



MODELOS DE DATOS

D.M.L. - CRUD



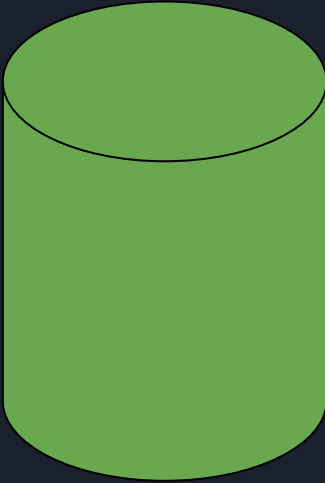
RECORDERIS...





¿QUÉ APRENDEREMOS HOY?

- Operaciones de modificación de datos (insert, update, delete)





¿Alguien que ya las conozca?



INSECCIÓN

Opción 1:

Especificar todos y cada uno
de los campos

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers values (21, 'MILEY', 'CYRUS', '1992-11-23', 'POP');
```

INSECCIÓN

Opción 2:

Omitir auto-incremental,
especificando los otros
campos

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers (first_name, last_name, birth_date, genre)
values ('MILEY', 'CYRUS', '1992-11-23', 'POP');
```

INSERTCIÓN

Opción 3:

Solo campos obligatorios,
especificando los nombres

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers (first_name, birth_date, genre)
values ('MILEY', '1992-11-23', 'POP');
```

INSECCIÓN

Opción 4:

Múltiples registros en una sola sentencia

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers (first_name, last_name, birth_date, genre) values  
( 'MILEY', 'CYRUS', '1992-11-23', 'POP' ),  
( 'LADY', 'GAGA', '1986-03-28', 'POP' ),  
( 'RIHANNA', null, '1988-20-02', 'POP' );
```


INSECCIÓN

Opción 5:

Resultado de una consulta
sobre datos existentes

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers (first_name, last_name, birth_date, genre)
select first_name, last_name, birth_date, genre
from another_singers
```

INSECCIÓN

Opción 6:

Reemplazar si la clave
primaria ya existe

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
insert into singers (id, first_name, last_name, birth_date, genre)
values (21, 'MILEY', '1992-11-23', 'POP')
on duplicate key update
    first_name = 'MILEY', last_name = 'CYRUS',
    birth_date = '1992-11-23', genre = 'POP';
```

INSECCIÓN

Opción 7:

DDL + insert

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
create table singers2 as
select *
from singers
```



INSECCIÓN

Opción 8, 9, 10, 11.....

Las anteriores son sólo algunas de las formas básicas de ejecutar comandos tipo "insert".

Con el tiempo aprenderás a conocer más y a dominarlas!

BORRADO

Opción 1:

Borrar de un registro
especificando la clave
primaria (la más común)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
delete
from   singers
where  id = 21;
```

BORRADO

Opción 2:

Borrar varios registros
especificando una condición

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
delete
from   singers
where  genre = 'POP';
```

BORRADO

Opción 3:

Borrar varios registros
especificando varias
condiciones

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
delete
from   singers
where  genre IN ('POP', 'ROCK')
and    birth_date >= '2001-01-01';
```

BORRADO

Opción 4:

La más riesgosa (y una de las mejores formas de quedarse sin trabajo)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
delete  
from   singers;
```



IMPORTANTE!!!

Repite conmigo:

"No debo ejecutar un delete sin
where, salvo que sea
estrictamente necesario!!!"



ACTUALIZACIÓN

Opción 1:

Actualizar un campo de un registro especificando la clave primaria (la más común)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
update singers
set first_name = 'MILEY'
where id = 21;
```

ACTUALIZACIÓN

Opción 2:

Actualizar varios campos de un registro especificando la clave primaria

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
update singers
set first_name = 'MILEY',
    last_name = 'CYRUS'
where id = 21;
```

ACTUALIZACIÓN

Opción 3:

Actualizar varios campos de
varios registros
especificando una condición

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
update singers
set genre = 'POPULAR'
where genre = 'POP';
```

