Base de Datos II

Unidad 2 – Seguridad e Integridad de BD

Objetivos

- Resguardo de BD
- Conceptos Básicos. Seguridad e integridad
- Control de acceso.
- Control de acceso y bloqueos.
- Aislamiento de Transacciones.
- Seguridad: Cifrado de Datos.

Backup: Características

Una **copia de seguridad** o ***backup***, en informática es una copia de información con el fin de que estas copias adicionales puedan utilizarse para restaurar el original después de una eventual pérdida de datos.

Fundamentalmente son útiles para dos cosas:

- Recuperarse de una catástrofe informática.
- Recuperar una pequeña cantidad de archivos que pueden haberse eliminado accidentalmente o corrompido.

La pérdida de datos es muy común: El 66% de los usuarios de internet han sufrido una seria pérdida de datos.

Backup BD: Tipos

- Backup Completo (Full): Guarda todos los archivos que sean especificados al tiempo de ejecutarse el respaldo. En el caso que se trate de una BD, realiza el respaldo completo: estructura, datos, procedimientos, triggers, etc.
- Backup Diferencial: Se guardan todos los cambios realizados desde el ultimo Backup Completo.
- Backup Logs: Se guardan las transacciones.

Estos tipos de Backup son implementados en formas diversas por cada DBMS.

Backup BD: Ventajas y desventajas

Tipo Backup	Ventajas	Desventajas
Full	Con este respaldo únicamente es posible recuperar toda la información	Tiempo de Ejecución
Diferencial	Sólo requiere del último Respaldo Completo y del último respaldo Diferencial. Velocidad	Incrementa su tamaño y tiempo cuando mas alejado se ejecute de un respaldo completo
Logs	Permite que la BD vuelva a un estado consistente. Velocidad	---

Duplicado de Información en línea

RAID ("Redundant Array of Inexpensive Disks") en términos sencillos es: un conjunto de 2 o más "Discos Duros" que operan como grupo y logran ofrecer una forma más avanzada de respaldo ya que:

- Es posible mantener copias *en línea* ("Redundancy").
- Agiliza las operaciones del Sistema.
- El sistema es capaz de recuperar información sin intervención de un Administrador.

Configuraciones Tipo RAID

- RAID-0: En esta configuración cada archivo es dividido ("Striped") y sus fracciones son colocadas en diferentes discos. Este tipo de implementación sólo agiliza el proceso de lectura de archivos, pero en ningún momento proporciona algún tipo de respaldo ("redundancy").
- RAID-1: este es el primer tipo de RAID que otorga cierto nivel de respaldo; cada vez que se vaya a guardar un archivo en el sistema éste se copiara *integro* a DOS discos (en linea), es por esto que RAID-1 también es llamado "Mirroring". También agiliza la lectura de archivos (si se encuentran ocupadas las cabezas de un disco "I/O") ya que otro archivo puede ser leído del otro disco y no requiere esperar a finalizar el "I/O" del primer disco.

Configuraciones Tipo RAID

- RAID-3: Esta configuración al igual que RAID-0 divide la información de todos los archivos ("Striping") en varios discos, pero ofrece un nivel de respaldo que RAID-0 no ofrece. *RAID-3* opera con un disco llamado "de paridad" ("parity disk"). Este "disco de paridad" guarda fracciones de los archivos necesarias para recuperar toda su Información, con esto, es posible reproducir el archivo que se perdió a partir de esta información de paridad.
- RAID-5: El problema que presenta RAID-3 es que el "disco de paridad" es un punto critico en el sistema; que ocurre si falla el disco de paridad ? *RAID-5*, no solo distribuye todos los archivos en un grupo de discos ("Striping"), sino también la información de paridad es guardada en todos los discos del sistema ("Striping"). Este configuración RAID suele ser usada en sistemas que requieren un *"alto nivel"* de disponibilidad.

Implementacion Backup

- **Horarios** Programar un horario de ejecución de las copias de seguridad aumenta considerablemente su efectividad y nivel de optimización.
- **Autentificación** Sobre el curso de operaciones regulares, las cuentas de usuario y/o los agentes del sistema que representan la copia de seguridad necesitan ser autentificados a cierto nivel. Utilizar un mecanismo de autentificación es una buena manera de evitar que el esquema de la copia de seguridad sea usado por actividades sin autorizar.
- **Cadena de confianza:** Los soportes de almacenamiento portátiles son elementos físicos y deben ser gestionados sólo por personas de confianza. Establecer una cadena de confianza individual es crítico para defender la seguridad de los datos.

