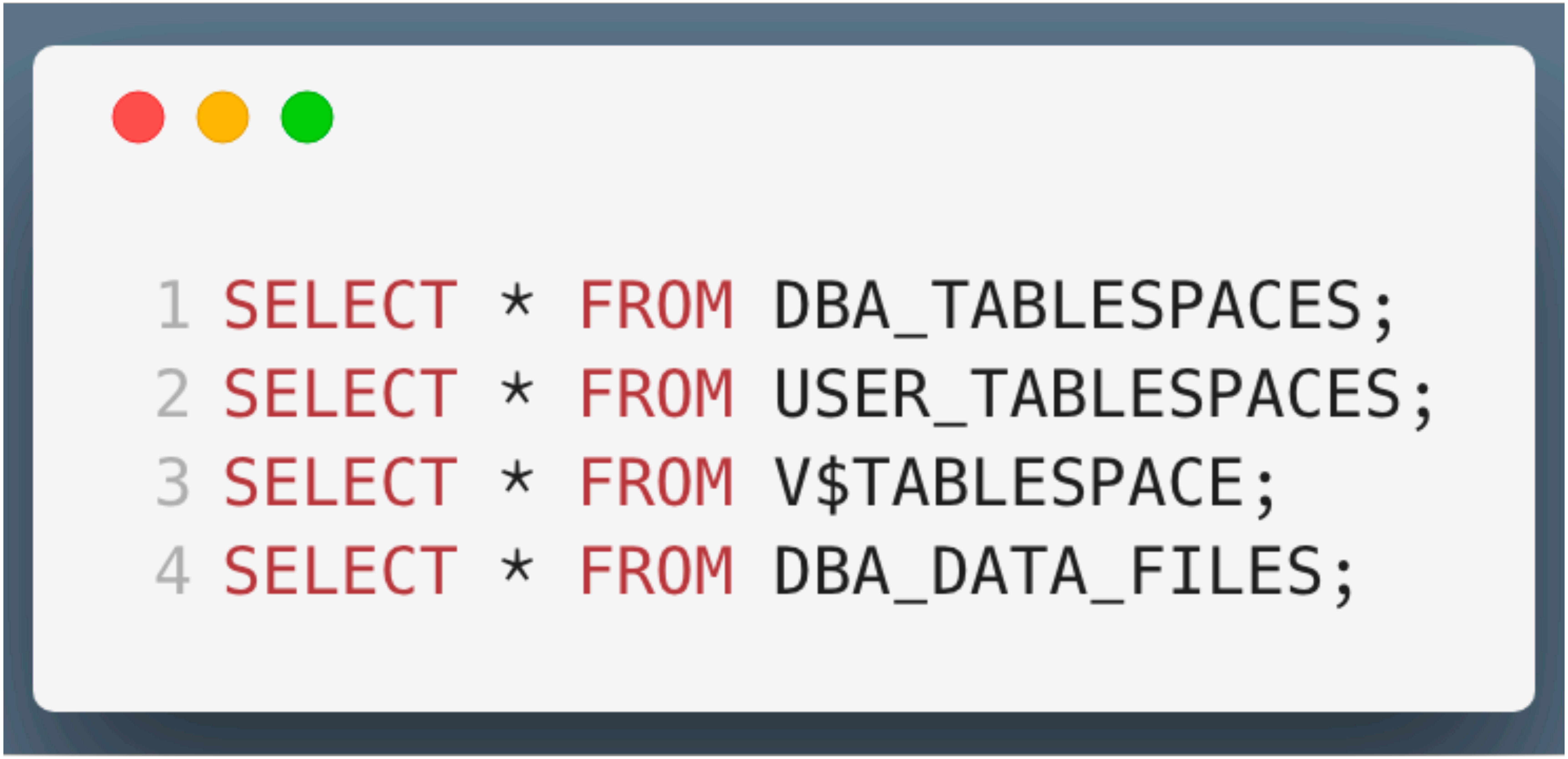
- You create tablespaces with **CREATE TABLESPACE**

- You drop tablespaces with **DROP TABLESPACE…**

- You alter a tablespace with **ALTER TABLESPACE**
    - Take the tablespace OFFLINE / ONLINE, ADD DATAFILES / TEMPFILES, READ ONLY.

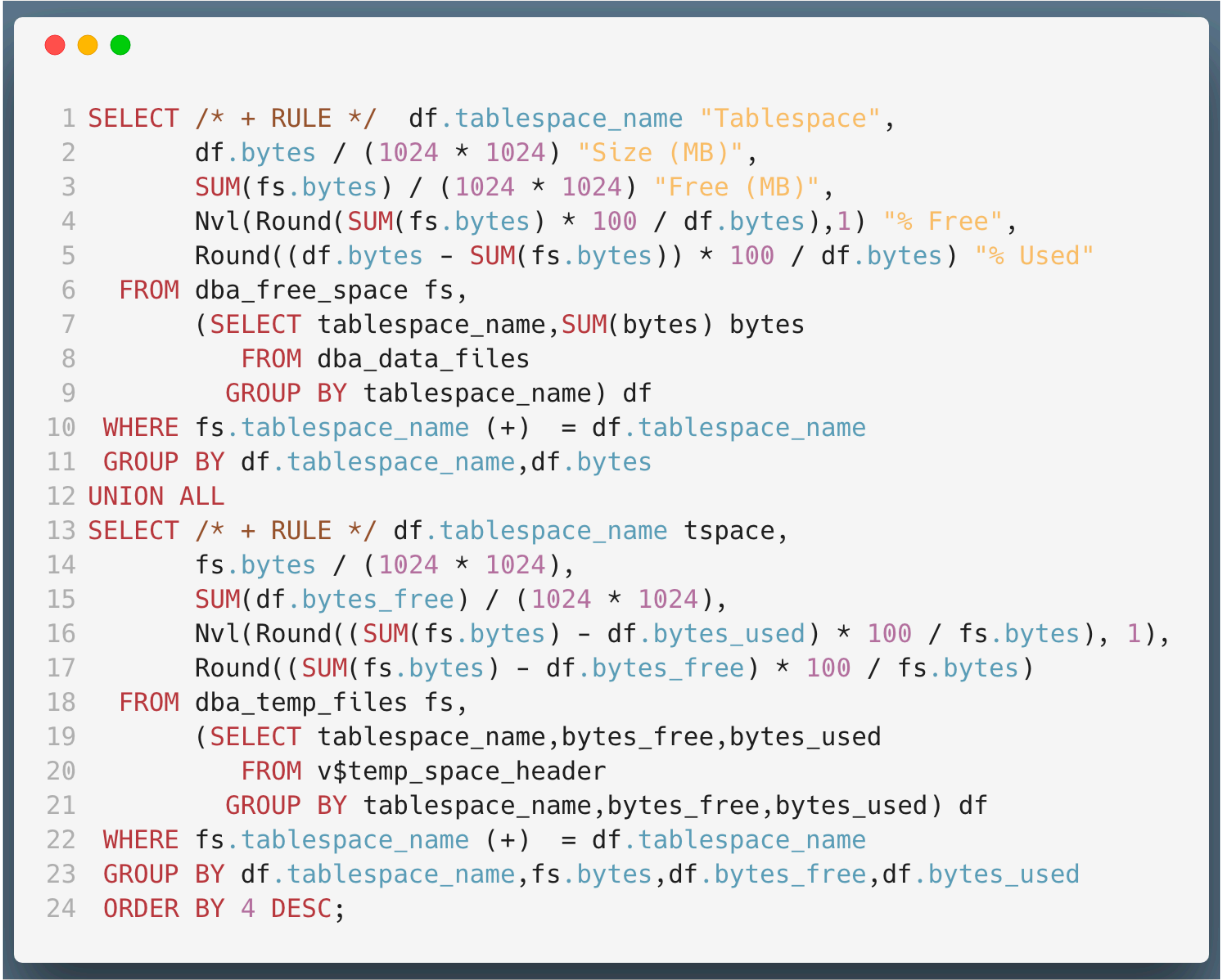
- You must have the **CREATE TABLESPACE** system privilege

Useful views provided by Oracle



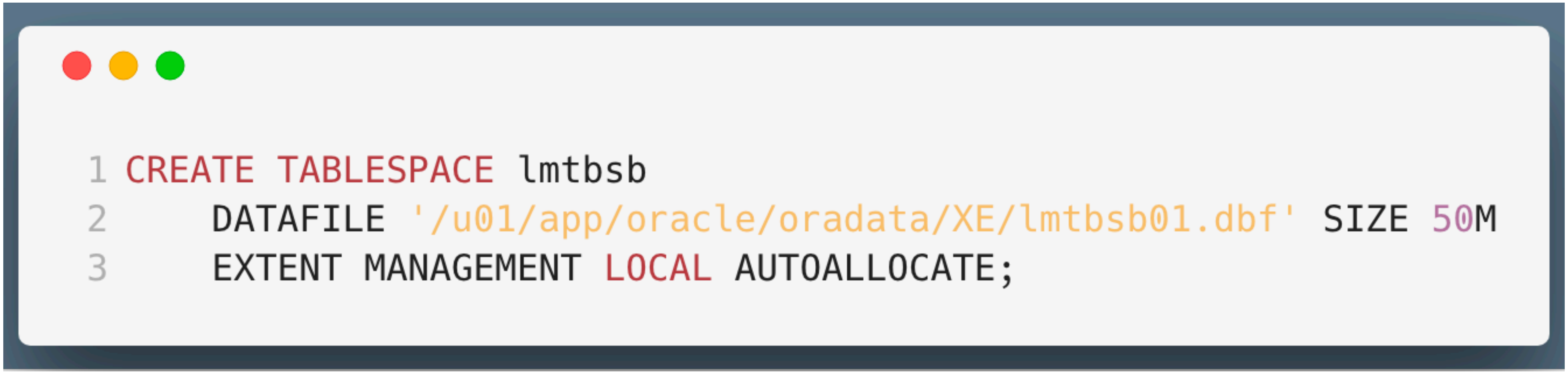
```sql
SELECT * FROM DBA_TABLESPACES;
SELECT * FROM USER_TABLESPACES;
SELECT * FROM V$TABLESPACE;
SELECT * FROM DBA_DATA_FILES;
```

How to check available space and % used - free



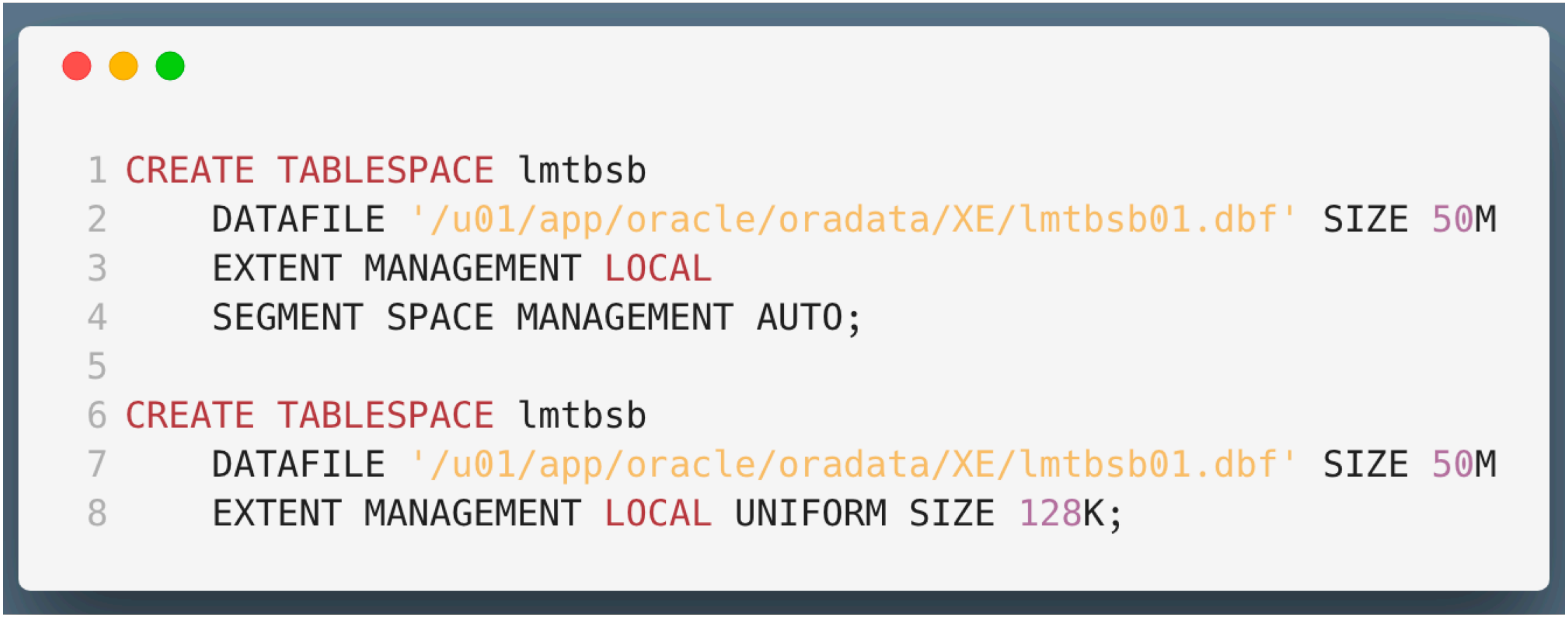
```sql
SELECT /* + RULE */  df.tablespace_name "Tablespace",
       df.bytes / (1024 * 1024) "Size (MB)",
       SUM(fs.bytes) / (1024 * 1024) "Free (MB)",
       Nvl(Round(SUM(fs.bytes) * 100 / df.bytes),1) "% Free",
       Round((df.bytes - SUM(fs.bytes)) * 100 / df.bytes) "% Used"
  FROM dba_free_space fs,
       (SELECT tablespace_name,SUM(bytes) bytes
          FROM dba_data_files
         GROUP BY tablespace_name) df
 WHERE fs.tablespace_name (+)  = df.tablespace_name
 GROUP BY df.tablespace_name,df.bytes
UNION ALL
SELECT /* + RULE */ df.tablespace_name tspace,
       fs.bytes / (1024 * 1024),
       SUM(df.bytes_free) / (1024 * 1024),
       Nvl(Round((SUM(fs.bytes) - df.bytes_used) * 100 / fs.bytes), 1),
       Round((SUM(fs.bytes) - df.bytes_free) * 100 / fs.bytes)
  FROM dba_temp_files fs,
       (SELECT tablespace_name,bytes_free,bytes_used
          FROM v$temp_space_header
         GROUP BY tablespace_name,bytes_free,bytes_used) df
 WHERE fs.tablespace_name (+)  = df.tablespace_name
 GROUP BY df.tablespace_name,fs.bytes,df.bytes_free,df.bytes_used
 ORDER BY 4 DESC;
```

Specifies AUTOALLOCATE

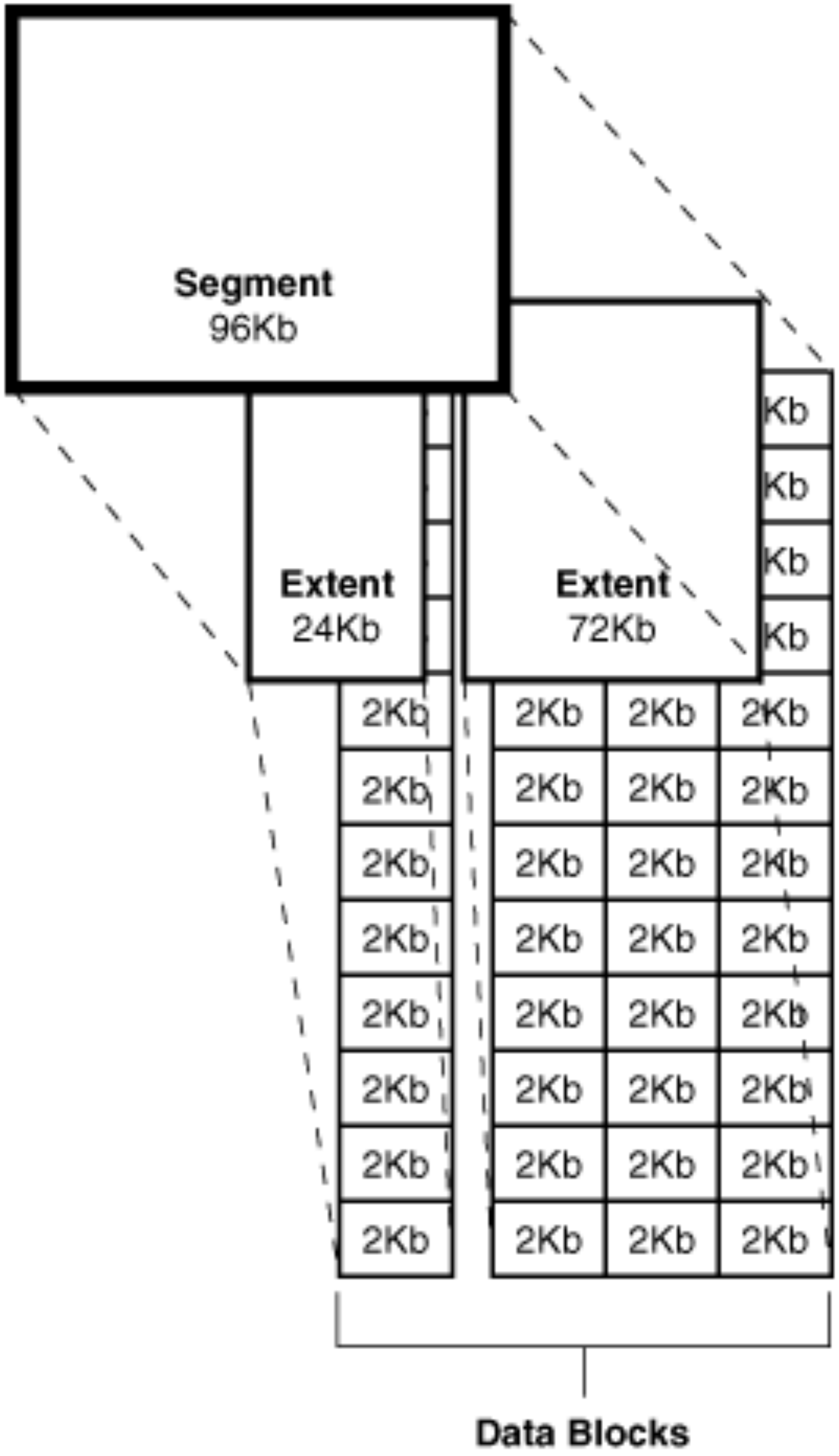


```
CREATE TABLESPACE lmtbsb
    DATAFILE '/u01/app/oracle/oradata/XE/lmtbsb01.dbf' SIZE 50M
    EXTENT MANAGEMENT LOCAL AUTOALLOCATE;
```

Specifies SEGMENT SPACE



```
CREATE TABLESPACE lmtbsb
    DATAFILE '/u01/app/oracle/oradata/XE/lmtbsb01.dbf' SIZE 50M
    EXTENT MANAGEMENT LOCAL
    SEGMENT SPACE MANAGEMENT AUTO;

CREATE TABLESPACE lmtbsb
    DATAFILE '/u01/app/oracle/oradata/XE/lmtbsb01.dbf' SIZE 50M
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 128K;
```


Segment
96Kb
Extent
24Kb
Extent
72Kb
2Kb
Data Blocks

extent_management_clause

- how the extents of the tablespace will be managed:

  - **1. LOCAL:** Manage a bitmap in the header of the first datafile of the tablespace.

