

**INTEGRIDAD DE DATOS.**

El concepto más básico y sencillo para integridad o consistencia de datos es la propiedad de los datos de reflejar fielmente la realidad exterior que representa. Esto es, si un dato dice que Almacenes la Negrita tiene 20'000,000.00 en su cuenta de cheques sea exacto y cierto, que solamente exista una sola vez este cliente en el sistema y, por supuesto, que la información pueda ser recuperada con exactitud.

Esto se antoja sencillo, pero en realidad es mucho más complejo de lo que se ve, primero tenemos que ver que los datos no hayan sido almacenado mal a consecuencia de un apagón sin UPS mientras se lleva a cabo la actualización física de datos al disco, que acciones tomar cuando la llave primaria de un dato está repetida, o peor aún, cuando una transacción es abortada y sabemos que cada actualización hecha dentro de ella debe ser regresada.

Como podemos ver el concepto de integridad es demasiado amplio, es por ello que existen diferentes niveles de integridad.

**INTEGRIDAD FISICA.**

Consiste en que los datos estén correctamente almacenados en el disco, que no hayan quedado a medio actualizar por culpa de un apagón al escribir datos actualizados al disco, o inclusive, el que se encime información entre dos chunks de información. Para garantizar este tipo de integridad a nivel de página usamos las marcas de tiempo o timestamps definidas en el capítulo de la estructura de disco.

	Offset		Chnk cks nslt flg frpt frcnt	
	next_pg		prev_pg	
=				=
			. . .  slt2 slt1 timestamp	

Cuando se lee la página se obtiene un checksum del timestamp ubicado al final de la página y se compara con el almacenado en el header de la página, de esta manera podemos garantizar que la información actualizada o creada no quedo a medias. Para ello antes de escribir o actualizar cualquier dato a la página se actualiza el checksum del timestamp (en la cabecera) se realiza la actualización y hasta que toda la información ha sido actualizada no se actualiza el timestamp al final de la página, ¿Qué quiere decir esto? Que cuando estemos en el "salto de la muerte", es decir escribiendo a disco, y algo muy malo sucede, dejando la escritura de la página "a medias", el RDBMS tiene una forma de detectar este tipo de corrupción si el checksum del timestamp no coincide con aquel que aparece en el header, evitando así problemas de integridad de este tipo, evitando encontrar nombres como "Pedroisco" en lugar de "Pedro" o "Francisco".

**INTEGRIDAD RELACIONAL.**

Aplica a la necesidad de asegurarse de que los datos almacenados cumplan con el modelo E-R planteado para representar la realidad, que quiere decir, que los campos definidos como llaves primarias sean únicos y no nulos, que las llaves foráneas existan como primarias en otra tabla, que los valores definidos como no duplicados y no nulos realmente lo sean y que los datos definidos realmente contengan el tipo de dato definido. Para ello nos valemos de "condiciones" o CONSTRAINTS, un constraint es una condición obligatoria definida a nivel de la base de datos y que es validada automáticamente por la base de datos. Generalmente se definen al crear la tabla, por ejemplo:

```
CREATE TABLE customer(  
    customer_num SERIAL,  
    fname CHAR(20),  
    PRIMARY KEY (customer_num)  
    CONSTRAINT pk_cnum  
);  
  
CREATE TABLE orders(  
    order_num SERIAL,  
    customer_num INTEGER,  
    FOREIGN KEY (customer_num)  
    REFERENCES customer  
    CONSTRAINT fk_cnum  
);
```

Los tipos de integridad relacional son:

- Integridad Referencial (Referential Integrity): ¿Están establecidas y garantizadas las relaciones entre tablas? FOREIGN KEY, PRIMARY KEY CONSTRAINTS.
- Integridad de Entidad (Entity Integrity): ¿Tiene cada registro en la tabla un identificador único? PRIMARY KEY CONSTRAINT.
- Integridad Semántica (Semantic Integrity): ¿Reflejan adecuadamente los datos en las columnas respectivas el tipo de información para el cual la columna fue diseñada? CHECK CONSTRAINT.

**INTEGRIDAD TRANSACCIONAL.**

Se refiere a la propiedad de una transacción de realizar todas las operaciones que la componen o ninguna. En caso de inconsistencias, quizá sea la más compleja y difícil de detectar de todas ya que las operaciones que componen a las transacciones pueden terminar exitosamente y sin embargo el RDBMS puede reflejar una realidad completamente falsa ¿Recuerdan el ejemplo de la transacción bancaria?

Ahora bien, para asegurar la integridad transaccional nos valemos de tres cosas:

- Los logical log files y
- Physical log file y
- Checkpoints.

El physical log es un espacio en disco donde guardamos la "imagen de antes" de una página que está siendo actualizada o borrada en memoria, la idea es que si por alguna falla el motor llegara a caerse con datos



actualizados en memoria, podamos recuperar al menos su contenido en el último punto de consistencia conocida (checkpoint).

Un checkpoint se define como un punto en el tiempo donde la consistencia física y transaccional están reconocidas. Estos checkpoints son periódicos y nos dan puntos de referencia donde la integridad transaccional es conocida.

Las acciones que tienen lugar durante un checkpoint son:

- Se detiene toda actividad de escritura o actualización en la instancia.
- Todas las páginas modificadas en memoria son bajadas a disco.
- Se vacía el physical log, ya que todas las páginas modificadas en memoria ya han sido escritas a disco.
- Se inserta un registro de checkpoint en el logical log file, mismo que contiene información sobre la fecha y hora del checkpoint e información sobre una y cada una de las transacciones que permanezcan abiertas al momento del checkpoint.
- Los buffers de los logical logs se vacían a disco.

Esto nos da un punto conocido de consistencia. Ya que sabemos que la información presente en los buffers (volátil en caso de falla) está reflejada en el disco y que tenemos toda la información relacionada con transacciones para dejar la base de datos perfectamente consistente.

Conceptos vistos:

Integridad.

Integridad Física.

Integridad Relacional.

Integridad referencial.

Integridad semántica.

Integridad de entidad.

Integridad Transaccional.

Physical Log File.

Checkpoint.