Implementacion Backup

- **Validación de copias de seguridad** El proceso por el cual los dueños de los datos pueden obtener información considerando como fueron copiados esos datos. El mismo proceso es también usado para probar conformidad para los cuerpos reguladores fuera de la organización. Terrorismo, complejidad de datos, valores de datos y aumento de la dependencia sobre volúmenes de datos crecientes, todos contribuyen a una ansiedad alrededor y dependencia sobre copias de seguridad satisfactorias.
- **Reportando** En configuraciones más largas, los reportes son útiles para monitorizar los medios usados, el estado de dispositivos, errores, coordinación de saltos y cualquier otra información sobre el proceso de copia de seguridad.
- **Registrando** En suma a la historia de los reportes generados por el ordenador, actividades y registros de cambio son útiles para así entender mejor la copia de seguridad.
- **Verificación** Muchos programas de copia de seguridad hacen uso de *Sumas de verifación* o *hashes*. Esto ofrece muchas ventajas. Primero, estos permiten a la integridad de los datos ser verificados sin hacer referencia al archivo original: si el archivo guardado en un medio de copia tiene la misma suma de verificación que el valor salvado, después es muy probable que sea correcto.

Conceptos Básicos: Seguridad e Integridad

La seguridad en Bases de datos frecuentemente se asocia con la integridad, pero ambos conceptos son bastante diferentes. La seguridad de los datos refiere a la protección de los datos contra su revelación, su alteración o destrucción no autorizada, mientras que la integridad se refiere a la precisión o validez de esos datos.

- Seguridad significa proteger los datos ante usuarios no autorizados.
- Integridad significa protegerlos de usuarios autorizados.

Seguridad – Consideraciones generales

Existen muchos aspectos sobre el problema de la seguridad, entre ellos se encuentran:

- Aspectos legales, sociales y éticos. (La persona que hace la petición ¿tiene derecho legal para conocer la información solicitada?) En Argentina: La ley 24.766 protege la información confidencial a través de acciones penales y civiles, considerando información confidencial aquella que cumple los siguientes puntos:
  - Es secreta en el sentido que no sea generalmente conocida ni fácilmente accesible para personas introducidas en los círculos en que normalmente se utiliza el tipo de información.
  - Tenga valor comercial para ser secreta.
  - Se hayan tomado medidas necesarias para mantenerla secreta, tomadas por la persona que legítimamente la controla.
  - Por medio de esta ley la sustracción de disquetes, acceso sin autorización a una red o computadora que contenga información confidencial será sancionado a través de la pena de violación de secretos.
- Controles físicos. (El lugar donde se encuentra la computadora ¿esta bajo llave o tiene alguna otra protección?).
- Cuestiones de política (Si se utiliza un esquema de contraseñas ¿Cómo se las mantiene en secreto? ¿Con que frecuencia son cambiadas?).
- Controles de hardware (¿La unidad de procesamiento posee características de seguridad como claves de protección de almacenamiento o un modo de operación protegido?).
- Soporte del sistema operativo. (¿El sistema operativo borra el contenido de la memoria principal y los archivos de disco cuando termina de utilizarlos?).

Seguridad de un DBMS

Actualmente, los DBMS soportan uno o ambos enfoques con respecto a la seguridad de los datos.

- Control discrecional: un usuario específico tendrá generalmente diferentes derechos de acceso (privilegios) sobre diferentes objetos. Los esquemas discrecionales son muy flexibles.
- Control Obligatorio: cada objeto de datos esta etiquetado con un nivel de clasificación determinado y a cada usuario se le da un nivel de acreditación. Un objeto de datos especifico solo puede ser accedido por los usuarios que tengan el nivel de acreditación adecuado. Los esquemas obligatorios tienden a ser jerárquicos y comparativamente rígidos.

Control de acceso discrecional

- Se necesita un lenguaje que soporte la definición de las restricciones de seguridad discrecionales. Generalmente se define que esta permitido en lugar de lo que no está, por lo tanto los lenguajes generalmente soportan la definición no de las restricciones de seguridad como tales, sino de las autorizaciones.

