

Una relación “Muchos a muchos” en un esquema lógico, cuando se pasa al esquema físico:

- Siempre se convierte en tabla

Una relación recursiva en un esquema lógico:

- Puede contener atributos opcionales monovalentes
No se porque -> Pregunta trampa: es porque dice “esquema lógico”
- no puede contener atributos
- puede contener atributos compuestos
- puede contener atributos opcionales polivalentes

Una interrelación en el modelo conceptual:

- Debe tener atributos.
- Debe tener al menos un identificador.
- Debe ser binaria.
- Ninguna de las anteriores

No es necesario tener atributos, si tiene atributos no necesita identificador y puede utilizar uno externo por ser entidad débil, no hay necesidad de que sea binaria si o si una interrelación.

Con respecto al mecanismo de abstracción en el “Diseño Conceptual”. ¿Cuáles de las siguientes afirmaciones son correctas?

- La abstracción de “Agregación” no se aplica en las relaciones.
- La abstracción de “Agregación” no se aplica en los atributos compuestos.
- La abstracción de “Agregación” no se aplica en las entidades.
- Ninguna de las anteriores.
Porque lo dice en la filmina 19/40 y ahí explica como es que representa.

El modelo conceptual posee la cualidad de que cada concepto tiene una sola interpretación, precisa y bien definida. Esa cualidad es:

- Formalidad.

Filmina 3/40 de la clase 2. La formalidad es eso.

En cuanto al proceso de normalización:

- Puede afectar el tiempo de acceso a los datos en las tablas en donde se producen cambios.
Si porque a veces tenes atributos repetidos para que vaya todo mas rápido, pero es mas difícil tener q andar actualizándolos, y a veces es costos actualizarlos.
- Si una tabla está en 3FN, no puede contener dependencias parciales.
Ósea estar en 2FN
- Si una tabla está en 3FN, no puede contener dependencias transitivas.
Si por definición.

Sea X, clave foránea en la tabla T1:

- X puede ser clave primaria en otra tabla.
- X debe ser clave primaria en otra tabla.
Por dificionion de clave foránea. Siempre es primaria en otra tabla
- Es posible usar X para aplicar integridad referencial.
La IR es la que permite que cuando borrás una tupla que es referenciada por otra tabla no queden cosas inconsistentes en la BD. Mantiene la INTEGRIDAD en las REFERENCIAS
- Se define un índice primario sobre X en T1
- Se define un índice secundario sobre X en T1
El índice primario es para las primary key, las foráneas tienen secundario q es bastante bueno igual para q sea rápido.

Un atributo compuesto en el modelo conceptual:

- Puede ser monovalente. (tener cardinalidad max 1)

- Puede ser obligatorio. (tener cardinalidad min 1)
- Puede ser polivalente. (tener cardinalidad max N)

Cuando se transforma un esquema conceptual en esquema lógico se debe:

- Eliminar atributos polivalentes.
- Eliminar atributos monovalentes compuestos
- Eliminar atributos opcionales compuestos
- Eliminar atributos opcionales.
- Eliminar atributos derivados.

Los opcionales se dejan y los derivados no siempre se eliminan, a veces se dejan por cuestiones de performance.

El proceso de diseño lógico de una BD:

- Es independiente del tipo de DBMS a utilizar.
Diseño lógico y físico dependen del TIPO de SGBD
- Es independiente del DBMS específico a utilizar.
También lo es el conceptual. El físico NO
- Es una descripción de alto nivel del contenido de información de la BD.
Esto es conceptual.
- Describe en detalle las estructuras de almacenamiento y método de acceso efectivo a los datos.
Esto es verdadero para físico.

Con respecto al proceso de "Validación" del esquema conceptual:

- Al verificar la cualidad de "Corrección" hay que hacerlo tanto sintácticamente como semánticamente.
- Para verificar la cualidad de "Minimalidad" basta con verificar que no se puede borrar un concepto del esquema sin perder información.
- Para verificar la cualidad de "Compleción" basta con verificar que los requerimientos estén representados en el esquema.
Esta es trampa porque no solo basta con eso sino también con Verificar en el esquema que cada concepto sea mencionado en los requerimientos

Dada una jerarquía de generalización en un esquema conceptual, en donde solo existe una entidad subconjunto:

- La Cobertura debe ser Parcial y exclusiva.
PORQUE TIENE SENTIDO

Un identificador de una entidad en un esquema conceptual:

- Puede ser interno
- Puede ser externo
- Puede ser simple
- Puede ser compuesto

UN esquema que está en 4FN:

- Necesariamente está en 3FN.
- Necesariamente está en BCFN
- No tiene atributos polivalentes

La cardinalidad de un identificador en un esquema lógico:

- Puede ser (0,1).
- Debe ser (0,1).
- Puede ser (1,1).