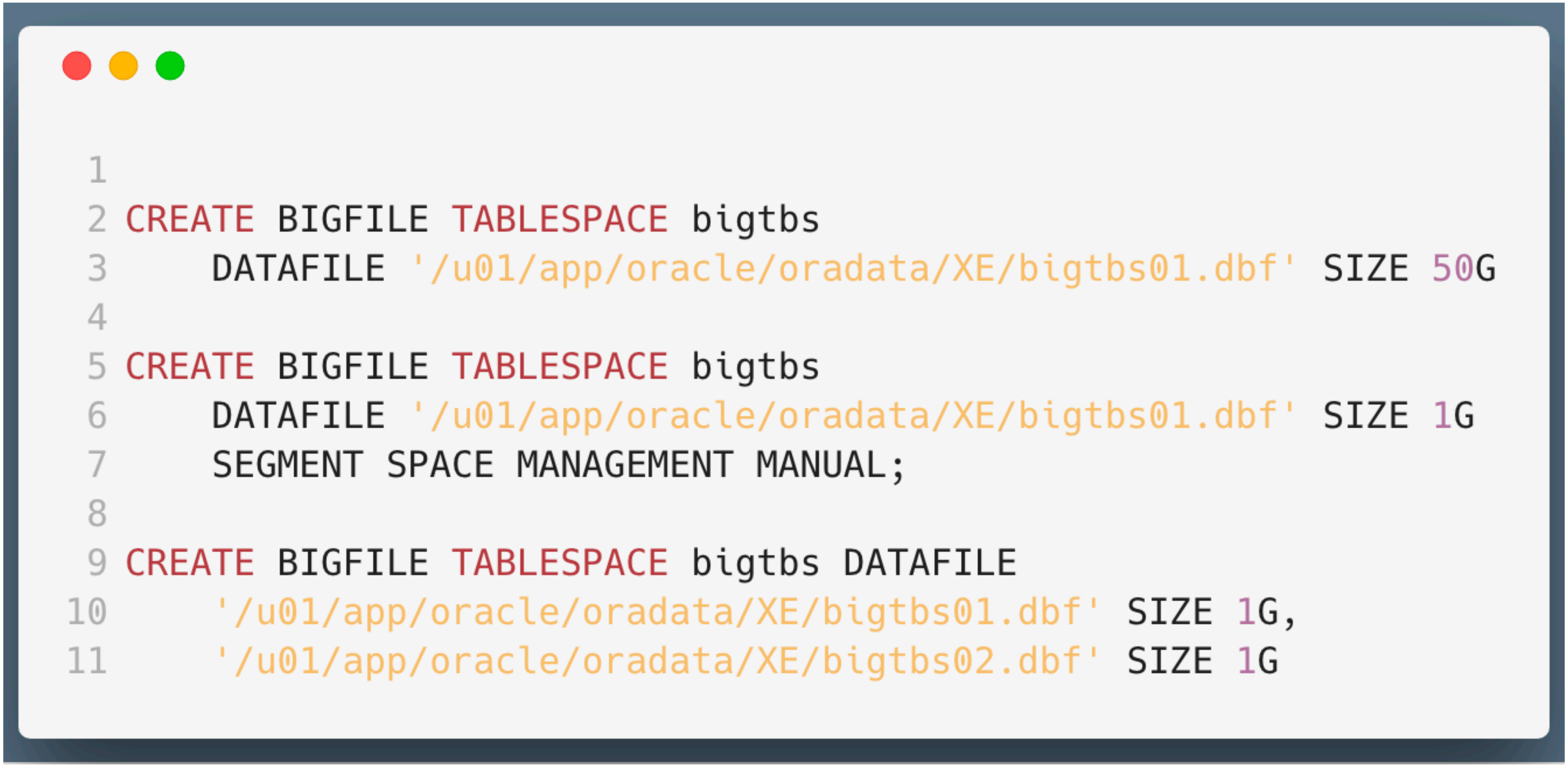
    - **1.1 AUTOALLOCATE:** System managed. User Can’t modify it. (64Kb, 1Mb, 64Mb)
    - **1.2 UNIFORM:** Tablespace is managed with uniform extents of xx bytes. Default size 1Mb

  - **2. DICTIONARY:** Use dictionary tables in SYS schema

segment_management_clause

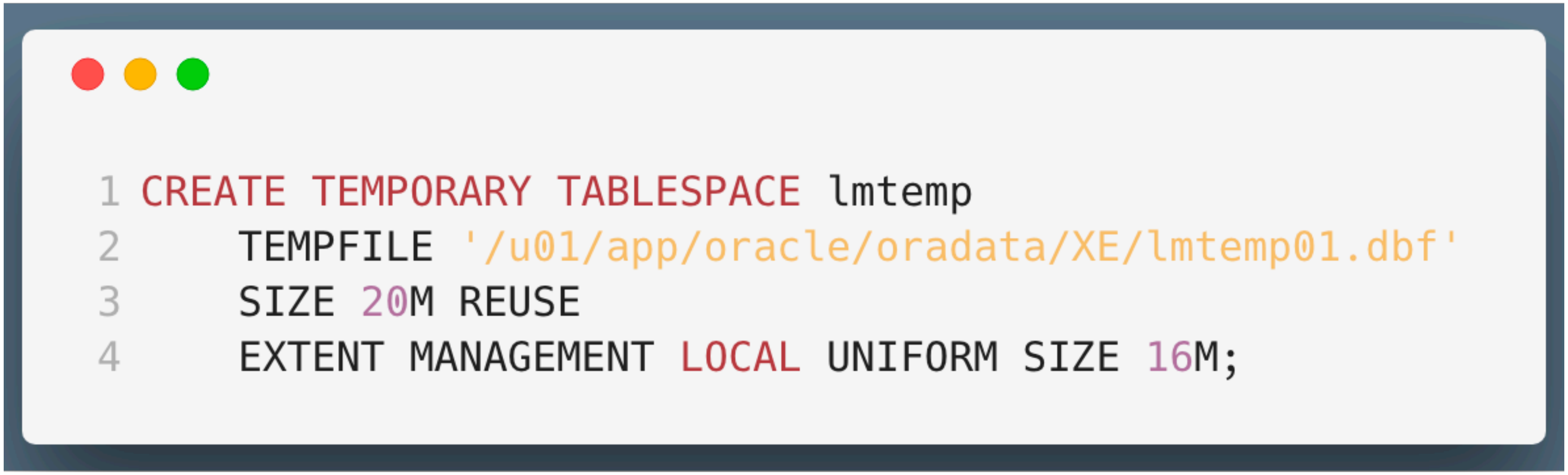
- Relevant only for permanent, **locally managed tablespaces.**
- Track used and free space in segments using:
  - **1. AUTO**: Manage the free space of segments in the tablespace using a **bitmap**
  - **2. MANUAL**: manage the free space of segments in the tablespace using **free lists**. ***Oracle strongly recommends that you do not use this setting and that you create tablespaces with automatic segment-space management.***

Create BIGFILE tablespaces



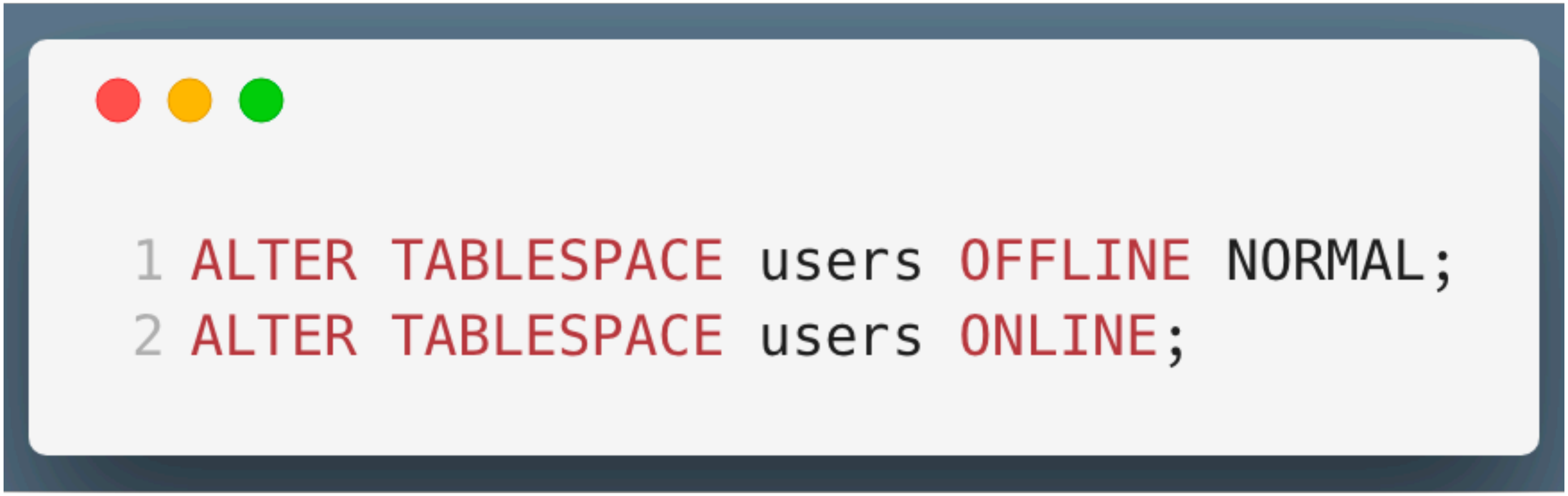
```sql

CREATE BIGFILE TABLESPACE bigtbs
    DATAFILE '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 50G

CREATE BIGFILE TABLESPACE bigtbs
    DATAFILE '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 1G
    SEGMENT SPACE MANAGEMENT MANUAL;

CREATE BIGFILE TABLESPACE bigtbs DATAFILE
    '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 1G,
    '/u01/app/oracle/oradata/XE/bigtbs02.dbf' SIZE 1G
```

Temporary tablespaces



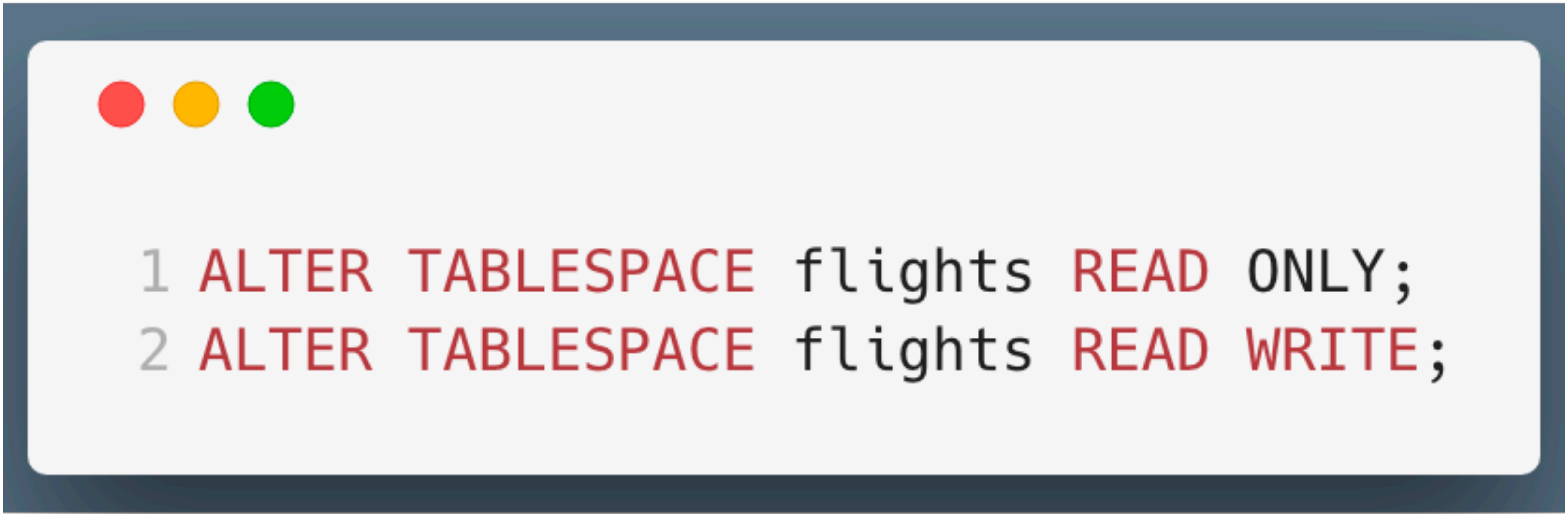
```
CREATE TEMPORARY TABLESPACE lmtemp
     TEMPFILE '/u01/app/oracle/oradata/XE/lmtemp01.dbf'
     SIZE 20M REUSE
     EXTENT MANAGEMENT LOCAL UNIFORM SIZE 16M;
```

Take a tablespace ONLINE or OFFLINE



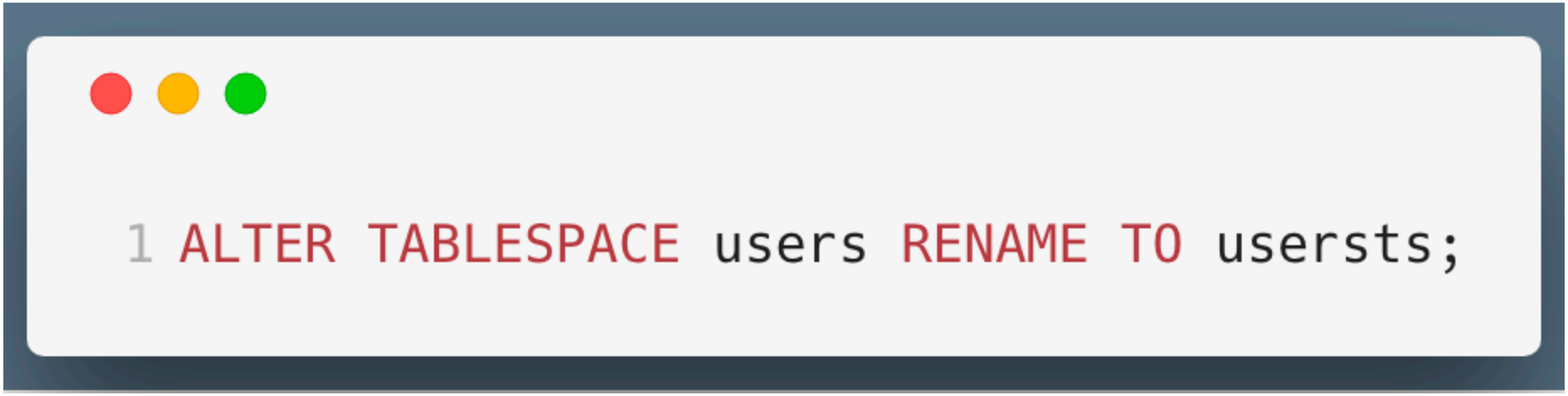
```sql
ALTER TABLESPACE users OFFLINE NORMAL;
ALTER TABLESPACE users ONLINE;
```

Making a Tablespace Read Only



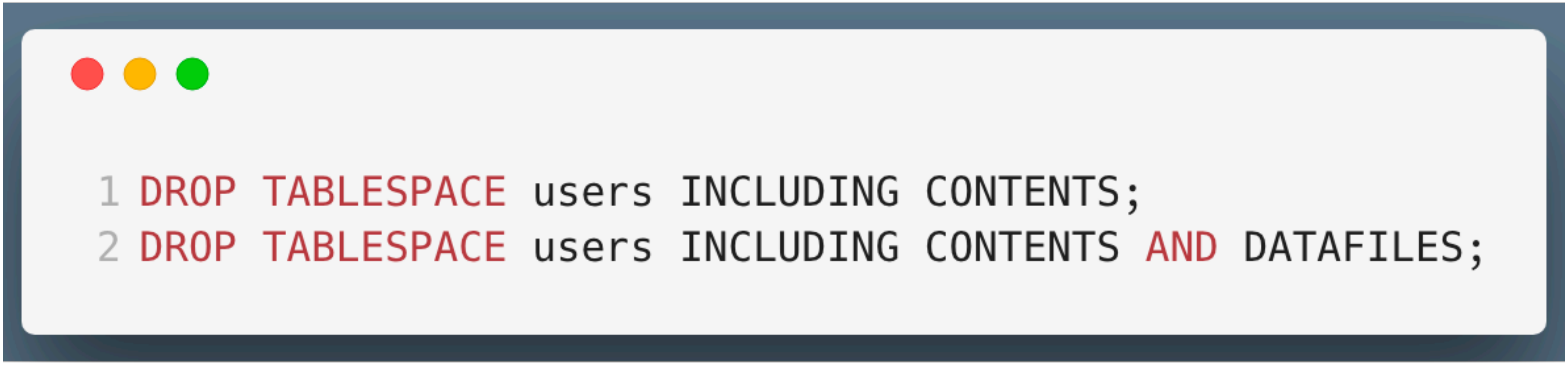
```sql
ALTER TABLESPACE flights READ ONLY;
ALTER TABLESPACE flights READ WRITE;
```

