

BD en MySQL - Repaso

Emmanuel Ayala Mexicano

LAMP

Concepto	Significado	Opciones
SISTEMA OPERATIVO	LINUX	WINDOWS, OS X, SOLARIS, ETC.
SERVIDOR WEB	APACHE	NGINX, LIGHTTPD,
DBMS	MYSQL	POSTGRE, SQLITE, ORACLE, DB2
LENGUAJE DE PROGRAMACIÓN	PHP Perl Python	JAVA, ASP, RUBY, C

Administrador de Base de Datos (dbms)

Crear / Eliminar Bases de Datos.

Crear / Eliminar Tablas.

Manejo de la información dentro de la BD.

Hacer respaldos de la BD.

Optimizar las BD.

USUARIOS

Función	Sintaxis	Ejemplo
Crear usuario	GRANT PRIVILEGIOS TO usuario IDENTIFIED BY password;	GRANT ALL PRIVILEGES ON *.* TO 'usuario'@'localhost' IDENTIFIED BY '000000';
Eliminar usuario	DROP USER usuario;	DROP USER 'usuario'@'localhost';

BD

Función	Sintaxis	Ejemplo
Crear Base de Datos	CREATE DATABASE mi_bd;	CREATE DATABASE ejemplo;
Eliminar Base de Datos	DROP DATABASE mi_bd;	DROP DATABASE ejemplo;
Mostrar Bases de Datos existentes	SHOW DATABASES;	SHOW DATABASES;
Utilizar Base de Datos	USE mi_bd;	USE ejemplos;
Mostrar las tablas de una Base de Datos	SHOW TABLES;	SHOW TABLES FROM ejemplo;*

* SE UTILIZA FROM nombre_de_la_base_de_datos cuando no está seleccionada una BD o cuando se quieren mostrar las tablas de otra base de datos que no está seleccionada.

TABLAS

Función	Sintaxis	Ejemplo
Crear tabla	CREATE TABLE mi_tabla (atributos, tipos);	CREATE TABLE usuarios (id INTEGER NOT NULL AUTO_INCREMENT PRIMARY KEY, nombre VARCHAR(255));
Eliminar tabla	DROP TABLE mi_tabla;	DROP TABLE usuarios;
Modificar Tabla	ALTER TABLE mi_tabla;	ALTER TABLE usuarios ADD rfc varchar(255) / ALTER TABLE usuarios DROP nombre
Describir tabla	DESCRIBE mi_tabla;	DESCRIBE usuarios;

DATOS

FUNCIÓN	MYSQL	SINTAXIS	EJEMPLO
AGREGAR	INSERT	INSERT INTO tabla VALUES();	INSERT INTO usuarios (email) VALUES ("Emmanuel");
SELECCIONAR	SELECT	SELECT * FROM tabla;	SELECT nombre FROM usuarios WHERE id > 1;
ACTUALIZAR	UPDATE	UPDATE tabla SET atributo = valor WHERE condicion;	UPDATE usuarios SET email = "amarrextro@yahoo.com", rol_id = 1 WHERE email = 'Emmanuel';
ELIMINAR	DELETE	DELETE FROM tabla WHERE condicion;	DELETE FROM usuarios WHERE id = 1;

WHERE

En MySQL podemos seleccionar datos que cumplan ciertas condiciones. Para lograr esto utilizamos el comando WHERE cuya función es comparar un atributo con un valor asignado.

La comparación la realizará un operador que designemos.

WHERE condición;

condición => atributo operador valor;

Resultados

Al obtener resultados de una consulta podemos agruparlos por un atributo.

GROUP BY atributo;

Al obtener resultados de una consulta podemos ordenarlos por uno o varios atributos de manera ascendente o descendente.

ORDER BY atributo (ASC / DESC);

Más condiciones - AND

Si queremos que varias condiciones se cumplan se pueden unir utilizando el conector AND.

SELECCIONAR TODOS LOS AUTOMOVILES CUYA FECHA DE FABRICACIÓN SEA MAS RECIENTE QUE 2005 Y QUE SEAN COLOR ROJO.

```
SELECT * FROM autos  
WHERE YEAR(fecha_de_fabricacion) > '2005' AND color = 'rojo';
```

Al menos una condición - OR

Si queremos filtrar los resultados que cumplan con un criterio preestablecido dentro de una lista podemos utilizar el conector OR.

SELECCIONAR TODOS LOS AUTOMÓVILES CUYO COLOR SEA ROJO O NEGRO.

```
SELECT * FROM autos  
WHERE color = 'rojo' OR color = 'negro';
```

Búsquedas anidadas

Podemos utilizar los resultados de una búsqueda para realizar otra búsqueda.

Selecciona el id de la especie donde el nombre es mamífero.

Selecciona todas las clasificaciones que sean mamíferos como mamíferos.