- Debe ser (1,1).
PORQUE TIENE SENTIDO (identificador necesita existir y ser solo 1 por tupla)
- Puede ser (1, N).
- Debe ser (1, N)

Un atributo derivado

- Puede existir en el esquema conceptual.
- Debe existir en el esquema lógico, si está en el esquema conceptual.
No debe, Puede
- No debe existir en el esquema lógico si está en el esquema conceptual.
MAL
- Debe existir en el esquema físico.
- MAL

Para que un esquema conceptual sea mínimo no debe tener:

- Ciclos entre entidades
Porque genera repetición de información.
- Atributos polivalentes
- Atributos compuestos
- Atributos derivados
- Ninguna de las anteriores

La conversión al modelo lógico de una jerarquía con cobertura parcial superpuesta, se puede resolver:

- Matando al padre y dejando solo a los hijos
- Dejando todo con una relación 'ES UN' entre hijos y padres.
- Matando a los hijos y dejando solo al padre.

El proceso de diseño físico de un BD:

- Es independiente del tipo de DBMS a utilizar.
- Es independiente del DBMS específico a utilizar.
- Es una descripción de alto nivel del contenido de información de la BD.
- Describe en detalle las estructuras de almacenamiento y método de acceso efectivo a los datos.

Enumere las posibilidades de integridad referencial que provee un SGBD:

- Restringir la operación
- Realizar la operación "en cascada"
- Establecer la clave foránea en nulo
- No hacer nada → se delega responsabilidad

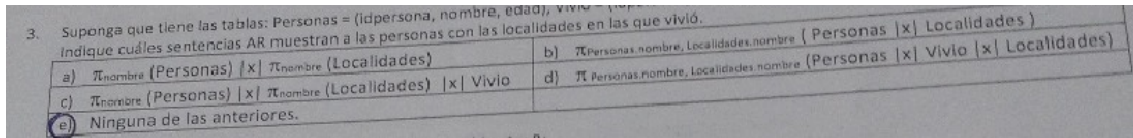
El proceso de diseño conceptual de BD:

- Es independiente del tipo de DBMS a utilizar.
 - Es independiente del DBMS específico a utilizar.
 - Es una descripción de alto nivel del contenido de información de la BD.
 - Describe en detalle las estructuras de almacenamiento y método de acceso efectivo a los datos.
-

En una consulta SQL:

- Es posible utilizar funciones de agregación en el GROUP BY
- Es posible utilizar funciones de agregación en el WHERE
- Es posible utilizar funciones de agregación en el WHERE, siempre y cuando se utilice un GROUP BY
- Ninguna de las anteriores (solo en el select y el having)

Suponga que tiene tablas **Personas(idpersona,nombre,edad)** **Vivio(idpersona,idlocalidad)** **Localidades(idlocalida,nombre,cp)** Indique cual sentencia de AR muestran personas con localidades en las que vivió



La primera no porque no cruza con Vivio

La B es Persona natural localidades, ósea nada q ver

La C no porque el cruzamiento está mal

La D lo cruza mal porque los nombres de persona y localidad son los mismos y los cruza, ahí tenes que hacer producto cartesiano con un where, o renombrar el atributo (que no me acuerdo como se hacia)

La función de agregación SUM de SQL:

- Debe usarse con al menos dos tablas involucradas EN la cláusula FROM
- Puede usarse junto a la cláusula HAVING o SELECT
- Debe usarse junto a la cláusula GROUP BY
- Devuelve un número que se corresponde con la cantidad de tuplas del resultado
- Ninguna de las anteriores

La función de agregación COUNT de SQL:

- Debe usarse con al menos dos tablas involucradas EN la cláusula FROM
- Puede usarse junto a la cláusula HAVING o SELECT
- Puede usarse dentro de la cláusula GROUP BY
- Puede usarse dentro de la cláusula WHERE
- Devuelve un número que se corresponde con la cantidad de tuplas del resultado
- Ninguna de las anteriores

