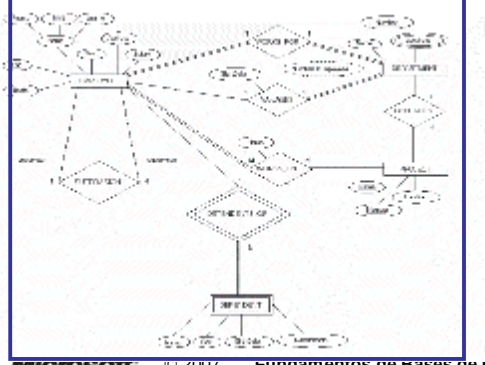
Fundamentos de Bases de Datos

Diseño de Bases de Datos Relacionales:
Mapeo ER al Modelo Relacional

Input:ERD Output: Esquema Relacional

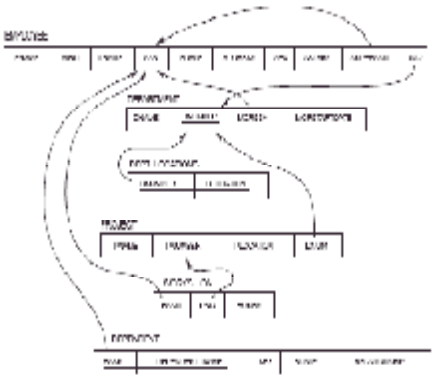
Entidades
Atributos Simples, Compuestos
Atributos Derivados, multivalor
Relaciones 1:1, 1:N, M:N,
Relaciones recursivas
Entidades débiles
Participación total
Participación opcional



Algoritmo De Mapeo

Tablas,
Llaves primarias PK
Llaves foráneas FK

Employee(Fname, ..
PK(SSN)
FK(SUPERSSN) refernces Employee(SSN)
Department(Dname, Dnumber, mgrSSN,...)
PK(Dnumber)
FK(mgrSSN) references Employee(SSN)
....
....



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

MAPEO ER A RELACIONES

- El enfoque ER representa un diseño conceptual que representa una situación real.
- Aquí consideramos el mapeo de entidades y relaciones de un diagrama ER a las relaciones (tablas) del modelo de datos relacional
- Aunque el mapeo es flexible de alguna forma, se definen varias heurísticas o reglas mapeo.

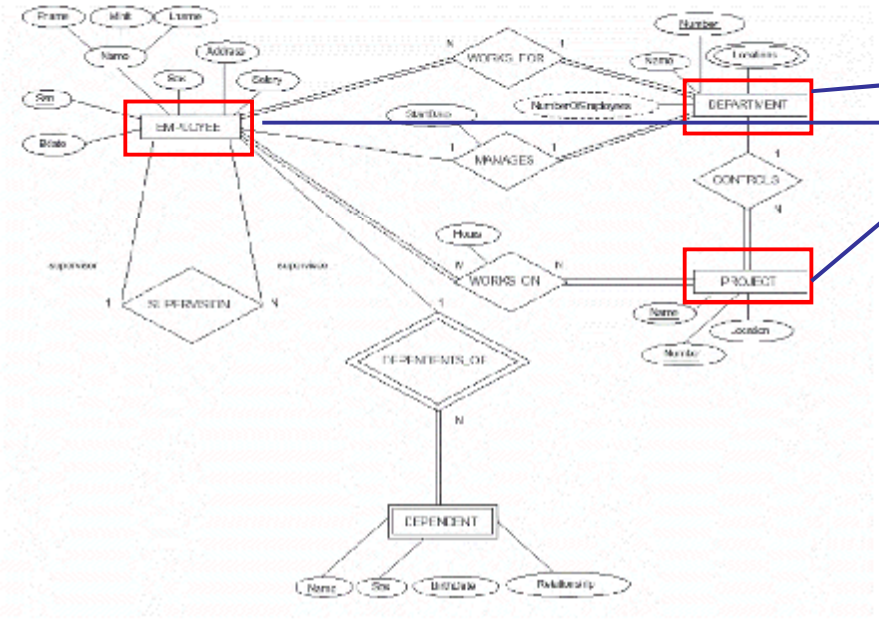
Reglas de Mapeo

- R1 Mapeo de Entidades
- R2 Mapeo de Atributos Simples y Compuestos
- R3 Mapeo de relaciones 1:1
- R4 Mapeo de relaciones 1:N
- R5 Mapeo de relaciones M:N
- R6 Mapeo de atributos multivalor
- R7 Mapeo de Entidades débiles
- R8 Mapeo de relaciones recursivas
- R9 Mapeo de atributos derivados

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

R1 Mapeo de entidades

- Por cada entidad no débil, crear una tabla

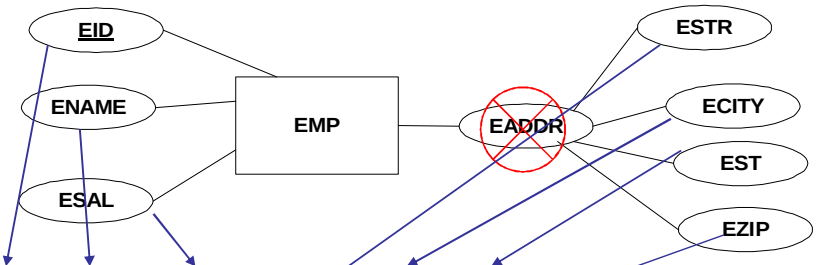


Tablas:

Department()
Employee()
Project()

R2. Atributos Simples y Compuestos

- **Cada Atributo Simple y cada nodo hoja de los atributos compuestos es una columna en R.**
- Atributos de la tabla
  - Los atributos simples de la entidad y
  - los componentes simples de atributos compuestos.
- Llave Primaria (Primary Key)
  - Llave primaria de la entidad.



emp(EID,ENAME,ESAL,ESTR,ECITY,EST,EZIP)

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

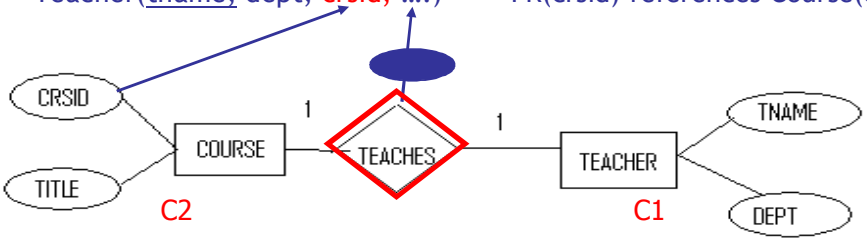
R3. CARDINALIDAD 1:1 (CARDINALITY RATIO)

Relación de 1:1 **entre clases C1 y C2 , seleccionar una de las clases (C1) e incluir la llave de C2 en C1 como una llave foránea. Incluir los atributos simples de la asociación entre ambas clases en C1**

Agregar a una de las tablas seleccionada (C1):

1. Atributos correspondientes a la llave primaria de la otra entidad involucrada en la relación (relationship). En este caso C2 o course
2. Si existen Atributos de la relación (relationship) C1:C2 se ponen en la tabla correspondiente a C1, TEACHER en este ejemplo.

