



## Boletín de ejercicios Tema 4

---

### Ejercicio 1

Crea la base de datos de la siguiente manera: **CREATE DATABASE colegio;**

Para que en las sucesivas consultas utilicemos esta base de datos tecleamos: **USE colegio;**

Crea la tabla, denominada **estudiantes**, con los siguientes campos:

- **Id** (valor numérico)
- **nombre** (secuencia de, como máximo, 50 caracteres). Esta columna no puede tomar valores nulos
- **nif** (secuencia de 9 caracteres)
- **nacimiento** (es una fecha)
- **observaciones** (secuencia de, como máximo, 100 caracteres)

Al final de la tabla se debe especificar:

- Con el nombre **pk\_alumno**, la restricción que especifica que una de las variables anteriores es clave primaria de la tabla.
- Con el nombre **uq\_nif** que los valores de la columna nif no pueden repetirse

Inserta los siguientes valores en la tabla acabas de crear especificando todas las columnas en las que se van a insertar:

- (100, 'Beatriz', '12345678A', '1992-01-01', NULL)
- (101, 'Yolanda', '23456789B', '1972-02-02', NULL)
- (102, 'Ebba', '34567890C', '1980-06-08', NULL)
- (103, 'Sonia', '45678901D', '1982-01-22', NULL)
- (104, 'Pedro', '56789012E', '1992-01-01', NULL)

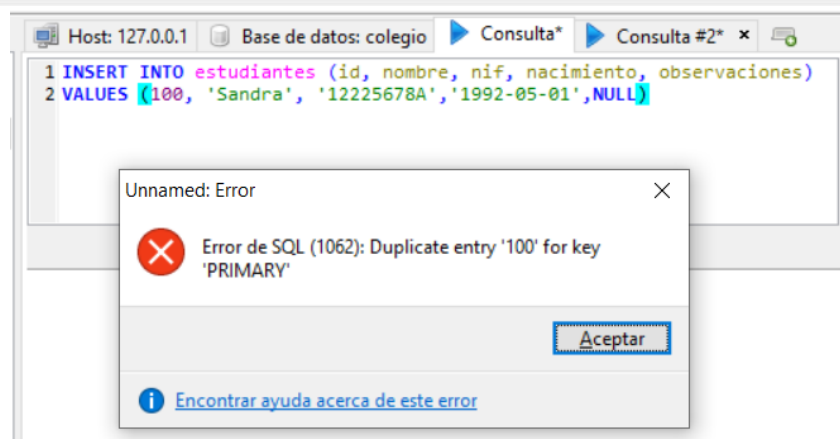


Genera un error al insertar un nuevo valor debido a la restricción **pk\_alumno** y realiza una captura de pantalla.

### Solución:

```
CREATE DATABASE colegio;  
  
USE colegio;  
  
CREATE TABLE estudiantes (  
    id            INTEGER,  
    nombre        VARCHAR(50) NOT NULL,  
    nif           CHAR(9),  
    nacimiento    DATE,  
    observaciones VARCHAR(100),  
  
    CONSTRAINT pk_estudiante PRIMARY KEY (id),  
    CONSTRAINT uq_nif UNIQUE KEY (nif)  
);
```

```
1 INSERT INTO estudiantes (id, nombre, nif, nacimiento, observaciones)  
2 VALUES (100, 'Beatriz', '12345678A', '1992-01-01', NULL),  
3         (101, 'Yolanda', '23456789B', '1972-02-02', NULL),  
4         (102, 'Ebba', '34567890C', '1980-06-08', NULL),  
5         (103, 'Sonia', '45678901D', '1982-01-22', NULL),  
6         (104, 'Pedro', '56789012E', '1992-01-01', NULL)  
7
```



### Ejercicio 2

Crea la tabla, denominada **docentes**, con los siguientes campos:



- Id (valor numérico)
- nombre (secuencia de, como máximo, 50 caracteres). Esta columna no puede tomar valores nulos
- nif (secuencia de 9 caracteres)
- nacimiento (es una fecha)
- observaciones (secuencia de, como máximo, 100 caracteres)