La cláusula ALTER TABLE de SQL:

- Es una cláusula de manipulación de datos
 - Permite alterar el valor de un dato en una fila
 - Permite alterar el valor de varios datos simultaneamente en diversas filas de la tabla
 - Ninguna de las anteriores
- Permite modificar la TABLA (columnas) no sus datos

El producto cartesiano del álgebra relacional entre dos tablas A y B:

- Relaciona todas las tuplas de B con todas las tuplas de A
- Relaciona las tuplas de B con todas las tuplas de A, solo si A y B tienen al menos un atributo en común

- Relaciona las tuplas de B con todas las tuplas de A, solo si un atributo en común entre A y B coincide en su valor
- Relaciona las tuplas de B con todas las tuplas de A, solo si todos los atributos en común entre A y B coinciden en su valor
- Ninguna de las anteriores

Se registran datos de empleados y sus hijos Empleado(nroEmp,nyapEmp,domicilio) Hijo(DNI,nyap,nroEmp). Se desea obtener listado donde aparezcan todos los nombres de los empleados y para cada empleado los nombres de sus hijos. Cuáles de las siguientes cláusulas cumplen lo deseado

- SELECT nyapEmp,nyap FROM Empleado, Hijo WHERE Empleado.nroEmp = Hijo.nroEmp
- SELECT nyapEmp, nyap FORM Empleado INNER JOIN Hijo ON Empleado.nroEmp = Hijo.nroEmp
- SELECT nyapEmp, nyap FORM Empleado LEFT OUTER INNER JOIN Hijo ON Empleado.nroEmp = Hijo.nroEmp
- SELECT nyapEmp, nyap FORM Empleado RIGHT OUTER INNER JOIN Hijo ON Empleado.nroEmp = Hijo.nroEmp

Con respecto al producto natural entre dos tablas T y S

- El tiempo teórico de rta de $T \times S$ y $S \times T$ es siempre el mismo
- El tiempo teórico de rta de $T \times S$ y $S \times T$ puede variar (una va a ser más eficiente)
- Genera el mismo resultado si se hace $T \times S$ que $S \times T$
- Una de las diferencias con el producto theta, es que el producto theta mantiene las columnas repetidas y el producto natural no.
- Si coinciden más de un atributo entre T y S se realiza por la coincidencia de todos los atributos comunes a la vez
- Ninguna de las anteriores

Dadas la tabla Emleados(nroEmp,nombreEmp,dni,tel) definida sin campos autoincrementales. Cuales de las siguientes sentencias son validas

- DELETE Empleados WHERE nroEmp=1204
- CREATE VIEW vistaEmpleados AS (SELECT nroEmp, nombreEmp FROM Empleados)
- INSERT INTO Empleados VALUES(2050,"Carlos gonzales",123214245)
- UPDATE Empleados SET tel='221-123232'
- Ninguna de las anteriores

Dadas dos tablas R1 y R2 al realizar $R1 \bowtie R2$

- Se necesita que el esquema de R1 esté incluido en el esquema de R2 (NO, al revés)
- Esquema($R1 \bowtie R2$) = Esquema(Esquema(R1)+Esquema(R2)) → No, es un "-" no un "+"
- El resultado son valores de atributos de R1 que se relacionan con todas las tuplas de R2
- Ninguna de las anteriores

Dada la tabla Personas(dni,nombre,domicilio), se debe presentar el nombre de todas las personas ¿Cuáles de las siguientes sentencias en CRT/CRD son válidas y cumplen con lo pedido?

- $\{t / (s \in \text{Personas}) \wedge (t[\text{nombre}] = s[\text{nombre}])\}$
- $\{\text{nom} / ((\text{dni}, \text{dom}) / (\text{dni}, \text{nom}, \text{dom}) \in \text{Personas})\}$
- $\{\text{nom} / ((\text{dni}, \text{nom}) / (\text{dni}, \text{nom}, \text{dom}) \in \text{Personas})\}$

- $\{t / (s/s \in \text{Personas}) \wedge (t \in \text{Personas})\}$
- $\{t[\text{nombres}] / (t \in \text{Personas})\}$
- Ninguna de las anteriores

Dadas las tablas: Alumnos(nombre,email,tel), Inscripciones(nombre,materia). Indicar las consultas que son válidas