En MySQL:

```
GRANT priv_type [(column_list)] [, priv_type [(column_list)] ...]
ON {tbl_name | * | *.* | db_name.*}
TO user_name [IDENTIFIED BY 'password']
[, user_name [IDENTIFIED BY 'password'] ...]
[WITH GRANT OPTION]

REVOKE priv_type [(column_list)] [, priv_type [(column_list)] ...]
ON {tbl_name | * | *.* | db_name.*}
FROM user_name [, user_name ...
```

- La granularidad permitida para el control de acceso discrecional, dependerá siempre de cada DBMS.

Control de acceso obligatorio

Los controles de acceso obligatorio son aplicables a las BD en la que los datos tienen una estructura bastante estática y rígida.

- Cada dato tiene un nivel de clasificación y cada usuario tiene un nivel de acreditación.
- El usuario *j* puede recuperar el objeto *i* solo si el nivel de acreditación de *j* es igual o mayor que el nivel de clasificación de *i*. (Propiedad de seguridad simple).
- El usuario *j* puede actualizar el objeto *i* solo si el nivel de acreditación de *j* es igual al nivel de clasificación de *i*. (Propiedad de estrella)
- Podemos pensarlo como una columna mas en cada tupla, en cada tabla, donde se coloca el nivel de clasificación del dato: 1, 2, 3, 4 o 5. Cada vez que se hace un una operación el DBMS agrega en la consulta: SELECT …. WHERE C= 'k' AND CLASE <= *acreditación del usuario*

Concurrencia

- Los DBMS permiten que muchas transacciones accedan a una misma BD a la vez. Para poder mantener la integridad de los datos, se necesita un mecanismo de control de concurrencia para asegurar que las transacciones concurrentes no interfieran entre si.
- Los tres problemas comunes de la concurrencia son:
  - El problema de la actualización perdida.
  - El problema de la dependencia no confirmada.
  - El problema del análisis inconsistente.

Concurrencia- Actualización perdida

- La transacción *A* recupera alguna tupla t en el tiempo *t1*.
- La transacción *B* recupera la misma tupla *t* en el tiempo *t2*.
- La transacción *A* actualiza la tupla *t* en el tiempo *t3*.
- La transacción *B* actualiza la tupla *t*, en el tiempo *t4*.
- La actualización realizada por la transacción A se pierde en el tiempo *t4*, ya que la transacción *B* la sobreescribe.

Concurrencia – Dependencia no confirmada

- La transacción *A* ve una actualización no confirmada en el tiempo *t2*.
- Esta actualización es posteriormente deshecha en el tiempo *t3.*
- La transacción *A* esta operando sobre la suposición falsa de que la tupla *t* tiene el valor visto en el tiempo *t2*, siendo que tiene el valor que tenía en el tiempo *t1*.
- La transacción *A* producirá una salida incorrecta.

Concurrencia – Análisis Inconsistente

- La transacción *A* esta sumando saldos de cuentas.
- La transacción *B* esta transfiriendo una cantidad *X* a la cuenta *Y*.
- El resultado producido por *A*, es obviamente incorrecto, y si *A* continuara, y escribiera ese dato en la BD, dejaría en efecto, la BD en estado inconsistente.
- Decimos que *A* vio un estado inconsistente de la BD, y por lo tanto, ha realizado un análisis inconsistente

Bloqueos

- Todos los problemas antes descriptos, pueden ser resueltos por medio de una técnica de control de concurrencia llamada bloqueo.
- **Definición:** cuando una transacción debe asegurarse de que algún objeto en el que esta interesado no cambiará de ninguna forma mientras lo esta usando, adquiere un bloqueo sobre ese objeto. De esta forma, la transacción tiene la certeza que el objeto permanecerá estable durante el tiempo que lo desee.

Tipos de bloqueo

- Bloqueo Exclusivo: (o de escritura X) Si una transacción pone un bloqueo exclusivo sobre la tupla *t*, entonces se rechazará una petición de cualquier otra transacción para un bloqueo de cualquier tipo sobre la tupla *t*.
- Bloqueo compartido: (o de lectura S) Si una transacción pone un bloqueo compartido sobre la tupla *t*, entonces:
  - Se rechazará la petición de cualquier transacción para un bloqueo exclusivo sobre *t*.
  - Se otorgará la petición de cualquier transacción para un bloqueo compartido sobre *t*.