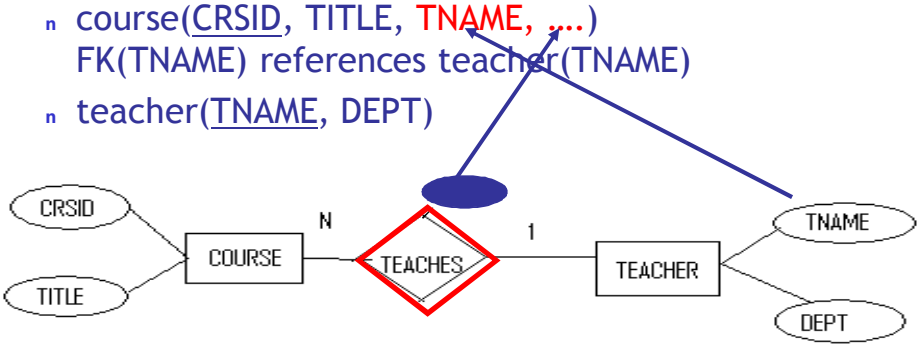
course(crsid, title)
Teacher(tname, dept, crsid, ….) FK(crsid) references Course(crsid)



Nota: Si es posible, seleccionar a la entidad que tiene participación total (total participation) en la relación (relationship)

R4. CARDINALIDAD 1:N

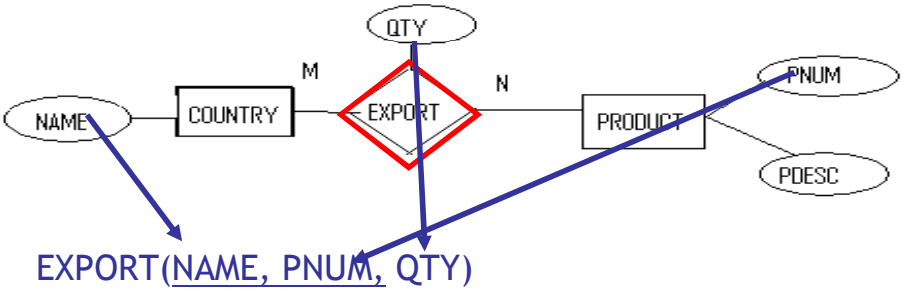
- Agregar a la relación o tabla (relation) de la entidad en el lado muchos (many) del tipo de relación (relationship) lo siguiente:
  - Los Atributos que forman la **llave primaria** de la entidad en el lado uno del tipo de relación(relationship)
  - Si existen atributos de la relacion (relationship), se ponen en la tabla que tiene N
    - course(CRSID, TITLE, TNAME, ….)
      FK(TNAME) references teacher(TNAME)
    - teacher(TNAME, DEPT)



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

R5. CARDINALIDAD M:N

- **R5. *Relaciones N:M* entre clases C1 y C2, crear una relación R para representar la relación entre clases. Incluir las llaves de C1 y C2 y todos los atributos simples de la relación entre las clases.**
- Llave Primaria
  - Combinación de los atributos que forman las llaves primarias de las entidades involucradas en la relación (relationship).



EXPORT(NAME, PNUM, QTY)

PK(NAME,PNUM)
FK(NAME) references COUNTRY(NAME)
FK(PNUM) references PRODUCT(PNUM)

R6. Atributos Multivalor (Multivalued)

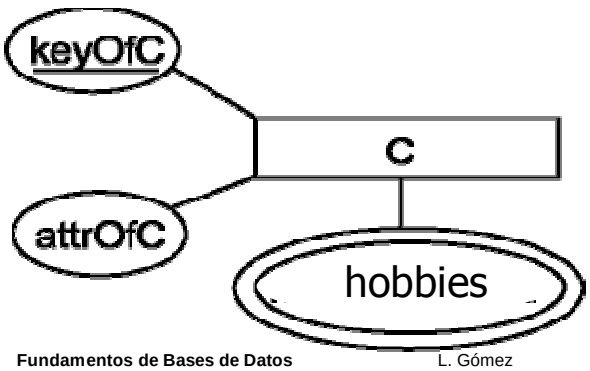
Crear una tabla para el atributo multivalor que incluya la llave de la entidad:

C(keyOfC, attrOfC)

hobbies(keyOfC, hobby)

FK(keyOfC) references C(keyOfC)

La llave de la relación *Hobbies* es una llave compuesta que consiste de la llave de la clase y el atributo multivalor. Notar que la restricción de integridad referencial se debe mantener



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

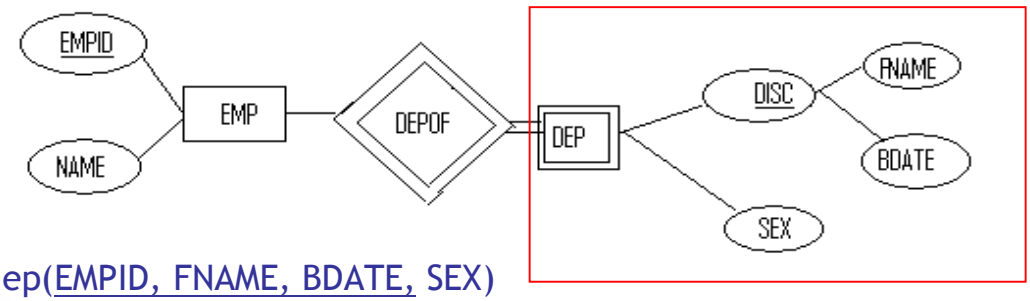
R7. Entidad Débil (Weak Entity)

- Atributos.

  Atributos simples y componentes simples de atributos compuestos de la misma entidad y los atributos correspondientes a la llave primaria de la entidad dueña (strong entity).

