

En este laboratorio usaremos la sentencia CREATE TABLE para crear las tablas del esquema que vamos a usar en las sesiones prácticas del curso.

## Preliminares

En algunos laboratorios se presentará una versión básica (simplificada) de la sintaxis de los comandos que se van a aplicar.

Para describir la sintaxis de esos comandos se usarán las siguientes convenciones tipográficas:

- Se usará una fuente diferente (ejemplo)
- La cursiva y negrita (**ejemplo**) indica marcadores de posición; son palabras que se deben reemplazar según las definiciones que se dan para cada caso
- Los elementos entre corchetes ( [ ejemplo ] ) son opcionales
- Las llaves y las líneas verticales indican que debe elegir una alternativa ( { ejemplo | otro } )
- Los puntos ( . . . ) significan que el elemento precedente se repite (si los puntos están entre corchetes, la repetición es opcional)

## CREATE TABLE

La sintaxis simplificada de la sentencia CREATE TABLE<sup>1</sup>, se muestra a continuación.

```
CREATE [GLOBAL TEMPORARY] TABLE [schema.]table
( { column data_type [inline_constraint [ ... ] ] | table_constraint } [,
... ] )
```

**schema** es el nombre del esquema al que pertenece la tabla

**table** es el nombre de la tabla

**column** es el nombre de la columna

**data\_type** es el tipo de dato (dominio) de la columna<sup>2</sup>.

**inline\_constraint** permite definir reglas de integridad<sup>3</sup> como parte de la definición de la columna. Éstas reglas pueden ser:

```
[ { NOT NULL | CHECK ( expression ) | DEFAULT default_expr |
  UNIQUE | PRIMARY KEY |
```

<sup>1</sup> <https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-TABLE.html#GUID-F9CE0CC3-13AE-4744-A43C-EAC7A71AAAB6>

<sup>2</sup> Los tipos de datos se pueden consultar en: <https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/Data-Types.html#GUID-7B72E154-677A-4342-A1EA-C74C1EA928E6>

<sup>3</sup> Las restricciones de integridad se pueden consultar en: <https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/constraint.html#GUID-1055EA97-BA6F-4764-A15F-1024FD5B6DFE>

```
REFERENCES [ refschema. ] reftable [ ( refcolumn ) ] [ ON DELETE { CASCADE | SET NULL } ] } ]
```

**table\_constraint** permite definir reglas de integridad que involucran varias columnas. Estas reglas pueden ser:

```
{ CHECK ( expression ) | UNIQUE ( column [, ... ] ) |
  PRIMARY KEY ( column [, ... ] ) |
  FOREIGN KEY ( column [, ... ] ) REFERENCES reftable [ ( refcolumn [, ... ]
) ]
  [ ON DELETE { CASCADE | SET NULL } ] }
```

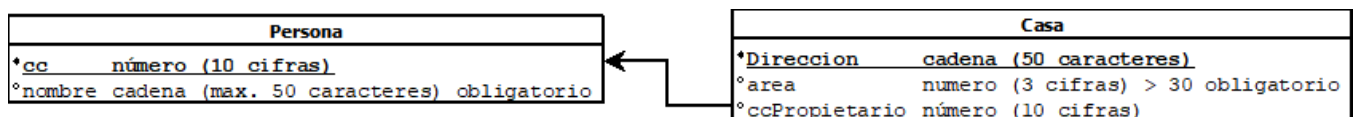
**expression** es una expresión que retorna un valor booleano

**default\_expr** es un valor que se asigna a la columna cuando se inserta un registro que no provee ese valor. La expresión puede ser un valor literal, una función, o el llamado a una función (ej. SYSDATE)

**reftable**, **refcolumn**, **refschema** son, respectivamente, los nombres de la tabla y columna a la que hace referencia, y del esquema donde está la tabla

## Ejemplo de Create Table

Dado el siguiente modelo relacional:



Los siguientes comandos crean las tablas Persona y Vehículo:

```
CREATE TABLE Persona (
    cc NUMBER(10) PRIMARY KEY,
    nombre VARCHAR2(50) NOT NULL);

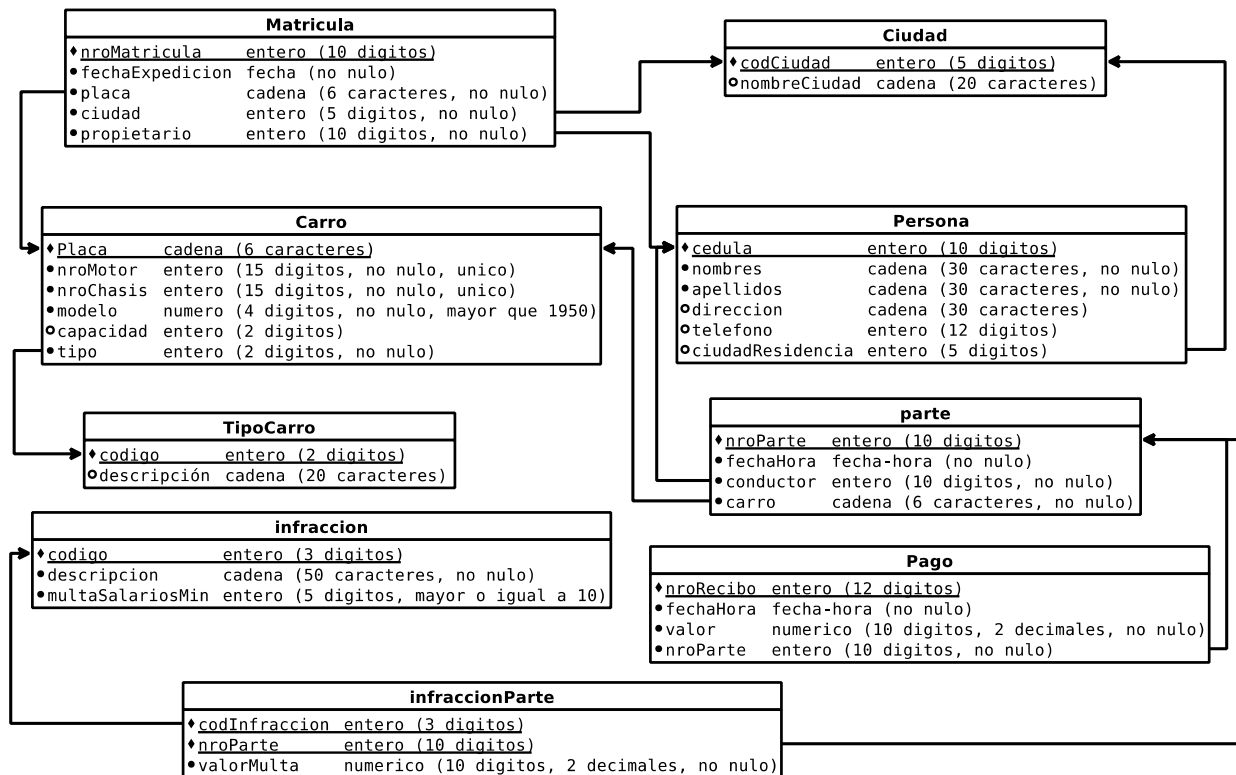
CREATE TABLE Casa (
    direccion VARCHAR2(50) PRIMARY KEY,
    area NUMBER(3) NOT NULL CHECK (area > 30),
    ccPropietario NUMBER(10) REFERENCES persona );
```

## Caso de Aplicación

En las sesiones prácticas usaremos el caso del manejo de información para administrar los partes y multas que se han dado a carros matriculados en Colombia. El sistema incluye los datos básicos del carro y la matrícula, la lista de infracciones que se tienen en Colombia, y la información de los propietarios de vehículos y de los conductores que han recibido un parte. Un carro puede tener varias matrículas, dado que puede ser vendido varias veces a lo largo de su vida útil. Cada matrícula se registra en una ciudad.

Cuando un conductor comete una infracción, se hace un parte, que incluye los datos del conductor, del vehículo que conducía, y de la infracción. El conductor puede ser una persona distinta al dueño del carro. Un parte puede incluir varias infracciones del mismo conductor que ocurrieron en el mismo

momento. El conductor debe pagar las multas que estén asociadas a las infracciones que cometió y se registraron en el parte. Esto lo puede hacer en uno o varios pagos. Cada pago se aplica a las infracciones más viejas que tiene el conductor, y si el pago no cubre toda la deuda, quedan las más recientes (o parte de la más reciente) sin cubrir.



**Nota:** se implementa una versión simplificada del sistema, ajustada al alcance necesario para la realización de las sesiones prácticas. Además, a propósito, algunos atributos no se han incluido, ya que se agregarán más adelante usando la cláusula ALTER TABLE.

## Creación de las tablas

El objetivo de este laboratorio es que usted cree las tablas del modelo del caso de aplicación (figura anterior).

Enviar el script con los comandos a [mcpabon@javerianacali.edu.co](mailto:mcpabon@javerianacali.edu.co)

Recuerde tomar copia de los scripts que ha trabajado en clase, ya que se pueden haber perdido cuando usted vuelva a ingresar al mismo computador. Adicionalmente, es posible que migremos a otra máquina, por lo cual serán necesarios los scripts.

**Referencias:** Manual de Oracle. En línea: <https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/CREATE-TABLE.html#GUID-F9CE0CC3-13AE-4744-A43C-EAC7A71AAAB6>

## Data Modeler

---

Una vez esté creado el esquema de su base de datos, lleve su modelo al Data Modeler, así:

- Elija en el menú: Archivo – Data modeler - Importar - Diccionario de Datos
- Elija la conexión
- Marque su esquema (bdXX)
- Seleccione las tablas
- Genere el diseño. Aparecerá la ventana del navegador del data modeler.
- Abra la ventana del explorador del data modeler: Ver – Data modeler – Explorador. Aparecerá entre los modelos relacionales el modelo que fue importado.
- Guarde el modelo en formato de imagen o pdf y envíelo por correo a [mcpabon@javerianacali.edu.co](mailto:mcpabon@javerianacali.edu.co)