Matriz de compatibilidad de tipos de bloqueo

	Exclusivo (X)	Compartido (S)	-
Exclusivo (X)	No	No	Si
Compartido (S)	No	Si	Si
-	Si	Si	Si

Un guion indica que no hay bloqueos (-)
Un No, indica conflictos (la petición no sera satisfecha y la transacción pasa a un estado de espera, y un Si indica compatibilidad.
Para garantizar que no ocurran los problemas de concurrencia, se tiene un protocolo de acceso a los datos o protocolo de bloqueo:

- Una transacción que desea recuperar una tupla debe primero adquirir un bloqueo S sobre esa tupla.
- Una transacción que desea actualizar una tupla, debe primero adquirir un bloqueo de tipo X sobre esa tupla. Si ya tiene un bloque S sobre la tupla, debe modificar ese bloqueo hacia el nivel X.
- Si una petición de bloqueo es rechazada porque entra en conflicto con un bloqueo ya existente, la transacción pasa a un estado de espera y permanecerá así hasta que la 1º transacción libere el bloqueo.

Los tres problemas de concurrencia y su resolución

- *El problema de la actualización perdida:*
  - Para solucionarlo, podemos decir que cuando la transacción *A* recupera la tupla *t*, adquiere un bloqueo de tipo S, en el tiempo *t1*. Cuando la transacción *B* en el tiempo *t2* recupera la tupla *t*, tambien adquiere un bloqueo de tipo S. Cuando en el tiempo *t3*, *A* intenta actualizar la tupla, y para esto, intenta modificar su bloqueo a X, no es aceptado, ya que existe un bloqueo de *B*. Por la misma razón, en el tiempo *t4*, cuando *B* quiere promocionar su bloqueo al tipo X, queda en estado de espera por ser bloqueos incompatibles. Ahora ambas actualizaciones no pueden continuar, y no existe la posibilidad de perder alguna actualización. Pero ahora, tenemos otro problema que se llama ***bloqueo mortal***.

Los tres problemas de concurrencia y su resolución

- *El problema de la dependencia no confirmada:*
  - En este caso, cuando la transacción *A* en el tiempo *t2* quiera realizar un bloqueo S para visualizar la tupla *t*, no podrá hacerlo, ya que la transacción *B* tiene un bloqueo X, debido a que esta actualizando la misma tupla. Por lo tanto *A* quedará en espera hasta que *B* finalice su bloqueo, llegando a su termino la transacción *B (*ya sea con un COMMIT O ROLLBACK), y recién en este momento *A* podrá continuar. De esta manera, *A* verá un valor confirmado y no depende de una actualización no confirmada.

Los tres problemas de concurrencia y su resolución

- *El problema del análisis inconsistente:*
  - La transacción A, que esta realizando la suma de saldos, realiza un bloqueo S sobre las tuplas. Cuando *B* quiera insertar un nuevo movimiento bancario, no podrá adquirir un bloqueo, ya que *A* tiene un bloqueo S, incompatible con el bloque *X* que quiere promover B. La transacción B pasará a un estado de espera, hasta que A finalice y libere el bloqueo. Por lo tanto, el bloqueo resuelve el problema original.

Bloqueos – Abrazo Mortal

- Es una situación donde dos o mas transacciones se encuentran en estado simultáneos de espera, cada una esperando que alguna de las demás libere un bloqueo para poder continuar.
- La detección de un bloqueo mortal implica la detección de un ciclo en el grafo de espera. La ruptura del abrazo mortal implica seleccionar una de las transacciones bloqueadas mortalmente como victima, y entonces deshacerla liberando por lo tanto sus bloqueos y permitiendo que continúen las demás transacciones.
- En la práctica no todos los DBMS detectan los abrazos mortales, usan un mecanismo de tiempos y asumen que una transacción que no ha realizado ningún trabajo durante cierto periodo preestablecido, esta bloqueada mortalmente.

Seriabilidad

- Es el criterio de corrección aceptado comúnmente para la ejecución de un conjunto dado de transacciones.
- Se considera que la ejecución de un conjunto dado de transacciones es correcta cuando es *seriable*; cuando produce el mismo resultado que una ejecución serial de las mismas transacciones, ejecutando una a la vez.
  - Las transacciones individuales son tomadas como correctas: se da por hecho que transforman un estado correcto de la BD en otro estado correcto.
  - También es correcta la ejecución de una transacción a la vez en cualquier orden serial.
  - Una ejecución intercalada es correcta cuando equivale a alguna ejecución serial, es decir, es seriable.

