

TAREA OPCIONAL BASE DE DATOS



INDICE

1. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma base de datos MySQL en dos terminales distintos, y que autocommit=0.....4
2. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma basededatos MySQL en dos terminales distintos, y que autocommit=0.....5
3. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma base de datos MySQL en dos terminales distintos, y que autocommit=0.....6

1. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma base de datos MySQL en dos terminales distintos, y que autocommit=0.

Dos terminales ejecutando comandos simultáneamente

Casos:

caso1: a b c d

a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'
b inicia una transacción en terminal2
c actualiza el precio a 500 dentro de la transacción
d realiza commit en terminal2

Resultado: Precio final es 500 (transacción b c d sobrescribe a).

caso2: a b c e

a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'
b inicia una transacción en terminal2
c actualiza el precio a 500 dentro de la transacción
e realiza rollback en terminal2

Resultado: Precio final es 1000 (a es la única sentencia que realiza commit).

caso4: b c d a

b inicia una transacción en terminal2
c actualiza el precio a 500 dentro de la transacción
d realiza commit en terminal2
a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'

Resultado: Precio final es 1000 (a sobrescribe b c d).

caso5: b c e a

b inicia una transacción en terminal2
c actualiza el precio a 500 dentro de la transacción
e realiza rollback en terminal2
a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'

Resultado: Precio final es 1000 (a es la única sentencia que realiza commit).

caso6: b c a d

b inicia una transacción en terminal2
c actualiza el precio a 500 dentro de la transacción
a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'
d realiza commit en terminal2

Resultado: Precio final es 500 (b c d se confirma después de a).

caso7: b a c d

b inicia una transacción en terminal2
a ejecuta update mascotas set precio=1000 donde tipo='perro' y raza='pitbull'
c actualiza el precio a 500 dentro de la transacción
d realiza commit en terminal2

Resultado: Precio final es 500 (b a c d sobrescribe a).

caso8: b a c e

b inicia una transacción en terminal2
a ejecuta update mascotas set precio=1000 where tipo='perro' and raza='pitbull'
c actualiza el precio a 500 dentro de la transacción
e realiza rollback en terminal2

Resultado: Precio final es 1000 (a es la única sentencia que realiza commit).

2. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma basededatos MySQL en dos terminales distintos, y que autocommit=0

Casos:

caso1: a e b f g d h

- a inicia una transacción en terminal1
- e inicia una transacción en terminal2
- b inserta un perro (caniche)
- f inserta un gato (siamés)
- g inserta un pájaro (loro)
- d realiza rollback en terminal1
- h realiza commit en terminal2

Resultado: 6 registros (2 inserciones de terminal2 se confirman).

caso2: a e b f g d i

- a inicia una transacción en terminal1
- e inicia una transacción en terminal2
- b inserta un perro (caniche)
- f inserta un gato (siamés)
- g inserta un pájaro (loro)
- d realiza rollback en terminal1
- i realiza rollback en terminal2

Resultado: 4 registros (ambos terminales realizan rollback).

caso3: a b e f g d h

- a inicia una transacción en terminal1
- b inserta un perro (caniche) en terminal1
- e inicia una transacción en terminal2
- f inserta un gato (siamés)
- g inserta un pájaro (loro)
- d realiza rollback en terminal1
- h realiza commit en terminal2

Resultado: 6 registros (2 inserciones de terminal2 se confirman).

caso4: a b e f g c i

- a inicia una transacción en terminal1
- b inserta un perro (caniche) en terminal1
- e inicia una transacción en terminal2
- f inserta un gato (siamés)
- g inserta un pájaro (loro)
- c realiza commit en terminal1
- i realiza rollback en terminal2

Resultado: 5 registros (terminal1 confirma inserción, terminal2 hace rollback).

3. Suponiendo que las siguientes órdenes se ejecuten simultáneamente sobre la misma base de datos MySQL en dos terminales distintos, y que autocommit=0.

Teniendo en cuenta que debemos utilizar, al menos, tres sentencias de cada terminal.

caso1:

¿En qué orden se deberían ejecutar para que el peso del jugador con dni 67543328E sea 68kilos?

caso2:

¿En qué orden se deberían ejecutar para que el peso del jugador con dni 67543328E sea 54kilos?

caso3:

¿En qué orden se deberían ejecutar para que el peso del jugador con dni 67543328E no cambie?

Casos:

caso1: Peso del jugador sea 68 kilos

Orden: e f g h a

- e - Iniciar transacción en terminal2
- f - Actualizar peso a 54 en terminal2
- g - Actualizar peso a 68 en terminal2
- h - Commit en terminal2
- a - Iniciar transacción en terminal1
- b - Actualizar peso a 105 en terminal1
- d - Rollback en terminal1

caso2: Peso del jugador sea 54 kilos

Orden: e f h a b

- e - Iniciar transacción en terminal2
- f - Actualizar peso a 54 en terminal2
- h - Commit en terminal2
- a - Iniciar transacción en terminal1
- b - Actualizar peso a 105 en terminal1
- d - Rollback en terminal1

caso3: Peso del jugador no cambie

Orden: e f i a b

- a - Iniciar transacción en terminal1
- b - Actualizar peso a 105 en terminal1
- d - Rollback en terminal1
- e - Iniciar transacción en terminal2
- f - Actualizar peso a 54 en terminal2
- i - Rollback en terminal2