- Llave Primaria (Primary Key)

  Combinación de los atributos de la llave primaria de la entidad dueña y el discriminador (llave parcial [partial key]) de la entidad débil.



dep(EMPID, FNAME, BDATE, SEX)

Rel. Recursivas y Atributos Derivados

- R8 Utilizar las reglas R3, R4 y R5 de acuerdo a la cardinalidad 1:1, 1:N o M:N. La llave foránea hace referencia a la misma tabla.
  - Tabla(llave, attr2, attr3, llaveforanea, attrel)
    PK(llave)
    FK(llaveforanea) references Tabla(llave)

- R9 Los attributos derivados no se representan en el modelo relacional. Se deben codificar dentro de la aplicación para calcularse.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Reglas de Mapeo

- R1 Mapeo de Entidades
  - Crear tabla
- R2 Mapeo de Atributos Simples y Compuestos
  - Columnas en la tabla
- R3 Mapeo de relaciones 1:1
  - Llave foránea en una de las tablas
- R4 Mapeo de relaciones 1:N
  - Llave foránea del lado N
- R5 Mapeo de relaciones M:N
  - Crear una tabla con 2 llaves foráneas
- R6 Mapeo de atributos multivalor
  - Una nueva tabla con PK y atributo multivalor
- R7 Mapeo de Entidades débiles
  - Una nueva tabla mas la llave primaria de entidad dueña
- R8 Mapeo de relaciones recursivas
  - Llaves foráneas que referencían a la misma tabla
- R9 Mapeo de atributos derivados
  - No se representan en el modelo

Resumen de reglas de Mapeo

R1. Cada *entidad o clase* se traduce como relación R directamente.

R2. Cada Atributo Simple y cada nodo hoja de los atributos compuestos es una columna en R.

R3. *Relación de 1:1* entre clases C1 y C2 , seleccionar una de las clases (C1) preferentemente aquella con participación total e incluir la llave de C2 en C1 como una llave foránea. Incluir los atributos simples de la asociación entre ambas clases en C1

R4. *Relaciones 1:N* entre clases C1 y C2, seleccionar la clase en el lado N de la relación (C1). Incluir la llave de C2 y atributos simples de la relación entre las clases en C1

R5. *Relaciones N:M* entre clases C1 y C2, crear una relación R para representar la relación entre clases. Incluir las llaves de C1 y C2 y todos los atributos simples de la relación entre las clases. La llave de la relación R es una llave compuesta que incluye las llaves de C1 y C2. Atributos MV de la relación entre clases se traducen en una relación separada (R6)

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Resumen de reglas de Mapeo

R6. Cada ***atributo multivalor (multivalued)*** (MV) de la clase C es puesto en una relación separada agregando la llave de C

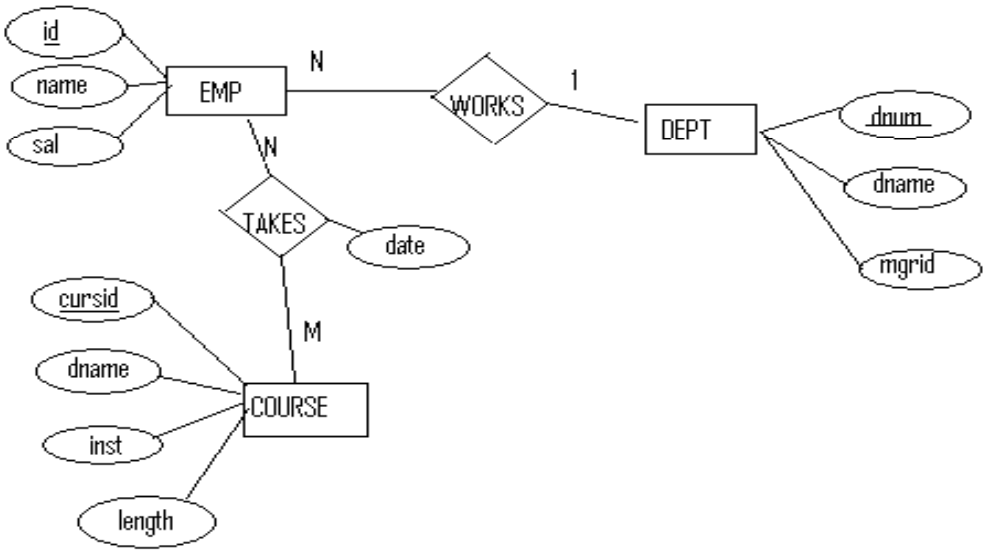
R7. Para cada ***entidad débil*** E1 con entidad dueña E2, crear una relación R. Incluir los atributos simples de la entidad débil y de la relación entre entidades.Incluir la llave de E2 y la llave parcial de E1. La llave de R es la combinación de las llaves de E1 y E2

R8. Utilizar las reglas R3, R4 y R5 de acuerdo a la cardinalidad 1:1, 1:N o M:N. La llave foránea hace referencia a la misma tabla.

R9. No se mapean al modelo relacional

EJERCICIO DE MAPEO

- Mapea el diagrama ER para cursos de entrenamiento de empleados al modelo relacional.