- **SELECT nombre FROM Alumnos WHERE EXIST (SELECT * FROM Inscripciones WHERE Alumnos.nombre = Inscripciones.nombre)**
- SELECT nombre FROM Alumnos WHERE nombre EXIST (SELECT nombre FROM Inscripciones WHERE Alumnos.nombre = Inscripciones.nombre)
- SELECT nombre FROM Alumnos WHERE nombre IN (SELECT * FROM Inscripciones WHERE Alumnos.nombre = Inscripciones.nombre)
Esta mal porque no devuelve nombres.
- **SELECT nombre FROM Alumnos**
- Ninguna de las anteriores

El lenguaje de definición de datos en un sistema de base de datos:

- Permite asignarle el valor NULL a un atributo de una tupla de una tabla
- Permite agregar una nueva fila a una tabla
- **Permite establecer si un atributo de una tabla acepta valores nulos**
- **Permite establecer el índice primario en una tabla**
- Permite agregar una tupla a una tabla
- **Permite agregar una nueva columna a una tabla**
- Ninguna de las anteriores

Con respecto al Cálculo Relacional de Tuplas (CRT) y al Cálculo Relacional de Dominios (CRD):

- **En CRT, es posible que una variable libre tome valores de una sola tabla**
- **En CRT, es posible que una variable libre tome valores de más de una tabla**
- En CRD, es posible utilizar el cuantificador universal, pero no es posible usarlo en CRT
- En CRT, es posible utilizar el cuantificador universal, pero no es posible usarlo en CRD

La cláusula MIN de SQL:

- Solo se puede utilizar en la cláusula HAVING
- Siempre requiere de una cláusula GROUP BY
- Calcula el valor mínimo para una fila de la tabla
- **Ninguna de las anteriores**

Se cuentan con las siguientes tablas: EMPLEADOS(nroEmp,nyap,fechanac); AREAS(nroArea,nombre,tamaño); TRABAJAN(nroEmp,nroArea). Cuales de estas consultas retorna como resultado los empleados que trabajan en todas las áreas

- TRABAJAN % ÁREAS
- ÁREAS % TRABAJAN
- EMPLEADOS % TRABAJAN
- **Ninguna de las anteriores (Trabajan % PI_nroArea Areas))**

Una sentencia de DML en SQL:

- **Puede no tener una cláusula FROM**
- Siempre debe tener una cláusula WHERE

- Siempre debe tener una cláusula SELECT
- Ninguna de las anteriores

Suponiendo que los atributos forman parte de las tablas indicadas en cada caso. Indique qué sentencias de SQL son válidas:

- **SELECT AVG(stock) as promedio FROM Productos WHERE Productos.categoria=3455**
- **SELECT rubro,nombre,SUM(cantidad) FROM Ventas GROUP BY rubro,nombre HAVING SUM (cantidad) > 100**
- **SELECT codigo,nombre FROM ventas1,ventas2 WHERE ventas1.codigo=ventas2.codigo**
- Ninguna de las anteriores

Cual de las siguientes operaciones de AR no son fundamentales (no añaden potencia):

- **División**
- Diferencia
- **Asignación**
- **Intesección**
- Selección
- Proyección
- Renombre
- Producto cartesiano
- Unión

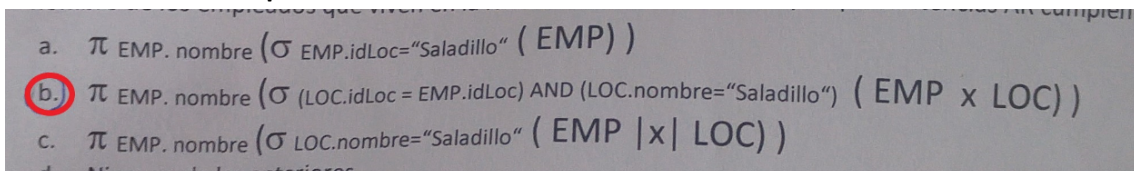
Con respecto al AR:

- Es un lenguaje de consulta no procedural → FALSO, es procedural
- Las operaciones pueden ser unarias, binarias o ternarias. → FALSO, ternarias no existe
- **El predicado de una operación de "Selección" puede tener conectivos lógicos y operadores de comparación**
- Ninguna de las anteriores

Con respecto a las vistas de SQL:

- **Una vista puede utilizarse como un mecanismo de seguridad**
- Una vista puede estar desactualizada
- **De una vista sólo se almacena su definición, no los datos que contiene**
- Ninguna de las anteriores

Se cuenta con las siguientes tablas: EMP(idEmp,dni,nombre,dirección,idLoc); LOC = (idLoc,nombre). Se desea tener el nombre de los empleados uqe viven en 'Saladillo'. Que sentencias de AR cumplen con lo solicitado.