Selecciona todas las mascotas que estén clasificadas como perros.

```
SELECT * FROM mascota WHERE clasificacion_id = (  
  SELECT id FROM clasificacion WHERE especie_id = (  
    SELECT id FROM especie WHERE nombre LIKE '%mamifero%'  
  ));
```

Funciones - Contar / Distintos

Función	Sintaxis	Ejemplo
Contar elementos que cumplan con las condiciones	COUNT();	SELECT COUNT(*) FROM publicaciones;
Seleccionar elementos distintos que cumplan con las condiciones	DISTINCT();	SELECT DISTINCT(categoria_id) FROM publicaciones;
Contar los distintos elementos que cumplan con las condiciones	COUNT(DISTINCT());	SELECT COUNT(DISTINCT(categoria_id)) FROM publicaciones;

Las condiciones son especificadas con WHERE

Funciones - Matemáticas

Función	Sintaxis	Ejemplo
Obtiene el promedio de una columna* que cumpla con determinadas condiciones	AVG();	SELECT AVG(visto) FROM publicaciones;
Obtiene el valor máximo de una columna* que cumpla con determinadas condiciones	MAX();	SELECT MAX(visto) FROM publicaciones;
Obtiene el valor mínimo de una columna* que cumpla con determinadas condiciones	MIN();	SELECT MIN(visto) FROM publicaciones;
Suma los valores numéricos de una columna* que cumpla con determinadas condiciones	SUM();	SELECT SUM(visto) FROM publicaciones;

* La columna debe contener valores numéricos
Las condiciones son especificadas con WHERE

Funciones - Fecha / Hora

Función	Sintaxis	Ejemplo
Fecha Actual	CURDATE();	SELECT CURDATE();
Tiempo Actual	CURTIME();	SELECT CURTIME();
Fecha / Tiempo Actual	NOW();	SELECT NOW();
Año	YEAR(fecha);	SELECT YEAR(CURDATE());
Mes	MONTH(fecha);	SELECT MONTH(CURDATE());
Día	DAY(fecha);	SELECT DAY(CURDATE());

Ejemplos - Fecha / Hora (2)

Función	Sintaxis	Ejemplo
Hora	HOUR(tiempo);	SELECT HOUR(CURTIME());
Minuto	MINUTE(tiempo);	SELECT MINUTE(CURTIME());
Segundo	SECOND(tiempo);	SELECT SECOND(CURTIME());

RespalDOS

```
mysqldump -u root -p mi_base_de_datos >  
respaldo_de_mi_base_de_datos.sql
```

```
mysqldump -u root --databases contenido > respaldo_de_contenido.sql
```

```
mysql < respaldo_de_mi_base_de_datos.sql
```

* Se le agregó la opción `--databases` para que se incluyera el `CREATE DATABASE` en caso de que se restaure la BD en un equipo donde no se ha creado la misma.

Batch

Es posible utilizar MySQL en modo “batch” (procesamiento por lotes). Es decir ejecutar una serie de comandos (predefinidos) sin ingresar a la consola.

Este modo de operación resulta útil cuando se realizan los mismos comandos de manera frecuente o se realizan acciones similares (se crea un script y se toma como base para otros scripts). También sirve para evitar tener que teclear los comandos (sobre todo si incluyen varias líneas) si es que existiera algún error.

Al ejecutarse desde el Sistema Operativo, se pueden utilizar comandos del mismo sistema para examinar los resultados.

Batch

Sintaxis	Descripción
mysql < script	MySQL “lee” y ejecuta los comandos que estén dentro del script. Si hay algún error, MySQL detiene el procesamiento de los demás comandos.
mysql -h localhost -u root -p < script	Tambien se le pueden asignar los parámetros de conexión (servidor, usuario, contrseña) para que el script lo ejecute con esos parámetros.
mysql -- force < script	Con la opción force se le indica a MySQL que continúe la ejecución de comandos hasta el final del archivo incluso si hay errores.
mysql < script less / mysql < script more	Utiliza los paginadores less o more para examinar los resultados de las consultas.
mysql < script > resultados-del-script	En lugar de mandar los resultados a pantalla (o utilizar paginador), con esta opción los resultados son enviados a un archivo de texto.
mysql -t < script	Este parámetro muestra la salida en modo interactivo.
mysql -vvv < script	Incluye en la salida los resultados de los comandos que se ejecutan (se muestra todo).
mysql>source /Users/dev/mysql_batch mysql>\. /Users/dev/mysql_batch	Con “source” o “\.” se pueden ejecutar scripts dentro de la consola de mysql.

mysql -u root -vvv < batch > resultado_consultas.txt

Administración de MySQL

Emmanuel Ayala Mexicano

Via Consola

```
c:\ruta_a_mysql> mysql -h equipo -u usuario -p
```

```
usuario$> ./mysql -h equipo -u usuario -p
```

Via software de administración

Además de utilizar la consola para administrar la BD contamos con distintas herramientas para la administración de las BD.