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

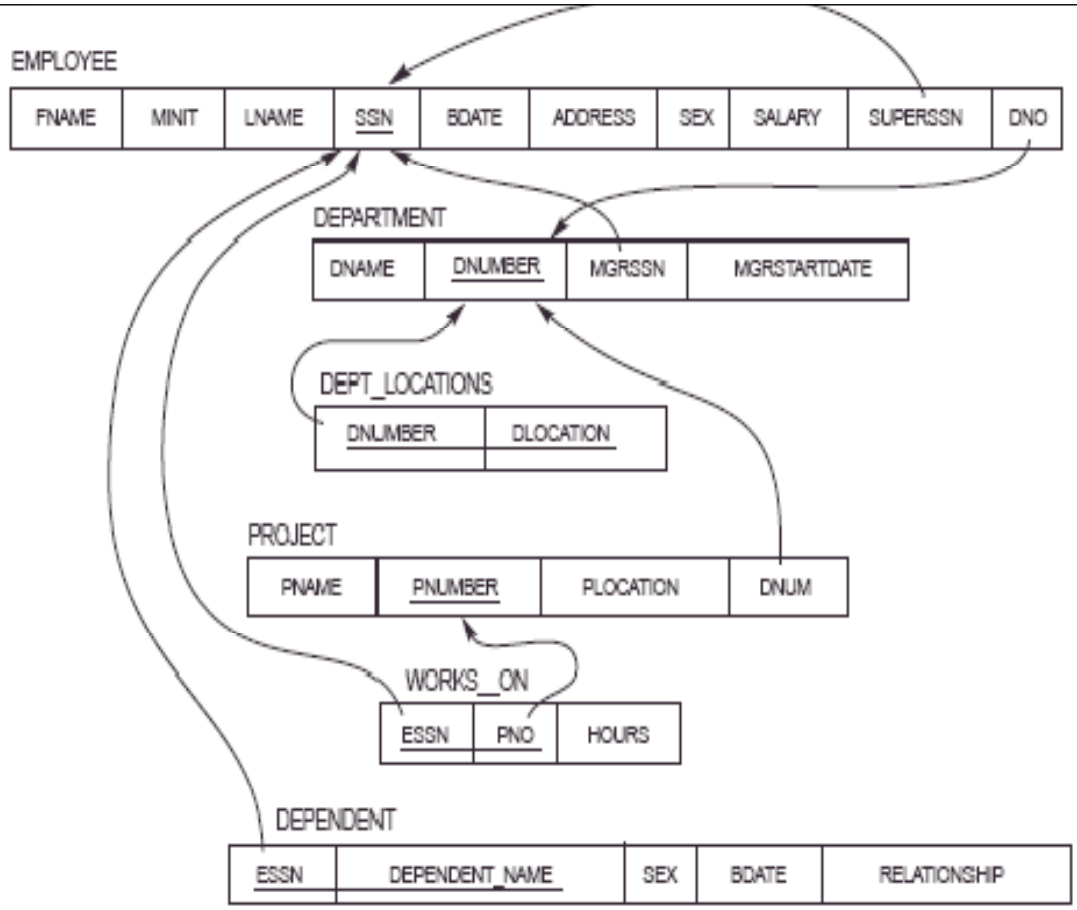
Solución

EMP(id, name, sal, dnum)
- PK(id)
- FK(dnum) references DEP(dnum)

DEP(dnum, dname, mgr)
- PK(dnum)

COURSE (cursid,dname, inst, length)
- PK(cursid)

TAKES (id, cursid, date)
- PK (id, cursid)
- FK(id) references EMP(id)
- FK(cursid) references COURSE(cursid)



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Diseño de Bases de datos Relacionales

Normalización

Diseño de Bases de Datos Relacionales

- Meta:
  Generar un conjunto de esquemas que permita almacenar la información eliminando redundancia innecesaria y que permita accesar la información fácilmente

- Los atributos deben agruparse en tablas (relaciones)
  - minimizando la redundancia de datos
  - reduciendo el espacio requerido para almacenar las tablas

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Redundancia de Datos (Data Redundancy)

Staff

staffNo	sName	position	salary	branchNo
SL21	John White	Manager	30000	B005
SG37	Ann Beech	Assistant	12000	B003
SG14	David Ford	Supervisor	18000	B003
SA9	Mary Howe	Assistant	9000	B007
SG5	Susan Brand	Manager	24000	B003
SL41	Julie Lee	Assistant	9000	B005

Branch

branchNo	bAddress
B005	22 Deer Rd, London
B007	16 Argyll St, Aberdeen
B003	163 Main St, Glasgow

StaffBranch

staffNo	sName	position	salary	branchNo	bAddress
SL21	John White	Manager	30000	B005	22 Deer Rd, London
SG37	Ann Beech	Assistant	12000	B003	163 Main St, Glasgow
SG14	David Ford	Supervisor	18000	B003	163 Main St, Glasgow
SA9	Mary Howe	Assistant	9000	B007	16 Argyll St, Aberdeen
SG5	Susan Brand	Manager	24000	B003	163 Main St, Glasgow
SL41	Julie Lee	Assistant	9000	B005	22 Deer Rd, London

Anomalías
Insert
Update
Delete

Diseño de Bases de Datos Relacionales

Propiedades NO Deseables en un diseño

- Repetición de Información
- Inhabilidad de representar cierta información
- Perdida de información

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Repetición de Información

suppliers(SNAME, SADDR, ITEM, PRICE)

La dirección del proveedor se repite para cada producto

- Se desperdicia espacio Cada tupla para un proveedor debe de actualizarse cuando hay un cambio de dirección.

Inabilidad para representar información

suppliers(SNAME, SADDR, ITEM, PRICE)

Inserción(Insertion):
No podemos grabar una dirección de un proveedor si ese proveedor no provee actualmente al menos un producto o parte (item)

Borrado (Deletion):
Si todos los productos que son proveidos por un proveedor son borrados, perdemos la dirección del proveedor.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Pérdida de Información

addresses(NAME, CITY, ZIP) ≠ addr1(NAME, CITY) join addr2(CITY, ZIP)

El join de tablas addr1 y addr2 no es igual a la tabla addresses

Addresses		
Name	**City**	**Zip**
bill	phx	85050
jane	phx	85051

Se divide en

Addr1	
Name	**City**
bill	phx
jane	phx

Addr2	
City	**Zip**
phx	85050
phx	85051

join

addr1 join addr2		
Name	**City**	**Zip**
bill	phx	85050
bill	phx	85051
jane	phx	85050
jane	phx	85051

Normalización

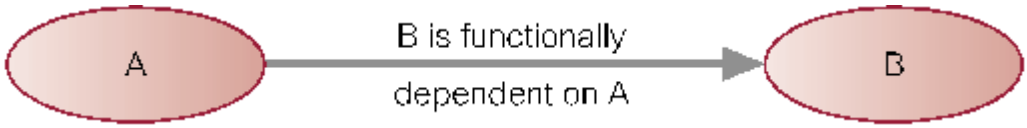
- Las cuatro formas normales más comunes son la primera (1NF), segunda(2NF) y tercera (3NF) forma normal y la forma normal de Boyce-Codd (BCNF)
- Basadas en las dependencias funcionales entre los atributos de una relación.
- Una relación puede normalizarse a una forma específica para prevenir la ocurrencia de
  - anomalías de actualización.
  - anomalías de inserción
  - anomalías de borrado

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

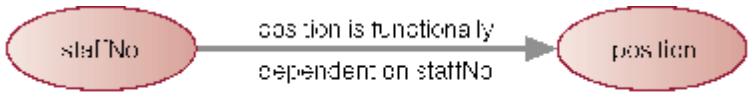
Dependencia Funcional

- Concepto principal asociado con la normalización.
- Dependencia Funcional (Functional Dependency)
  - Describe una relacion entre atributos
  - A, B son atributos de la relacion R,

  A -> B se lee:
  - A determina los valores de B
  - B es funcionalmente dependiente A si cada valor de A en R esta asociado con exactamente un valor de B en R.