Al final de la tabla se debe especificar:

- Con el nombre pk\_docente, la restricción que especifica que una de las variables anteriores es clave primaria de la tabla.
- Con el nombre uq\_nif que los valores de la columna nif no pueden repetirse

Inserta los siguientes valores en la tabla que acabas de crear sin especificar las columnas en las que se van a insertar:

- (10, 'Andrea', '25874469A', '1968-05-01', NULL)
- (11, 'Gerardo', '78552369B', '1960-11-02', NULL)
- (12, 'María', '36046789C', '1965-02-25', NULL)
- (13, 'Janet', '36124418D', '1970-03-22', NULL)
- (14, 'Javier', '52876631F', '1970-06-11', NULL)

Genera un error en los campos nombre y nif al insertar filas. Realiza capturas de pantalla de los mismos.

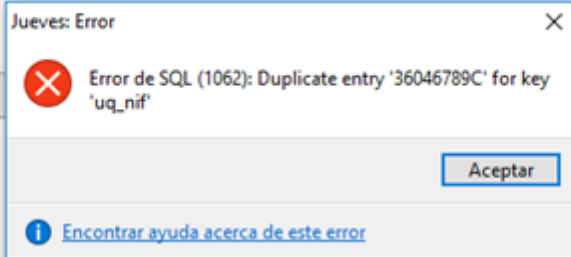
### **Solución:**



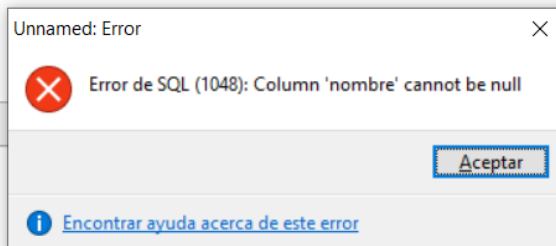
```
1 USE colegio;
2
3 CREATE TABLE docentes (
4     id            INTEGER,
5     nombre        VARCHAR(50) NOT NULL,
6     nif           CHAR(9),
7     nacimiento    DATE,
8     observaciones  VARCHAR(100),
9
10    CONSTRAINT pk_docentes PRIMARY KEY (id),
11    CONSTRAINT uq_nif UNIQUE KEY (nif)
12 );
```

```
1 INSERT INTO docentes
2 VALUES (10, 'Andrea', '25874469A', '1968-05-01', NULL),
3         (11, 'Gerardo', '78552369B', '1960-11-02', NULL),
4         (12, 'María', '36046789C', '1965-02-25', NULL),
5         (13, 'Janet', '36124418D', '1970-03-22', NULL),
6         (14, 'Javier', '52876631F', '1970-06-11', NULL)
7
```

```
1 INSERT INTO docentes (id,nombre,nif,nacimiento,observaciones)
2 VALUES (25,'Jessica','36046789C','1972-11-22',NULL)
3
```



```
1 INSERT INTO docentes (id,nombre,nif,nacimiento,observaciones)
2 VALUES (13, NULL, '12333678A', '1972-05-01', NULL)
3
```





### **Ejercicio 3**

Crea una tabla, denominada **aulas**, con los siguientes campos:

- **id** (valor numérico)
- **ubicación** (secuencia de tres caracteres)
- **capacidad** (valor numérico)

Para que todo sea correcto, añade las restricciones que consideres y en donde consideres.

