Lenguaje de Definición de Datos

Tipos de Datos
Operadores
Sentencias
CREATE
INSERT
DELETE
DROP

1. Tipos de Datos en SQL

NUMBER(p,s) Representa números
- **p**
  - Precisión, número de dígitos
  - máx 38, mín 1
- **s**
  - Escala, número de cifras decimales
  - máx 127, mín -84)

VARCHAR2(size) Representa cadena de caracteres de longitud variable
- **Size**
  - Longitud
  - máx 4000, mín 1

DATE Tipo Fecha

LONG Cadena de caracteres de longitud variable de hasta 2 gigabytes

2. Tipos de Operadores en SQL

+,–, ||	Sumar, Restar, Concatenar
***, /**	Multiplicar, Dividir
=, =>, <=, <, >	Comparadores clásicos
IS NULL,BETWEEN, IN,LIKE	Comparadores especiales
NOT, AND, OR	Operadores Lógicos Clásicos

Comparadores Especiales

- Comparador **IS NULL** Detecta valores nulos
  - (A=10), A **IS NULL** ⇒ false, A is **NOT NULL**⇒true

- Comparador **BETWEEN** Detecta valores entre dos constantes
  - A **BETWEEN** X and Y ⇔ A >= X y A <=Y

- Comparador **IN** Detecta pertenencia a conjunto
  - A **IN** (1,2,3) ⇒ true si A=1 o A=2 o A=3

- Comparador **LIKE** Sirve para utilizar "máscaras" en cadenas de caracteres

  _ sustituye cualquier carácter
  - x **LIKE** '_A__' ⇒ true si x='1A23' ⇒false si x='1A234'

  % sustituye cualquier cadena
  - x **LIKE** '%A%' ⇒ true si x='1AX"%' ⇒true si X='ABLA'

3. Sentencias

Sentencia CREATE TABLE

CREATE TABLE *[usuario.]nombre_tabla* **(**
Nombre_atributo1 tipo1,
Nombre_atributo2 tipo2,

Nombre_atributon tipon,
[**CONSTRAINT** *nombre_de_restricción* Tipo de restricción (atributo[s]) afectados]
);

RESTRICCIONES DE TABLA

PRIMARY KEY Identifica la Clave Primaria

Sintaxis

CONSTRAINT *nombre_de_restricción* **PRIMARY KEY** (atributo o atributos implicados),

Ejemplos**:**

CONSTRAINT CP_ejemplo **PRIMARY KEY** (nombre_ejemplo),
CONSTRAINT PK_empleados **PRIMARY KEY** (nombre,apellido1,apellido2),

UNIQUE El atributo que tenga esta restricción ha de tomar valores únicos
Si hay más de un atributo, la combinación ha de ser única

Sintaxis

CONSTRAINT *nombre_de_restricción* **UNIQUE** (nombre de atributo o atributos),

Ejemplos:

CONSTRAINT AU_ejemplo **UNIQUE** (nom_dep),
CONSTRAINT unicos **UNIQUE** (nomprof,nomasig,numaula),

CHECK Comprueba si el atributo implicado pertenece a un conjunto o rango de valores

Sintaxis

CONSTRAINT *nombre_de_restricción* **CHECK** (nombre_atributo **IN** (conjunto de valores)),

CONSTRAINT *nombre_de_restricción* **CHECK** (nombre_atributo **BETWEEN** valor1 **AND** valor2),

Ejemplos:

CONSTRAINT CHIN_valor_crédito **CHECK** (numcred **IN** (3,6,9,12);
CONSTRAINT Chequea_nota **CHECK** (nota **BETWEEN** 0 **AND** 10);

FOREIGN KEY Obliga a la existencia de los valores de ese atributo en el atributo clave primaria de otra tabla a la que referencia

Sintaxis

CONSTRAINT *nombre_de_restricción* **FOREIGN KEY** (nombre_atributo/s)
REFERENCES nombre_tabla_ref (nombre_atributo/s),

Ejemplos:

CONSTRAINT FK_matricula **FOREIGN KEY** (*cod_est*) **REFERENCES** *estudiantes* (*dni*),

En la tabla que se está creando existe un atributo llamado *cod_est* que hace referencia al atributo *dni* de la tabla *estudiantes*, esto permite unir la tabla actual con la tabla e*studiantes*.

