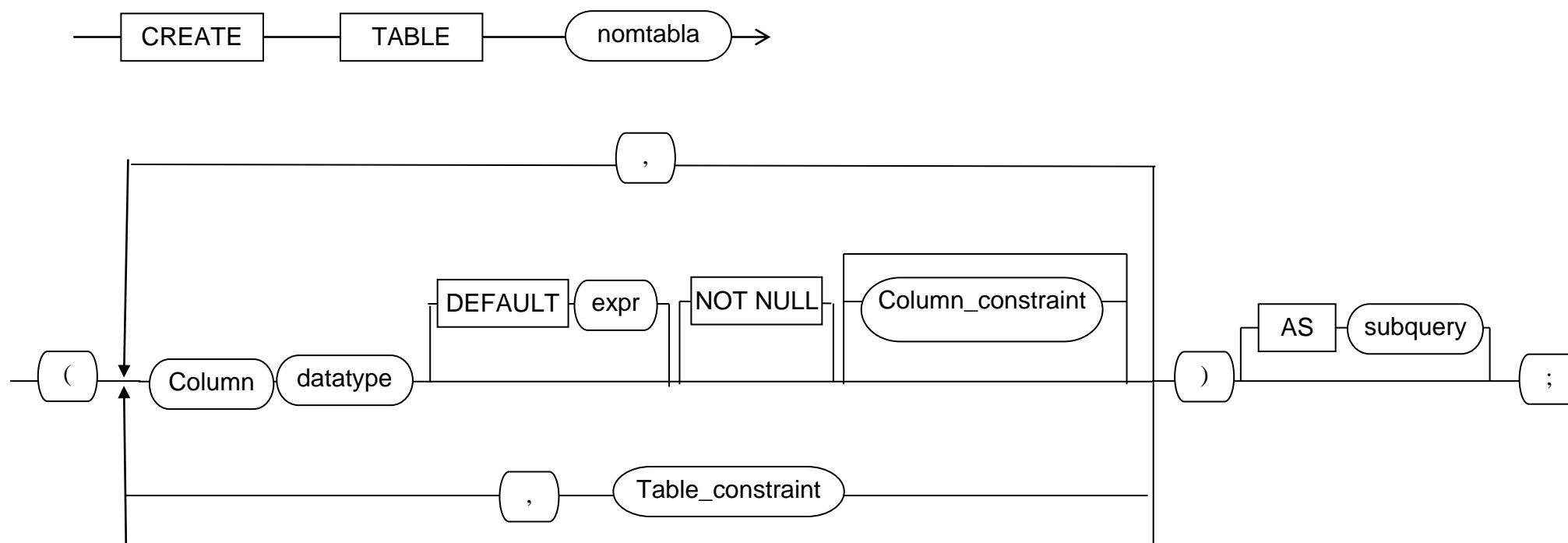


Sesión 1 – Repaso de FBD (2).

Sentencias CRUD.

CREATE TABLE para crear una tabla

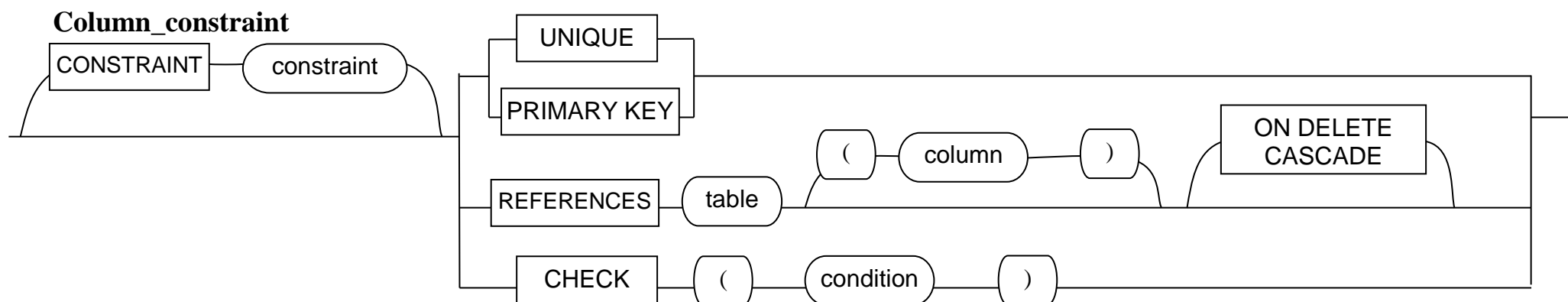


donde:

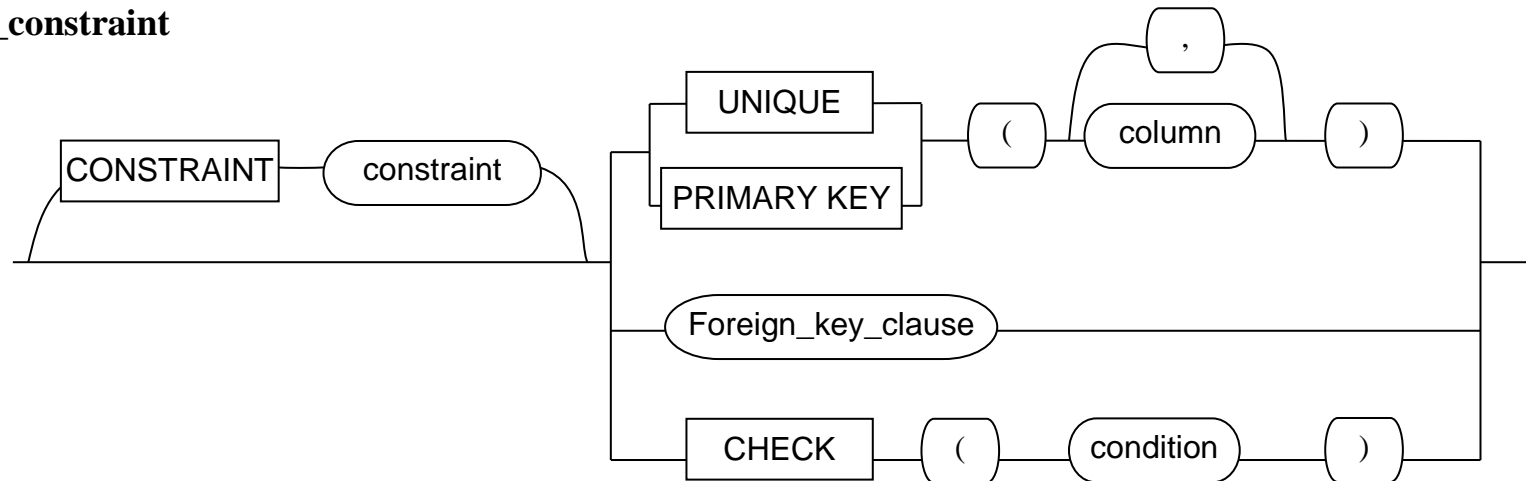
- nomtabla: Es el nombre de una tabla.
- column: Es el nombre de una columna de la tabla.
- datatype: Especifica el tipo de datos de la columna pudiendo ser éste uno de los definidos por el sistema (se pueden encontrar en la ayuda, normalmente utilizaremos char(n), varchar2(n), number(n,m), integer, date)
- Table-constraint /Column-constraint: Restricciones a nivel de tabla (table-constraint) o de columna (column-constraint).

Se puede definir: una CLAVE PRIMARIA por tabla, restricciones UNIQUE, CLAVES AJENAS, una restricción DEFAULT por columna, y restricciones CHECK. Todas ellas pueden ser introducidas en la misma sentencia, según la sintaxis que se muestra en la siguiente página. Se debe tener en cuenta los siguientes aspectos:

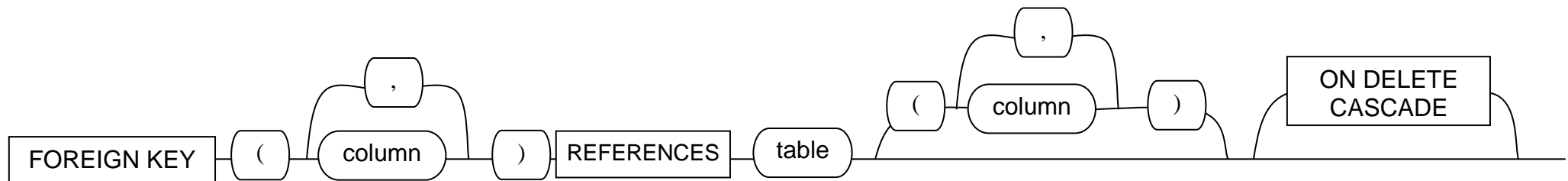
- ~ Las columnas definidas como clave primaria no podrán contener nulos y no habrá combinaciones repetidas de sus valores.
- ~ La combinación de valores de las columnas enumeradas en un UNIQUE no se podrán repetir. Las columnas que tengan una restricción UNIQUE sí pueden contener nulos.
- ~ Al definir la integridad referencial, los tipos de datos de las columnas que constituyen la clave ajena y los de las columnas referenciadas deben ser compatibles. Cuando la restricción se está definiendo a nivel de columna se omiten las palabras FOREIGN KEY y la enumeración de columnas.
- ~ La restricción DEFAULT especifica el valor que se le coloca a una columna por defecto cuando al insertar una fila en la tabla esa columna carece de valor explícito.
- ~ La restricción CHECK limita los posibles valores que pueden tomar una columna o conjunto de columnas. Se pueden definir múltiples restricciones CHECK para una tabla, sin embargo sólo se puede definir una a nivel de columna. Cuando para una columna existe más de una restricción CHECK se evalúan todas. La restricción CHECK que afecte a más de una columna debe ser definida a nivel de tabla.
- ~ En la sintaxis que se muestra a continuación, donde pone "CONSTRAINT constraint", se refiere a que a las restricciones se les da un nombre. Si el nombre no lo proporcionamos nosotros, el sistema le da automáticamente uno. Si queremos darle nosotros un nombre a la restricción, pondremos la palabra CONSTRAINT y a continuación el nombre que queramos. El hecho de que un usuario dé nombre a las restricciones es para facilitar posteriormente el trabajo con las mismas, por lo tanto el usuario debe definirse un criterio a la hora de definirse los nombres.



