



E.T.S.
INGENIERÍA
INFORMÁTICA

Gestión de la Información



UNIVERSIDAD
DE MÁLAGA



LENGUAJES Y
CIENCIAS DE LA
COMPUTACIÓN
UNIVERSIDAD DE MÁLAGA

Tema 3

Diseño de transacciones en bases de datos

José Luis Pastrana Brincones (pastrana@lcc.uma.es)



OCW UMA

Delgado Peña, J.; Godoy Castillo, R. (2009) Elaboración de Cartografía Física Elemental. OCW-Universidad de Málaga. <http://ocw.uma.es>
Bajo licencia Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Spain



Diseño de Transacciones

Las transacciones en una BD se utilizan para representar cómo evoluciona el sistema.

- ▶ *Operaciones de acceso a los datos:*
 - inserción de ocurrencias de entidad o relación.
 - borrado de ocurrencias de entidad o relación.
 - modificación del valor de los atributos de ocurrencias de entidad o relación.

En su nivel más bajo consiste en definir cómo se insertan, borran y modifican ocurrencias de todos los objetos del sistema

Diseño de Transacciones

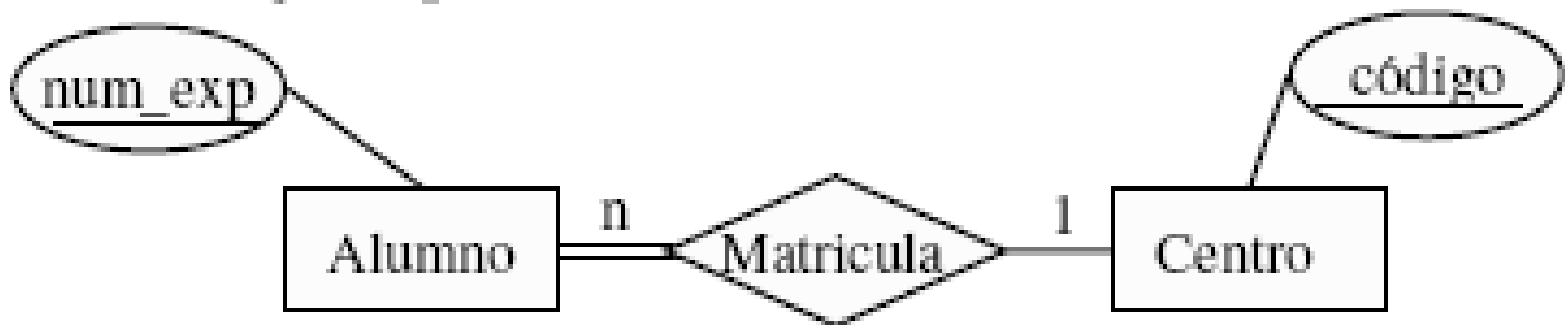
▶ Una transacción es una secuencia de operaciones de acceso a los datos que constituye una unidad lógica de ejecución.

- ▶ Atomicidad, (*Atomicity*)
- ▶ Consistencia, (*Consistency*)
- ▶ Aislamiento, (*Isolation*)
- ▶ Persistencia (*Durability*)



Diseño de Transacciones

- ▶ ¿Por qué son necesarias las transacciones?



Transacción Nuevo_alumno

- Insertar en *Alumno*
- Insertar en *Matricula*

(Previamente debe existir el centro en el que se matricula)

Diseño de Transacciones

Metodología de diseño de transacciones:

- ▶ **Objetivo:** Obtener un conjunto de *transacciones mínimas a partir del* diagrama ER.
- ▶ Este conjunto incluirá, para cada entidad y para cada relación del diagrama, una transacción mínima de **inserción, una de borrado y varias de modificación.**
- ▶ **Transacciones mínimas:** única forma permitida de modificar la información almacenada. Podrán ser utilizadas en transacciones más complejas.

