

Revisión de errores frecuentes

Sitio: [Agencia de Aprendizaje a lo largo de la Vida](#)
Curso: Administración de Base de Datos 1° G
Libro: Revisión de errores frecuentes

Imprimido por: Sebastian Puche
Día: sábado, 21 de septiembre de 2024, 20:44

Tabla de contenidos

- 1. Introducción
 - 1.1. Análisis de una sentencia
 - 1.2. ¿Cuáles son los errores frecuentes?

Introducción



En este libro vamos a repasar los [errores](#) que se cometen en la etapa de análisis del modelo y en la [etapa de manipulación de los datos](#) que contiene.

Puede ocurrir que en el desempeño de tus actividades te encuentres cubriendo los dos roles, entonces, la mecánica que apliques en la construcción de las consultas, [procedures](#), [triggers](#), [vistas](#), (estos últimos conceptos los veremos muy pronto), estén reflejados en la [normalización](#).

No obstante, podemos cometer los siguientes errores:

- [Ignorar el propósito de los datos](#), es decir, la importancia que estos tienen en la regla de negocios que se debe llevar al modelo. Fijate como se analizó el modelo propuesto en el libro "[Revisión de modelos](#)".
- [Mala normalización](#), es decir, como sabemos que se aplica lógica y podemos tener más de un diseño para el mismo caso, nos quedamos con tablas que no llegan a cumplir la 3era forma normal.
- [Permitir una redundancia de datos](#) superior a la aceptada.
- [Ignorar la integridad referencial](#). Esta es una de las herramientas más valiosas que garantizan los motores de base de datos para mantener la calidad de los datos en su mejor forma. Si no se implementan restricciones o muy pocas restricciones desde la etapa de diseño, la integridad de los datos tendrá que depender totalmente de la lógica de negocios, lo que la hace susceptible a errores humanos.



En los próximos libros veremos un análisis de una sentencia y una lista de errores frecuentes.

Análisis de errores en una sentencia



En la práctica de la semana 6 está la siguiente consigna:

Usa la tabla “libro” y “libroautor” para mostrar el título del libro y la cantidad de autores que tiene cada uno.

El error que se comete en este tipo de consulta es **olvidar el group by**, la consulta la formulan de la siguiente manera:

```
MariaDB [biblioteca]> select titulo, count(*)
-> from libro l inner join libroautor la on l.codlibro = la.codlibro;
```

titulo	count(*)
El libro del PC	47

1 row in set (0.02 sec)

Annotations:

- El contador arroja un valor
- Es el primer titulo

¿Dónde está el error?

Si observas aparece un solo título, y no significa que hay un solo insert: esto sucede porque la segunda columna, **count** , da como resultado **un solo valor** y esto anula el resto de las filas.

```
MariaDB [biblioteca]> select titulo, count(*)
-> from libro l inner join libroautor la on l.codlibro = la.codlibro
-> group by l.codlibro;
```

titulo	count(*)
El libro del PC	1
Los secretos de Visual Basic 5	1
Hackers	3
NovellIntranetWare/NetWare5	1
Linux a Fondo	1
La Biblia de Linux	1
Sepa como usar su PC al maximo	1
S.O.S. PC	1
Sitios web bajo Linux	1
Manual de Diseño Web 2001	2
Blogs	1
CLIPPER 5.2	1
Aplicaciones ofimáticas	3
Tecnologías de Interconectividad de Redes	1
Sistemas Operativos Monopuesto	1
AutoCAD 2010	1
DYANE Version 4	1
Quark Xpress 8	1
Guia completa de Excell version 2002	1
Las inteligencias multiples en el aula	1
Administracion Organizacional	2
El trabajo en grupo y la diversidad en el aula	3
Arquitectura de equipos y sistemas informaticos	2
Analisis y Diseño Detallado de Aplicaciones Informaticas de Gestion	4
Etica informática y Etica e internet	1
Programacion en turbo pascal	1
Redes Locales	1
Data Centers hoy	1
Web 2.0:Nuevas Formas de Aprender y Participar	2
Delphi 5	1
El Libro de Excel 2010	2
Windows Me Milleniu Edition	1
Pascal esencial	1

33 rows in set (0.00 sec)

Annotation:

- Se ve el título con la cantidad de autores

¿Cuáles son los errores frecuentes?



A continuación compartimos una lista con los posible errores que te podrás encontrar.

- En [consultas con máximos y mínimos](#), buscar en la muestra de datos esos valores y colocarlos directamente en la consulta en lugar de usar las funciones. [No olvides que la base es accedida por varios usuarios y desconoces los dominios](#).
- En [los productos cartesianos](#), es decir cuando utilizamos más de una tabla, [olvidar que se debe igualar los atributos en común](#).
- Colocar un [igual, mayor o menor antes de una subconsulta](#) que da por resultado una lista de dominios.
- [Olvidar anteponer al atributo el nombre o alias de la tabla](#), cuando, el atributo aparece con el mismo nombre en más de una de ellas. Tal vez en el producto cartesiano se coloca, pero no si se usa en el [where](#), [en funciones](#), o en los grupos.
- No [respetar la integridad referencial](#) cuando se insertan datos en tablas relacionadas.
- No [respetar el tipo de dominios](#) declarados.
- Realizar los [insert en orden distinto](#) a lo indicado en la creación de la tabla.
- [Olvidar que es un modelo relacional](#) cuando creamos las sentencias.