Renaming a tablespace



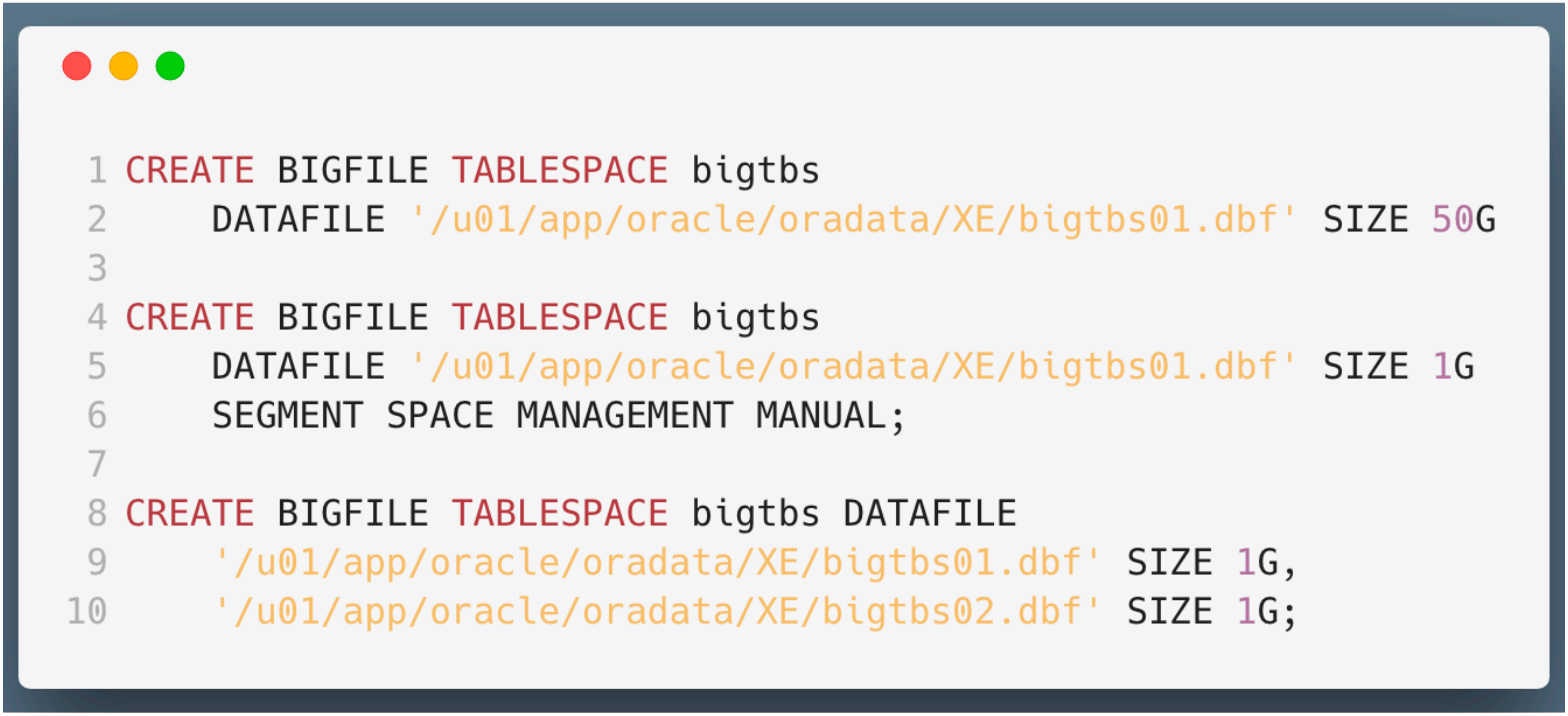
```
ALTER TABLESPACE users RENAME TO usersts;
```

Dropping a tablespace



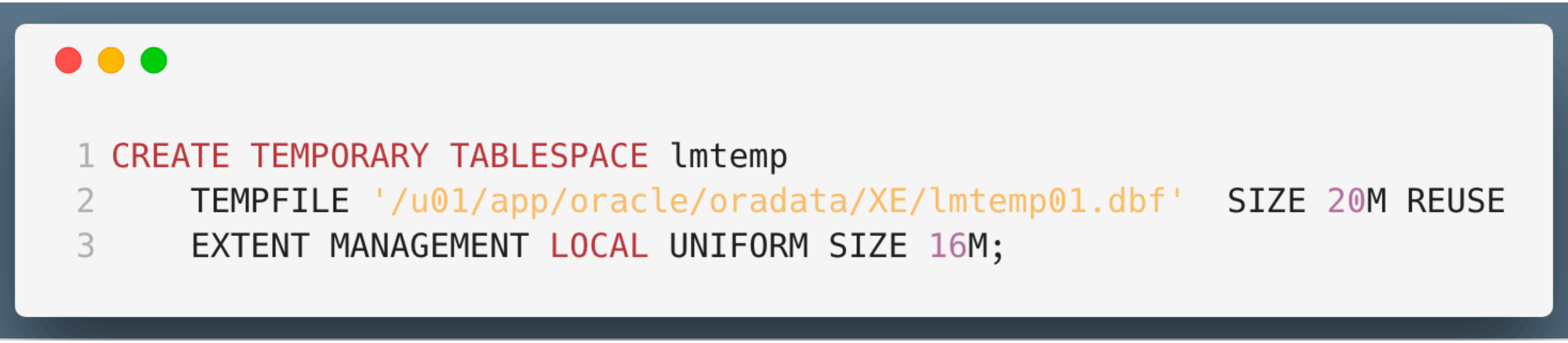
```sql
DROP TABLESPACE users INCLUDING CONTENTS;
DROP TABLESPACE users INCLUDING CONTENTS AND DATAFILES;
```

Creating a Bigfile tablespace



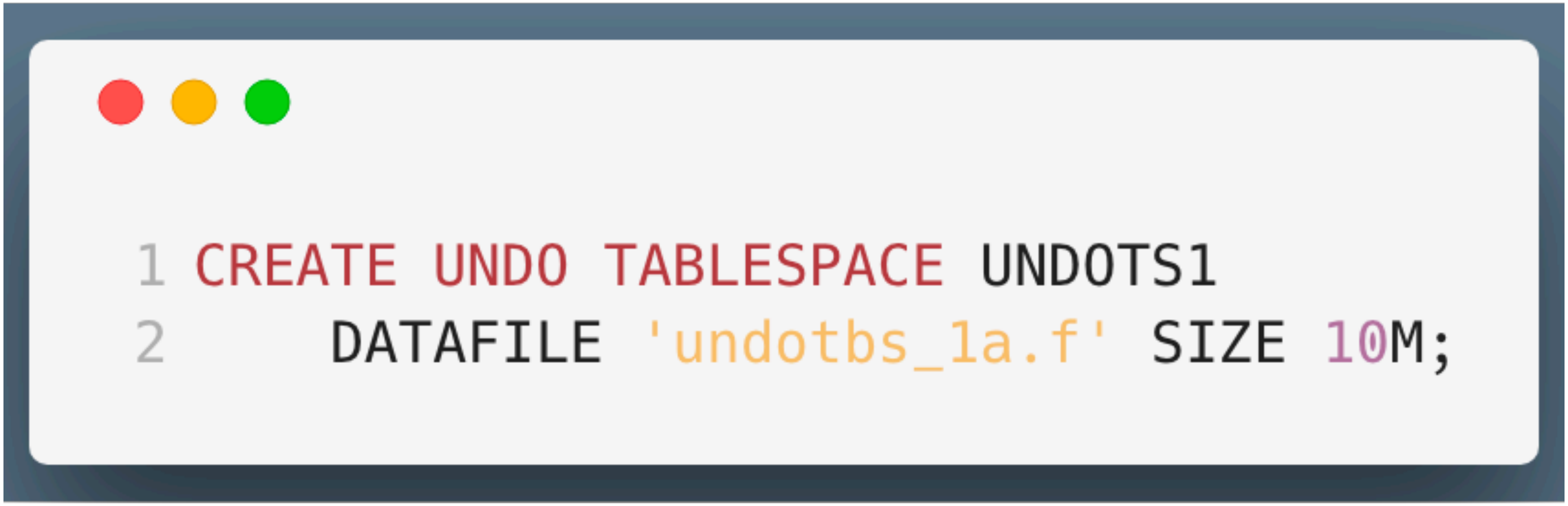
```
CREATE BIGFILE TABLESPACE bigtbs
    DATAFILE '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 50G

CREATE BIGFILE TABLESPACE bigtbs
    DATAFILE '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 1G
    SEGMENT SPACE MANAGEMENT MANUAL;

CREATE BIGFILE TABLESPACE bigtbs DATAFILE
    '/u01/app/oracle/oradata/XE/bigtbs01.dbf' SIZE 1G,
    '/u01/app/oracle/oradata/XE/bigtbs02.dbf' SIZE 1G;
```

Creating a temporary tablespace



```sql
CREATE TEMPORARY TABLESPACE lmtemp
    TEMPFILE '/u01/app/oracle/oradata/XE/lmtemp01.dbf'  SIZE 20M REUSE
    EXTENT MANAGEMENT LOCAL UNIFORM SIZE 16M;
```

Creating an UNDO tablespace



```
CREATE UNDO TABLESPACE UNDOTS1
    DATAFILE 'undotbs_1a.f' SIZE 10M;
```

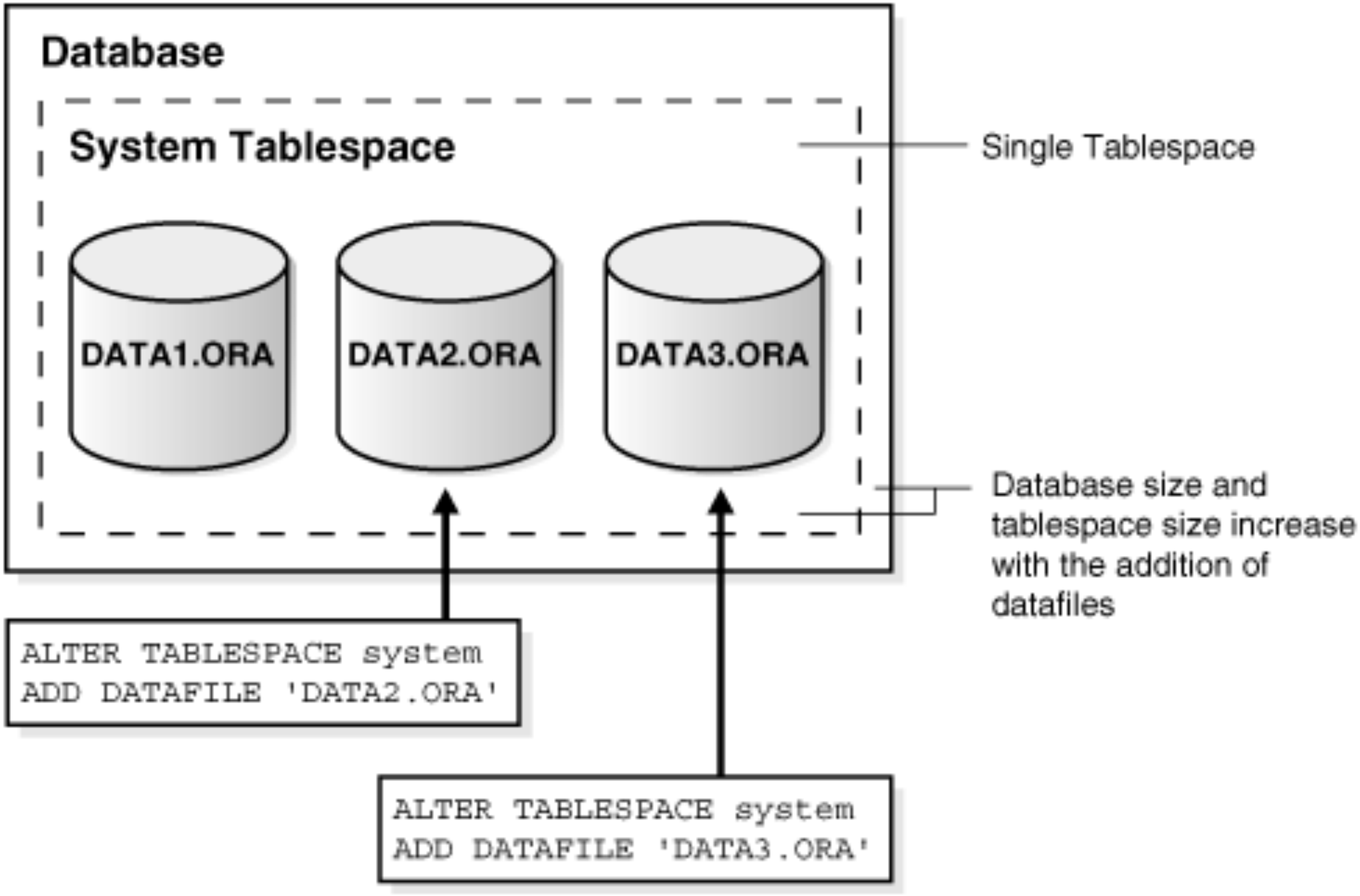
Allocate more space for a Database

Allocate more space

- There are 3 options:

1. Add a datafile to a tablespace
2. Add a new tablespace
3. Increase a size of a datafile

Add a datafile to a tablespace



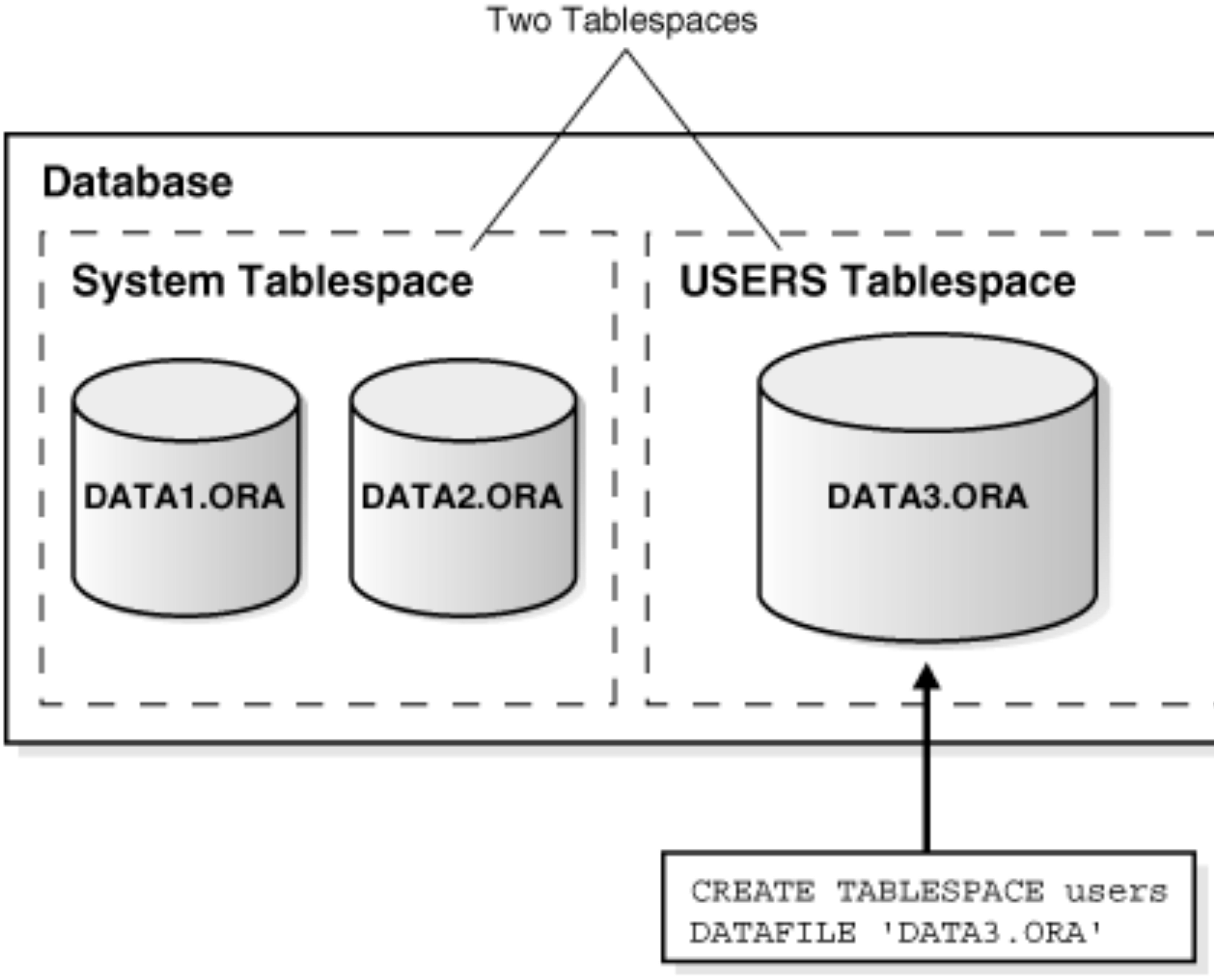
Single Tablespace

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'
```

Add a new tablespace



Two Tablespaces

Database

System Tablespace

DATA1.ORA

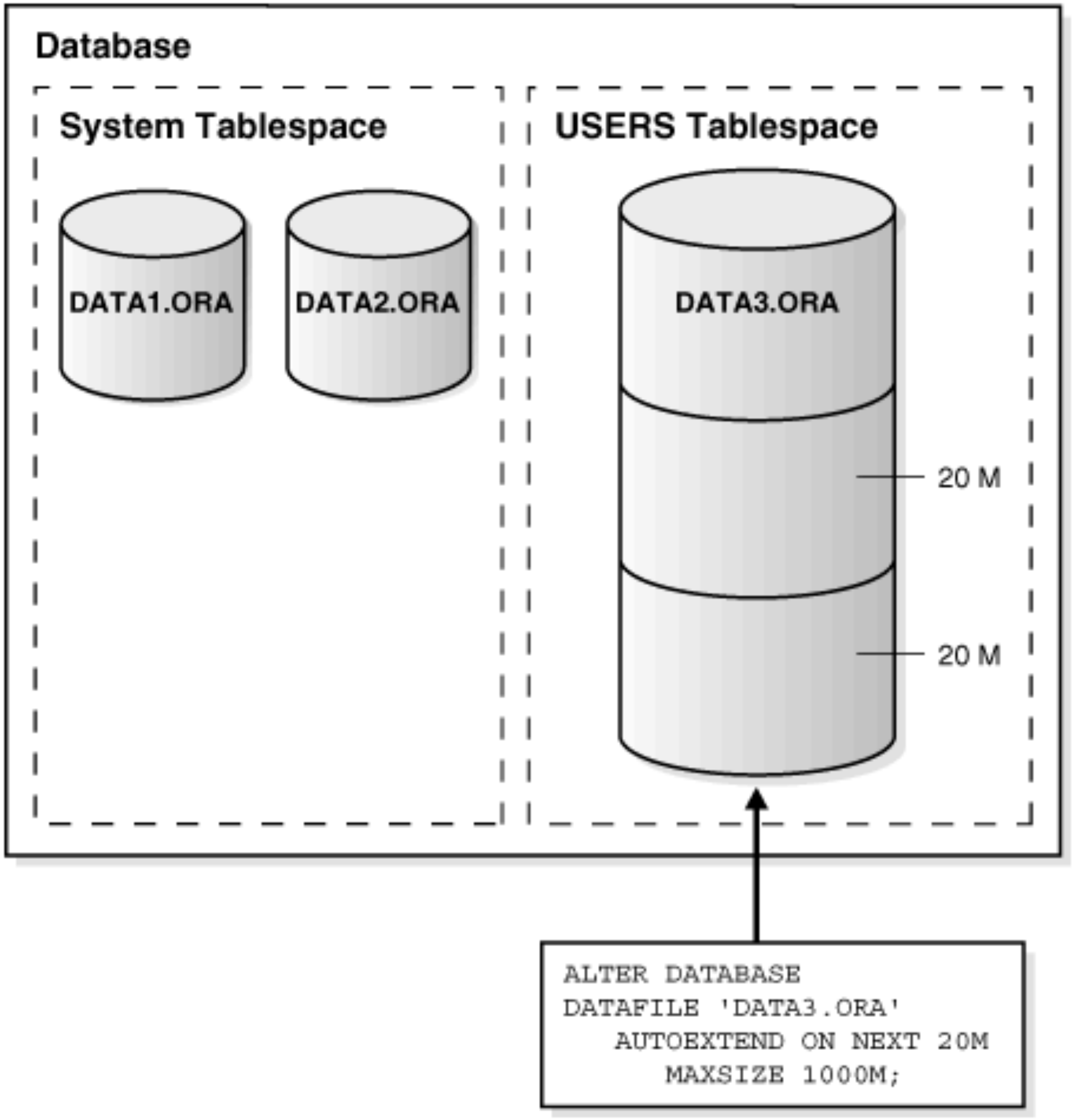
DATA2.ORA

USERS Tablespace

DATA3.ORA

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

Enlarge by dynamically sizing datafiles



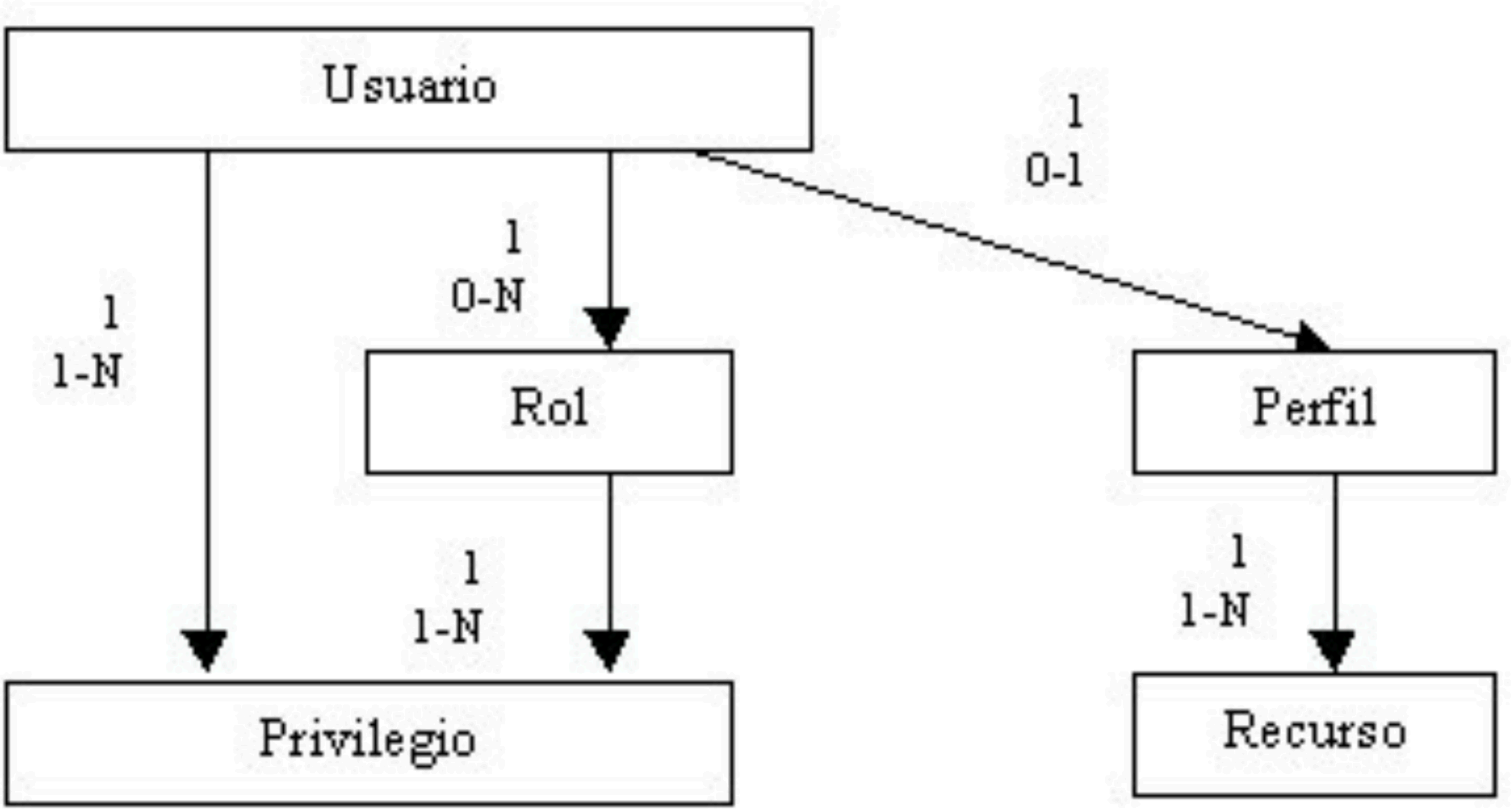
```
ALTER DATABASE
DATAFILE 'DATA3.ORA'
   AUTOEXTEND ON NEXT 20M
      MAXSIZE 1000M;
```

Administration of users and privileges

http://www.redcientifica.com/oracle/c0004p0004.html

https://docs.oracle.com/cd/B28359_01/network.111/b28531/authorization.htm#DBSEG004

https://blogdeaitor.wordpress.com/2008/10/30/comandos-oracle-%E2%80%93-tercera-parte-%E2%80%93/



Usuario

1
0-1

1
0-N

1
1-N

Rol

Perfil

1
1-N

1
1-N

Privilegio

Recurso

USER

- Object able to **connect** to the database instance using a valid user name defined in the database

ROLE

- Grouping of **privileges** to assign to a user of group of users

PRIVILEGE

- Right to **run** a particular type of SQL statement.

- Right to **access** an object belonging to another user.

- Right to **run** a PL/SQL package, and so on

PROFILE

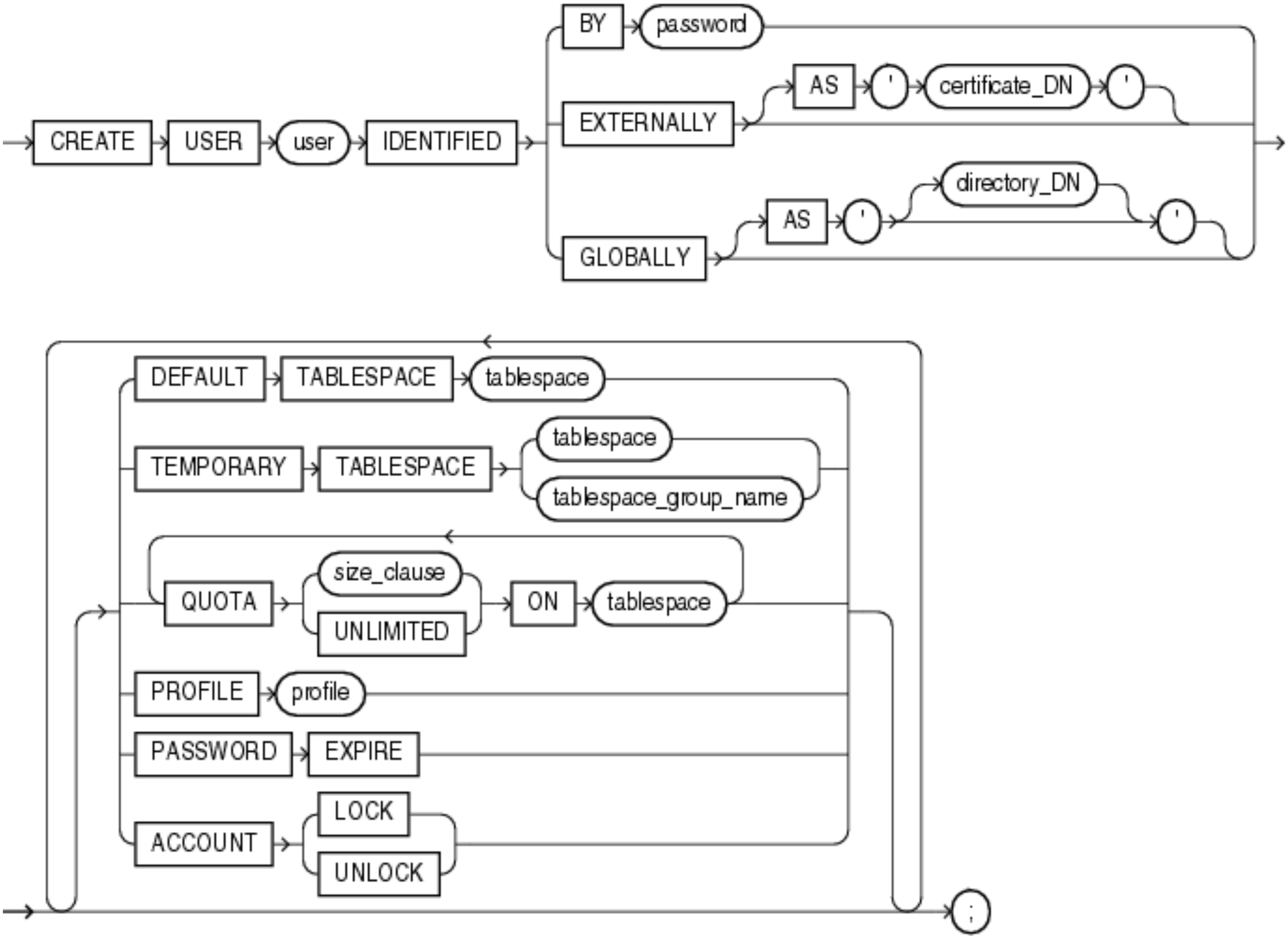
- Set of **limits (resources)** on database resources and password access to the database
    - (**Resource**: Restriction on how user behaves during the session (CPU, time, sessions)

Summarizing

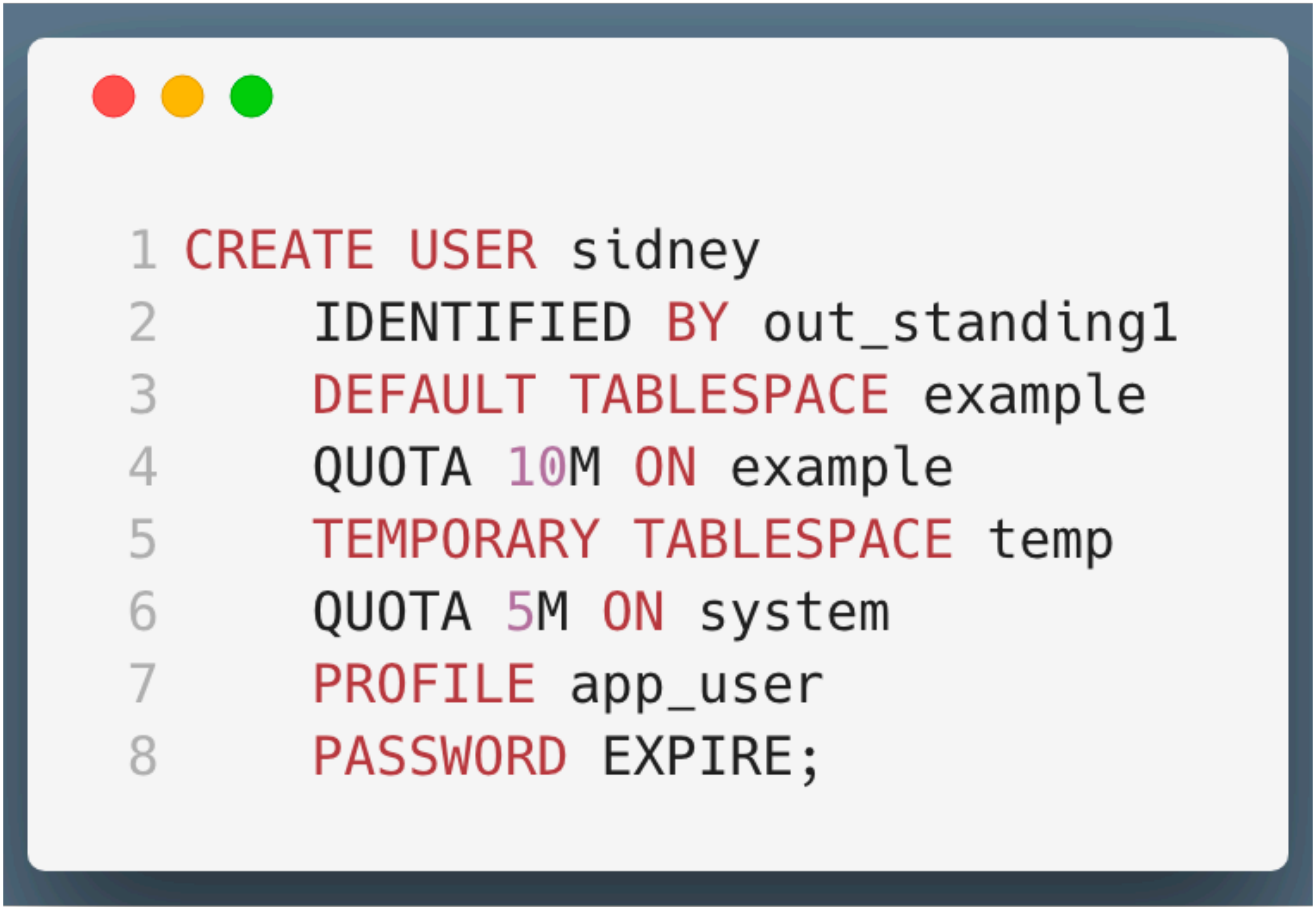
Concepto	**Significado**
Privilegio	Permiso para realizar una acción, asignable a un rol o a un usuario
Rol	Conjunto de privilegios, asignables a un rol o a un usuario
Usuario	Colección de objetos y privilegios identificados con un nombre de usuario y contraseña
Perfil	Conjunto de restricciones relativas al uso de recursos y asignables a un usuario. **Sólo** es posible asignar un perfil a un usuario.
Recurso	Uso susceptible de ser restringido, asignable a un **perfil**

Summarizing

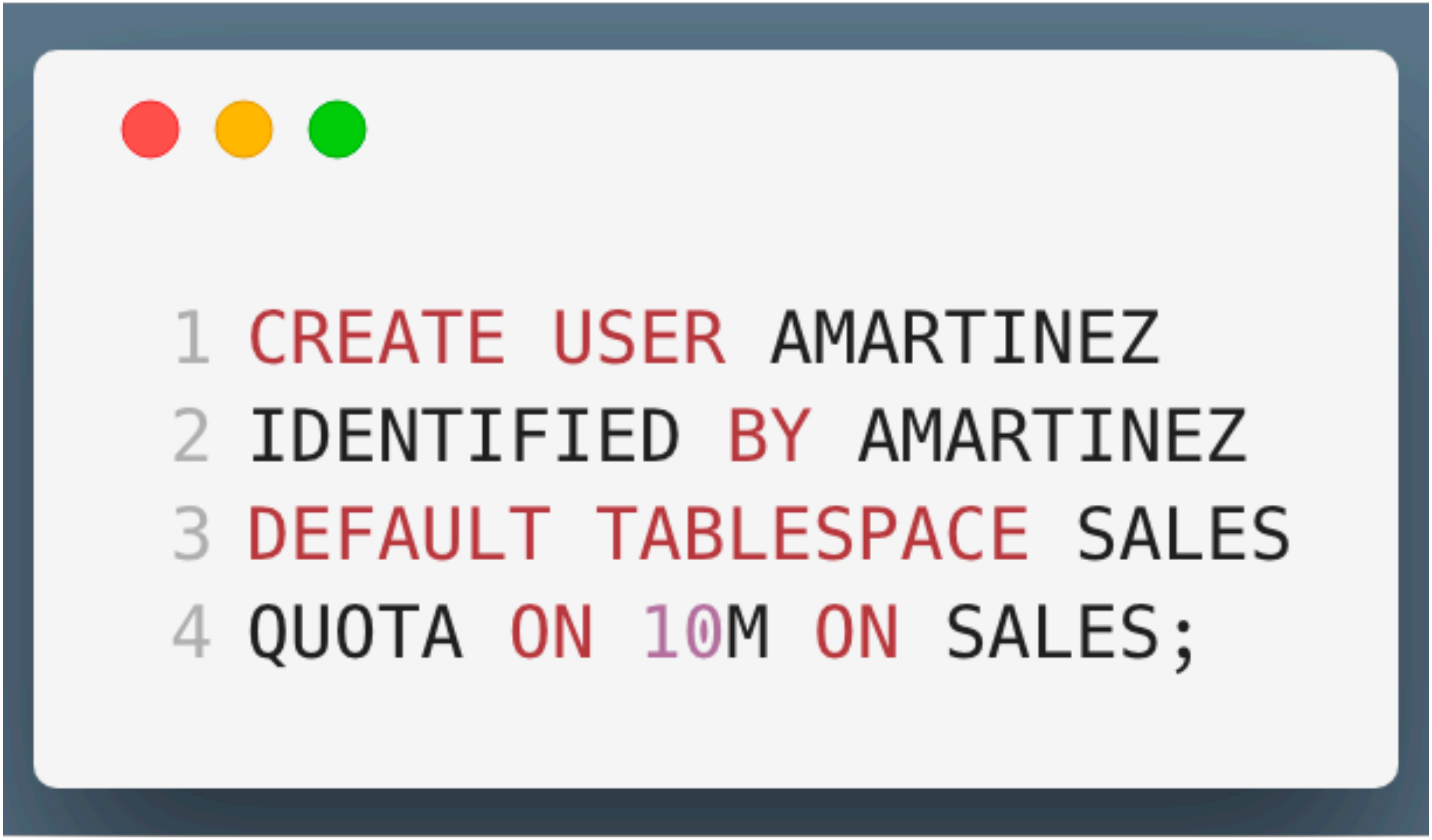
Concepto	**Commands**
Privilegio	GRANT - REVOKE
Rol	(CREATE | ALTER | DROP | SET) ROLE (GRANT | REVOKE) TO name_of_role
Usuario	(CREATE | ALTER | DROP) USER (GRANT | REVOKE) TO name_of_user
Perfil	(CREATE | ALTER | DROP) USER ALTER USER … PROFILE CREATE USER … PROFILE


CREATE
USER
user
IDENTIFIED
BY
password
EXTERNALLY
AS
'
certificate_DN
'
GLOBALLY
AS
'
directory_DN
'
DEFAULT
TABLESPACE
tablespace
TEMPORARY
TABLESPACE
tablespace
tablespace_group_name
QUOTA
size_clause
UNLIMITED
ON
tablespace
PROFILE
profile
PASSWORD
EXPIRE
ACCOUNT
LOCK
UNLOCK
;

CREATE USERS

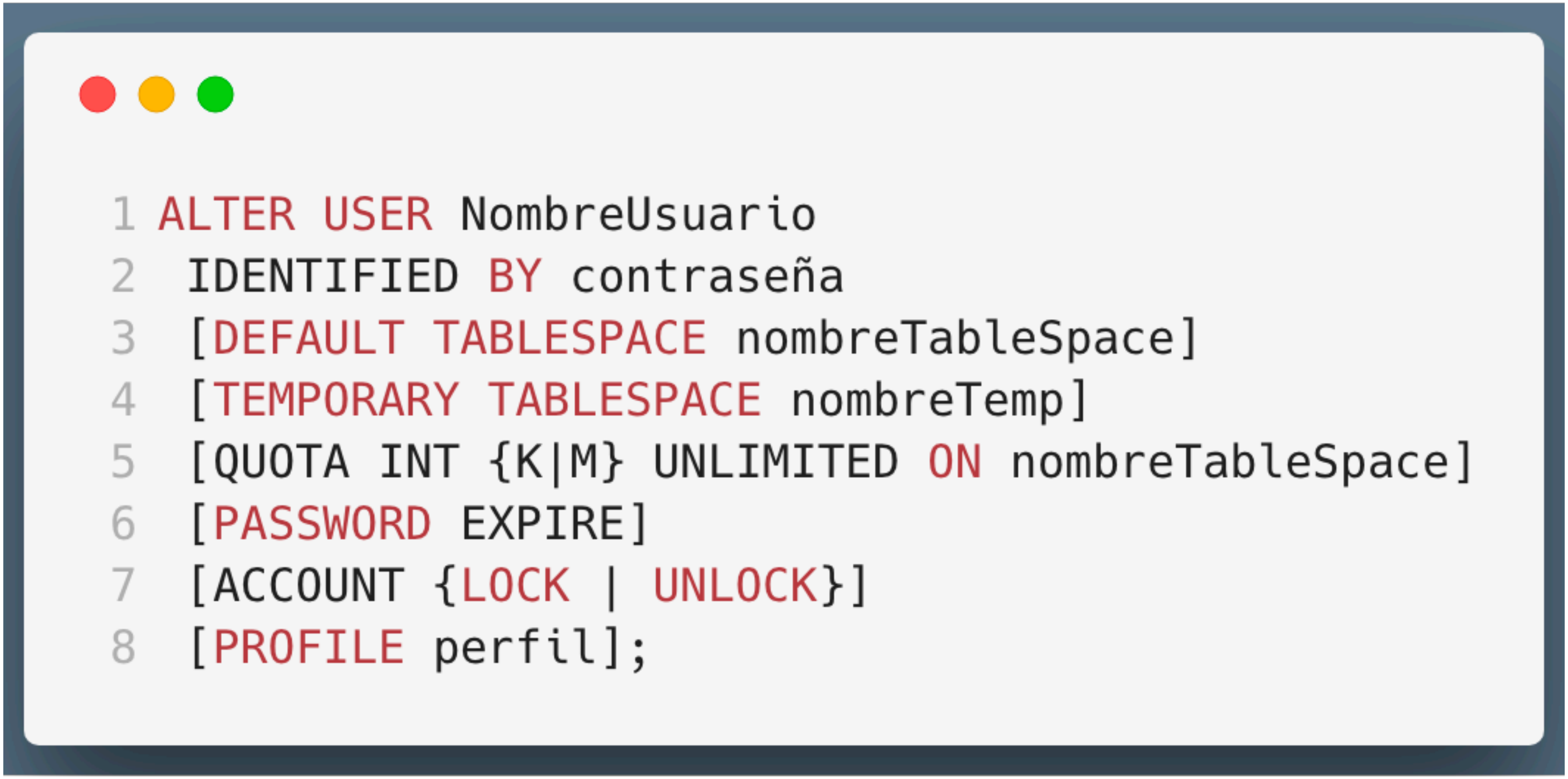


