

TEMA 6: LENGUAJE DE DEFINICIÓN DE DATOS (DDL)

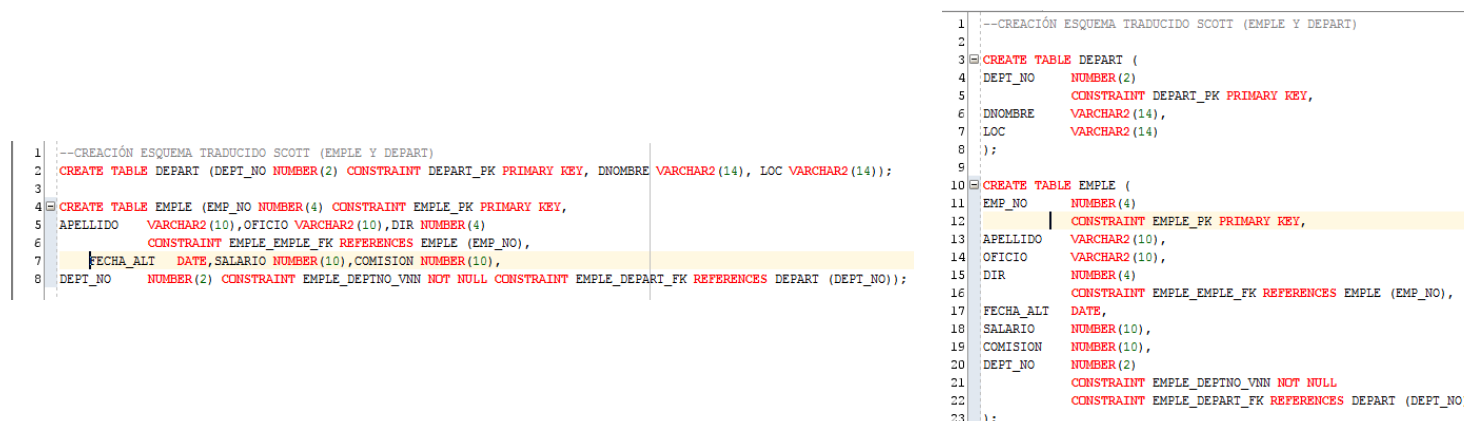
RECOMENDACIONES PARA LA REALIZACIÓN DE SCRIPTS SQL

Tal y como hemos visto en las prácticas del tema, la realización de un script SQL atiende fundamentalmente a la reutilización de este en distintas ocasiones, es por esto que, a la hora de escribir el script SQL conviene tener en cuenta las siguientes **recomendaciones**:

1.- Escribir el script con el adecuado sangrado o indentación para facilitar su legibilidad

No es lo mismo escribir un script con conciencia de leerlo posteriormente que sin ella. Un script **debe leerse con facilidad**, tanto en el momento de escribirlo como en el futuro cuando vuelve a leerse.

Para muestra un botón:



```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2  CREATE TABLE DEPART (DEPT_NO NUMBER(2) CONSTRAINT DEPART_PK PRIMARY KEY, DNOMBRE VARCHAR2(14), LOC VARCHAR2(14));
3
4  CREATE TABLE EMPL (EMP_NO NUMBER(4) CONSTRAINT EMPL_PK PRIMARY KEY,
5  APELLIDO VARCHAR2(10), OFICIO VARCHAR2(10), DIR NUMBER(4)
6  CONSTRAINT EMPL_EMPL_FK REFERENCES EMPL (EMP_NO),
7  FECHA_ALT DATE, SALARIO NUMBER(10), COMISION NUMBER(10),
8  DEPT_NO NUMBER(2) CONSTRAINT EMPL_DEPTNO_VNN NOT NULL CONSTRAINT EMPL_DEPART_FK REFERENCES DEPART (DEPT_NO));
  