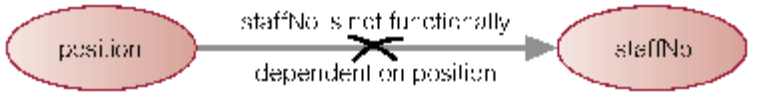


Ejemplo – Dependencia Funcional



Staff number S_21 → Manager

(a)



(b)

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

FD axiomas de Armstrong (reglas de inferencia)

1. Reflexivity
   If B is a subset of A, then A -> B
2. Augmentation
   If A -> B, then A,C -> B,C
3. Transitivity
   If A-> B and B -> C, then A -> C

Resumen de equivalencias en FD

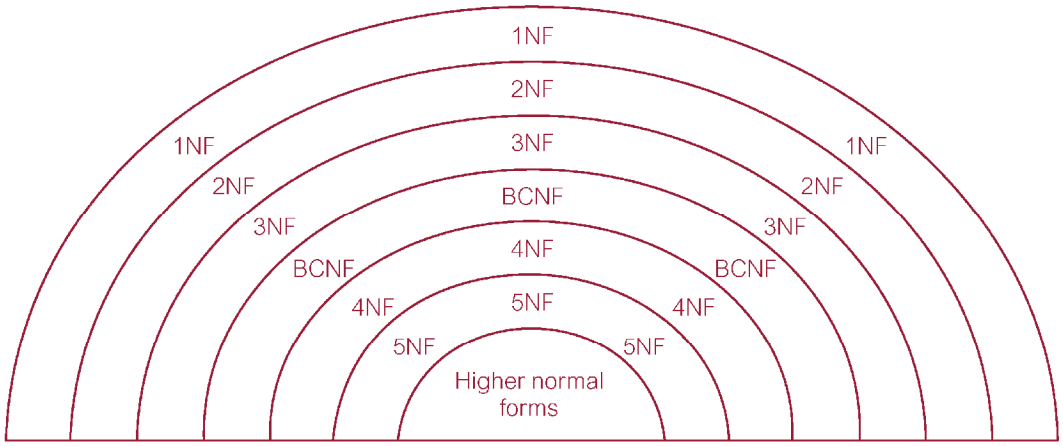
A, B y C son atributos

- if A->B then A,C->B,C
- if A->B,C then A->B y A->C
- if A->B y B->C then A->C

- A->B no es lo mismo que B->A
- If A,B->C no necesariamente A->C and B->C

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Relaciones entre formas normales



Forma no normalizada (UNF)

- Una tabla que contiene uno o mas grupos repetidos.

StaffPropertyInspection

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-00 22-Apr-01 1-Oct-01	10.00 09.00 12.00	Need to replace crockery In good order Damp rot in bathroom	SG37 SG14 SG14	Ann Beech David Ford David Ford	M931 JGR M533 HDR N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-01 24-Oct-01	13.00 14.00	Replace living room carpet Good condition	SG14 SG37	David Ford Ann Beech	M533 HDR N721 HFR

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

De UNF a 1NF

- En la tabla no normalizada (un-normalized table):
  - Selecionar la llave
  - Identificar los grupos repetidos
  - Eliminar los grupos repetidos

    almacenando datos apropiados en las celdas vacías (aplanando “flattening” la tabla)

Primera Forma Normal (1NF)

StaffPropertyInspection

propertyNo	pAddress	iDate	iTime	comments	staffNo	sName	carReg
PG4	6 Lawrence St, Glasgow	18-Oct-00	10.00	Need to replace crockery	SG37	Ann Beech	M231 JGR
		22-Apr-01	09.00	In good order	SG14	David Ford	M533 HDR
		1-Oct-01	12.00	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	5 Novar Dr, Glasgow	22-Apr-01	13.00	Replace living room carpet	SG14	David Ford	M533 HDR
		24-Oct-01	14.00	Good condition	SG37	Ann Beech	N721 HFR

- **Una relación en la cual la intersección de cada renglon y columna contiene uno y solo un valor.**

StaffPropertyInspection

propertyNo	iDate	iTime	pAddress	comments	staffNo	sName	carReg
PG4	18-Oct-00	10.00	6 Lawrence St, Glasgow	Need to replace crockery	SG37	Ann Beech	M231 JGR
PG4	22-Apr-01	09.00	6 Lawrence St, Glasgow	In good order	SG14	David Ford	M533 HDR
PG4	1-Oct-01	12.00	6 Lawrence St, Glasgow	Damp rot in bathroom	SG14	David Ford	N721 HFR
PG16	22-Apr-01	13.00	5 Novar Dr, Glasgow	Replace living room carpet	SG14	David Ford	M533 HDR
PG16	24-Oct-01	14.00	5 Novar Dr, Glasgow	Good condition	SG37	Ann Beech	N721 HFR

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Segunda Forma Normal (2NF)

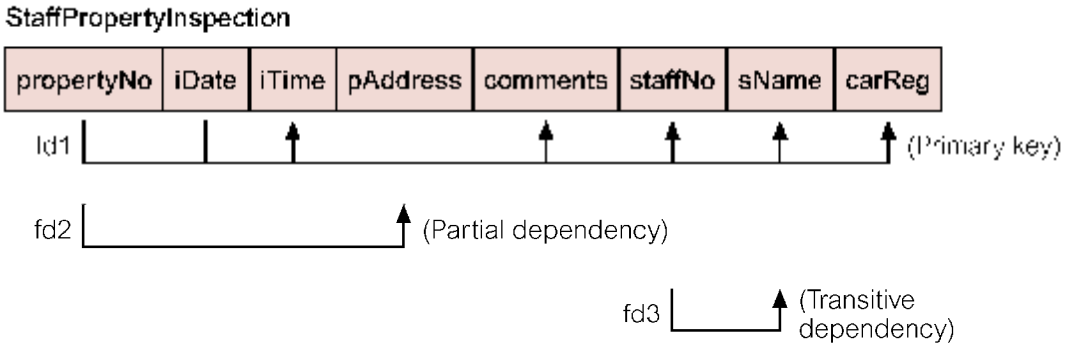
- Basada en el concepto de **dependencia funcional completa:**
  - A y B son atributos de una relación,
  - B es completamente dependiente de A si
    B es funcionalmente dependiente de A pero no de un subconjunto propio de A.
- Una relacion que esta en 1NF y que cada atributo que no es parte de la llave primaria es funcionalmente dependiente de la llave primaria.
- **NO EXISTEN DEPENDENCIAS PARCIALES**

1NF a 2NF

- Identificar la llave primaria de la relación en 1NF.
- Identificar las dependencias funcionales de la relación.
- Si existen dependencias parciales en la llave primaria hay que eliminarlas al colocarlas en una nueva relacion junto con una copia de su determinante.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