```
CREATE USER sidney
       IDENTIFIED BY out_standing1
       DEFAULT TABLESPACE example
       QUOTA 10M ON example
       TEMPORARY TABLESPACE temp
       QUOTA 5M ON system
       PROFILE app_user
       PASSWORD EXPIRE;
```



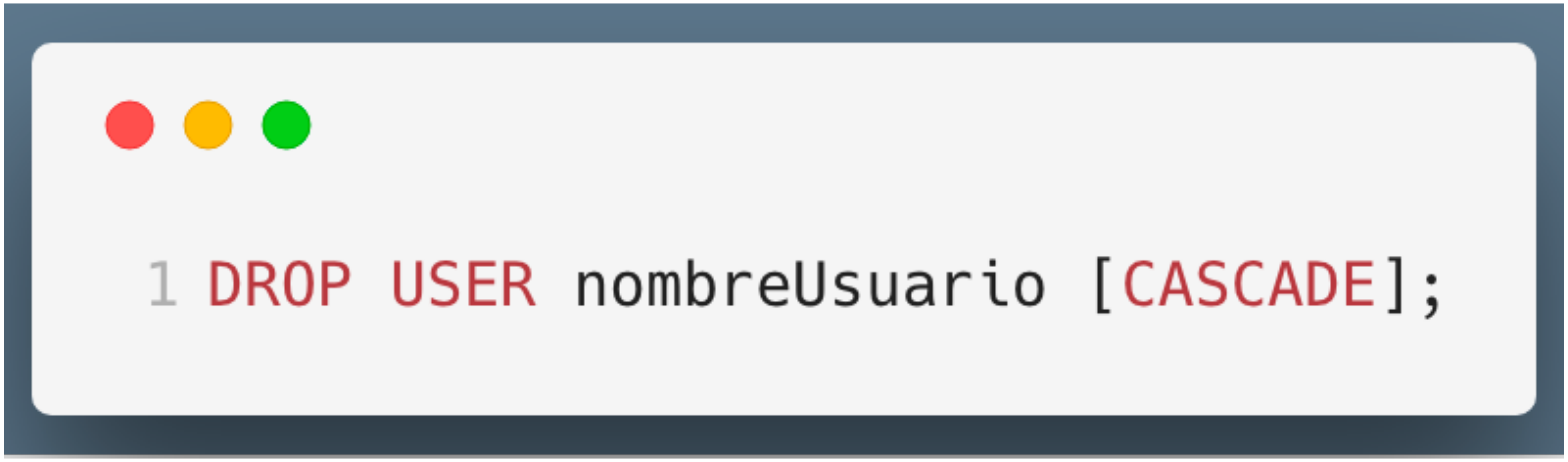
```
CREATE USER AMARTINEZ
IDENTIFIED BY AMARTINEZ
DEFAULT TABLESPACE SALES
QUOTA ON 10M ON SALES;
```

ALTER USERS



```
ALTER USER NombreUsuario
 IDENTIFIED BY contraseña
 [DEFAULT TABLESPACE nombreTableSpace]
 [TEMPORARY TABLESPACE nombreTemp]
 [QUOTA INT {K|M} UNLIMITED ON nombreTableSpace]
 [PASSWORD EXPIRE]
 [ACCOUNT {LOCK | UNLOCK}]
 [PROFILE perfil];
```

DROP USERS



```
DROP USER nombreUsuario [CASCADE];
```

- Use Oracle **DROP USER CASCADE** command If a user owns any database objects

- If objects should remain, then revoke **CREATE SESSION** privilege

PRIVILEGES

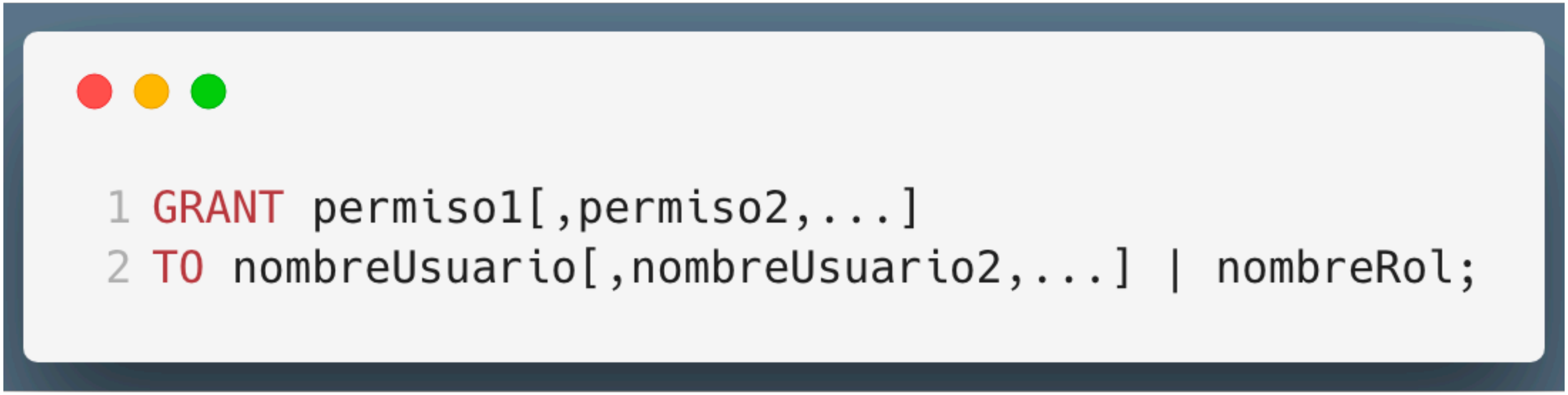
PRIVILEGES

- Way to authorize users.
- It controls if a user can modify an object owned by another user.
- In order to GRANT or REVOKE privileges you need either:
  - Be an administrator
  - Have the **ADMIN** privilege
  - Be the owner of the object (certain objects)

TYPES OF PRIVILEGES

- **System privileges:** Tables, views, indexes, sequences, PL/SQL functions, procedures.
- **Object privileges:** Access another user’s object. Tables, views, indexes, sequences.
- **Privilege hierarchy:** Privileges which confer other privileges.
- **PUBLIC role:** By default you can **SELECT** and **EXECUTE** privileges on system tables and views, functions and packages.

1. SYSTEM PRIVILEGES



```sql
GRANT permiso1[,permiso2,...]
TO nombreUsuario[,nombreUsuario2,...] | nombreRol;
```

1. SYSTEM PRIVILEGES

- **create**
  - **session:** Permite conectarse a la base de datos
  - **table:** Permite crear tablas
  - **sequence:** Permite crear secuencias
  - **view:** Permite crear vistas
  - **trigger:** Permite crear disparadores
  - **procedure:** Permite crear procedimientos
  - **profile:** Permite crear perfiles
  - **synonym:** Permite crear sinónimos

1. SYSTEM PRIVILEGES

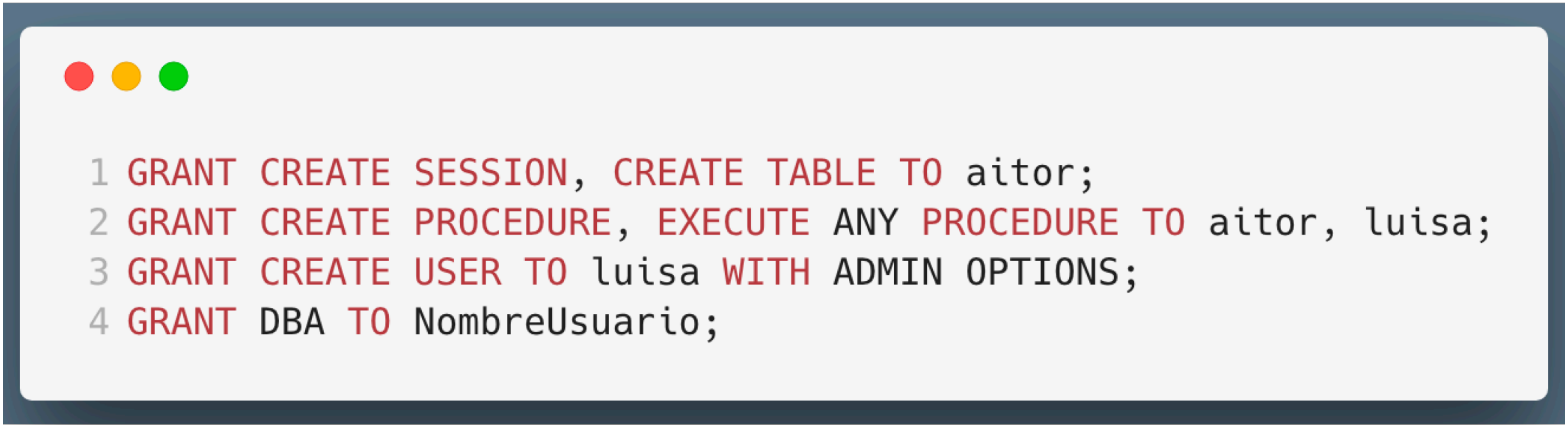
- **execute any procedure:** Permite ejecutar cualquier procedimiento
- **create**
  - **user:** Permite crear usuarios. **WITH ADMIN OPTIONS** permite que el nuevo usuario tenga permisos administrativos, por ejemplo, para crear nuevos usuarios.
  - **role:** Permite crear roles

1. SYSTEM PRIVILEGES

- **drop**
  - **table:** Permite eliminar tables
  - **sequence:** Permite eliminar secuencias
  - **view:** Permite eliminar vistas
  - **trigger:** Permite eliminar disparadores
  - **procedure:** Permite eliminar procedimientos
  - **profile:** Permite eliminar perfiles
  - **synonym:** Permite eliminar sinónimos
  - **user:** Permite eliminar usuarios
  - **role:** Permite eliminar roles
  - **session:** Permite eliminar sesiones

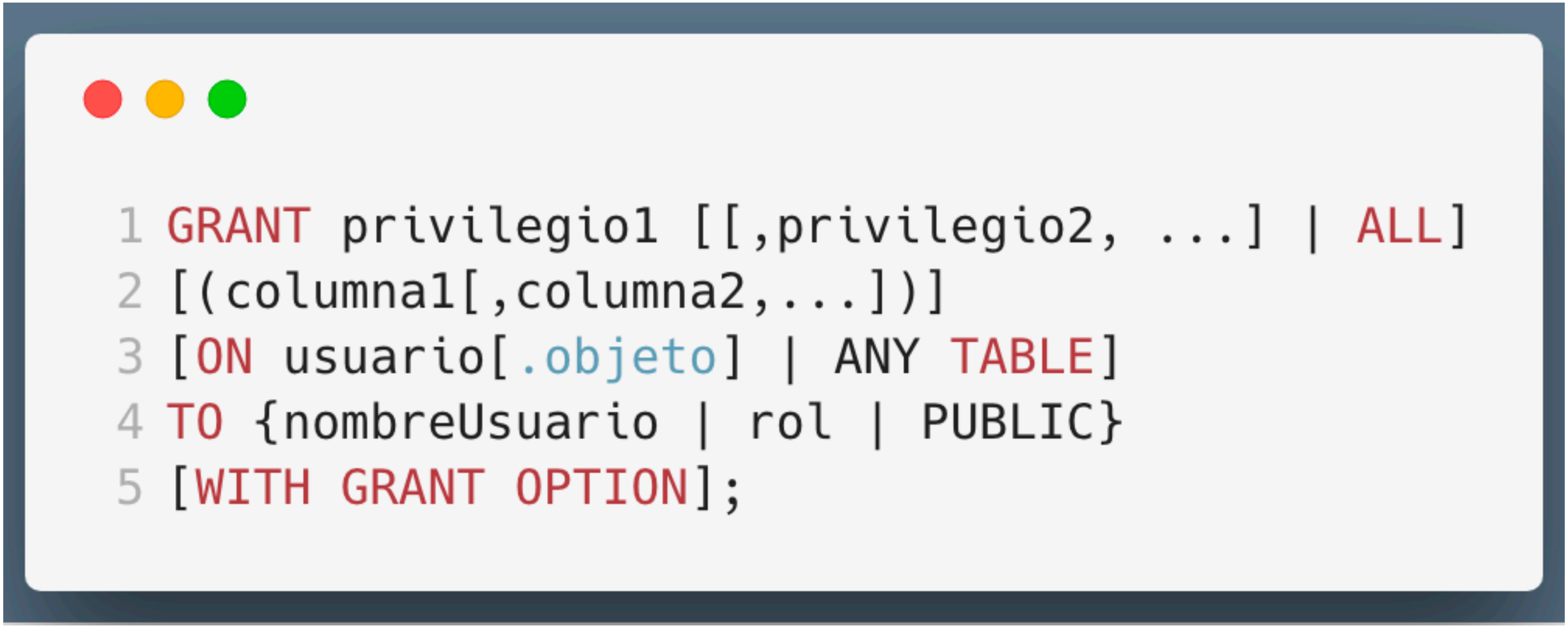
1. SYSTEM PRIVILEGES

- **grant**
  - **privilege:** Permite asignar privilegios
  - **role:** Permite asignar roles



```
GRANT CREATE SESSION, CREATE TABLE TO aitor;
GRANT CREATE PROCEDURE, EXECUTE ANY PROCEDURE TO aitor, luisa;
GRANT CREATE USER TO luisa WITH ADMIN OPTIONS;
GRANT DBA TO NombreUsuario;
```

2. OBJECT PRIVILEGES

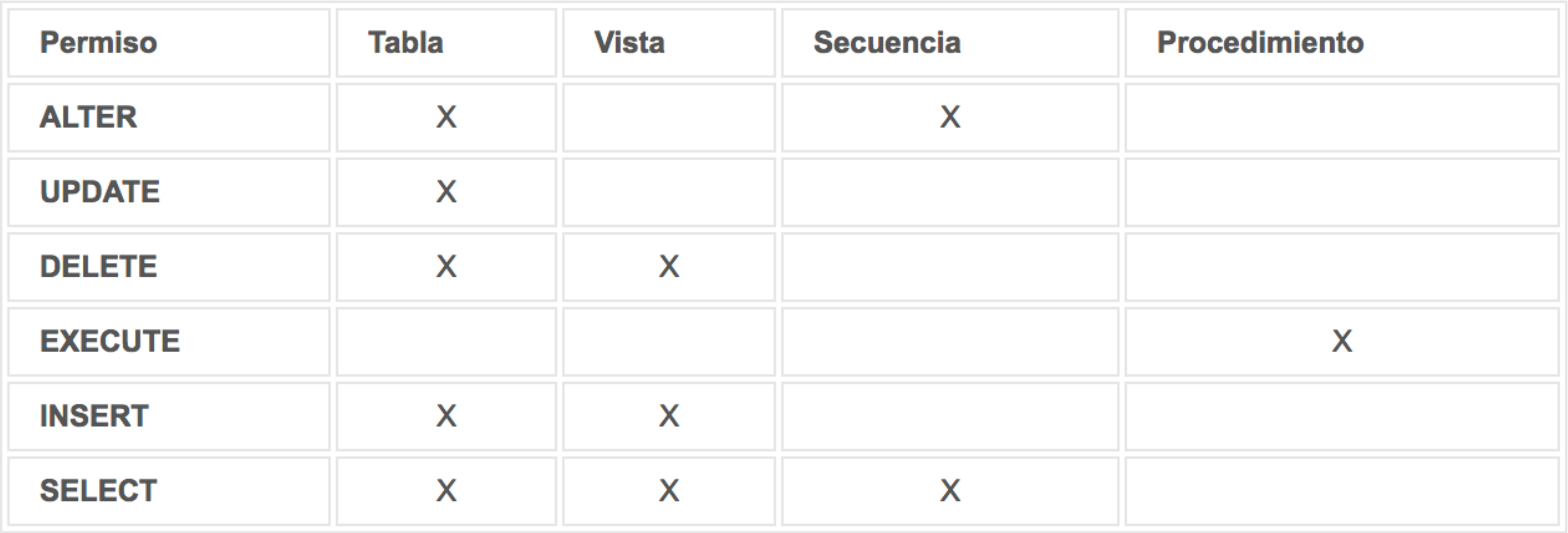


