

Siguiendo con el esquema de base de datos creado en el laboratorio 2, en este laboratorio realizaremos modificaciones en el esquema de la base de datos, usando la sintaxis básica de la sentencia **ALTER TABLE**. Adicionalmente, realizaremos consultas usando **OUTER JOIN**.

Sentencia ALTER TABLE

La sintaxis básica del ALTER TABLE es:

```
ALTER TABLE [schema.] table_name action
```

action es:

```
RENAME TO new_table_name
RENAME COLUMN column TO new_name
ADD ( column data_type [ inline_constraint [ ... ] ] [, ...] )
DROP( column [, ...] ) [CASCADE CONSTRAINTS]
MODIFY( column [ data_type ] [ inline_constraint [ ... ] ] [, ...] )
ADD table_constraint [, ...]
DROP PRIMARY KEY [ CASCADE ]
DROP UNIQUE ( column [, ...] )
DROP CONSTRAINT constraint_name [ CASCADE ]
```

data_type es el tipo de dato (dominio) de la *columna*¹.

inline_constraint permite definir reglas de integridad² como parte de la definición de la columna. Estas reglas pueden ser:

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

¹ Los tipos de dato se pueden consultar en: <https://docs.oracle.com/en/database/oracle/oracle-database/18/sqlrf/Data-Types.html#GUID-A3COD836-BADB-44E5-A5D4-265BA5968483>

² 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>

(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

Ejemplos: dada la tabla prueba,

```
CREATE TABLE prueba ( col1 NUMBER(5,2),
                        col2 VARCHAR2(5),
                        col3 VARCHAR(6) );
```

- Cambiar el nombre de la tabla:
`ALTER TABLE prueba RENAME TO nuevaPrueba;`
- Cambiar el nombre de una columna:
`ALTER TABLE nuevaPrueba RENAME COLUMN col1 TO colNew;`
- Borrar una columna:
`ALTER TABLE nuevaPrueba DROP (colNew) CASCADE CONSTRAINTS;`
- Cambiar la definición de una columna:
`ALTER TABLE nuevaPrueba MODIFY (col2 NUMBER(5) UNIQUE);`
- Agregar una llave primaria compuesta:
`ALTER TABLE nuevaPrueba ADD PRIMARY KEY (col2, col3);`

Cláusula FROM con OUTER JOIN

La sintaxis básica es:

```
FROM { table_reference | join_clause | ( join_clause ) } [ , ... ]
```

table_reference es:

```
{ [ schema. ] table | ( subquery ) } [ alias ]
```

join_clause es:

```
table_reference { inner_cross_join_clause | outer_join_clause }...
```

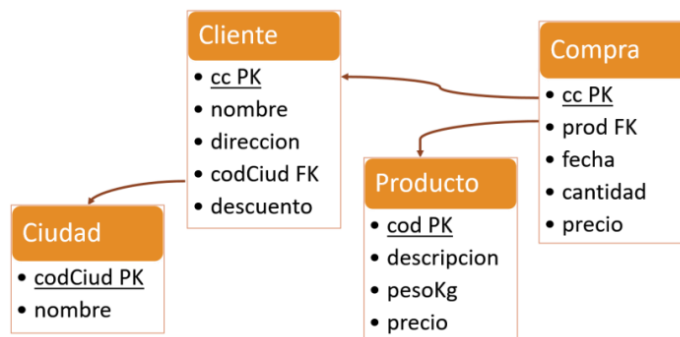
inner_cross_join_clause es:

```
{ [ INNER ] JOIN table_reference { ON condition | USING ( column
[ , column ]... ) } | { CROSS | NATURAL [ INNER ] }
JOIN table_reference }
```

outer_join_clause es:

```
[ NATURAL ] { FULL | LEFT | RIGHT } [ OUTER ] JOIN table_reference
[ ON condition | USING ( column [ , ... ] ) ]
```

Ejemplo: dado el siguiente esquema relacional,



- Listar todos los productos, y si fueron sido comprados, la fecha y cantidad comprada:

```
SELECT cod, descripcion, fecha, cantidad
FROM producto LEFT OUTER JOIN compra ON (cod=prod);
```

- Listar todos los clientes, y si han hecho compras, sido comprados, las fechas en que compraron:

```
SELECT cc, nombre, fecha
FROM compra NATURAL RIGHT OUTER JOIN cliente;
```

Ejercicios

Usando el esquema de base de datos creado en los laboratorios anteriores, escriba una sentencia SQL para cada uno de los requerimientos de ésta sección.

Nota. La referencia de la sentencia ALTER TABLE se encuentra en: [ALTER TABLE \(oracle.com\)](https://www.oracle.com/technetwork/database/enterprise/alter-table-0891907.html)

- Se requiere agregar a la información de los carros el atributo **Color**, que tiene un texto.
- Se requiere agregar en la tabla **TipoCarro** una restricción para asegurar que el atributo **descripción** no sea nulo.
- Modificar la tabla **Pago**, cuyo atributo **fechaHora** es no nulo, para que use como valor por defecto, cuando no se provea ese atributo, la fecha actual del sistema.
- Agregar a la tabla **Infraccion** el atributo **id** que es un número de 5 dígitos. Cambie la clave primaria de la tabla para que en adelante sea el **id**. Para ello, actualice los datos de la tabla, asignando el id a partir de **1000**, e incrementando de 1 en 1, según el orden en que aparezca cada registro.
- Liste todas las ciudades y, si tienen personas que residen en ellas, el nombre y dirección de los residentes.
- Liste todas las ciudades y si tienen partes registrados, el valor total de los pagos recibidos para esa ciudad

7. Liste todas las infracciones definidas en el sistema, y si están relacionadas con un parte que ha tenido algún pago, el número de recibo y la fecha del pago.

Al finalizar la sesión, cada estudiante debe enviar el script a mcpabon@javerianacali.edu.co.