RESTRICCIONES DE COLUMNA

Hay un tipo de restricciones, que sólo afectan a un atributo y se indican en la sentencia de creación a continuación del atributo correspondiente:

CREATE TABLE *[usuario.]nombre_tabla* (
……
Nombre_atributok tipok **TIPO_RESTRICCIÓN**,
……
);

Las restricciones específicas de columna son **NOT NULL**, **DEFAULT**.

No obstante cualquier otra restricción que afecte a un único atributo puede ser descrita en la sentencia de creación como si fuera una restricción de columna. **No es recomendable**, impide darle nombre a la restricción.

NOT NULL Impide insertar valores **NULOS** en el atributo correspondiente
Por defecto **NULL**

Sintaxis

Nombre_atributok tipok **NOT NULL**,

Ejemplo:

CREATE TABLE estudiante(
……
apellido VARCHAR2(15) **NOT NULL**,
……
);

DEFAULT *valor* Si no se inserta ningún valor en ese atributo, por defecto valdrá *valor*

Sintaxis

Nombre_atributok tipok **DEFAULT valor_tipok** ,

Ejemplo:

CREATE TABLE estudiante(
……
curso NUMBER(1) **DEFAULT** 1,
……
);

Sentencia INSERT INTO ... VALUES

Esta sentencia se utiliza para insertar valores en las tablas creadas

Sintaxis

INSERT INTO *nombre_tabla* [*(atributo1, atributo2,..., atributon)*] **VALUES** (valor1,valor2,...,valorn)**;**

- *nombre_tabla* Nombre de la tabla en la que se quieren insertar los datos.
- [*(atributo1, atributo2,..., atributon)*] es **optativa** cuando se inserta una fila completa y ordenada. Cuando sólo se insertan valores para algunos atributos, o estos atributos están desordenados es **obligatoria**.
- (valor1,valor2,...,valorn) valores del mismo tipo que los atributos correspondientes.

Ejemplos:

INSERT INTO estudiante **VALUES** ('1342567',3344,'Juan','Pérez',2);
INSERT INTO estudiante (dni,nia,apellido) **VALUES** ('723564',3443,'López');

Sentencia DELETE FROM

Esta sentencia se utiliza para borrar todas o algunas filas de una tabla

Sintaxis

DELETE FROM *nombre_tabla* [**WHERE** *condición_de_selección*]**;**

- Sin el corchete se borran todas las filas de la tabla.
- **WHERE** *condición_de_selección* En este caso sólo se borrarán las filas seleccionadas.

Ejemplos:

DELETE FROM estudiante;
Borra todas las filas de la tabla estudiante.

DELETE FROM estudiante
WHERE curso=4;
Borra de la tabla estudiante las filas que en el atributo curso tienen el valor 4.

Sentencia DROP TABLE

Esta sentencia se utiliza para borrar una tabla

Sintaxis

DROP TABLE *nombre_tabla* [**CASCADE CONSTRAINTS**]**;**

- **CASCADE CONSTRAINTS** Se utiliza para poder borrar una tabla que es referenciada por otra u otras para las que su clave primaria es clave ajena.

Ejemplos:

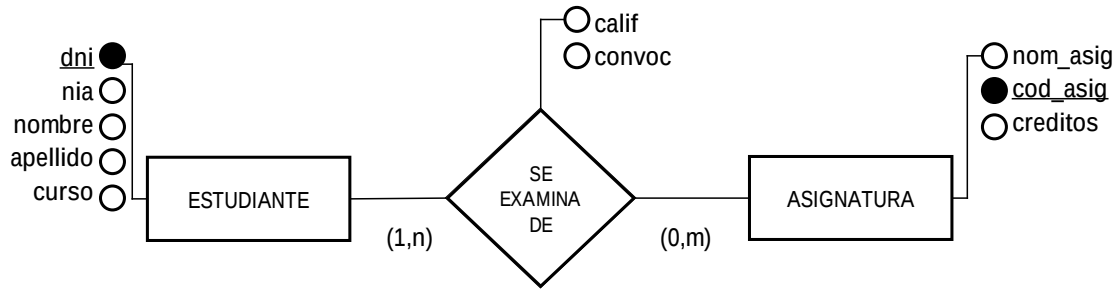
DROP TABLE notas;
Los atributos: *cod_est,cod_asig,convoc* que constituyen la clave primaria de la tabla *notas* no están referenciados por ninguna otra tabla.