```

```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2
3  CREATE TABLE DEPART (
4  DEPT_NO      NUMBER(2)
5              CONSTRAINT DEPART_PK PRIMARY KEY,
6  DNOMBRE     VARCHAR2(14),
7  LOC         VARCHAR2(14)
8  );
9
10 CREATE TABLE EMPL (
11 EMP_NO      NUMBER(4)
12             CONSTRAINT EMPL_PK PRIMARY KEY,
13 APELLIDO    VARCHAR2(10),
14 OFICIO      VARCHAR2(10),
15 DIR         NUMBER(4)
16             CONSTRAINT EMPL_EMPL_FK REFERENCES EMPL (EMP_NO),
17 FECHA_ALT   DATE,
18 SALARIO     NUMBER(10),
19 COMISION    NUMBER(10),
20 DEPT_NO     NUMBER(2)
21             CONSTRAINT EMPL_DEPTNO_VNN NOT NULL
22             CONSTRAINT EMPL_DEPART_FK REFERENCES DEPART (DEPT_NO)
23 );
  
```

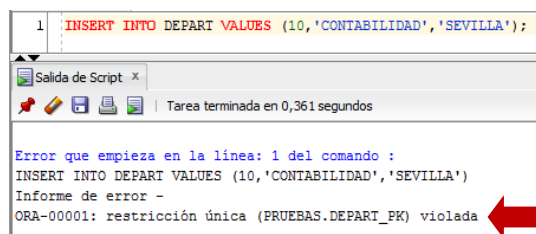
¿Cuál se lee mejor?

Aunque parezca mentira:

- Estos scripts contienen exactamente las mismas sentencias.
- Se me entregan scripts como el primero...

2.- Todas las restricciones deben tener un nombre.

Tal y como se indica en el tema, **todas las restricciones en Oracle deben tener un nombre**. Este nombre es utilizado por Oracle al dar el mensaje de error cuando una restricción es violada. Es recomendable poner nuestro propio nombre, ya que, si no se indica, Oracle lo elegirá por nosotros (sysxxxx).



```

1  INSERT INTO DEPART VALUES (10, 'CONTABILIDAD', 'SEVILLA');
  
```

Salida de Script x

Tarea terminada en 0,361 segundos

Error que empieza en la línea: 1 del comando :

```

INSERT INTO DEPART VALUES (10, 'CONTABILIDAD', 'SEVILLA')
  
```

Informe de error -

ORA-00001: restricción única (PRUEBAS.DEPART_PK) violada

Conviene seguir las recomendaciones indicadas en el tema para dar nombre a las restricciones.

3.- El orden de creación de las tablas del esquema es importante.

Todo script de creación de un esquema lógico completo **debe tener en cuenta el orden adecuado de creación de las tablas para que no se produzcan errores**. Existen tablas que referencian a otras mediante claves ajenas y no podrán crearse si las tablas referenciadas no están ya creadas.

- Por ejemplo, en el esquema lógico de las tablas EMPL y DEPART, no podremos crear nunca la tabla EMPL si no hemos creado previamente la tabla DEPART debido a que EMPL tiene una clave ajena que referencia a DEPART.

```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2
3  CREATE TABLE DEPART (
4      DEPT_NO      NUMBER(2)
5      CONSTRAINT DEPART_PK PRIMARY KEY,
6      DNOMBRE      VARCHAR2(14),
7      LOC          VARCHAR2(14)
8  );
9
10 CREATE TABLE EMPL (
11     EMP_NO      NUMBER(4)
12     CONSTRAINT EMPL_PK PRIMARY KEY,
13     APELLIDO     VARCHAR2(10),
14     OFICIO       VARCHAR2(10),
15     DIR          NUMBER(4)
16     CONSTRAINT EMPL_EMP_FK REFERENCES EMPL (EMP_NO),
17     FECHA_ALT    DATE,
18     SALARIO      NUMBER(10),
19     COMISION     NUMBER(10),
20     DEPT_NO      NUMBER(2)
21     CONSTRAINT EMPL_DEPTNO_VNN NOT NULL
22     CONSTRAINT EMPL_DEPART_FK REFERENCES DEPART (DEPT_NO)
23 );

```

4.- Se debe tener clara la diferencia entre un error de sintaxis y un error de ejecución.