2NF



Property
(PropertyNo, Paddress)
PropertyInspection
(propertyNo, iDate,iTime, comments, StaffNo, Sname, carReg)

Tercera Formal Normal (3NF)

- Basada en el concepto de dependencia transitiva:
  - A, B y C son atributos de una relación tal que if A -> B and B -> C,
  - Entonces C es dependiente transitivamente de A a través de B. (A no es funcionalmente dependiente de B o C).
- 3NF – Una relación que está en 1NF y 2NF y cuyos atributos que no forman parte de la llave primaria no son transitivamente dependiente de la llave primaria

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

2NF to 3NF

Property
(PropertyNo, Paddress)
PropertyInspection
(propertyNo, iDate, iTime, comments, StaffNo, Sname, carReg)

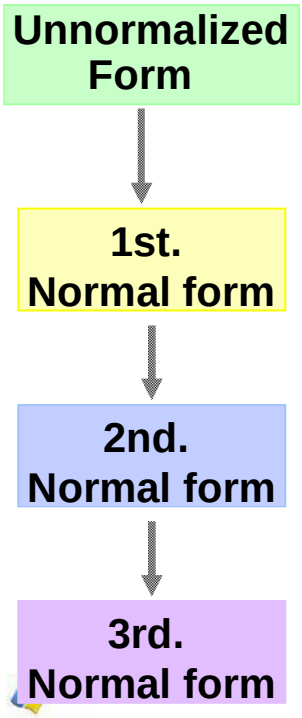
- Identificar la llave primaria en la relación en 2NF
- Copiar dependencias transitivas existentes a una nueva tabla.

Property(PropertyNo, Paddress)

PropertyInspection
(propertyNo, iDate, iTime, comments, StaffNo, carReg)

Staff(StaffNo, Sname)

En resumen



Eliminar grupos repetidos

Todos los atributos no llave deben de depender completamente de la llave primaria (Fully Dependency on PK for all not PK).

No dependencies parciales

Eliminar dependencias transitivas

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Normalizar la siguiente tabla

Matrícula, IdLibro, FechaVencimiento, NombrePersona, Teléfono, Clasificación, TítuloLibro

Dependencias funcionales

(Matrícula, IdLibro, FechaVencimiento, NombrePersona, Teléfono, Clasificación, TítuloLibro)

PK (Matrícula, IdLibro)

Dependencias Funcionales

- Matrícula **à** NombrePersona, Teléfono
- IdLibro **à** TítuloLibro, Clasificación
- Clasificación **à** Título
- Matrícula, IdLibro **à** FechaVencimiento
- Matrícula, IdLibro **à** NombrePersona, Teléfono, Clasificación, TítuloLibro

- QA76.9.d26 c66 2002 Database Systems
- QA76.9.d26 c648 2005 Database Systems

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

2NF

Matrícula	NombrePersona	Teléfono

IdLibro	Clasificación	TítuloLibro

Matrícula	IdLibro	FechaVencimiento

3NF

Matrícula	NombrePersona	Teléfono

IdLibro	Clasificación

Clasificación	TítuloLibro

Matrícula	IdLibro	FechaVencimiento

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Transacciones

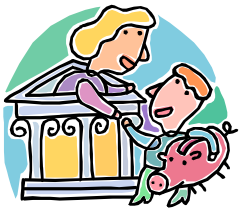
Transacciones

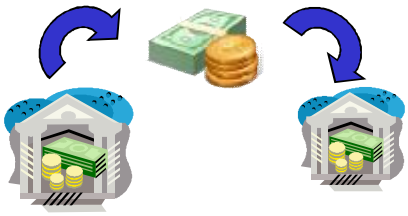
Recuperación

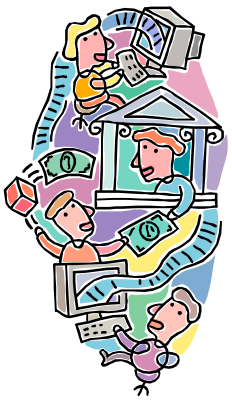
Control de concurrencia

Transacciones

- Ejemplos







Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Falla en una transacción

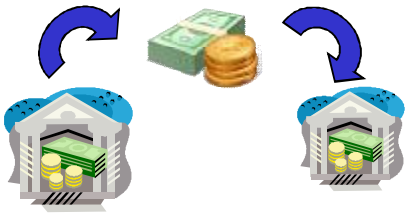
- Problemas potenciales causados por fallas del sistema.

```
RetiroATM (cantidad)
Read cantidad
If saldo > cantidad
   {
   saldo <- saldo – cantidad
            /* Ocurre una falla en el sistema */
   Cajero entrega el dinero
   }
```

Transaccion

Un programa tomado como unidad atómica (todo o nada) que realiza accesos o actualizaciones a una base de datos llevando a la base de datos de un estado consistente (correcto) a otro estado consistente.

```
Base de datos en estado consistente
Begin transaction; /* Transfer fund */
   SaldoCuenta1 <- SaldoCuenta1– cantidad
            Base de datos en estado inconsistente aqui
    SaldoCuenta2 <- SaldoCuenta2 + cantidad
End transaction;  /* Transfer funds */
Base de datos en estado consistente
```



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

ACID

- **Propiedadades ACID de una transaccion**
- Atomicidad (Atomicity)
- Consistencia (Consistency)
- Aislamiento (Isolation)
- Durabilidad (Durability).

ATOMICIDAD (ATOMICITY)

- "El sistema bajo prueba debe garantizar que las transacciones son ATOMICAS, el sistema realizara todas las operaciones individuales en los datos o asegurarar que ninguna operación completada parcialmente tenga ningun efecto en los datos"… "TODO O NADA"

```
RetiroATM (cantidad)
Read cantidad
If saldo > cantidad
   {
   saldo <- saldo – cantidad
         /* Ocurre una falla en el sistema */
   Cajero entrega el dinero
   }
```


Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

CONSISTENCIA (CONSISTENCY)

- "Consistencia es la propiedad que requiere para cualquier ejecución de la transacción que la base de datos pase de un estado consistente a otro estado consistente"

```
RetiroATM (cant1)
Read saldo
If saldo > cant1
   {
    saldo <- saldo – cant1
    Cajero entrega el dinero
   }
```

```
RetiroATM (cant2)
Read saldo
If saldo > cant2
   {
    saldo <- saldo – cant2
    Cajero entrega el dinero
   }
```


AISLAMIENTO (ISOLATION)

- Las operaciones de transacciones concurrentes deben llevar a resultados exactamente iguales a los resultados que se hubieran obtenido al ejecutar las transacciones de manera serial
- La transacción NO debe revelar sus resultados parciales (uncommitted) a otras transacciones.
- Las transacciones se ejecutan sin interferencias de otras transacciones concurrentes. las transacciones son independientes.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Isolation

- Cuál es el saldo final en la cuenta si el saldo inicial es $500 y cant1= 100 y cant2=50

RetiroATM (cant1)
Read saldo
If saldo > cant1
{
saldo <- saldo – cant1
 	
transaccion aborta
Cajero entrega el dinero
}
	RetiroATM (cant2)
	
	Read saldo
	If saldo > cant2
	{
	saldo <- saldo – cant2
	Cajero entrega el dinero
	}

$350 ? $400?