Inserta en dicha tabla los siguientes valores:

- (1001,'A01',15)
- (1002,'A02',15)
- (1014,'A14',16)

### **Solución:**

```
1 CREATE TABLE aulas (  
2   id          INTEGER,  
3   ubicación   CHAR(3),  
4   capacidad   INTEGER,  
5  
6   CONSTRAINT pk_aula PRIMARY KEY (id),  
7   CONSTRAINT uq_ubicación UNIQUE KEY (ubicación));
```

```
1 INSERT INTO aulas  
2 VALUES (1001,'A01',15),(1002,'A02',15),(1014,'A14',16)  
3
```

### **Ejercicio 4**

Crear las siguientes tablas teniendo en cuenta que dependen de otras (no considerar en este ejercicio modificación o eliminación de datos). Añade las restricciones que consideres y donde consideres:

- Tabla, denominada **cursos**, con los siguientes campos:



- **id** (valor numérico)
  - **docente\_id** (valor numérico)
  - **aula\_id** (valor numérico)
  - **nombre** (secuencia de, como máximo 50 caracteres)
  - **fecha\_inicio** (almacena una fecha)
  - **fecha\_fin** (almacena una fecha)
  - **horario** (secuencia de, como máximo 50 caracteres)
- Tabla, denominada **clases**, con los siguientes campos:
- **id** (valor numérico)
  - **curso\_id** (valor numérico)
  - **fecha** (almacena una fecha)
  - **contenido** (secuencia de caracteres de, como máximo, 100 caracteres)
- Tabla **estudiantes\_cursos**
- **estudiante\_id** (valor numérico)
  - **curso\_id** (valor numérico)

**Solución:**



```
1 CREATE TABLE cursos (  
2   id          INTEGER,  
3   docente_id  INTEGER,  
4   aula_id     INTEGER,  
5   nombre      INTEGER,  
6   fecha_inicio DATE,  
7   fecha_fin   DATE,  
8   horario     VARCHAR(50),  
9  
10  CONSTRAINT pk_curso PRIMARY KEY (id),  
11  CONSTRAINT fk_docente FOREIGN KEY (docente_id)  
12    REFERENCES docentes(id),  
13  CONSTRAINT fk_aula FOREIGN KEY (aula_id)  
14    REFERENCES aulas(id)  
15 );  
16
```

```
1 CREATE TABLE clases (  
2   id          INTEGER,  
3   curso_id    INTEGER,  
4   fecha       DATE,  
5   contenido   VARCHAR(100),  
6  
7   CONSTRAINT pk_clase PRIMARY KEY (id),  
8   CONSTRAINT uq_clase UNIQUE KEY (curso_id, fecha),  
9   CONSTRAINT fk_curso FOREIGN KEY (curso_id)  
10    REFERENCES cursos(id)  
11 );  
12
```

```
1 CREATE TABLE estudiantes_cursos (  
2   estudiante_id  INTEGER,  
3   curso_id       INTEGER,  
4  
5   CONSTRAINT pk_estudiantecurso PRIMARY KEY (estudiante_id, curso_id),  
6   CONSTRAINT fk_estudiante FOREIGN KEY (estudiante_id)  
7     REFERENCES estudiantes(id),  
8   CONSTRAINT fk_curso2 FOREIGN KEY (curso_id)  
9     REFERENCES cursos(id)  
10 );  
11
```

## Ejercicio 5

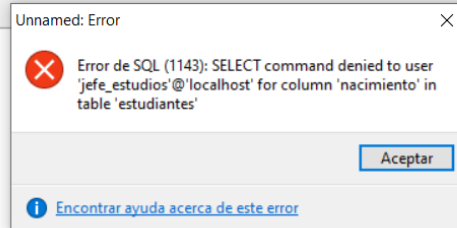
Utilizando el icono de usuarios de HeidiSQL, crea un usuario denominado **jefe\_estudios** de modo que pueda acceder a la tabla **estudiantes** pero sólo puede consultar el nombre y el NIF del estudiante. Configura este usuario de modo que puede delegar este permiso en otro usuario. Demuestra que no puede acceder a otra tabla o a otra columna de la tabla estudiantes.



### **Solución:**

```
1 GRANT SELECT (nif,nombre)
2 ON estudiantes
3 TO jefe_estudios@localhost
4 WITH GRANT OPTION
5
```

```
1 SELECT nacimiento
2 FROM estudiantes
```



```
1 SELECT *
2 FROM docentes
```