Una de ellas (muy común en las instalaciones XAMPP) es phpMyAdmin.

phpMyAdmin: http://www.phpmyadmin.net/home_page/index.php

Otras herramientas de administración de MySQL.

Navicat: <http://www.navicat.com/>

SQLyog: <http://www.webyog.com/en>


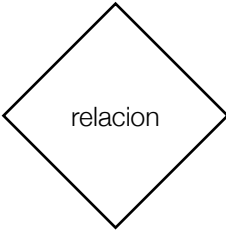
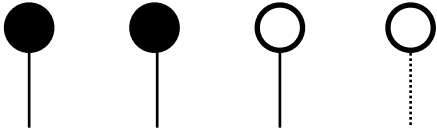
Workbench: <http://www.mysql.com/products/workbench/>

SequelPro: <http://www.sequelpro.com/>

Modelo Entidad Relación

Emmanuel Ayala Mexicano

Elementos Básicos

Elemento	Descripción	Representación
Entidad	Conjunto de elementos con existencia propia, se caracterizan por las mismas propiedades. Conceptos sobre los que necesitamos almacenar información y distinguibles de los demás objetos.	
Relación	Asociaciones en el mundo real entre una o más entidades	
Atributo	Características o cualidades propias de las entidades que queremos recoger dentro del diseño de datos. Estas propiedades no tienen existencia propia, sólo tienen sentido en el esquema de la BD ya que son parte de una entidad o de una relación	 AIP IA Obligatorio Opcional
Dominio	Atributo el cuál solo puede aceptar ciertos valores predefinidos. (Sexo, Estado Civil, Escolaridad)	En el diagrama se representa como un atributo y se indica que representa un dominio.

AIP: Atributo Identificador Primario
AI: Identificador Alternativo
Obligatorio.
Opcional.

Pasos para realizar un Diagrama E / R

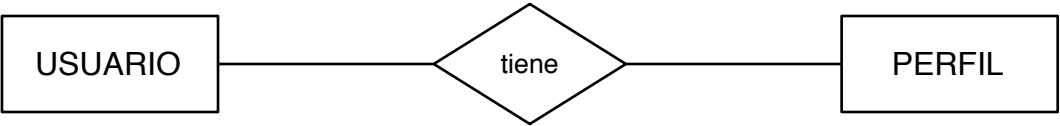
Paso	Descripción
1	Identificar y enumerar las posibles entidades teniendo en cuenta lo siguiente: en general, un tipo de entidad es un sustantivo dentro de una oración con una serie de propiedades o características.
2	Identificar y enumerar las posibles relaciones, teniendo en cuenta lo siguiente: en general, una relación viene reflejada por un verbo dentro de una oración que relaciona dos objetos.
3	Dibujar las relaciones (estudiando el tipo de correspondencia y las cardinalidades) y los tipos de entidad con los atributos correspondientes.

Diagrama Entidad Relación - Modelo Relacional

Emmanuel Ayala Mexicano

Transformación de Entidades

1:1



Cada tipo de entidad se transforma en una relación.

USUARIO
Todavía no transformamos atributos

PERFIL
Todavía no transformamos atributos

Transformación de Atributos de Entidades

Todo atributo de una entidad se transforma en un atributo(columna) de la relación en la cual se ha transformado la entidad (si el atributo estaba definido sobre un dominio, en el modelo relacional queda también definido sobre el mismo dominio).

Los atributos identificadores principales se transforman en la clave primaria de la relación. Los atributos identificadores alternativos se transforman en claves alternativas del modelo relacional.

Para los atributos multivaluados se crea una nueva relación formada con la clave primaria de la entidad y el atributo multivaluado siendo ambos clave primaria de la nueva relación.

Los atributos opcionales se transforman en una columna de la relación en la cual se ha transformado la entidad, admitiendo la posibilidad de valores nulos.

USUARIO			
id	nombre	email	password
PK	AK	AK	NOT NULL

SocialNetworking		
usuario_id	id	nombre
FK	PK	

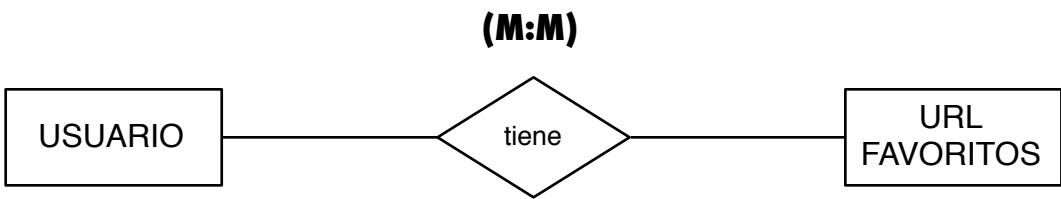
PERFIL					
id	usuario_id	foto	direccion	sexo	fecha_nacimiento
PK	FK			DOM.	NULL

