

Sistemas de Información para la Gestión

ICN292 - 2024/02

Eloy Alvarado Narváez 

eloy.alvarado@usm.cl

Universidad Técnica Federico Santa María

23 de octubre de 2024

Introducción a SQL

Recursos recomendados

Referencia

- SQL W3 School
- MySQL

Aplicaciones

Learn SQL & Database :



Solo learn :



Khan Academy :



Certificación gratis

Oracle MySQL Explorer



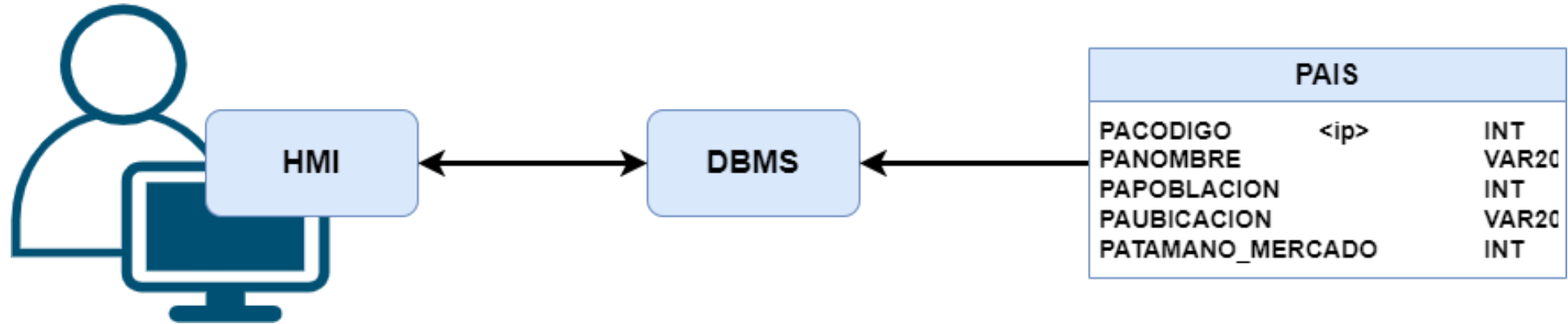
- Duración aproximada: 1 hr 30 min
- Incluye una pequeña certificación

Structured Query Language

- El lenguaje de Consulta Estructurado o SQL (*Structured Query Language*) es un lenguaje de programación para DBMS (o RDBMS), que permite crear bases de datos y especificar diversos tipos de operaciones en éstas.
- Es propuesto en 1974 por Donald Chamberlin y Raymond Boyce, se caracteriza por el manejo del álgebra y el cálculo relacional, así como por su simplicidad semántica y su aplicación multiplataforma de acceso a todo tipo de bases de datos relaciones.
 - Se convierte en estándar mundial (ISO) en 1987, de la ANSI en 1986.
- Permite crear tablas y relaciones, efectuar consultas y acciones, con el fin de guardar y recuperar fácilmente información de interés de una base de datos, así como también hacer cambios sobre ella.

Structured Query Language

- Siendo un **lenguaje de programación declarativo**, la operación se gestiona a través de comandos en un editor de la **DBMS**.



Structured Query Language

- Si bien cualquier editor puede ser usado para esta función, normalmente cada DBMS posee un propio editor, el cual provee la función de escritura y revisión de la semántica (no la lógica) de la sentencia SQL.
- El editor del DBMS no diferencia entre estilos de escritura (mayúsculas, minúsculas, cursiva, saltos de línea, etc) en las declaraciones SQL (preferencias del programador), pero los datos si deben ser registrados segun sus propias reglas.
- De todas formas existen reglas semánticas, como el uso de los paréntesis, comas, etc., y toda sentencia debe concluir con el caracter punto y coma (;) para ser ejecuta.
- Las sentencias son interpretadas y ejecutadas por el DBMS según sus propias características.

Structured Query Language

Escribiendo:

```
1 CREATE TABLE pais (pacodigo INTEGER NOT NULL, panombre VARCHAR(20) NOT NULL,
2  pablocacion INTEGER, paubicacion VARCHAR(20) DEFAULT 'América', patamano_mercado INTEGER, PRIMARY KEY (pacodigo));
```

DBMS interpreta:

Field	Type	Null	Key	Default	Extra
pacodigo	int(11)	NO	PRI	NULL	
panombre	varchar(20)	NO		NULL	
papoblacion	int(11)	YES		NULL	
paubicacion	varchar(20)	YES		América	
patamano_mercado	int(11)	YES		NULL	

Que ejecuta:

PAIS	
PACODIGO <ip>	INT
PANOMBRE	VAR20
PAPOBLACION	INT
PAUBICACION	VAR20
PATAMANO_MERCADO	INT

Funciones Básicas

- DDL (*Data Definition Language*):
 - Comandos de definición de la estructura de la BD relacional, como por ejemplo: crear, eliminar y alterar tablas; crear vistas, etc.
- DML (*Data Manipulation Language*):
 - Comandos de selección y extracción de datos, y de inserción, modificación y eliminación de campos o registros.
- DCL (*Data Control Language*):
 - Comandos de control y seguridad del sistema.

Comandos DDL

- **CREATE**: Utilizado para Crear tablas y atributos.

→ Sigue la estructura:

```
1 CREATE TABLE [nombre] ([atributo N] [características atributo N], ...);
```

- Ejemplo:

```
1 CREATE TABLE cliente (  
2   rut VARCHAR(10),  
3   nombre VARCHAR(20) NOT NULL,  
4   apellidos VARCHAR(20) NOT NULL,  
5   direccion VARCHAR(40),  
6   telefono VARCHAR(7),  
7   PRIMARY KEY (rut)  
8 );
```

- Si la clave primaria es compuesta se declara como primary key (clave 1, clave 2, ...)