Diseño de Transacciones

Transacción mínima:

- ▶ Incluye todas las operaciones que son necesarias para que la modificación del sistema de información sea **válida**.
- ▶ Constará siempre de **una operación básica (de inserción, borrado o modificación)** sobre el objeto (entidad o relación) para el que se está definiendo la transacción.
- ▶ Puede incorporar operaciones sobre otras entidades o relaciones con las que esté vinculado.

Diseño de Transacciones



▶ Transacción Nuevo_alumno

- Insertar en *Alumno*
- Insertar en *Estudia*

Transacción
Mínima

Diseño de Transacciones



Transacción Nuevo_alumno

- Insertar en *Alumno*
- Insertar en *Estudia*
- Mientras se quiera seguir
 - Insertar en *Estudia*
- Fin_mientras

Transacción
No Mínima

Diseño de Transacciones

Análisis de transacciones:

- ▶ Se decidirá, para cada entidad o relación del DER, qué operaciones sobre otros objetos serán necesarias para conseguir añadir, borrar o modificar una ocurrencia de esa entidad o relación.
- ▶ La determinación de dichas operaciones **se deduce lógicamente del DER**, al considerar las restricciones representadas en el mismo.
- ▶ En diagramas que no tienen restricciones de integridad añadidas, se concreta de la forma siguiente:

Diseño de Transacciones

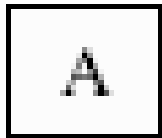
Inserción

- ▶ ***Entidades:*** Si una entidad tiene restricción de existencia en alguna relación R , hay que insertar en R . En casos particulares hay que insertar en varias entidades a la vez.
- ▶ ***Relaciones:*** Si la relación R en la que se está insertando participa como objeto agregado con restricción de existencia en otra relación S , hay que insertar en S .

Diseño de Transacciones

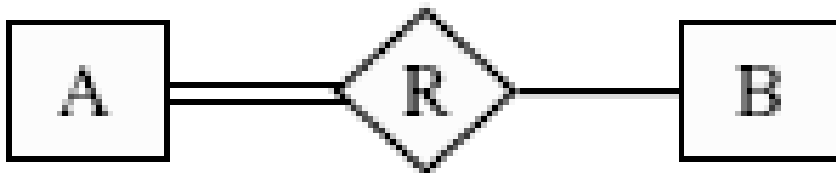
▶ Transacciones de inserción

Entidades:



Transacción Nuevo_A

* Insertar en A



Transacción Nuevo_A

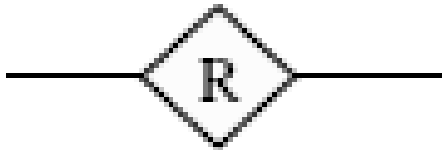
* Insertar en A

* Insertar en R

Diseño de Transacciones

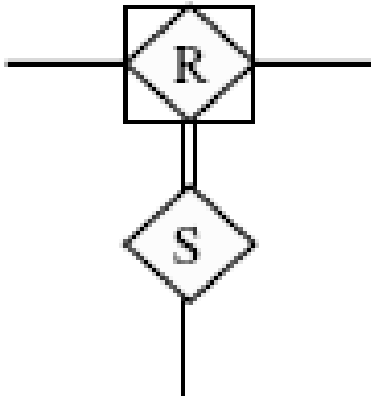
▶ Transacciones de inserción

Relaciones:



Transacción Nuevo_R

* Insertar en R



Transacción Nuevo_R

* Insertar en R

* Insertar en S

Diseño de Transacciones

Borrado: Se considerarán dos tipos: restrictivo y en cascada

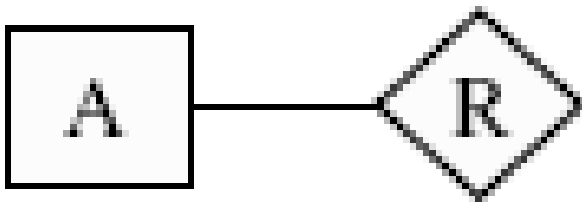
Entidades:

- ▶ **Restrictivo respecto a una relación R :** la transacción sólo incluye la operación de borrado de la entidad.
- ▶ **En cascada respecto a una relación R :** la transacción incluye, además de la operación de borrado de la entidad, el borrado de las ocurrencias de R en las que interviene la entidad. Si la entidad tiene restricción de existencia respecto a R , el borrado siempre tendrá que ser de este tipo.