Niveles de Aislamiento

Definición

- El grado de interferencia que una transacción dada es capaz de tolerar por parte de transacciones concurrentes.
- Si queremos garantizar la seriabilidad, no podemos aceptar ningún tipo de interferencia, es decir, el nivel de aislamiento deber ser el máximo imposible.
- En teoría se pueden definir 5 niveles de aislamiento, pero el SQL estándar define solo 4.

Bloqueo por Aproximación

Granularidad del bloqueo

- Hasta el momento, supusimos bloqueos a nivel de tuplas, pero se pueden bloquear unidades mas grandes o mas pequeñas. A esto lo llamamos granularidad del bloqueo.
- Definimos 3 nuevos niveles de bloque llamados bloqueos por aproximación, que tiene sentido sobre las tablas pero no sobre las tuplas.
  - Bloqueo de aproximación compartida (IS): bloqueos S sobre tuplas individuales de R.
  - Bloqueo de aproximación exclusiva (IX): Igual que IS, pero ademas puede actualizar tuplas individuales, por lo que adquiere bloqueos X sobre esas tuplas.
  - Bloqueo de aproximación compartida exclusiva (SIX): Combina S e IX, la transacción puede tolerar lectores concurrentes , pero no actualizadores concurrentes en R. Puede actualizar tuplas individuales en R y por lo tanto pondra bloqueos X sobre esas tuplas.

Matriz de compatibilidad de bloqueos por aproximación

	X	SIX	IX	S	IS	-
X	No	No	No	No	No	Si
SIX	No	No	No	No	Si	Si
IX	No	No	Si	No	Si	Si
S	No	No	No	Si	Si	Si
IS	No	Si	Si	Si	Si	Si
-	Si	Si	Si	Si	Si	Si

- Antes de que una transacción dada pueda adquirir un bloqueo S sobre una tupla dada deber adquirir un bloqueo IS o una mas fuerte sobre la tabla que contiene a esa tupla.
- Antes de que una transacción dada pueda adquirir un bloqueo X sobre una tupla dada, primero debe adquirir un bloqueo IX o uno mas fuerte sobre la tabla que contiene esa tupla.

Aislamiento de transacciones

- Los niveles posibles son: READ UNCOMMITTED, READ COMMITTED, REPETEABLE READ o SERIALIZABLE.
- El nivel predeterminado es SERIALIZABLE (el mas alto), pero se puede especificar uno menor.
- Si alguna transacción se efectúa a un nivel menor de aislamiento, la seriabilidad podría ser violada.
- Se definen tres formas específicas en las que se puede violar la seriabilidad:
  - Lectura sucia.
  - Lectura no repetible.
  - Fantasmas.

Lectura sucia

- La transacción T1 realiza una actualización sobre alguna fila, la transacción T2 recupera esa fila y la transacción T1 termina con una transacción deshacer. La transacción T2 ha visto una fila que ya no existe, y que en cierto sentido nunca existió.

Lectura no repetible

- T1 recupera una fila, luego T2 actualiza esa fila y después la transacción T1 recupera nuevamente la misma fila. La transacción T1 ha recuperado la misma fila dos veces, pero ve dos valores diferentes en ella.

Fantasmas

- La transacción T1 recupera el conjunto de las filas que satisfacen alguna condición. La transacción T2 inserta entonces una nueva fila que satisface la misma condición. Si la transacción T1 repite ahora su petición de recuperación, vera una fila que antes no existía, un “fantasma”.

Niveles de Aislamiento de SQL

Nivel de Aislamiento	Lectura Sucia	Lectura no repetible	Fantasmas
READ UNCOMMITED	SI	SI	SI
READ COMMITED	NO	SI	SI
REPEATABLE READ	NO	NO	SI
SERIALIZABLE	NO	NO	NO

Un SI indica que puede ocurrir la violación indicada.
Podemos decir que la única forma de evitar los fantasmas es bloquear la ruta de acceso que se utiliza para acceder a los datos en consideración. Este bloqueo puede ser a nivel de un índice.