- PK – Clave (Llave) Primaria
- AK – Clave (Llave) Alternativa (UNIQUE en definición de datos en MySQL)
- DOM – Dominio – Solo puede aceptar valores preestablecidos (M,F en este caso).
- SocialNetworking – Supongamos que queremos saber en que redes sociales el usuario tiene cuenta (atributo multivaluado) / Podemos incluso definir un dominio sobre el atributo multivaluado si SOLO permitimos ciertos valores (Facebook, Twitter) y, si el usuario tiene cuenta en Hi5, MySpace, Orkut, etc no la van a poder ingresar al sistema.
- FK – Clave (Llave) Foránea – Hace referencia a un registro en otra tabla.
- fn – Fecha de Nacimiento – En este caso es un atributo opcional (no es indispensable tener la fecha de nacimiento del usuario) por eso puede ser NULO.

Transformación de relaciones M:M

Este tipo de relación se transforma en una relación que tendrá como clave primaria la concatenación de los AIP de los tipos de entidad que asocia.

Los atributos que forman la clave primaria de esta relación son clave foránea respecto a cada una de las tablas donde este atributo es clave primaria.



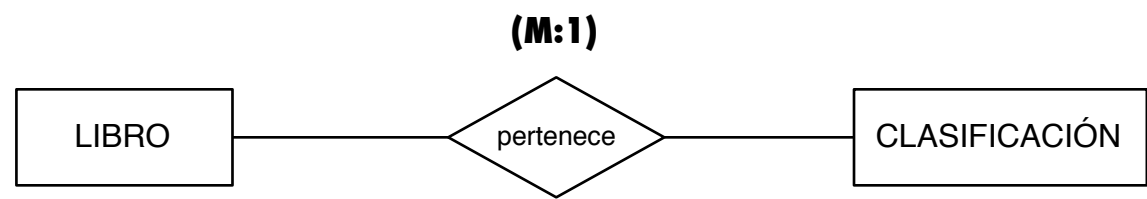
USUARIO			
id	nombre	email	password

usuario_favoritos	
usuario_id	favorito_id

FAVORITO		
id	url	nombre

Transformación de relaciones 1:M

Para este tipo de relación hay 2 posibles transformaciones:



a) Progagar el AIP del tipo de entidad que tiene cardinalidad máxima 1 a la que contiene M.

LIBRO				
id	clasificacion_id	nombre	autor	ISBN

CLASIFICACIÓN	
id	nombre

b) Transformarlo en una relación como si se tratara de una relación N:M

LIBRO			
id	nombre	autor	ISBN

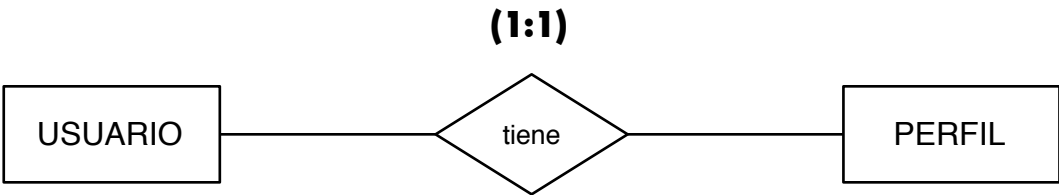
clasificado	
libro_id	clasificacion_id

CLASIFICACIÓN	
id	nombre

b) En este tipo de transformación sólo va a existir un elemento en clasificado por libro. Habría que validar vía llenguaje de programación.

Transformación de relaciones 1:1

Para este tipo de relación hay 3 posibles transformaciones:



a) Analizar cuál de las entidades es débil (o "pertenece" a la otra) y mezclar sus atributos junto con la entidad fuerte en el nombre de la entidad fuerte.

USUARIO							
id	nombre	contraseña	email	foto	dirección	sexo	fecha_nacimiento

b) Progagar el AIP del tipo de entidad fuerte al tipo de la entidad débil.

USUARIO			
id	nombre	contraseña	email

PERFIL					
id	usuario_id	foto	direccion	sexo	fecha_nacimiento

c) Crear una nueva relación, en la que la clave primaria estará formada por los AIP de los tipos de entidad que la interrelación asocia.

USUARIO			
id	nombre	contraseña	email

usuario_perfil	
usuario_id	perfil_id

PERFIL				
id	foto	direccion	sexo	fecha_nacimiento

Personalmente optaría por la opción “b”. La opción “c” necesita hacer 2 consultas para obtener el perfil. La opción “a” la es demasiado rígida para aceptar cambios (sobre todo en la parte de la codificación).