Diseño de Transacciones

▶ Transacciones de borrado

Entidades:



Restrictivo

Transacción Borrar_A

* Borrar en A

Cascada

Transacción Borrar_A

* Borrar en A

* Borrar en R

Diseño de Transacciones

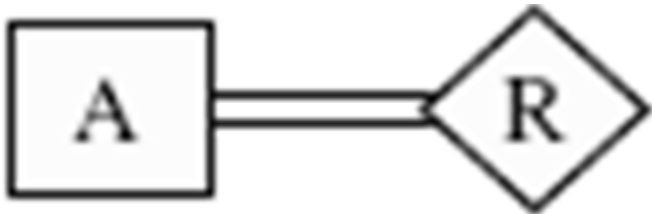
Relaciones:

- ▶ *Restringido respecto a una entidad o relación: no hay que añadir ninguna operación.*
- ▶ *En cascada: si alguna de las entidades que participan en la relación tienen restricción de existencia en ella, hay que incluir también el borrado de dicha entidad para los casos en los que la ocurrencia de relación borrada sea la última en la que participaba la entidad.*
- ▶ Además, si la relación participa como objeto agregado en otra relación, sea S, en la transacción se incluye el borrado de S para las ocurrencias en las que interviene el agregado.

Diseño de Transacciones

▶ Transacciones de borrado

Relaciones:



Restrictivo

Transacción Borrar_R

* Borrar en R

Cascada

Transacción Borrar_R

* Borrar en R

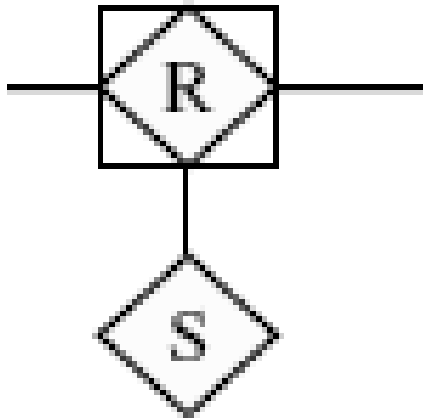
* Borrar en A

(si era el último en R)

Diseño de Transacciones

▶ Transacciones de borrado

Relaciones:



Restrictivo

Transacción Borrar_R

* Borrar en R

Cascada

Transacción Borrar_R

* Borrar en R

* Borrar en S

Diseño de Transacciones

- ▶ En el borrado en cascada o en la inserción de entidades con restricciones de existencia, las operaciones añadidas a la transacción pueden requerir a su vez una propagación, que también hay que incluir en la transacción.