DURABILIDAD (DURABILITY)

- "Todos los cambios hechos por transacciones terminadas (committed transactions) se vuelven permanentes en la base de datos y no deben ser alterados como consecuencia de fallas subsecuentes.

- Es responsabilidad del subsistema de recuperación de fallas el asegurar la durabilidad.
  - Falla permanente irrecuperable de un medio durable conteniendo datos de la base de datos o bitácoras de recuperación.
  - Falla del sistema (System crash/hang) que requiera re-inicialización del sistema (system reboot)
  - Falla de memoria (Memory failure) total o parcial

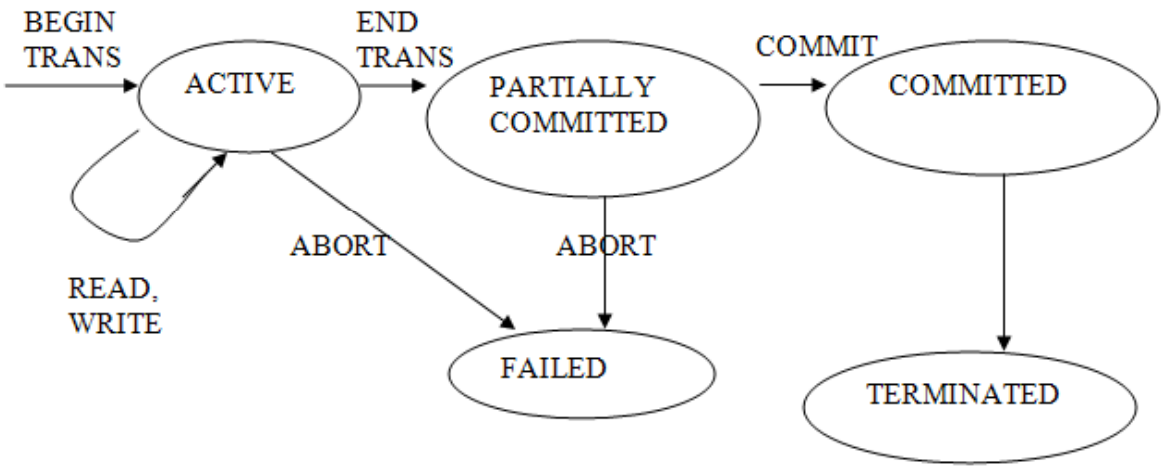
Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Recuperación (RECOVERY CONTROL)

- Parte integral del DBMS, responsable de
  - detección de fallas
  - restauración de la BD a un estado consistente que existía previo a la falla.
  - responsabilidad del esquema de recuperación para asegurar atomicidad

Estados de una Transacción

Inicio



Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Bitácora del Sistema (SYSTEM LOG)

- Registra todas las operaciones de las transacciones que afectan los valores de los elementos de la base de datos para recuperarlos en caso de fallas.
- La bitácora (Log) se mantiene en disco y periódicamente se respalda en otras unidades.

  Sea T una transacción única y X un elemento de la base de datos.
- Registros en la BITACORA (LOG ENTRIES)
  - [start-transaction, T]
  - [write-item, T, X, new_value], o
    [write-item, T, X, old_value, new_value]
  - [read-item, T, X]
  - [commit, T]

Suposiciones

- Las transacciones no están anidadas.
- Todas los cambios permanentes a la BD ocurren dentro de las transacciones, de tal forma que las operaciones en la bitácora se pueden
  - des-hacer (undo)
  - re-hacer (redo)

  basándose en estados de la transacción.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Protocolo Incremental con Updates Inmediatos

- Este protocolo usa
  - [start-transaction, T]
  - [read-item, T, X]
  - [write-item, T, X, old_value, new_value]
  - [commit, T]
- Registros en la bitácora deben ser escritos a almacenaje estable (stable storage) antes de ejecutar una actualización inmediata (immediate update).
- Esquema de recuperación.
  - Undo(Ti) donde Ti es una transacción que no ha hecho COMMIT (uncommitted transaction)
  - Redo(Tj) donde Tj es una transacción que YA ha hecho COMMIT (committed transaction)
- Re-hacer y des-hacer (Redo, Undo) deben ser idempotentes
  - (idempotent) – al ejecutarlas varias veces es equivalente a ejecutarlas una sola vez

Protocolo Incremental con Updates diferidos

- Durante la ejecución de la transacción, todas las operaciones de WRITE se hacen solo en memoria, se escriben en el log y en el espacio de la transacción hasta que se llega al estado de partially commits.
- Al momento del commit, el LOG se escribe en disco y las transacciones y los WRITE que se pospusieron (deferred writes) son ejecutados..
- This protocol uses entries:
  - [start-transaction, T]
  - [write-item, T, X, new_value]
  - [commit, T]

Esquema de recuperación

  - IgnoraRegistro(Ti) donde Ti es una transacción que no ha hecho COMMIT (uncommitted transaction)
  - Redo(Tj) donde Tj es una transacción que YA ha hecho COMMIT (committed transaction)
- Re-hacer (Redo) deben ser idempotente.

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Checkpoints

- Se utiliza en el archivo de log para mejorar el rendimiento del proceso de recuperación
- El registro **[Checkpoint]** indica que el sistema ha escrito en disco todas las operaciones de transacciones que ya hicieron commit
- Las transacciones que ya hicieron commit, antes del checkpoint no requieren el proceso de REDO en caso de una falla del sistema

Ejercicio

[start-trans T1]
...
[commit T1]
[start-trans T2]
...

A: T1 committed, T2 uncommitted

???