Cuales enunciados responden a la siguiente consulta SQL (la función month devuelve el nro de mes de una fecha):

**SELECT F.idCliente, SUM(F.monto) FROM Renglones R RIGHT OUTER JOIN Facturas F ON
R.idFactura = F.idFactura WHERE (month(F.fecha) =1) AND (F.monto<270) GROUP BY
F.idCliente**

- Mostrar para cada cliente las facturas que realizó en el mes de enero que superaron los 270 pesos
- Mostrar para cada cliente la suma final de sus facturas con monto final superior a 270 pesos del mes de Enero
- Mostrar para cada cliente la suma final de sus facturas con monto final superior a 270 pesos del mes de Enero y que tengan al menos un renglon.
- Ninguna de las anteriores. (creo que Mostrar para cada cliente la suma final de sus facturas con monto inferior a 270 pesos del mes de Enero).

En AR la expresión $A - (A-B)$ es equivalente a :

- $A \cap B$
- $A \times (B-A)$
- $A \cup B - ((A-B) \cup (B-A))$
- Ninguna de las anteriores

“La ejecución aislada de la transacción conserva el estado consistente de la BD”, corresponde con la propiedad de:

- Consistencia

¿Cuáles de las siguientes trazas de ejecución corresponde a una bitácora con modificación inmediata de BD en un entorno monousuario?

a) T0 Start T0 A, 22, 52 T0 B, 35, 11 T0 C, 46, 12 T0 Commit	b) T0 Start T0 A, 20, 52 T0 B, 30, 11 T0 Abort	c) T0 Start T0 A, 52 T0 B, 35 T0 Commit	d) T0 Start T0 A, 58 T0 B, 35 T0 C, 45 T0 Abort	e) T0 Start T0 A, 22, 52 T1 Start T0 B, 18, 35 T1 Abort T0 Commit
---	--	---	--	---

deben tener el start, luego se debe guardar el valor anterior ya que los cambios se ejecutan en la base de datos, es decir debe tener esta forma ($\langle T_i, E, V_a, V_n \rangle$), y siempre terminan con un commit, la e esta mal porque es un monousuario, por lo tanto, debe terminar una para que arranque la otra.

Una ejecución de dos transacciones en un ambiente monousuario:

- A veces puede producir deadlock.
- A veces puede producir inanición.
- Necesita un mecanismo de recuperación en caso de error.

Con respecto al uso de doble paginación como método de recuperación de integridad de una BD:

- Es necesario dividir la bitácora en páginas. (no se usa bitacora)
- Los ABORT se realizan automáticamente.
- Utiliza dos tablas (la actual y la sombra) que se guardan en memoria principal. (la otra en almacenamiento estable)
- En algún momento puede haber mas de una copia de una página determinada.

Una transacción que alcanza el estado de ABORTADA:

- Puede haber cometido parcialmente.
- Puede haber cometido.
- Tiene que haber fallado

En un entorno concurrente, y si las transacciones involucradas son correctas (transforman la base de datos de un estado correcto a otro correcto):

- La ejecución de las transacciones en cualquier orden serial es correcto.
- Una ejecución intercalada será correcta si el estado de la BD resultante es el mismo que el obtenido por una ejecución serial.
- Cualquier ejecución intercalada dejara un estado inconsistente.

El protocolo de bloqueo de dos fases:

- En la fase de crecimiento es posible pedir bloqueos, pero no liberarlos.
- En la fase de decrecimiento solo es posible liberar bloqueos.
- Permite ejecutar transacciones concurrentes sin que se genere deadlock.
- Permite ejecutar transacciones concurrentes sin que se generen inaniciones.
- Evita que se produzca perdida de consistencia de los datos.
- En ciertas ocasiones puede provocar una planificación serial
- Asegura atomicidad en el procesamiento de transacciones concurrentes
- Asegura aislamiento en el procesamiento de transacciones concurrentes
- Asegura aislamiento en el procesamiento de transacciones monousuario

Al utilizar el protocolo basado en HDE. ¿Cuáles de estas condiciones realmente debe cumplir para que una transacción T1 puede escribir el dato D?

