



Nombre de alumno:

Teresa Méndez Pérez

Nombre del profesor:

Andrés Alejandro Reyes Molina

Nombre del trabajo:

Súper nota

Materia:

Base de datos II

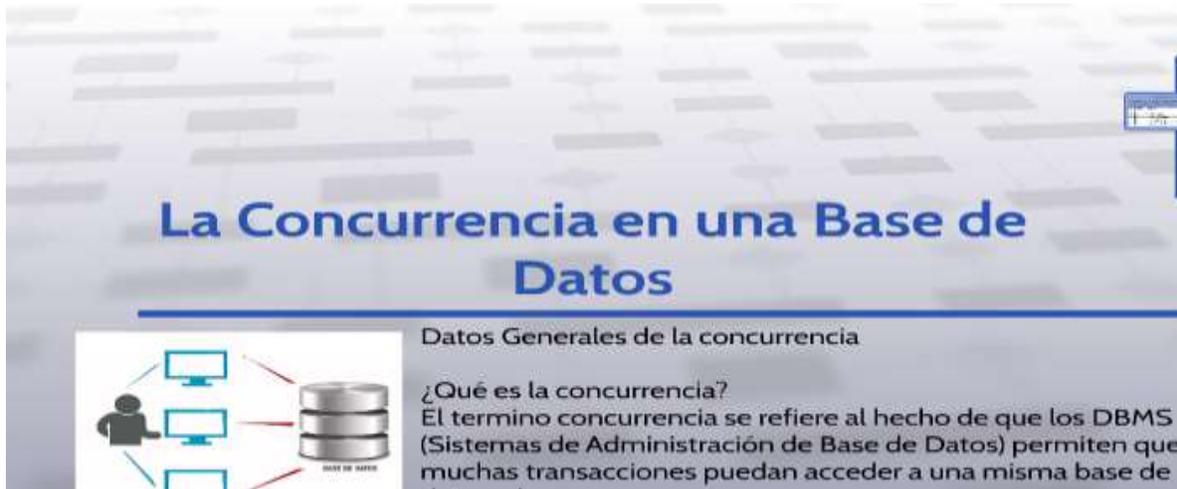
Grado: 8 cuatrimestre

Comitán de Domínguez Chiapas a 28 de enero de 2022.

CONCURRENCIA

DEFINICION

El termino concurrencia se refiere a la capacidad de los Sistemas de Administración de Base de Datos, de permitir que múltiples procesos sean ejecutados al mismo tiempo, y que también puedan interactuar entre sí.



La Concurrencia en una Base de Datos

Datos Generales de la concurrencia

¿Qué es la concurrencia?
El termino concurrencia se refiere al hecho de que los DBMS (Sistemas de Administración de Base de Datos) permiten que muchas transacciones puedan acceder a una misma base de datos.

The slide features a background of a grid with faint database icons. The title 'La Concurrencia en una Base de Datos' is centered in a large blue font. Below the title, there is a horizontal line. To the left of the text is a diagram showing a person icon connected to three computer monitors, which are in turn connected to a database cylinder icon. To the right of the diagram, the text 'Datos Generales de la concurrencia' is followed by a question '¿Qué es la concurrencia?' and a paragraph explaining that DBMS allow multiple transactions to access the same database.

PROBLEMAS QUE SE PRESENTAN

Protocolo basado en técnicas de bloqueo

Bloqueos binarios

Bloque de lectura/escritura

Problema de bloqueos en dos fases: interbloqueo y espera indefinida

Exclusión mutua

No apropiación

Espera circular

Bloqueo mutuo o dead lock

Serial schedules

SERIABILIDAD

> Se encarga de la administración de las transacciones en la Base de Datos.

> Para eso recibe las instrucciones que los programas pretenden ejecutar y se toma la libertad de reordenarlas, sin cambiar nunca el orden relativo de los Read y Write.

> Puede llegar a agregar instrucciones (nunca R/W) por su cuenta.

> Puede llegar a demorar la ejecución de las instrucciones.

* Si las transacciones se ejecutaran siempre en forma serial, entonces no habría concurrencia pero los datos siempre serían correctos.

* Si las historias son entrelazadas, podría suceder que queden datos erróneos que no se puedan corregir o que si una transacción aborta otra también tenga que abortar.

* EJ:

- T1: $r1(x), w1(x), a1$; T2: $r2(x), r2(y), w2(x), c2$ (o $a2$).

- H1: $r1(x), w1(x), r2(x), r2(y), w2(x), c2, a1$ (Historia no recuperable)

- H2: $r1(x), w1(x), r2(x), r2(y), w2(x), a1, a2$ (Historia con abortos en cascada)

Es aquella que es equivalente a una historia serial con las mismas transacciones. Hay varias nociones de equivalencia (o serializabilidad):

Intuitiva. Dos historias son equivalentes si dejan la base en el mismo estado.