DROP TABLE estudiante **CASCADE CONSTRAINTS**;
Como el atributo *dni* de *estudiante* es referenciada por la clave ajena *cod_est* en la tabla *notas* es obligado añadir la cláusula CASCADE CONSTRAINTS

Ejemplo de Creación de Tablas

Consideremos el siguiente supuesto:

- Se desea conocer la información sobre los estudiantes matriculados en determinadas asignaturas y las calificaciones que obtienen.
- De los estudiantes se quiere conocer su nombre, su apellido, su DNI y su número de identificación académica, ambos son únicos. Se quiere saber en qué curso está matriculado el estudiante, pudiendo estar matriculado en un solo curso.
- Un estudiante no podrá figurar si no se conoce su apellido. Por otro lado un estudiante siempre ha de estar matriculado en algún curso, al menos en 1º.
- Las asignaturas tienen un nombre, pudiendo existir más de una asignatura con el mismo nombre, un número de créditos múltiplo de 3, siempre inferior o igual a 12, y un código que las identifica de manera única.
- No puede aparecer ninguna asignatura si no se conoce su nombre. La mayoría de asignaturas tienen 6 créditos.
- Los estudiantes pueden estar matriculados en varias asignaturas, y al menos en una.
- Pueden existir asignaturas en las que no esté matriculado ningún estudiante.
- Interesa almacenar la calificación que obtienen los estudiantes en las distintas convocatorias de las asignaturas en las que está matriculado. La calificación es siempre positiva e igual o inferior a 10 pudiendo tener 1 decimal.
- Como se quiere tener una lista antes de los exámenes de las asignaturas, debe permitirse la posibilidad de tener la lista sin la calificación, pero con la convocatoria.



Se elige **dni** como *clave primaria*, sabiendo que ***nia*** es *clave candidata*, para la entidad ***estudiante***.

Paso a tablas

Tabla **estudiante**:

dni	nia	Nombre	Apellido	Curso
cadena de caracteres clave única	numero valor único	cadena de caracteres	cadena de caracteres obligatorio	Numero obligatorio

Tabla **asignatura**:

cod_asig	nom_asig	creditos
numero clave única	cadena de caracteres	numero entero toma los valores 3,6,9,12 obligatorio, por defecto 6

Tabla de la relación *"SE EXAMINA DE"* **notas**:

cod_est	**cod_asig**	calif	**convoc**
cadena de caracteres forma parte de la CP clave ajena referencia a *dni* de *estudiante*	numero forma parte de la CP clave ajena referencia a *cod_asig* de *asignatura*	numero decimal <=0 y <= 10	numero forma parte de la CP obligatorio

IMPLEMENTACIÓN

```
CREATE TABLE estudiante(
dni             VARCHAR2(10),
nia             NUMBER(4) UNIQUE,
nombre                  VARCHAR2(15),
apellido                VARCHAR2(15) NOT NULL,
curso           NUMBER(1) DEFAULT 1,
CONSTRAINT PK_estudiante PRIMARY KEY (dni)
);
```

ALTERNATIVA
EN LA SIGUIENTE CREACIÓN LA RESTRICCIÓN DE **UNIQUE** *PARA* **nia** *ESTÁ NOMINADA*

```
CREATE TABLE estudiante(
dni             VARCHAR2(10),
nia             NUMBER(4),
nombre                  VARCHAR2(15),
apellido                VARCHAR2(15) NOT NULL,
curso           NUMBER(1) DEFAULT 1,
CONSTRAINT PK_estudiante PRIMARY KEY (dni),
CONSTRAINT Unica_estudiante UNIQUE (nia)
);
```

```
CREATE TABLE asignatura(
cod_asig        NUMBER(3),
nom_asig        VARCHAR2(15) NOT NULL,
creditos                NUMBER(2) DEFAULT 6,
CONSTRAINT PK_asignatura PRIMARY KEY (cod_asig),
CONSTRAINT CH_asignatura CHECK (creditos IN (3,6,9,12))
);


CREATE TABLE notas(
cod_est                 VARCHAR2(10),
cod_asig        NUMBER(3),
convoc          NUMBER(1) NOT NULL,
calif           NUMBER(2,1),
CONSTRAINT PK_notas PRIMARY KEY (cod_est,cod_asig,convoc),
CONSTRAINT FK1_notas FOREIGN KEY (cod_est) REFERENCES estudiante (dni),
CONSTRAINT FK2_notas FOREIGN KEY (cod_asig) REFERENCES asignatura
(cod_asig),
CONSTRAINT CH_notas CHECK (calif BETWEEN 0 AND 10)
);
```