ACTUALIZACIÓN

Opción 4:

La más riesgosa (y otra de las mejores formas de quedarse sin trabajo)

Field	Type	Null	Key	Default	Extra
id	int(11)	NO	PRI	NULL	auto_increment
first_name	varchar(30)	NO		NULL	
last_name	varchar(30)	YES		NULL	
birth_date	date	NO		NULL	
genre	varchar(15)	NO		NULL	

5 rows in set (0,004 sec)

```
update singers
set genre = 'ROCK';
```



IMPORTANTE!!!

Repite conmigo:

"No debo ejecutar un update
sin where, salvo que sea
estrictamente necesario!!!"



EJERCICIOS

Para realizar los siguientes ejercicios se debe usar la base de datos "world".

Si aún no la tiene en su entorno de trabajo por favor referirse a la documentación de la clase anterior

city
123 ID
ABC Name
ABC CountryCode
ABC District
123 Population

countrylanguage
ABC CountryCode
ABC Language
ABC IsOfficial
123 Percentage

country
ABC Code
ABC Name
ABC Continent
ABC Region
123 SurfaceArea
123 IndepYear
123 Population
123 LifeExpectancy
123 GNP
123 GNPOld
ABC LocalName
ABC GovernmentForm
ABC HeadOfState
123 Capital
ABC Code2



EJERCICIOS (1 / 6)

Ejercicio 1

Crear una copia de la tabla "country" con el nombre "country2". Debe quedar igual tanto en campos y tipos de datos como en el contenido

Ejercicio 2:

Crear una copia de la tabla "city" con el nombre "city2". Debe quedar igual tanto en campos y tipos de datos como en su contenido



EJERCICIOS (2 / 6)

Ejercicio 3

Escribir una sentencia sql para insertar en la tabla "country2" los datos de las siguientes micro-naciones, preferiblemente usando una única sentencia. El código del país queda a libre elección.

Nombre, Continente, Idioma, Superficie, Habitantes

Sealand, Europa, Inglés, 0.05km², 50 habitantes

Christiania, Europa, Danés, 0.34 km², 1000 habitantes

República de Molissia, Norteamérica, 45000 km², 35 habitantes



EJERCICIOS (3 / 6)

Ejercicio 4

Escribir una sentencia sql para actualizar la tabla "country2", de modo que se aumente en un 20% la población de todos los países asiáticos.

Ejercicio 5

Escribir una sentencia sql para actualizar los nombres de los países en la tabla "country2", de forma que todos los nombres queden en mayúsculas.



EJERCICIOS (4 / 6)

Ejercicio 6

Escribir una sentencia sql para actualizar los nombres de las ciudades en la tabla "city2", de forma que todos los nombres queden en mayúsculas.

Ejercicio 7

Escribir una sentencia sql para eliminar de la tabla "city2" las ciudades europeas de la región de europa central, con población menor a 10 millones de habitantes y expectativa de vida menor a 73 años



EJERCICIOS (5 / 6)

Ejercicio 8

Escribir una sentencia sql para eliminar de la tabla "city2" las ciudades cuyo nombre sea palíndromo

Ejercicio 9

Escribir una sentencia sql para actualizar en la tabla "city2" el distrito a 'N/A' para las ciudades cuyo distrito sea vacío (ojo, no tener en cuenta valores nulos)



EJERCICIOS (6 / 6)

Ejercicio 10

Escribir una sentencia sql para actualizar en la tabla "country2" la capital a 'N/A' únicamente para los países cuya capital sea null (ojo, no tener en cuenta valores vacíos)

Ejercicio 11

Insertar el país con código "CCK" y nombre "Islas Cocos", de modo que si ya existe con el nombre "Cocos (Keeling) Islands", se le actualice el nombre.



¿PREGUNTAS?