```
GRANT privilegio1 [[,privilegio2, ...] | ALL]
[(columna1[,columna2,...])]
[ON usuario[.objeto] | ANY TABLE]
TO {nombreUsuario | rol | PUBLIC}
[WITH GRANT OPTION];
```

2. OBJECT PRIVILEGES

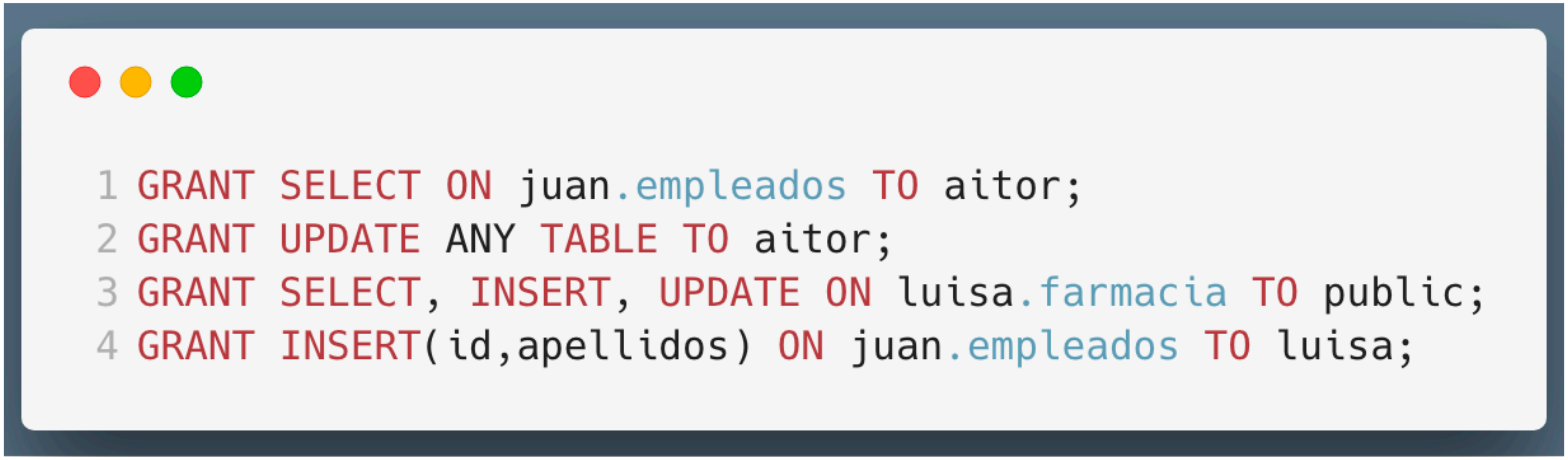
- **ON:** Objeto sobre el que aplico los privilegios
- **TO:** Usuario al que concedo los privilegios
- **ALL:** Permite asignar todos los permisos
- **PUBLIC:** Asigna el privilegio o privilegios a todos los usuarios del sistema (también a los futuros)
- **WITH GRANT OPTION:** Permite que el usuario que lo reciba pueda conceder permisos a otros usuarios

2. OBJECT PRIVILEGES



Permiso	Tabla	Vista	Secuencia	Procedimiento
ALTER	X		X	
UPDATE	X			
DELETE	X	X		
EXECUTE				X
INSERT	X	X		
SELECT	X	X	X	

2. OBJECT PRIVILEGES



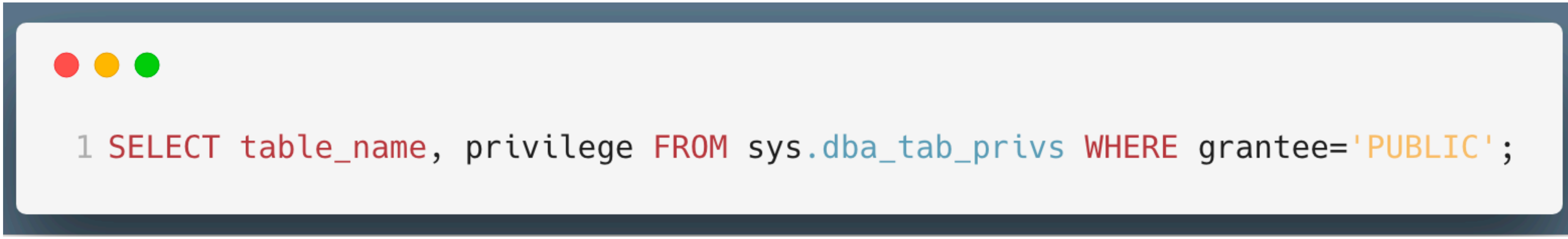
```sql
GRANT SELECT ON juan.empleados TO aitor;
GRANT UPDATE ANY TABLE TO aitor;
GRANT SELECT, INSERT, UPDATE ON luisa.farmacia TO public;
GRANT INSERT(id,apellidos) ON juan.empleados TO luisa;
```

3. PUBLIC ROLE

- Accessible to every database user, all privileges and roles granted to PUBLIC are accessible to every database user.
- Security administrators and database users should grant a privilege or role to PUBLIC only if every database user requires the privilege or role
- **Each database user should have only the privileges required to accomplish the current group tasks successfully.**

3. PUBLIC ROLE

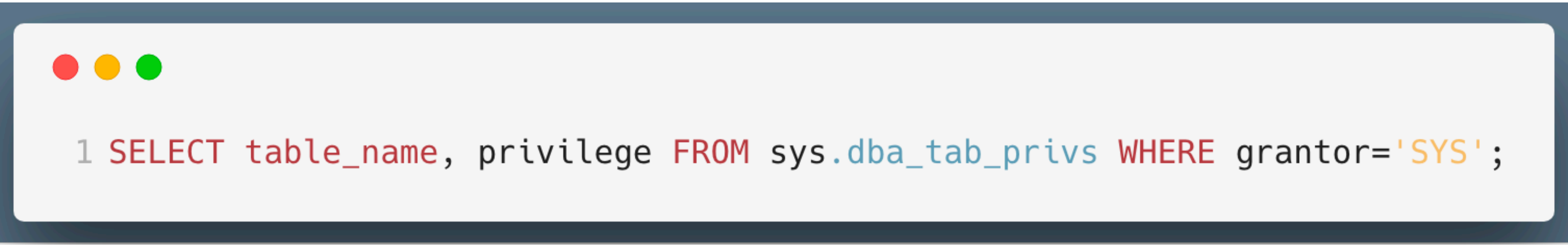
- By default PUBLIC has SELECT and EXECUTE privileges on various system tables and views and PL/SQL functions, procedures and packages. You can see the list of objects by using this query:



```sql
SELECT table_name, privilege FROM sys.dba_tab_privs WHERE grantee='PUBLIC';
```

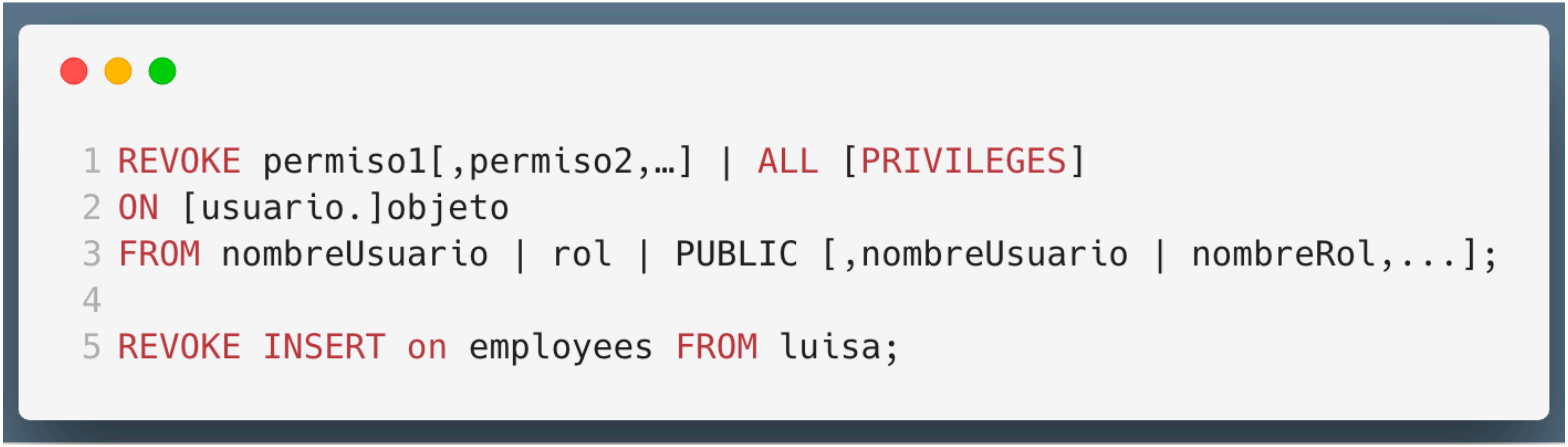
3. PUBLIC ROLE

- Privileges that are granted to PUBLIC as part of database creation cannot be revoked. To see a list of these privileges, use this query:



```sql
SELECT table_name, privilege FROM sys.dba_tab_privs WHERE grantor='SYS';
```

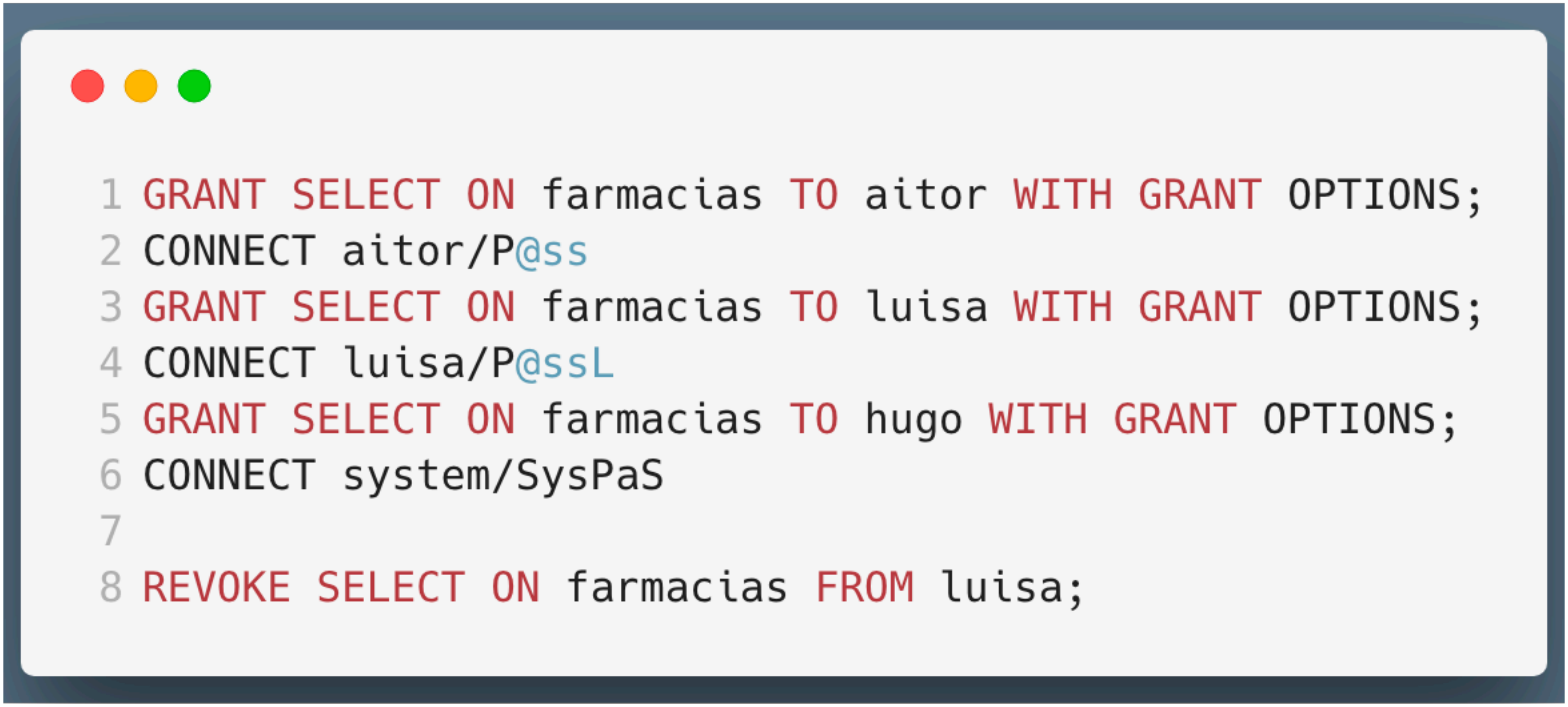
DROP PRIVILEGES (objects)



```sql
REVOKE permiso1[,permiso2,…] | ALL [PRIVILEGES]
ON [usuario.]objeto
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];

REVOKE INSERT on employees FROM luisa;
```

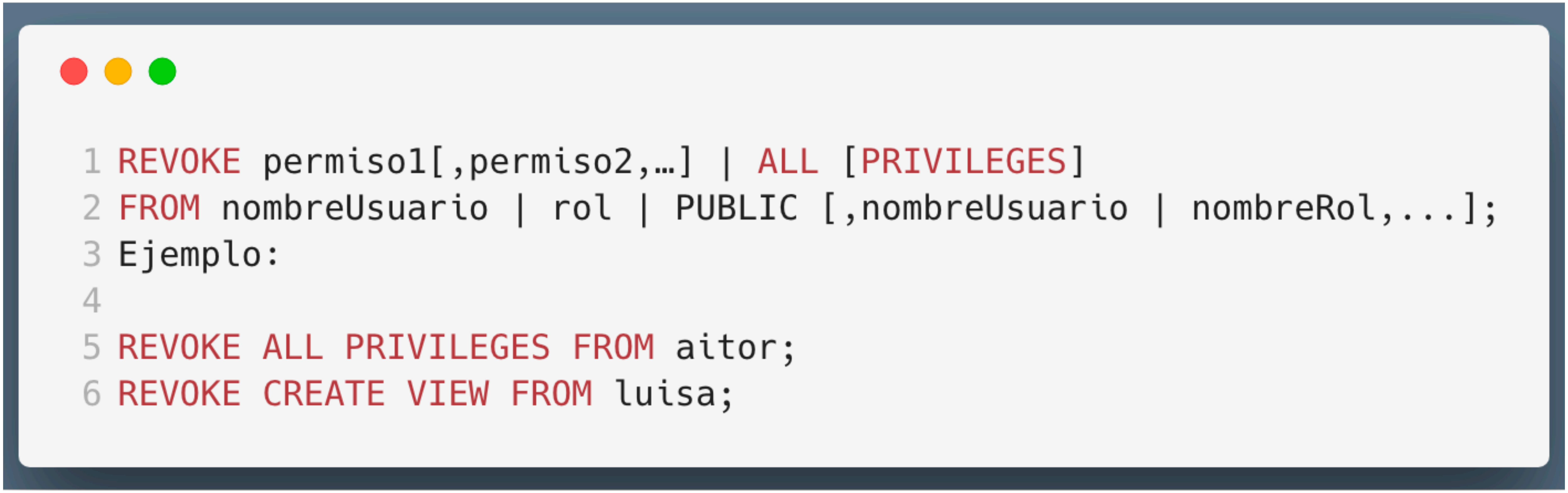
- **Nota**: En los privilegios sobre objetos, **SI** se quita el permiso de select sobre employees a luisa y a todos los usuarios a los que ésta les ha concedido dicho permiso.

DROP PRIVILEGES (objects)



```
GRANT SELECT ON farmacias TO aitor WITH GRANT OPTIONS;
CONNECT aitor/P@ss
GRANT SELECT ON farmacias TO luisa WITH GRANT OPTIONS;
CONNECT luisa/P@ssL
GRANT SELECT ON farmacias TO hugo WITH GRANT OPTIONS;
CONNECT system/SysPaS

REVOKE SELECT ON farmacias FROM luisa;
```

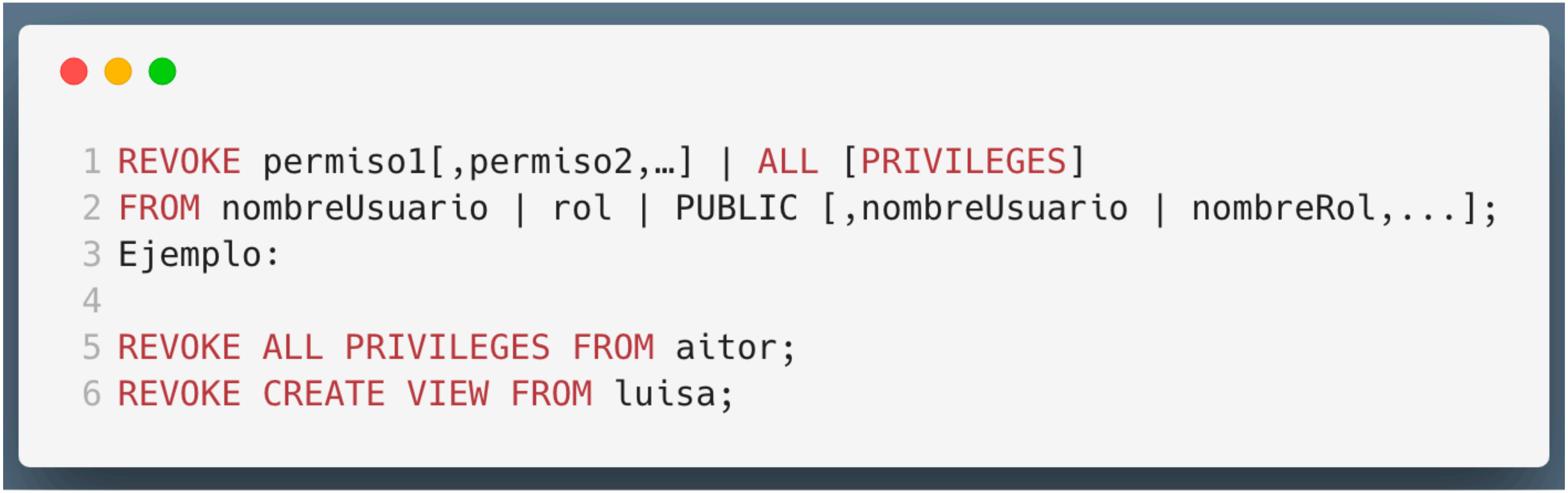
DROP PRIVILEGES (system)