Problema: difícil de garantizar. Puede suceder de casualidad.

Por conflicto. Si tienen todas las operaciones en conflicto en el mismo orden.

Por vistas. Si cada T_i lee de las mismas T_j, \dots, T_n , en H y H' .

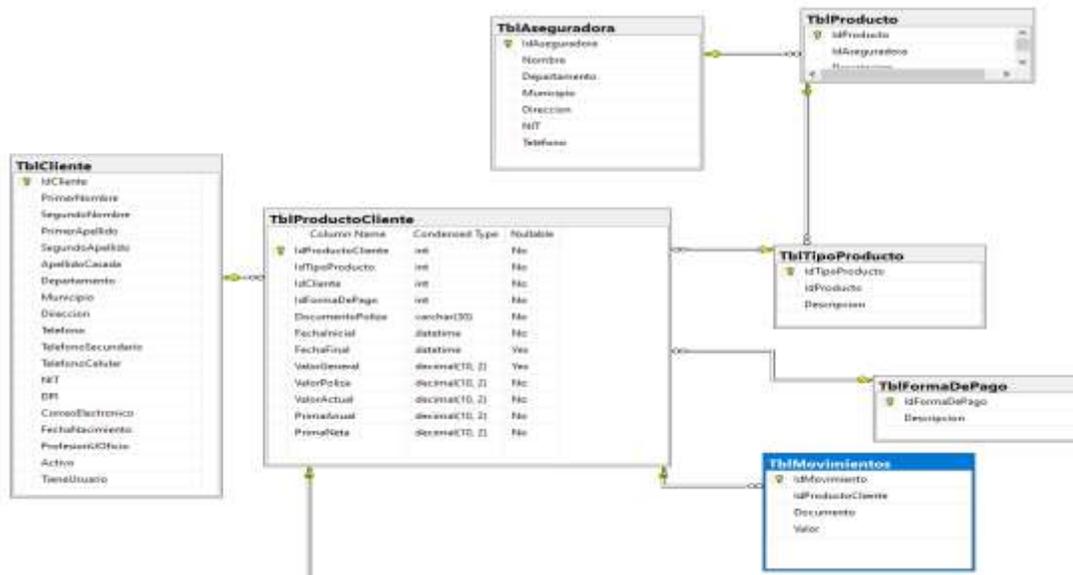
Schedules seriales y no seriales (1)

- **Schedule serial limita la concurrencia:**
 - Si T espera una operación I/O no se puede pasar la ejecución a otra transacción.
 - Si T es muy larga, el resto de las transacciones deben esperar a que T termine.
 - En general, los *schedules* seriales son considerados inaceptables en la práctica.
- Algunos *schedules* no seriales dan un resultado correcto: determinar cuales dan siempre un resultado correcto.
- Un *schedule* S de n transacciones es serializable si es equivalente a algún *schedule* serial de las mismas n transacciones (dos grupos disjuntos).
 - Hay $n!$ posibles *schedules* seriales.
 - Muchos más *schedules* no seriales.

MECANISMOS DE SEGUROS



TIPOS DE SEGUROS



PROTOCOLOS

Protocolos basados en grafos

Protocolo de árbol

Protocolo de ordenación por marcas temporales

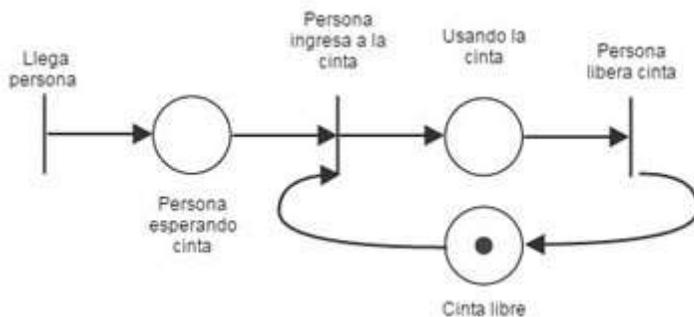
Protocolos optimistas



DEAD LOCK

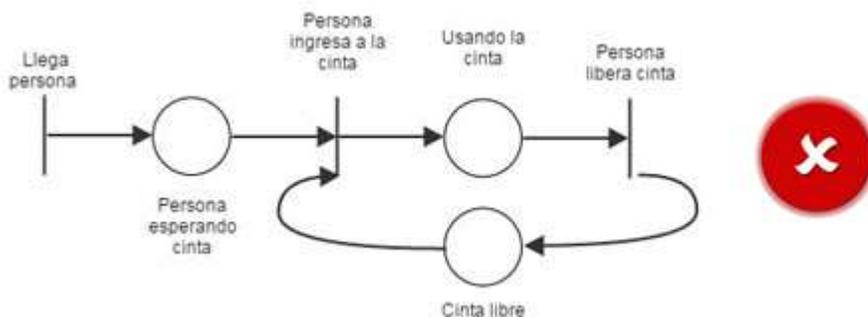
En sistemas operativos, el deadlock, es el bloqueo permanente de un conjunto de procesos en un sistema concurrente que compiten por recursos del sistema o bien se comunican entre ellos.