Table_constraint



Foreign_key_clause



Ejemplo:

Para ilustrar la sintaxis, se muestran las restricciones definidas de distinto modo en cada tabla para así ilustrar distintos modos de especificarlas.

```
CREATE TABLE pieza (numpieza number(10) CONSTRAINT cppiezas PRIMARY KEY,  
                    nompieza varchar2(30),  
                    preciovent number(6) CONSTRAINT chprecioven CHECK (preciovent >0));
```

```
CREATE TABLE vendedor (numvend number(5),  
                        nomvend varchar2(30), nombrecomer varchar2(30), direccion varchar2(30),  
                        telefono varchar2(15), poblacion varchar(20), provincia varchar2(20),  
                        PRIMARY KEY(numvend));
```

```
CREATE TABLE preciosum (numvend integer, numpieza varchar2(16) CONSTRAINT fkprepie REFERENCES pieza,  
                        preciounit number(6), diassum number(3) DEFAULT 2 CONSTRAINT chdiassum CHECK (diassum>0),  
                        CONSTRAINT fkpreven FOREIGN KEY (numvend) REFERENCES vendedor,  
                        CONSTRAINT cppreciosum PRIMARY KEY (numpieza, numvend));
```

DROP TABLE para borrar una tabla

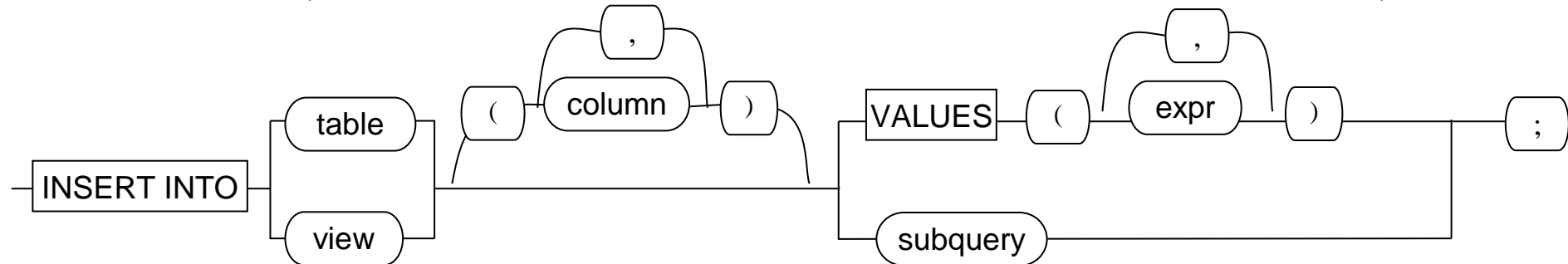


En muchas ocasiones para borrar las tablas se deberá seguir un orden adecuado, ya que puede que no se nos permita el borrado de alguna tabla para no interferir en la integridad referencial de la base de datos.

INSERT para insertar filas en una tabla

Si no se especifica la lista de columnas se supondrá que se va a insertar una fila con valor para todas las columnas de la tabla.

Se debe tener en cuenta que el orden de la lista de valores debe coincidir con el orden dado a las columnas, si se enumeran, o con el que tienen en la tabla.