```
REVOKE permiso1[,permiso2,…] | ALL [PRIVILEGES]
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];
Ejemplo:

REVOKE ALL PRIVILEGES FROM aitor;
REVOKE CREATE VIEW FROM luisa;
```

- **Nota:** En los privilegios de sistema, se quita el permiso de crear vistas a luisa pero no a los usuarios a los que éste haya concedido el privilegio

DROP PRIVILEGES (system)

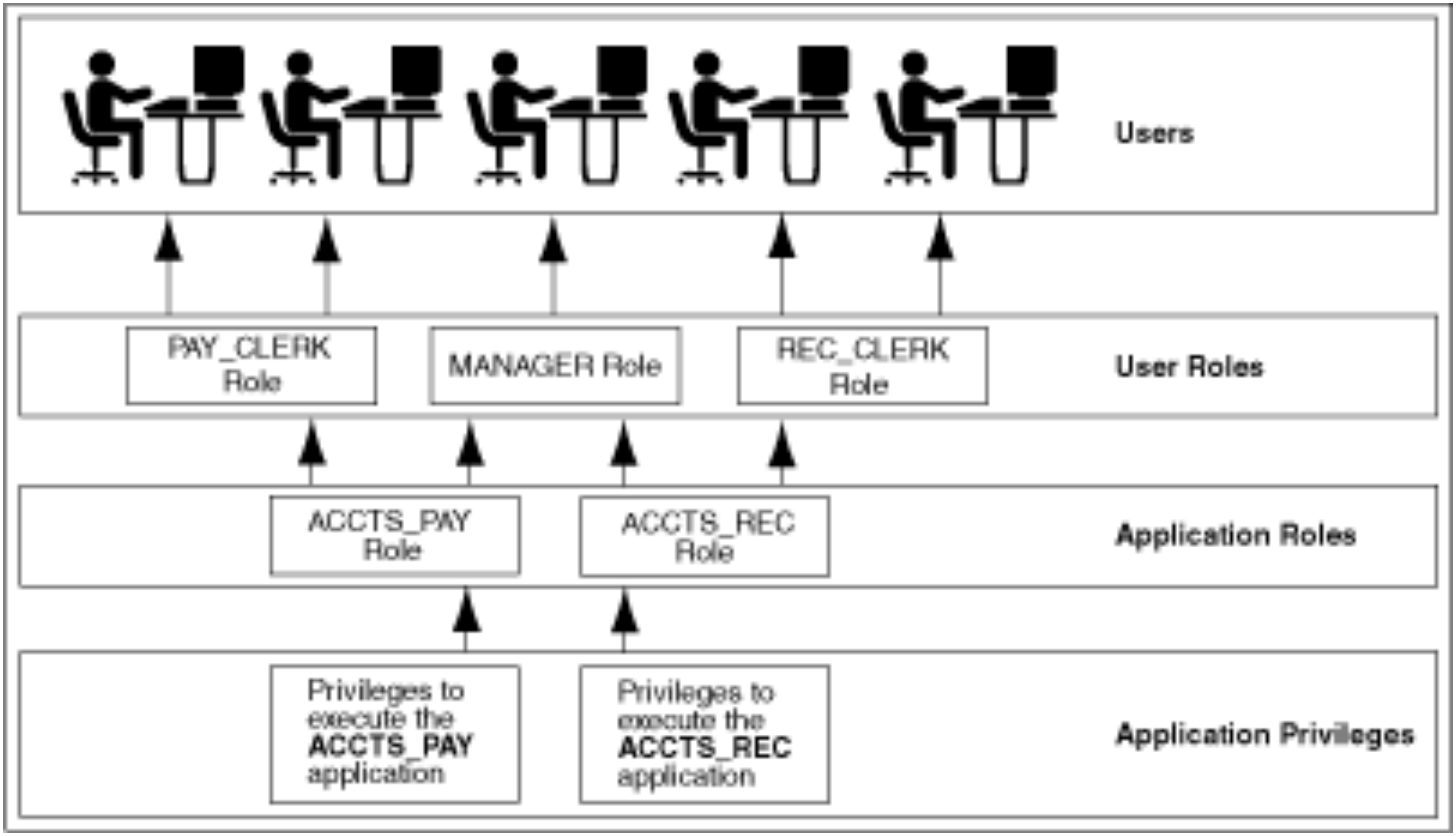


```
REVOKE permiso1[,permiso2,…] | ALL [PRIVILEGES]
FROM nombreUsuario | rol | PUBLIC [,nombreUsuario | nombreRol,...];
Ejemplo:

REVOKE ALL PRIVILEGES FROM aitor;
REVOKE CREATE VIEW FROM luisa;
```

- **Nota:** En los privilegios de sistema, se quita el permiso de crear vistas a luisa pero no a los usuarios a los que éste haya concedido el privilegio

ROLES

COMMON USES OF ROLES



Users

PAY_CLERK Role

MANAGER Role

REC_CLERK Role

User Roles

ACCTS_PAY Role

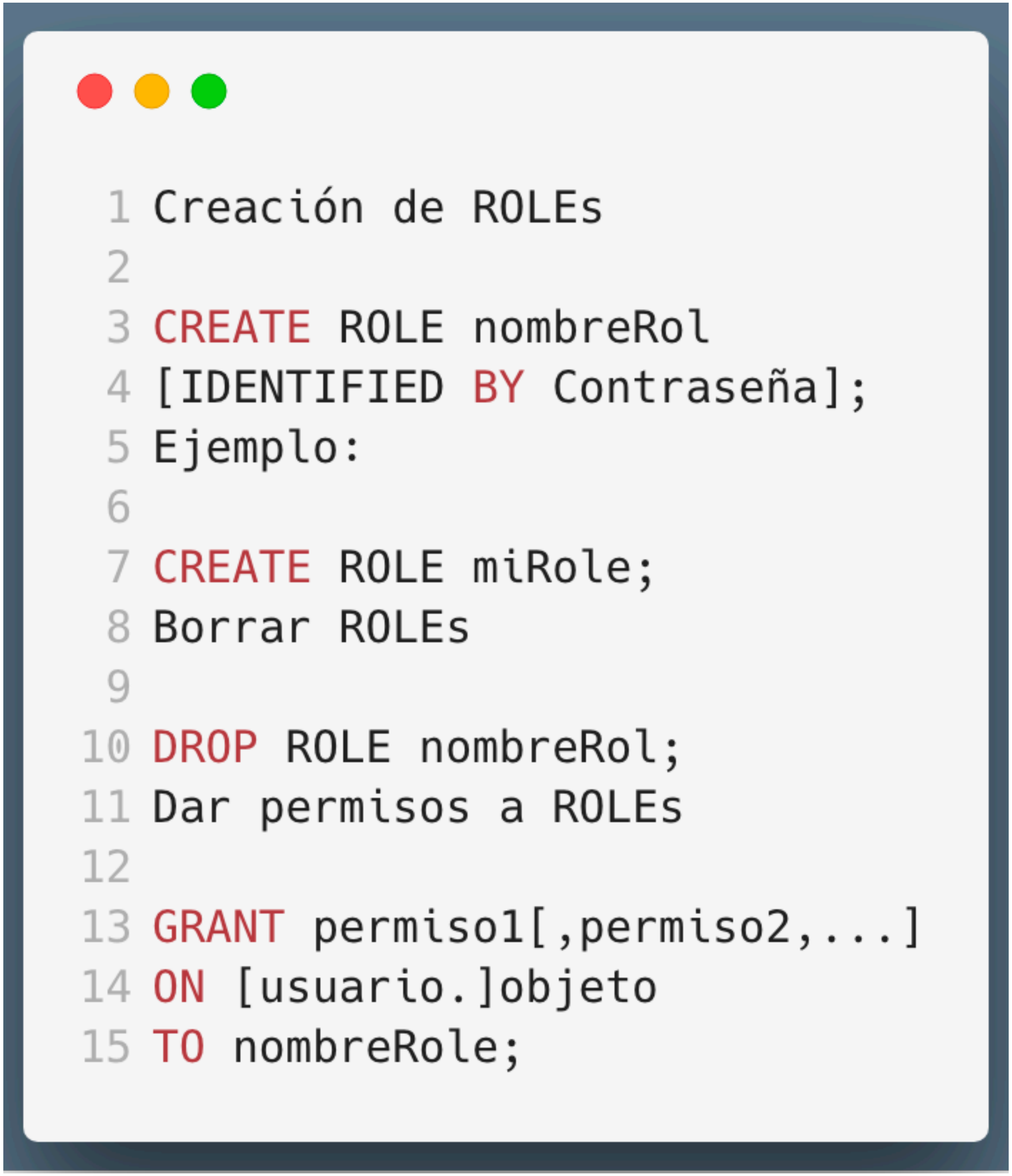
ACCTS_REC Role

Application Roles

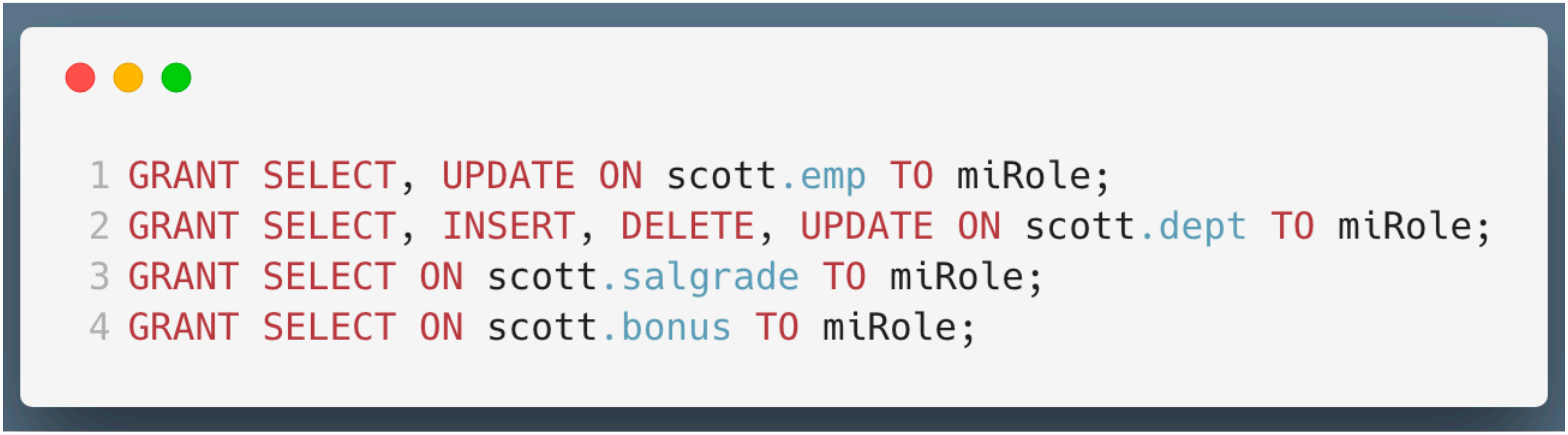
Privileges to execute the **ACCTS_PAY** application

Privileges to execute the **ACCTS_REC** application

Application Privileges



```
Creación de ROLEs

CREATE ROLE nombreRol
[IDENTIFIED BY Contraseña];
Ejemplo:

CREATE ROLE miRole;
Borrar ROLEs

DROP ROLE nombreRol;
Dar permisos a ROLEs

GRANT permiso1[,permiso2,...]
ON [usuario.]objeto
TO nombreRole;
```



```sql
GRANT SELECT, UPDATE ON scott.emp TO miRole;
GRANT SELECT, INSERT, DELETE, UPDATE ON scott.dept TO miRole;
GRANT SELECT ON scott.salgrade TO miRole;
GRANT SELECT ON scott.bonus TO miRole;
```

TABLAS A TENER EN CUENTA

- **Roles**:
  - **dba_roles**: Contiene todos los roles disponibles
  - **dba_role_privs**: Contiene el mapeado de roles y usuarios
  - **dba_sys_privs**: Privilegios asignados a cada role (incluso a los ya predefinidos Oracle)
  - **ROLE_TAB_PRIVS:** Privilegios de tablas garantizados a roles.
  - **ROLE_SYS_PRIVS:** Privilegios del sistema asignados a roles
  - **ROLE_ROLE_PRIVS:** Roles que se han asignado a otros roles

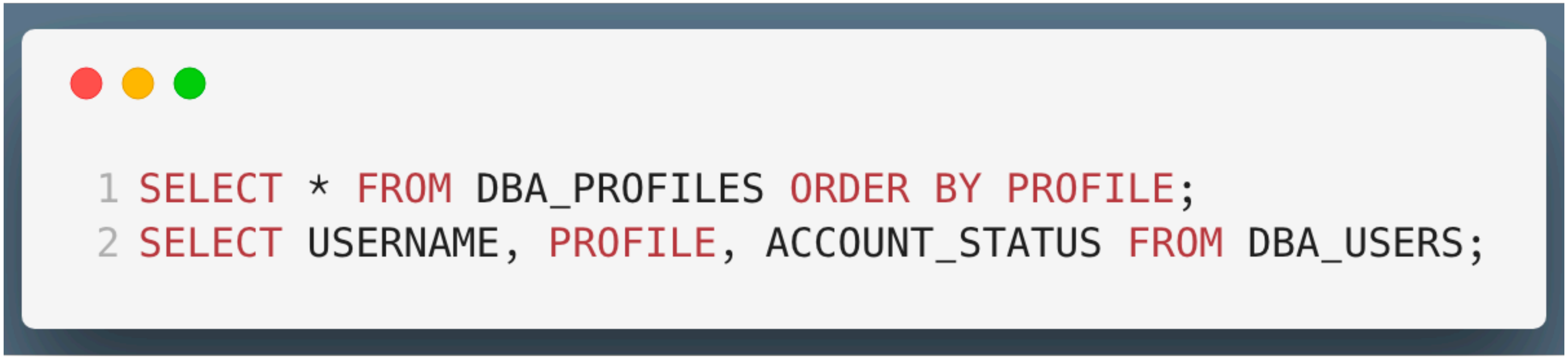
- **Recursos**:
    - **dba_ts_quotas**: Límites de uso de espacio en disco
    - **user_resource_limits**: Limites de recursos en Oracle para el usuario actual

- **Usuarios**:
  - **dba_users**: Vista que apunta realmente a la tabla sys.user$. Almacena información sobre todos los usuarios de la base de datos
  - **user_users**: Vista que muestra información sobre el usuario actualmente conectado
  - **user_resource_limits**: Limites de recursos en Oracle para el usuario actual
  - **all_tables**: Contiene todas las tablas accesibles por el usuario (las propias más las que tiene permisos sobre ellas)

PROFILES

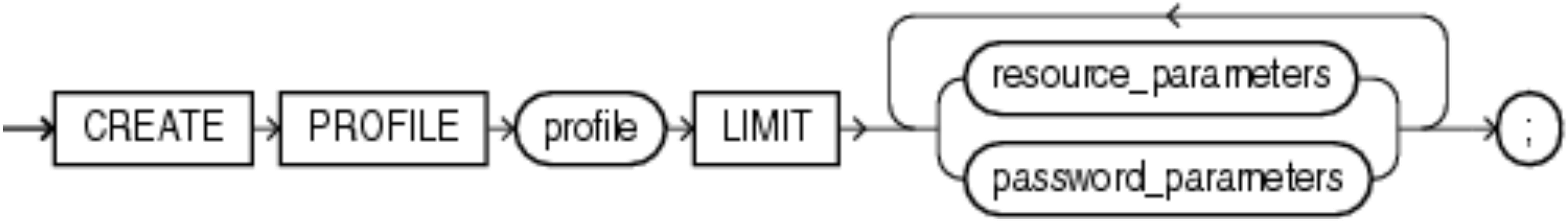
PROFILES

- A profile is a set of limits on database resources and password access to the database.
- If no profile is specified, then the user is assigned a default profile.

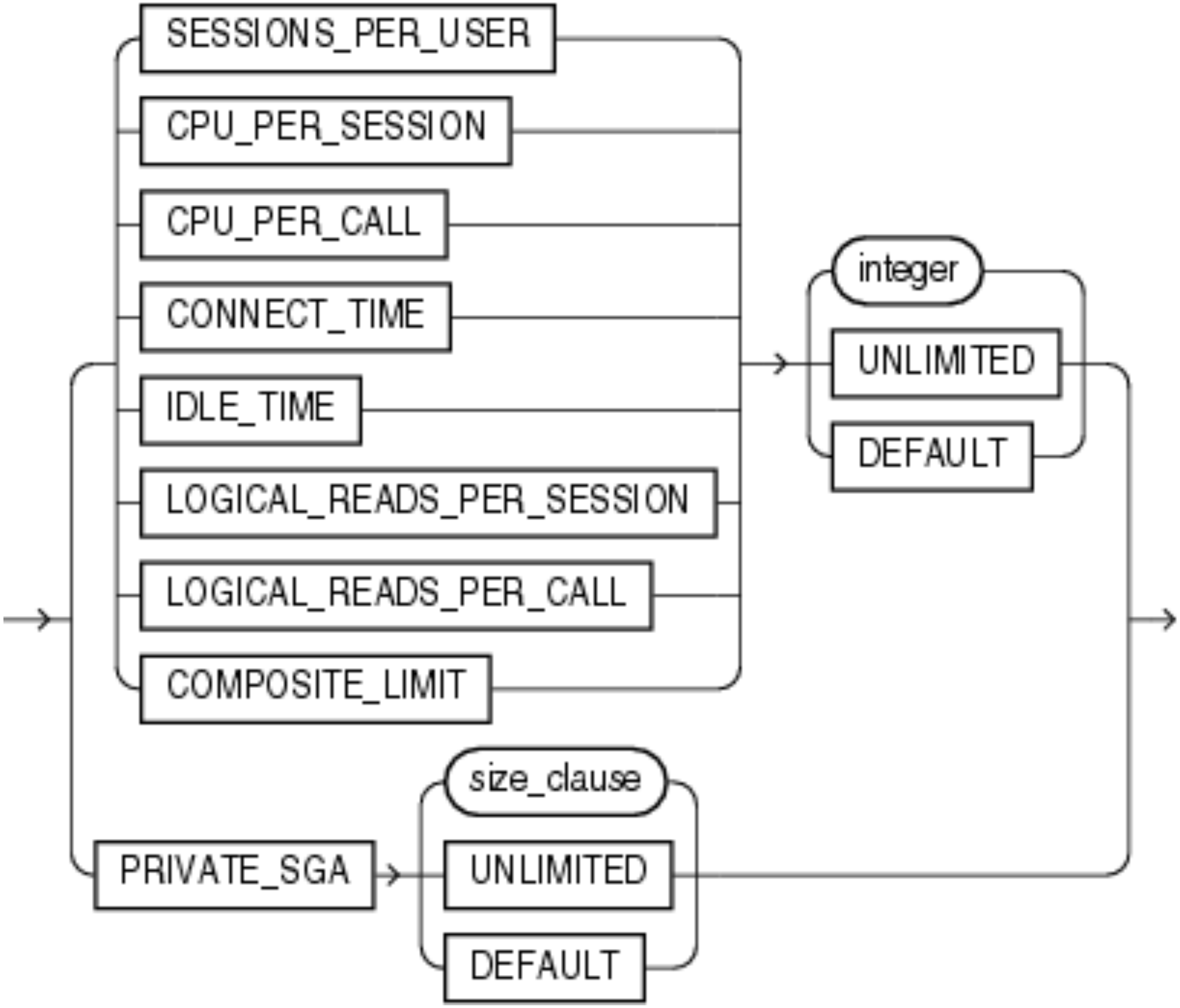