- $HDE(T1) > HL(D)$
- $HDE(T1) > HE(D)$

En un entorno concurrente donde se utiliza el protocolo basado en bitácora:

- Se requiere del uso de checkpoints de otra forma sería imposible recuperarse ante un error.
- Se requiere del uso de checkpoints con lista de transacciones activas, de otra forma sería imposible recuperarse ante un error.
- El fallo de una transacción puede llevar a un retroceso en cascada de otras transacciones.

Cuales de estos problemas de concurrencia se pueden dar a causa de la interferencia entre transacciones

- Inanición
- Deadlock
- actualización perdida
- dependencia no confirmada
- análisis inconsistente

Dadas dos instrucciones X1 y X2, correspondientes de las transacciones T1 y T2 respectivamente. T1 y T2 se ejecutan concurrentemente:

- X1 y X2 están en conflicto si operan sobre los mismos datos y ambas son escrituras.
- X1 y X2 están en conflicto si operan sobre los mismos datos y ambas son lecturas.
- X1 y X2 están en conflicto si operan sobre los mismos datos y una de ellas es de escritura y otra de lectura.

La ocurrencia de inanición de transacciones en un ambiente monousuario:

- Nunca puede ocurrir.

Cuando se utiliza el método de bitácora con modificación diferida, para asegurar la integridad de la BD:

- En la bitácora no es necesario registrar el commit de la transacción.
- En la bitácora no es necesario registrar el valor nuevo del dato a modificar.
- En la bitácora no es necesario registrar el valor viejo del dato a modificar.

Una transacción que alcanza el estado de cometida:

- Pudo haber fallado durante su ejecución.
- Alcanzo el estado de parcialmente cometida.
- Pudo haber abortado durante su ejecución.

Sea Ti y Tj transacciones. Al armar un grafo de precedencia de transacciones como prueba de seriabilidad, existirá una arista dirigida de Ti hacia Tj si:

- Ti ejecuta WRITE(Q) antes que Tj un WRITE(Q).
- Ti ejecuta READ(Q) antes que Tj un WRITE(Q).
- Ti ejecuta WRITE(Q) antes que Tj un READ(Q).
- Ti ejecuta READ(Q) antes que Tj un READ(Q).

Sea una transacción cuya HDE es 1:28, si quiere acceder a un dato Q cuya hora de ultima lectura es 1:26 y de ultima escritura es 1:32, entonces:

- T puede leer Q
- T no puede leer Q
- T puede escribir Q
- T no puede escribir Q

Teniendo en cuenta la traza de ejecución que se muestra a la derecha, indique cuales de las sentencias son verdaderas:

```
T0 Start
T0 A, 19
T0 B, 79
Checkpoint
T0 Commit
```

- Utiliza el protocolo de bitácora con modificación inmediata
- Utiliza el protocolo de bitácora con modificación diferida
- La traza de ejecución no es válida. (es por el checkpoint el parcial sabio tiene la verdad)

¿Cuáles de las siguientes son desventajas del método de doble paginación?

- Produce fragmentación de los datos debido a los constantes cambios de ubicación de los mismos
- Es necesario un “Garbage Collector” para manejar las páginas que quedan sin referenciar.
- Existe una sobrecarga, ya que en un momento dado puede haber mas de una copia de la misma página.
- División de la BD en páginas
- Complicada en ambientes concurrentes / distribuidos

Los puntos de verificación (checkpoints) en un entorno concurrente:

- Indica cual pagina debe utilizarse cuando se emplea el método de doble paginación
- Indica desde donde se debe revisar la bitácora
- Son imprescindibles para rehacer (REDO) o deshacer (UNDO) el trabajo efectuado por una transacción

La seguridad de una base de datos es un concepto definible a nivel:

- Físico
- Sistema Operativo
- Humano
- SGBD
- Admin BD

La técnica de bitácora en transacciones:

- Garantiza el aislamiento en las transacciones
- Garantiza que la transacción no va a tener fallas (pipu y fede bancan este)
- Garantiza la atomicidad de la transacción (este es el que supuestamente esta bien en el parcial)