Que hacer si la falla ocurre en A?

a) Protocolo Updates Inmediatos

b) Protocolo updates diferidos

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Ejercicio

[start-trans T1]
...
[commit T1]
[start-trans T2]
...
T1 committed, T2 uncommitted
[checkpoint]
B: T1 committed, T2 uncommitted

[commit T2]
T1 & T2 committed

Que hacer si la falla ocurre en B?

a) Protocolo Updates Inmediatos

b) Protocolo updates diferidos

Ejercicio

[start-trans T1]
...
[commit T1]
[start-trans T2]
...
T1 committed, T2 uncommitted
[checkpoint]
T1 committed, T2 uncommitted
[commit T2]
C: T1 & T2 committed

Que hacer si la falla ocurre en C?

a) Protocolo Updates Inmediatos

b) Protocolo updates diferidos

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Recuperación

crash at point	a) IMMEDIATE	b) DEFERRED
A	REDO T1, UNDO T2	REDO T1
B	UNDO T2	--
C	REDO T2	REDO T2

Control de concurrencia

Motivación

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Problemas potenciales debido a la concurrencia

- Update perdido (lost update)
  - Las transacciones tratan de modificar un mismo valor, la última que ejecuta es la que deja el registro
- Lectura de valores ya no válidos (dirty read)
  - Una transacción lee un valor que fue modificado por otra transacción que fue cancelada (rollback)
- Análisis Inconsistente
  - Valores inconsistentes determinados por la ejecución intercalada de transacciones

Lost Update

- Un UPDATE de un usuario puede ser re-escrito por otro usuario

Time	T1	T2	X
t1		Begin_transaction	100
t2	Begin_transaction	read(X)	100
t3	read(X)	X = X + 100	100
t4	X = X - 10	write(X)	200
t5	write(X)	commit	90
t6	commit		90

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Dirty Read

- Se le permite a una transacción ver los resultados intermedios de otra transacción que no ha hecho COMMIT

Time	T3	T4	X
t1		Begin_transaction	100
t2		read(X)	100
t3		X = X + 100	100
t4	Begin_transaction	write(X)	200
t5	read(X)	...	200
t6	X = X - 10	rollback	100
t7	write(X)		190
t8	commit		190

Análisis Inconsistente

T5 lee varios valores, T6 modifica algunos valores durante la ejecución de T5

Time	T5	T6	X	Y	Z	Sum
t1		Begin_transaction	100	50	25	
t2	Begin_transaction	sum = 0	100	50	25	0
t3	read(X)	read(X)	100	50	25	0
t4	X = X - 10	sum = sum + X	100	50	25	100
t5	write(X)	read(Y)	90	50	25	100
t6	read(Z)	sum = sum + Y	90	50	25	150
t7	Z = Z + 10		90	50	25	150
t8	write(Z)		90	50	35	150
t9	commit	read(Z.)	90	50	35	150
t10		sum = sum + Z	90	50	35	185
t11		commit	90	50	35	185

175?

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

Mecanismos de Control de Concurrencia

- No existe EL MEJOR, es decir el que sea más eficiente en todas las situaciones
- Bloqueo (LOCKS)
  - Usa candados de lectura y escritura
  - Desventaja: Puede producir Deadlock (procesos se quedan esperando por eventos que nunca ocurrirán)
- Marcas de tiempo (TIMESTAMP)
  - Usan registros de tiempo, se aborta la transacción si el valor ya fue modificado por una transacción más reciente.
  - Desventaja: Puede producir demasiados Rollback

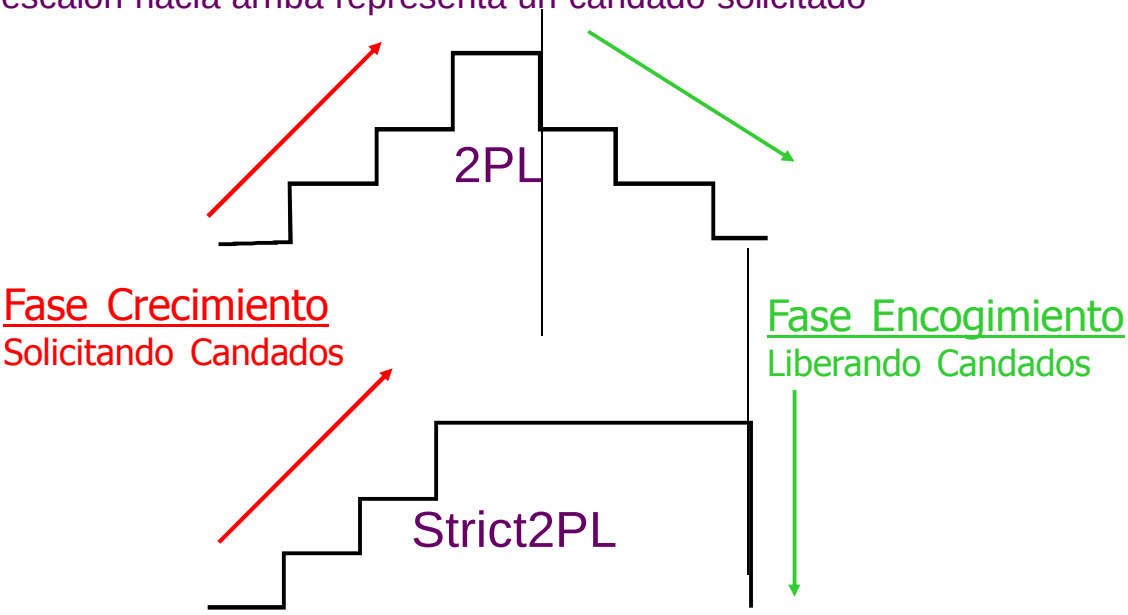
Bloqueo de 2 fases (2PL)

- Fase 1 Crecimiento
  - La transacción solo puede solicitar candados
  - Un candado por cada elemento a leer o modificar
  - Candados de lectura, varias transacciones pueden obtenerlo
  - Candado de escritura, solo una transacción a la vez, si el candado está ocupado, la transacción se espera en la fila automáticamente y se activa cuando le toca su turno para el recurso
- Fase 2 Encogimiento
  - La transacción solo libera candados en esta fase, una vez que libera un candado, ya no puede solicitar más
  - Al liberarse un candado, puede activar una transacción en espera
- No garantiza Aislamiento

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com

2PL vs Strict2PL

Cada escalón hacia arriba representa un candado solicitado



Bloqueo de 2 fases estricto (strict2PL)

- Igual que 2PL, solo que los candados son liberados hasta que la transacción va a hacer commit
- SI garantiza AISLAMIENTO

Printed with FinePrint trial version - purchase at www.fineprint.com
PDF created with pdfFactory trial version www.pdffactory.com