```sql
SELECT * FROM DBA_PROFILES ORDER BY PROFILE;
SELECT USERNAME, PROFILE, ACCOUNT_STATUS FROM DBA_USERS;
```

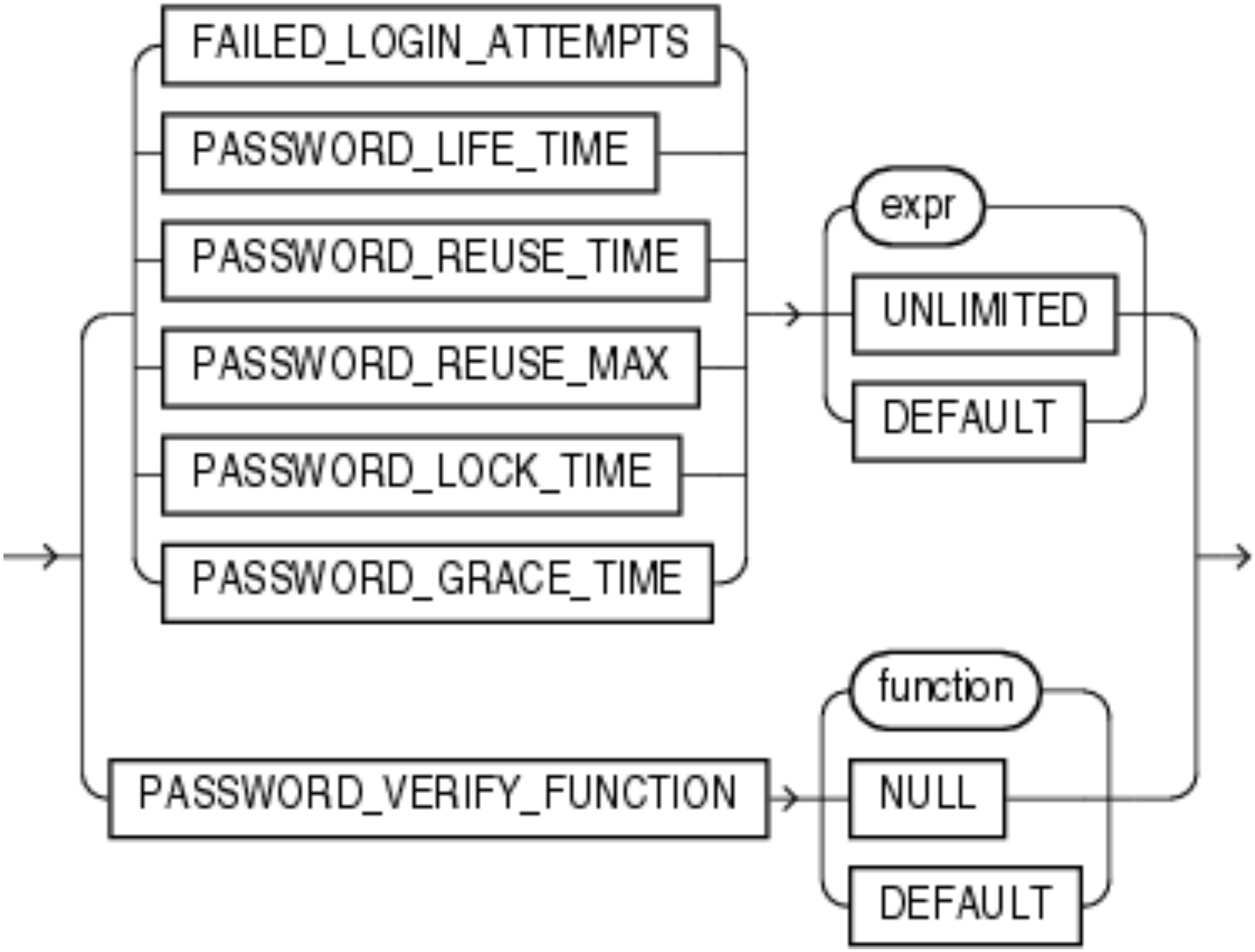
PROFILES


CREATE
PROFILE
profile
LIMIT
resource_parameters
password_parameters
;

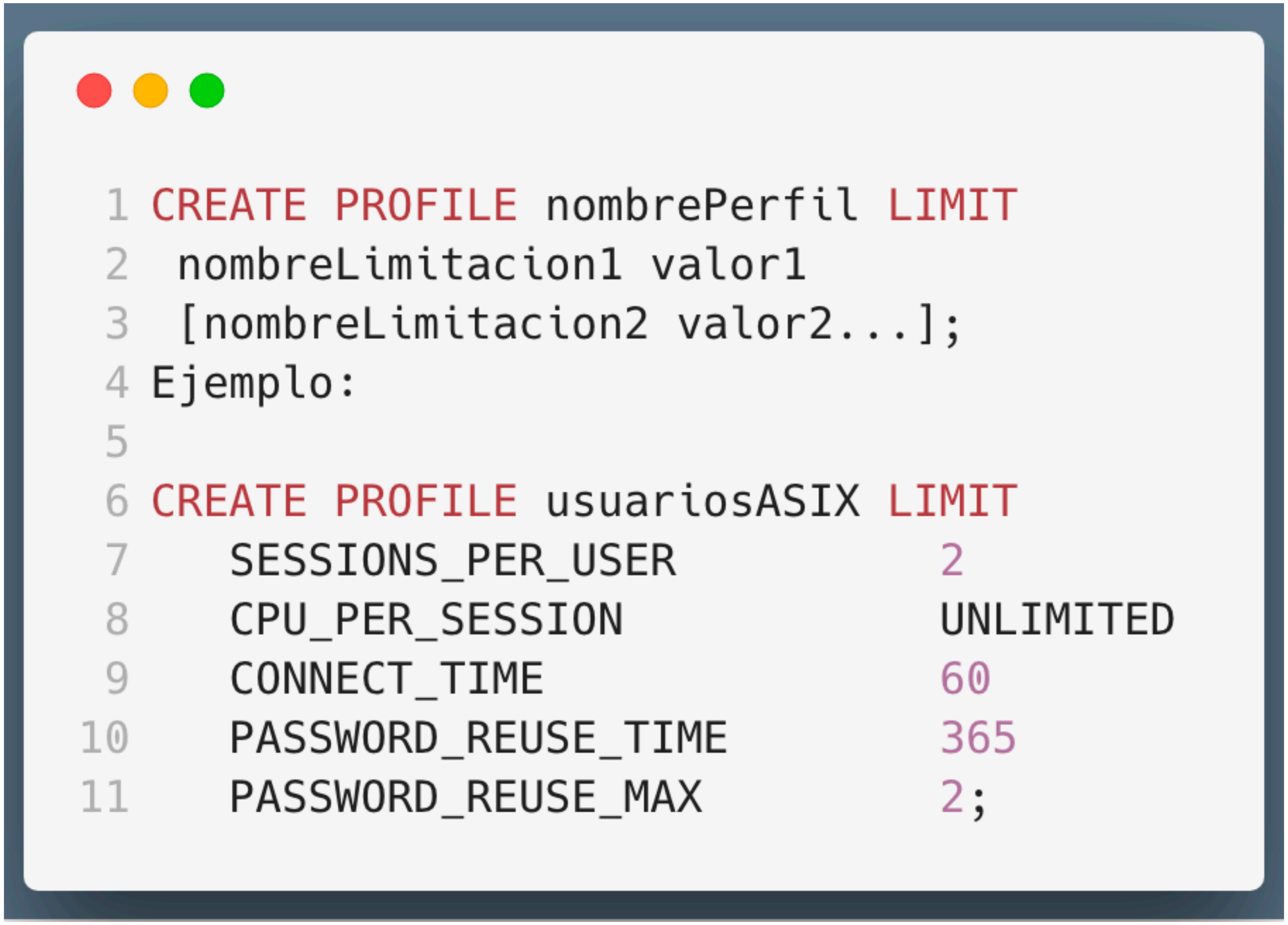
RESOURCE PARAMETERS



PASSWORD PARAMETERS

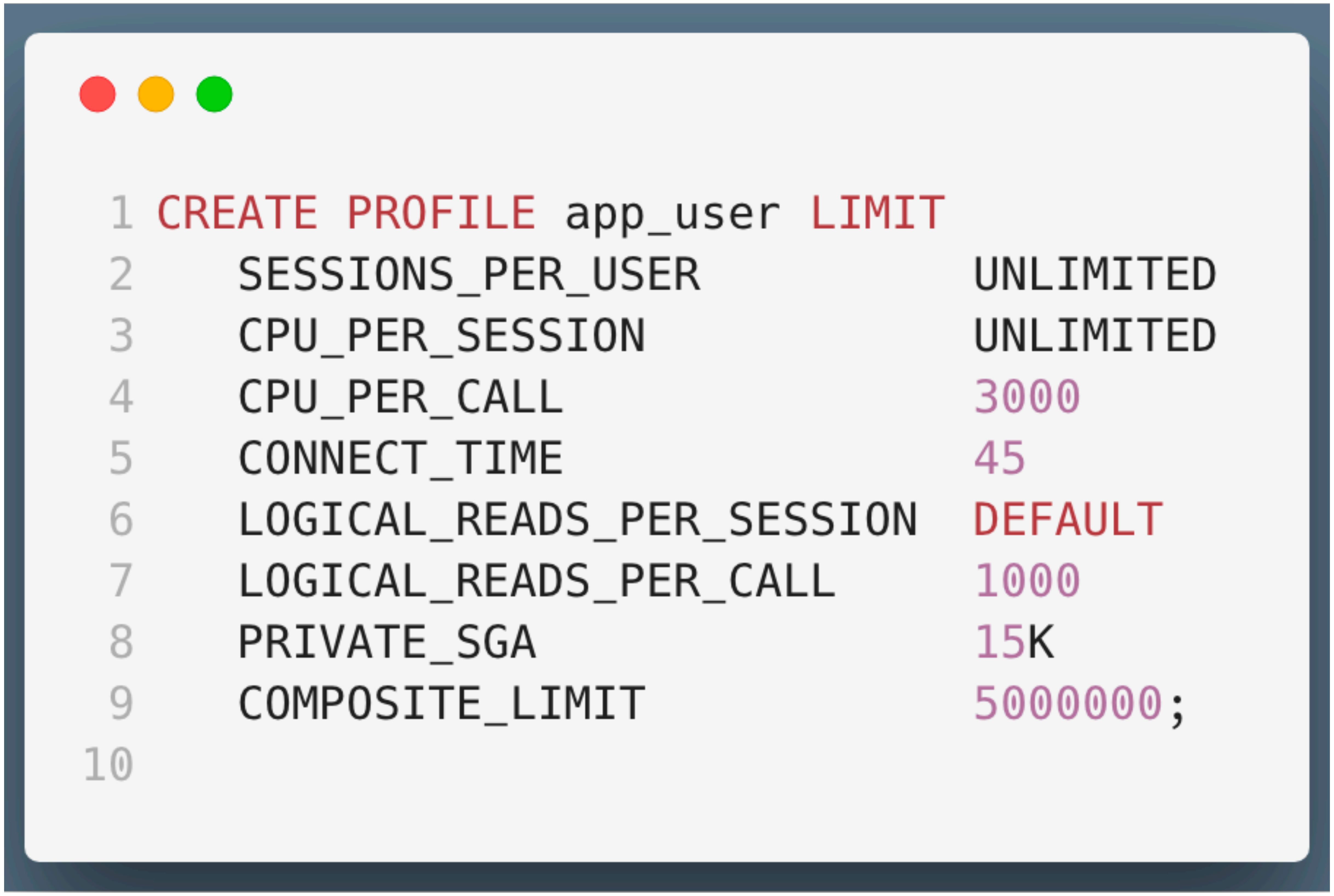


PROFILES



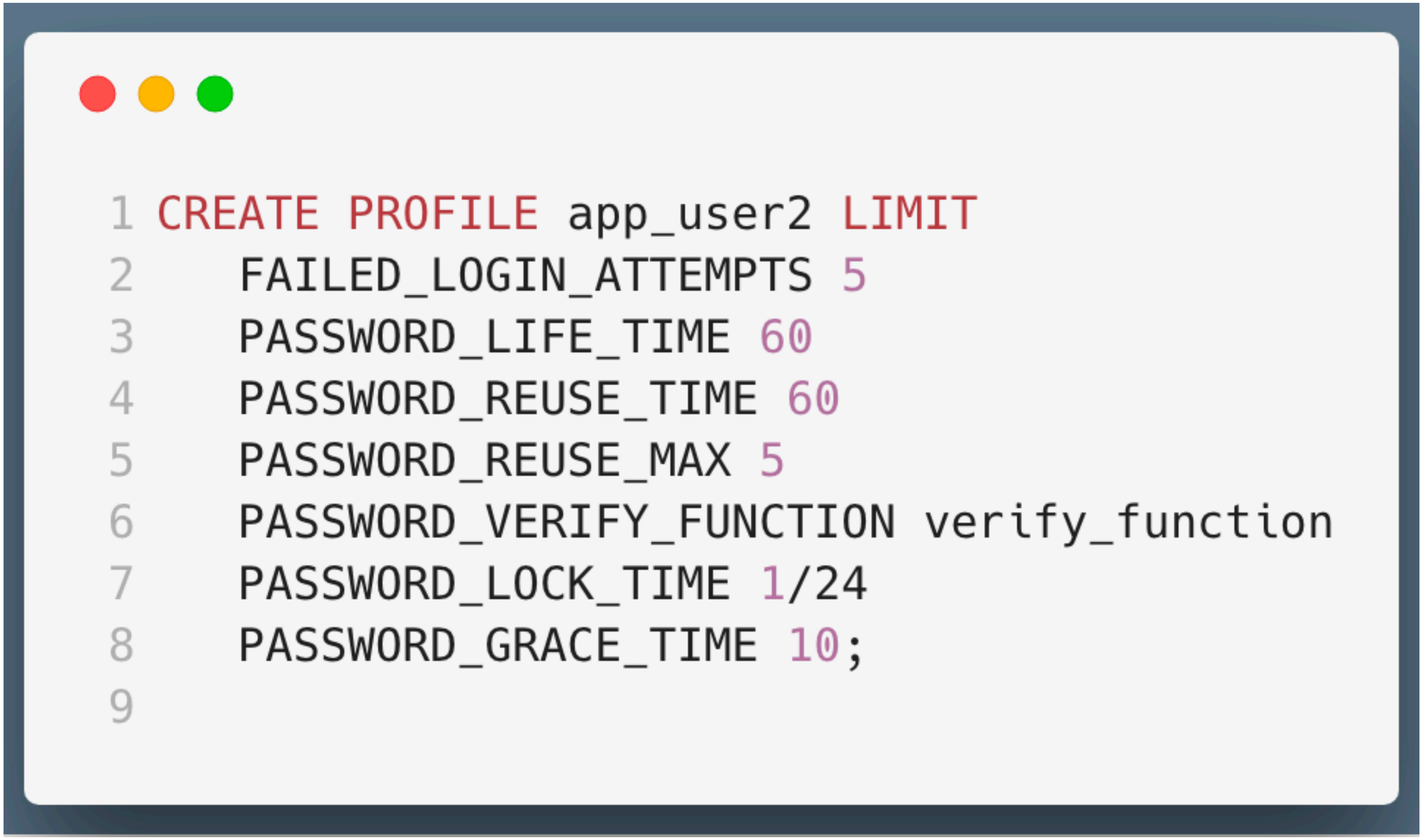
```
CREATE PROFILE nombrePerfil LIMIT
 nombreLimitacion1 valor1
 [nombreLimitacion2 valor2...];
Ejemplo:

CREATE PROFILE usuariosASIX LIMIT
   SESSIONS_PER_USER          2
   CPU_PER_SESSION            UNLIMITED
   CONNECT_TIME               60
   PASSWORD_REUSE_TIME        365
   PASSWORD_REUSE_MAX         2;
```

PROFILES



```
CREATE PROFILE app_user LIMIT
   SESSIONS_PER_USER          UNLIMITED
   CPU_PER_SESSION            UNLIMITED
   CPU_PER_CALL               3000
   CONNECT_TIME               45
   LOGICAL_READS_PER_SESSION  DEFAULT
   LOGICAL_READS_PER_CALL     1000
   PRIVATE_SGA                15K
   COMPOSITE_LIMIT            5000000;
```

PROFILES



```
CREATE PROFILE app_user2 LIMIT
    FAILED_LOGIN_ATTEMPTS 5
    PASSWORD_LIFE_TIME 60
    PASSWORD_REUSE_TIME 60
    PASSWORD_REUSE_MAX 5
    PASSWORD_VERIFY_FUNCTION verify_function
    PASSWORD_LOCK_TIME 1/24
    PASSWORD_GRACE_TIME 10;

```

BEST PRACTICES

- Plan Your Access Carefully
- Grant Privileges to Roles Instead of Specific Users
- Assign the Least Access Needed for Users to Do Their Jobs
- Create Users for Individual DBAs Instead of Generic DBA Users
- Split DBA Role from Developer Role
- Do Not Use Oracle Roles Such as CONNECT

- Roles Can Be Nested
  - An HR role that allows HR users to view salary data.
  - A DEVELOPER role that allows developers to create tables.
  - A MANAGER role that combines both HR and DEVELOPER.

- Avoid Granting privileges to PUBLIC

- Only Grant Necessary System Privileges to Trusted Users

Thank you!