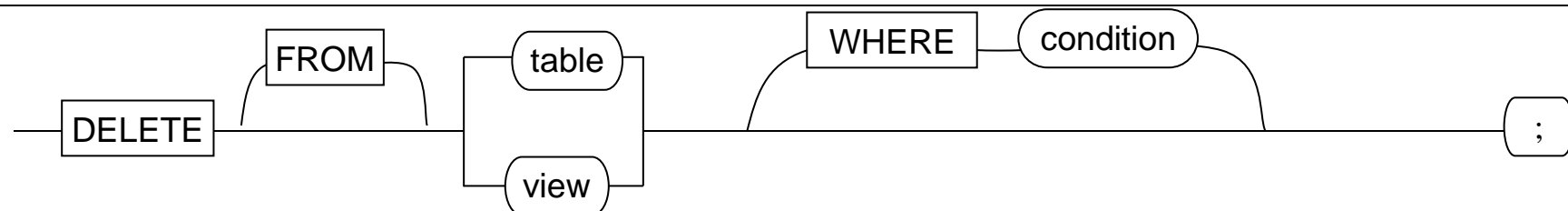


INSERT INTO pieza VALUES ('A25D', 'DESTORNILLADOR MULTIPLE', 1500);

INSERT INTO pieza(numpieza) VALUES ('A29C');

INSERT INTO subpiezas(num, precio) SELECT numpieza, preciovent FROM pieza WHERE preciovent>150;

DELETE para borrar filas de una tabla



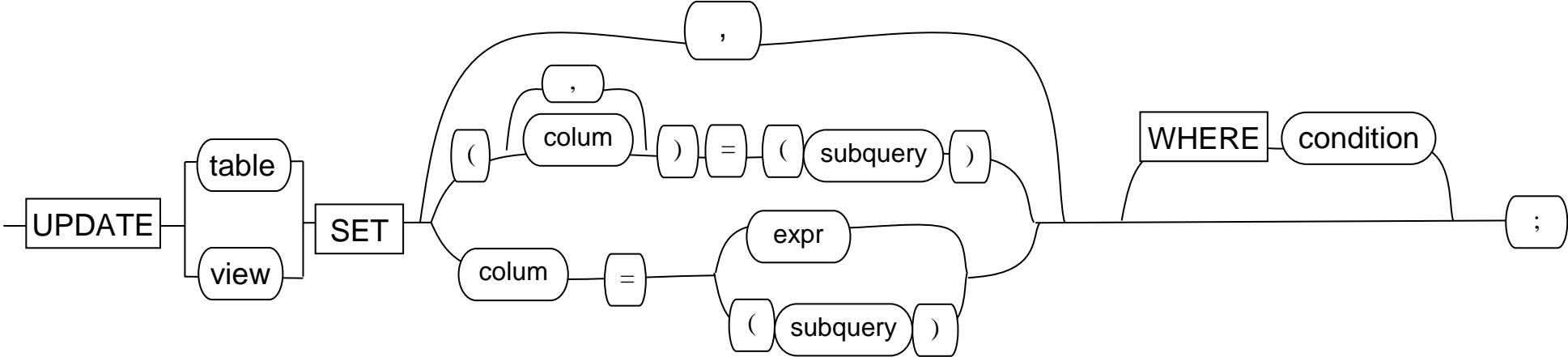
Si no se especifica ninguna condición (WHERE), se borrarán todas las filas de la tabla o vista.

Ejemplos

DELETE FROM pieza WHERE numpieza='A29D'

DELETE FROM pieza WHERE numpieza LIKE '%DESTORNILLADOR%'

UPDATE para modificar los datos de una tabla



Al igual que en el caso de la sentencia DELETE, si no se especifica ninguna condición, la modificación afectará a todas las filas de la tabla.

Ejemplo

```
UPDATE piezas
SET preciovent= 2000
WHERE numpieza='A29D';
```

```
UPDATE vendedor
SET telefono='6565656565'
WHERE nombrecomer LIKE 'HARW%';
```

```
UPDATE vendedor
SET direccion='OLMO,19',
    telefono='666567686'
WHERE numvend= 2;
```

```
UPDATE PIEZA
SET precioent=(select max(precioent)
                FROM pieza)
WHERE numpieza='A-1001-L';
```

Utilidad práctica de los conceptos vistos en la sesión

- Sentencias Create table, drop table:

Son fundamentales para crear y borrar la estructura de la información de nuestro modelo de datos.

- Sentencias Insert, update, y delete:

Nos permiten incluir nueva información en las tablas de nuestro sistema, modificar datos ya existentes (filas completas, o datos individuales), o borrar filas.