Comandos DDL

CLIENTE		
RUT *	<ip>	VAR10
NOMBRE *		VAR20
APELLIDOS *		VAR20
DIRECCION		VAR40
TELEFONO		VAR7

Comandos DDL

- **TRUNCATE**: Empleado para eliminar el contenido de una tabla (manteniendo su estructura)

→ Sigue la estructura:

```
1 TRUNCATE TABLE [nombre];
```

- **DROP**: Empleado para eliminar una tabla (estructura y contenido).

→ Sigue la estructura:

```
1 DROP TABLE [nombre];
```

Comandos DDL

- **ALTER**: Utilizado para modificar la estructura de una tabla.
→ Sigue la estructura:

```
1 ALTER TABLA [nombre] ADD|DROP|CHANGE|MODIFY [modificaciones];
```

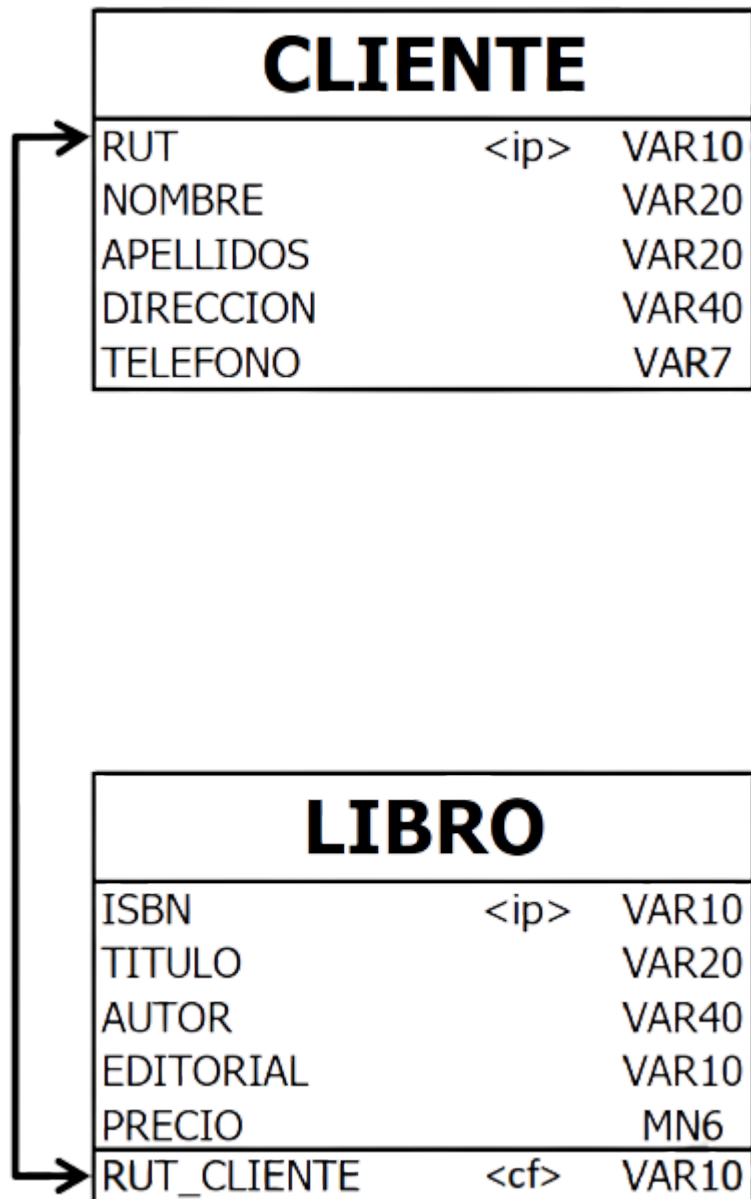
- **ADD**: Agrega un campo o columna.
- **DROP**: Elimina un campo o columna.
- **MODIFY**: Modifica el tipo de datos de un campo o columna.
- **CHANGE**: Modifica nombre y tipo de datos de un campo o columna.
- Ejemplo:

```
1 ALTER TABLE cliente ADD genero CHAR(1);
```

Comandos DDL

CLIENTE		
RUT	<ip>	VAR10
NOMBRE		VAR20
APELLIDOS		VAR20
DIRECCION		VAR40
TELEFONO		VAR7

Generación de Tablas y Relaciones



Creación tabla cliente

```

1 CREATE TABLE cliente (
2   rut VARCHAR(10),
3   nombre VARCHAR(20),
4   apellidos VARCHAR(20),
5   direccion VARCHAR(40),
6   telefono VARCHAR(7),
7   PRIMARY KEY (rut)
8 );

```

Creación tabla libro

```

1 CREATE TABLE libro (
2   isbn VARCHAR(10),
3   titulo VARCHAR(20),
4   autor VARCHAR(40),
5   editorial VARCHAR(10),
6   precio MN(6,0),
7   PRIMARY KEY (isbn)
8 );

```

Relaciones

```

1 ALTER TABLE libro ADD rut_cliente VARCHAR(10)
2
3 ALTER TABLE libro ADD FOREIGN KEY (rut_cliente)
4 REFERENCES cliente(rut) ON UPDATE RESTRICT;

```

Práctica



SQLZOO: Wiki-based tutorial.