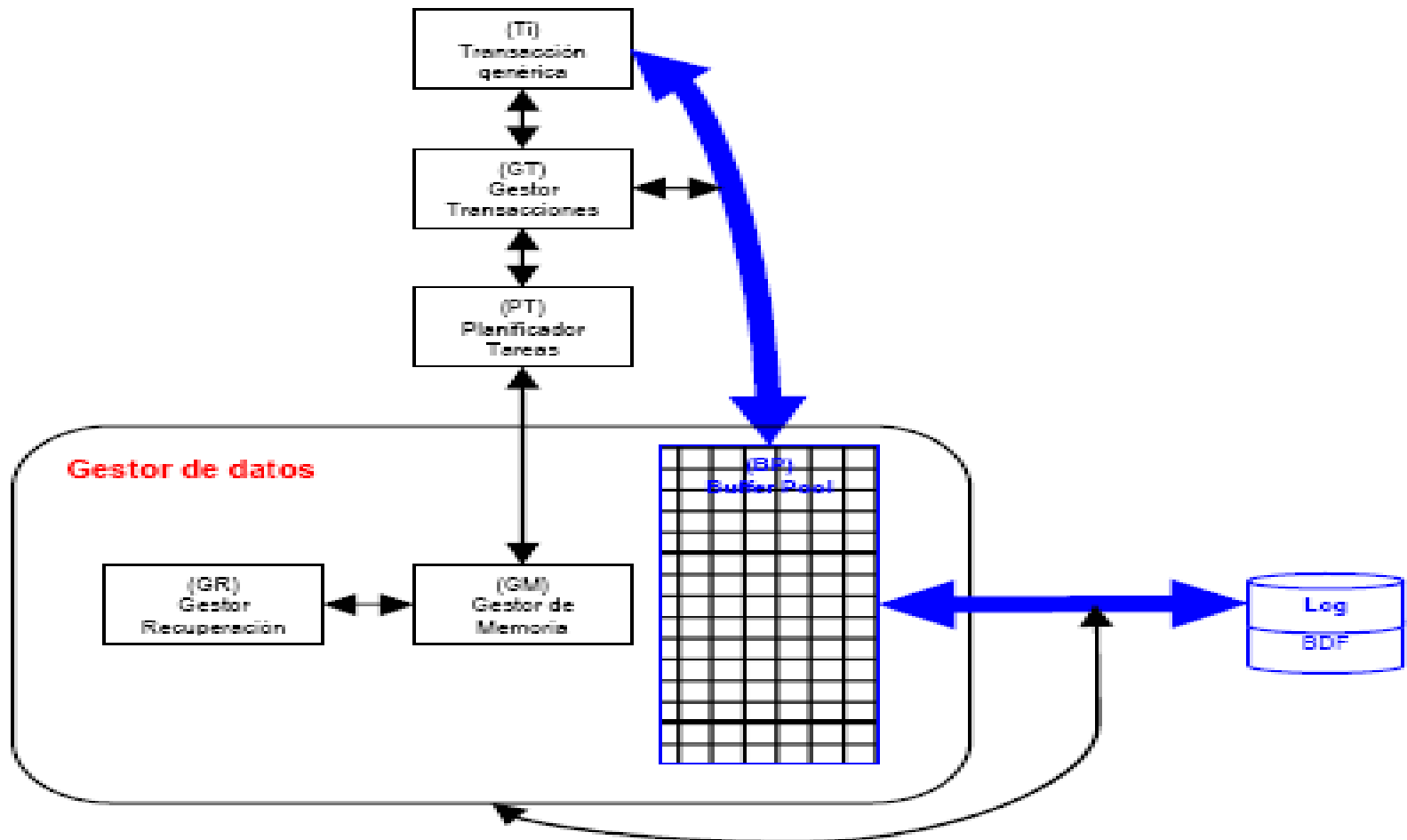
Modificación:

- ▶ Puede haber muchas transacciones de modificación diferentes.
- ▶ Deben diseñarse las transacciones que posibiliten modificar cualquier atributo de cualquier objeto que sea necesario

Arquitectura de gestión de transacciones

- ▶ El SGBD actúa como interfaz entre los programas(transacciones) y el sistema operativo (métodos de acceso a ficheros) para realizar la recuperación física de las páginas o bloques físicos de datos (unidades mínimas de transferencia entre la memoria externa y la memoria principal).
- ▶ En el esquema siguiente se presentan los distintos subsistemas en que puede descomponerse la funcionalidad del SGBD para dar servicios a estas transacciones y, simultáneamente, garantizar la consistencia de representación de estados en la base de datos física.

