PHP y SQLite

Diseño físico

Francisco Arce
www.pacoarce.com

Diseño físico

- El diseño físico de la base de datos optimiza el rendimiento a la vez que asegura la integridad de los datos al evitar repeticiones innecesarias de datos.
- Durante el diseño físico, se transforman las entidades en tablas, las instancias en filas y los atributos en columnas.

Diseño físico

- Cómo convertir entidades en tablas físicas
- Qué atributos utilizar para las columnas de las tablas físicas
- Qué columnas de las tablas deben definirse como llaves primarias
- Qué índices deben definirse en las tablas
- Qué vistas deben definirse en las tablas
- Cómo desnormalizar las tablas (si es necesario)
- Cómo resolver relaciones de muchos a muchos

Diseño físico

- Diseño de base de datos con desnormalización
  - Las reglas de normalización no consideran el rendimiento.
  - En algunos casos, es necesario considerar la desnormalización para mejorar el rendimiento.

PHP y SQLite

Crear las tablas

Francisco Arce
www.pacoarce.com

Diseño físico

id	Titulo	pag	Año	idEditorial
1	Cien Años de soledad	600	1980	1
2	La guerra del fin del mundo	700	1989	2
3	La región más transparente	190	1995	3
4	El lenguaje de programación C	210	1999	4
5	El amor en los tiempos de cólera	544	1985	3

idAutor	idLibro
1	1
2	2
3	3
4	4
5	4
1	5

id	Autor
1	Gabriel García Márquez
2	Mario Vargas Llosa
3	Carlos Fuentes
4	Dennis Ritchie
5	Brian Kernighan

id	Editorial
1	Nueva Era
2	Alfaguara
3	Diana
4	Prentice Hall

PHP y SQLite

Armar las llaves

Francisco Arce
www.pacoarce.com

Diseño físico

id	Libros	tipo
1	id	Integer
2	nombre	text
3	anio	int
4	idEditorial	int

idAutor	Libros Autor	Tipo
1	id	Integer
2	idLibro	Integer
3	idAutor	Integer

id	Autor	Tipo
1	id	Integer
2	nombre	text
3	pais	text
4	fechaNac	text
5	nobel	boolean

id	Editoriales	Tipo
1	id	Integer
2	nombre	text
3	pais	text

Francisco Arce
www.pacoarce.com

Armar las llaves con Alias

Una manera tradicional de “armar las llaves” es por medio del alias.

A cada columna le corresponde el alias de su tabla para evitar “choque de nomenclaturas”.

Tenemos que relacionar la llave primaria con las llave foránea.

Opcionalmente podemos añadir un alias a la columna.

PHP y SQLite

Joins

Francisco Arce
www.pacoarce.com

Joins

Por medio de la cláusula JOIN podemos unir dos o más tablas por medio de campos comunes.

Los tipos de JOINS con los que contamos en SQLite son:

- CROSS JOIN
- INNER JOIN
- OUTER JOIN

(SQLite no cuenta con RIGHT y FULL, sólo con LEFT)

PHP y SQLite

CROSS JOIN

Francisco Arce
www.pacoarce.com

CROSS JOIN

- A CROSS JOIN empata cada fila de la primera tabla con cada fila de la segunda tabla.
- Si las tablas de entrada tienen “x” y columnas “y”, respectivamente , la tabla resultante tendrá “x * y” columnas, un producto cartesiano.
- Debido a que CROSS JOIN genera resultados muy grandes. Debe tener cuidado de utilizar sólo cuando sea apropiado.

CROSS JOIN

```
select l.*,e.*
from libros as l
cross join editoriales as e;
```

PHP y SQLite

INNER JOIN

Francisco Arce
www.pacoarce.com

Inner Join

- Un INNER JOIN da como resultado la combinación de valores de las columnas de dos tablas (Tabla1 y Tabla2) con base en la “expresión de unión”.
- La consulta compara cada fila de la “Tabla 1” con cada fila de la “Tabla 2” para encontrar todos los pares de filas que satisfacen la “expresión de unión”.