El hecho de disponer de la sintaxis SQL orienta a la hora de escribir el script y ayuda a evitar errores de sintaxis. No es lo mismo un error de sintaxis que un error de ejecución:

- Un **error de sintaxis** se produce debido a que la sentencia se ha escrito mal según las normas del lenguaje SQL (sintaxis o forma del lenguaje). Una sentencia sintácticamente incorrecta suele detectarse por el intérprete y no se envía al servidor.

```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2
3  CREATE TABLE DEPART
4      DEPT_NO      NUMBER(2)
5      CONSTRAINT DEPART_PK PRIMARY KEY,
6      DNOMBRE      VARCHAR2(14),
7      LOC          VARCHAR2(14)
8  );

```

Salida de Script x

Tarea terminada en 0,29 segundos

Error que empieza en la línea: 3 del comando :

```

CREATE TABLE DEPART
DEPT_NO      NUMBER(2)
CONSTRAINT DEPART_PK PRIMARY KEY,
DNOMBRE      VARCHAR2(14),
LOC          VARCHAR2(14)
)

```

Informe de error -

ORA-00922: falta la opción o no es válida

00922. 00000 - "missing or invalid option"

*Cause:

*Action:

- Un **error de ejecución** se puede producir por diferentes motivos una vez la sintaxis es correcta, es decir, cuando la sentencia **escrita correctamente** se ejecuta en el servidor.

```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2
3  CREATE TABLE DEPART (
4      DEPT_NO      NUMBER(2)
5      CONSTRAINT DEPART_PK PRIMARY KEY,
6      DNOMBRE      VARCHAR2(14),
7      LOC          VARCHAR2(14)
8  );

```

Salida de Script x

Tarea terminada en 0,241 segundos

Error que empieza en la línea: 3 del comando :

```

CREATE TABLE DEPART (
DEPT_NO      NUMBER(2)
CONSTRAINT DEPART_PK PRIMARY KEY,
DNOMBRE      VARCHAR2(14),
LOC          VARCHAR2(14)
)

```

Informe de error -

ORA-00955: este nombre ya lo está utilizando otro objeto existente

00955. 00000 - "name is already used by an existing object"

*Cause:

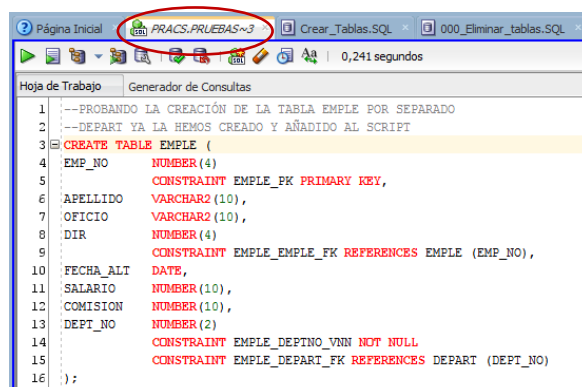
*Action:

5.- Es recomendable escribir el script poco a poco y no todo de golpe

Precisamente por lo comentado en el punto anterior, es recomendable minimizar la posibilidad de acumular errores de sintaxis, sobre todo cuando se está empezando. Aunque el editor dirigido por la sintaxis nos ayuda a localizar errores de sintaxis, en ocasiones puede confundirnos también, por lo que si realizamos un script largo de golpe su depuración puede ser muy confusa.

Un consejo útil es realizar el script tabla a tabla, es decir, realizar la sentencia de creación de cada tabla por separado en una hoja de trabajo de pruebas e ir completando el script final en otra hoja de trabajo con las sentencias ya verificadas. Para comprobar al final que el script completo funciona correctamente tan solo hay que realizar un script de borrado de todas las tablas (que se han creado correctamente por separado) y ejecutar de nuevo el script completo.