Arquitectura de gestión de transacciones



Arquitectura de gestión de transacciones

Se diferencian los siguientes subsistemas:

- ▶ (GT) Gestor de transacciones.
- ▶ (PT) Planificador de tareas o gestor de transacciones concurrentes.
- ▶ (GM) Gestor de memoria caché.
- ▶ (GR) Gestor de recuperación de la base de datos.

Arquitectura de gestión de transacciones

(GT) Gestor de transacciones

- ▶ Recibe las peticiones de las transacciones (lecturas, inserciones, actualizaciones, eliminaciones, etc.)
- ▶ Es capaz de servir los datos a dichas transacciones como retorno a su petición así como los códigos de estado (códigos de retorno) que informan del nivel de finalización de dicha petición.

Arquitectura de gestión de transacciones

(PT) Planificador de tareas

- ▶ Gestiona el ámbito de concurrencia de dichas transacciones .
- ▶ En función de las políticas y algoritmos implementados para resolver los conflictos de concurrencia que pueden originarse
 - Bloqueo
 - Métodos de ordenación de transacciones
 - Gestión optimista de transacciones
 - etc.
- ▶ El Planificador (Scheduler) puede decidir, y en qué orden, si acepta o rechaza determinadas peticiones que recibe del gestor de transacciones, implementando primitivas específicas de cancelación (ABORT) o reinicio (RESTART) de transacciones.

Arquitectura de gestión de transacciones

(GM) Gestor de memoria caché.

- ▶ Trata de optimizar el rendimiento global de transacciones por segundo) del sistema.
- ▶ Para ello la política se basa en acaparar la memoria rápida disponible (buffer pool(BP) del SGBD).
- ▶ En el (BP) se alojan los gránulos o páginas intercambiados con el almacenamiento externo.
- ▶ El (GM) procura retener estas páginas una vez cargadas en el (BP)

Arquitectura de gestión de transacciones

- ▶ Las transacciones que generan modificaciones de estas páginas generan en el (BP) una nueva versión de dicha página denominada página sucia (dirty page).
- ▶ Por otro lado, dado que el (BP) se aloja en memoria volátil, es imposible mantener indefinidamente una página sucia en el mismo, teniendo lugar, antes o después, una descarga de dichas páginas a la memoria externa o base de datos física (BDF).
- ▶ El (GM) tiene que gestionar todas estas operaciones buscando el mejor rendimiento global del sistema.

Arquitectura de gestión de transacciones

(GR) Gestor de recuperación de la base de datos.

- ▶ Tiene que implementar los procedimientos adecuados para garantizar la representación de estados persistentes (almacenados en la BDF) consistentes.
- ▶ (GR) requiere de una estructura adicional de datos en memoria externa para garantizar sus actuaciones:
 - el diario (Log) de transacciones,
 - manteniendo en el mismo un registro de la actividad de todas las transacciones
 - y de los procesos de descarga masiva de páginas del (BP) a la (BDF).

Arquitectura de gestión de transacciones

Los registros del Log están representados por:

- ▶ Puntos de sincronismo de transacciones: Begin(T), Commit(T), Abort(T). Marcan el inicio y la terminación, normal o anormal, de una transacción.
- ▶ Puntos de control del sistema(CheckPoint): Marcan la actividad de descarga masiva de páginas sucias del (BP) a la (BDF).
- ▶ Preimágenes de páginas: Es el estado de una página antes de la actualización. Las preimágenes permitirán devolver, en caso de fallo, el estado de la base de datos al estado consistente inicial o anterior al inicio de la transacción (Backward Recovery).

Arquitectura de gestión de transacciones

- ▶ Postimágenes de páginas: Es el estado de una página (página sucia en principio) después de la actualización (inserción, modificación o eliminación) generada por una transacción.
- ▶ Las postimágenes permitirán reconstruir, en caso de fallo, la actividad de una transacción hacia delante (Forward recovery) hasta el punto de confirmación de la misma, si este se produjo.
- ▶ Todos estos registros se almacenan en el Log cronológicamente, asociando a los mismos la transacción correspondiente.