Seguridad – Cifrado de Datos

- Cuando hablamos de seguridad, también tenemos que tener en cuenta al usuario que trata de causar daño sobre el sistema (por ejemplo: eliminando físicamente parte de la BD o interviniendo una línea de comunicación).
- La medida mas efectiva contra este tipo de amenazas es el cifrado de datos (o encriptación), que implica guardar y transmitir los datos sensibles en forma cifrada.

Cifrado de datos

- Un texto plano es cifrado sometiéndolo a un algoritmo de cifrado.
- Los detalles del algoritmo de cifrado son públicos, pero la clave de cifrado se mantiene en secreto.
- El texto cifrado, que debe ser ininteligible para cualquiera que no posea la clave de cifrado, es lo que se guarda en la BD o se transmite por la línea de comunicación.
- El esquema de cifrado debería ser tal, que el trabajo involucrado en romperlo sobrepasara cualquier ventaja potencial que pudiera obtenerse al hacerlo.
- *Fundamental en seguridad: El costo de romperla sea significativamente mayor que el beneficio potencial.*

Estándar de cifrado de datos

- Sustitución: se usa una clave de cifrado para determinar para cada carácter del texto plano un carácter de texto cifrado que va a sustituir a ese carácter.
- Permutación: los caracteres del texto plano son simplemente reorganizados en una secuencia diferente.
- Los algoritmos que usan una combinación de ambos, brindan un alto grado de seguridad.
- Uno de estos algoritmos es el Estándar de Cifrado de Datos (DES), desarrollado por IBM y adoptado como estándar federal de los EU.

Cifrado de Clave Pública

En el esquema de clave pública, tanto el algoritmo de cifrado como la clave de cifrado están disponibles y por lo tanto, cualquier persona puede convertir texto plano en texto cifrado. Pero la ***clave de descifrado*** correspondiente se mantiene en secreto. Todo esquema de clave pública involucra dos claves, una para el cifrado y otra para el descifrado.

Además, la clave de descifrado no puede ser deducida de manera realista a partir de la clave de cifrado.

Esquema RSA de clave pública

- Existe un algoritmo rápido para determinar si un numero dado es primo.
- No existe ningún algoritmo rápido para encontrar los factores primos de un número dado compuesto (es decir, no primo) dado.

El esquema RSA funciona de la siguiente forma:

1. Se seleccionan al azar dos números primos grandes, *p* y *q*, diferentes, y se calcula $r = p * q$.
2. Se selecciona al azar un número entero grande, *e*, que sea primo relativo (no tiene factores en común con respecto al producto $(p - 1) * (q - 1)$. El número *e* es la clave de cifrado.
3. Se toma la clave de descifrado, *d*, para que sea el "multiplicativo inverso" único de *e* modulo $(p - 1) * (q - 1)$, es decir $d * e = 1 \text{ modulo } (p - 1) * (q - 1)$
4. Se multiplican los enteros *r* y *e*, pero no *d*.
5. Para cifrar un texto plano *P* (por simplicidad, suponemos que es un entero menor que r), lo reemplazamos por el texto cifrado *C*, el cual se calcula : $C = P^e \text{ modulo } r$.
6. Para descifrar una parte del texto cifrado *C* lo reemplazamos por el texto plano *P*, que se calcula de la siguiente forma: $P = C^e \text{ modulo } r$.

Esquema RSA de clave pública

Este esquema funciona: el descifrado de C usando d, recupera el P original. Sin embargo calcular d, conociendo únicamente r y e (y no p o q) no es factible (o es demasiado costoso). Por lo tanto, cualquier persona puede cifrar texto plano, pero solo los usuarios autorizados (que tiene d) pueden descifrar el texto cifrado.

Ejemplo RSA

Hagamos p=3, q=5, entonces r=15. El producto (p-1)*(q-1)=8. Hagamos e=11 (un numero primo mayor que p y q). Para calcular d hacemos:

d * 11 = 1 modulo 8 , por lo tanto d = 3.

El texto plano P consiste en el entero 13. Entonces el texto cifrado es:

$C = P^e$ modulo r

$=13^{11}$ modulo 15

=1.792.169.394.037 modulo 15

=7

Ahora el texto plano P original esta dado por:

$P = C^d$ modulo r

$= 7^3$ modulo 15

= 343 modulo 15

= 13

Como e y d, son inversos entre si, los esquemas de cifrado de clave pública también permiten que los mensajes cifrados sean firmados, de forma tal que el receptor pueda estar seguro que el mensaje se origino en la persona que dice haberlo hecho