Al modelar una red, siempre se debe chequear que su ejecución no esté bloqueada por un mal diseño. Por ejemplo, consideremos la siguiente imagen:



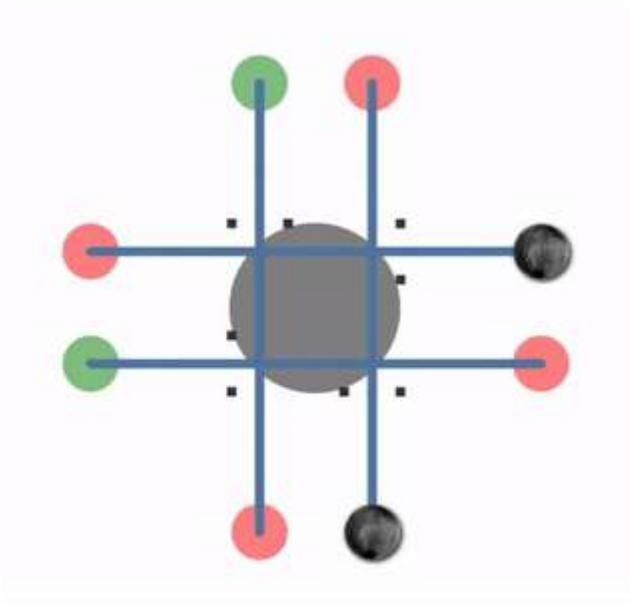
Supongamos que no ponemos la marcación inicial en el sitio "Cinta libre". En ese caso, la transición "Persona ingresa a la cinta" no estará nunca habilitada, ya que es necesario que reciba un token por cada flecha de entrada.

En este caso, la red será INCORRECTA, ya que ninguna persona podrá usar la cinta.



TECNICAS PARA PREVENIRLO

El interbloqueo se produce si cada proceso retiene un recurso y solicita el otro. Puede parecer que es un error de programación en lugar de un error del diseño del sistema operativo. Sin embargo, se ha visto que el diseño de un programa concurrente entraña gran dificultad. Se producen interbloqueos como éste y la causa está frecuentemente en la compleja lógica del programa, lo que hace más difícil su detección. Una posible estrategia para resolver estos interbloqueos es imponer restricciones en el diseño del sistema sobre el orden en el que se solicitan los recursos



TECNICAS PARA DESHACERLO

Las soluciones del interbloqueo se presentan a través de tres estrategias: prevención, detección y predicción del mismo. ... Predicción: se realiza mediante procesos dinámicos que establecen si la petición de un recurso podrá conducir a interbloqueo.

- » Este tema trata de ver como es posible evitar los interbloqueos para no llegar a un "punto de no retorno" de acuerdo al orden de peticiones realizadas por diversos procesos.

Proceso P₁

Solicita(C)
Solicita(I)
Uso de rec.
Libera(I)
Libera(C)

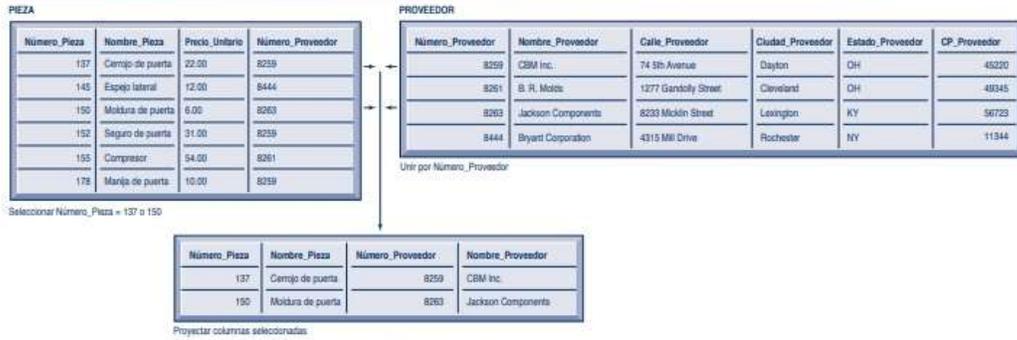
Proceso P₂

Solicita(I)
Solicita(C)
Uso de rec.
Libera(C)
Libera(I)

Predicción de interbloqueo

OPERACIONES DE SEGUROS EMPLEANDO DLM CON DBMS COMERCIAL

FIGURA 6-5 LAS TRES OPERACIONES BÁSICAS DE UN DBMS RELACIONAL



Las operaciones seleccionar, unir y proyectar permiten combinar datos de dos tablas distintas y mostrar sólo los atributos seleccionados.