Inner Join

Sintaxis:

SELECT ... FROM tabla1 [INNER] JOIN tabla2 ON expresión_unión;

Inner Join

```
SELECT  l.id AS id,
        l.nombre AS titulo,
        l.anio AS anio,
        l.idEditorial AS idEditorial,
        e.nombre AS editorial
FROM libros as l
INNER JOIN editoriales as e
ON e.id = l.idEditorial;
```

PHP y SQLite

OUTER JOIN

Francisco Arce
www.pacoarce.com

OUTER JOIN

- OUTER JOIN es una extensión de INNER JOIN .
- Aunque el SQL estándar define tres tipos de combinaciones externas : LEFT, RIGHT y FULL, SQLite sólo admite el LEFT OUTER JOIN .
- OUTER JOIN tienen una condición que es idéntica a la LEFT JOIN, expresada mediante una palabra clave como ON, USING o NATURAL.
- Una vez que la primera operación JOIN es calculada, una operación OUTER JOIN concatenará los registros faltantes de ambas tablas, llenando los campos faltantes con valores nulos (NULL) en caso de que sea necesario.

LEFT OUTER JOIN

Sintaxis:

SELECT ... FROM tabla1 LEFT OUTER JOIN tabla2 ON expresión_condicional

Para evitar redundancia podemos utilizar la palabra reservada USING(), donde se especifica una lista de una o más columnas.

SELECT ... FROM tabla1 LEFT OUTER JOIN tabla2 USING (columna1 ,...)

LEFT OUTER JOIN

```
SELECT  l.id AS id,
        l.nombre AS titulo,
        l.anio AS anio,
        l.idEditorial AS idEditorial,
        e.nombre AS editorial
FROM libros as l
LEFT OUTER JOIN editoriales as e
ON e.id = l.idEditorial;
```

PHP y SQLite

UNION

Francisco Arce
www.pacoarce.com

UNION

La cláusula UNION SQLite se utiliza para combinar los resultados de dos o más instrucciones SELECT sin devolver las filas duplicadas .

Para utilizar la unión, cada SELECT debe tener el mismo número de columnas seleccionadas, el mismo número de expresiones de columna, el mismo tipo de datos y tenerlos en el mismo orden, pero no no es necesario que tengan la misma longitud

Francisco Arce
www.pacoarce.com

UNION

SELECT columna1 [, columna2]
FROM tabla1 [, tabla2]
[WHERE condición]

UNION

SELECT columna1 [, columna2]
FROM tabla1 [, tabla2]
[WHERE condición]

Francisco Arce
www.pacoarce.com

PHP y SQLite

Unir tablas relación muchos a muchos

Francisco Arce
www.pacoarce.com

Diseño físico

id	Libros	tipo
1	id	Integer
2	nombre	text
3	anio	int
4	idEditorial	int

idAutor	Libros Autor	Tipo
1	id	Integer
2	idLibro	Integer
3	idAutor	Integer

id	Autor	Tipo
1	id	Integer
2	nombre	text
3	pais	text
4	fechaNac	text
5	nobel	boolean

id	Editoriales	Tipo
1	id	Integer
2	nombre	text
3	pais	text

Francisco Arce
www.pacoarce.com

Unir tablas relación muchos a muchos

Por medio de dos sentencias JOIN podemos unir tablas con relación “muchos a muchos”, las cuales cuentan con una tabla intermedia.

Por lo general necesitaremos una función de agregado como “GROUP_CONCAT ()” porque vamos a recibir un columna con más de un resultados.

Unir tablas relación muchos a muchos

```
SELECT
        l.nombre as titulo,
        l.anio as anio,
        GROUP_CONCAT(a.nombre) as autores
FROM libros as l
        INNER JOIN librosAutores la ON l.id = la.idLibro
        INNER JOIN autores a ON la.idAutor = a.id
GROUP BY l.nombre;
```

Francisco Arce
www.pacoarce.com