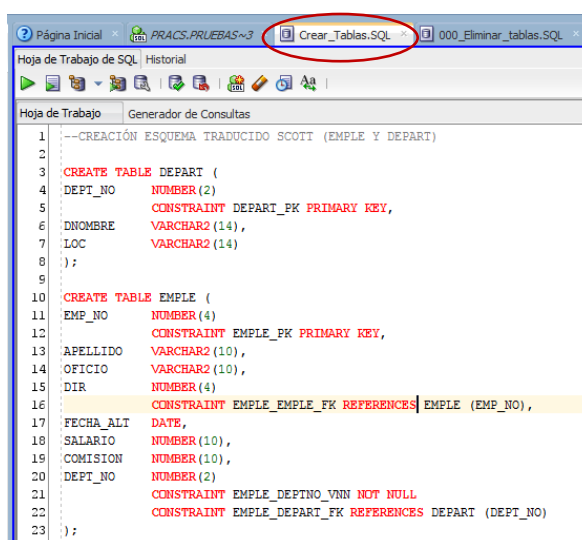
Prueba individual. Creación de las tablas por separado.



```

1  --PROBANDO LA CREACIÓN DE LA TABLA EMPLE POR SEPARADO
2  --DEPART YA LA HEMOS CREADO Y AÑADIDO AL SCRIPT
3  CREATE TABLE EMPLE (
4      EMP_NO          NUMBER(4)
5      CONSTRAINT EMPLE_FK PRIMARY KEY,
6      APELLIDO        VARCHAR2(10),
7      OFICIO          VARCHAR2(10),
8      DIR             NUMBER(4)
9      CONSTRAINT EMPLE_EMPLE_FK REFERENCES EMPLE (EMP_NO),
10     FECHA_ALT        DATE,
11     SALARIO          NUMBER(10),
12     COMISION         NUMBER(10),
13     DEPT_NO          NUMBER(2)
14     CONSTRAINT EMPLE_DEPTNO_VNN NOT NULL
15     CONSTRAINT EMPLE_DEPART_FK REFERENCES DEPART (DEPT_NO)
16 );
  
```

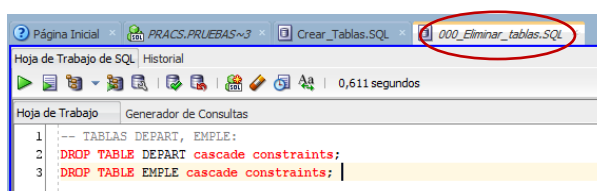
Script completo. Se van añadiendo las sentencias ya probadas



```

1  --CREACIÓN ESQUEMA TRADUCIDO SCOTT (EMPLE Y DEPART)
2
3  CREATE TABLE DEPART (
4      DEPT_NO          NUMBER(2)
5      CONSTRAINT DEPART_FK PRIMARY KEY,
6      DNOMBRE          VARCHAR2(14),
7      LOC              VARCHAR2(14)
8  );
9
10 CREATE TABLE EMPLE (
11     EMP_NO            NUMBER(4)
12     CONSTRAINT EMPLE_FK PRIMARY KEY,
13     APELLIDO          VARCHAR2(10),
14     OFICIO            VARCHAR2(10),
15     DIR               NUMBER(4)
16     CONSTRAINT EMPLE_EMPLE_FK REFERENCES EMPLE (EMP_NO),
17     FECHA_ALT         DATE,
18     SALARIO           NUMBER(10),
19     COMISION          NUMBER(10),
20     DEPT_NO           NUMBER(2)
21     CONSTRAINT EMPLE_DEPTNO_VNN NOT NULL
22     CONSTRAINT EMPLE_DEPART_FK REFERENCES DEPART (DEPT_NO)
23 );
  
```

Script de borrado de las tablas. Permite probar el script de creación varias veces.



```

1  -- TABLAS DEPART, EMPLE:
2  DROP TABLE DEPART cascade constraints;
3  DROP TABLE EMPLE cascade constraints;
  
```