

INNODB LOCKING EXPLAINED WITH STICK FIGURES

Bill Karwin

October 29, 2015

Bill Karwin

- Senior Database Architect at School Messenger (West Corporation)
- Author of *SQL Antipatterns: Avoiding the Pitfalls of Database Programming*
- 20+ years experience with SQL databases, software development, MySQL consulting and training
-

why locking?



- When multiple clients access the same data, they have to avoid clobbering each others' work.
- Databases must restrict access to one client at a time for a given table or row.

why locking?

- The DBMS creates locks against tables and rows, and gives them to clients, first-come, first-serve.
- When a client requests a exclusive lock, but a different client currently holds it, the requestor waits until the holder releases its lock.
- Most locks last until the end of the transaction.

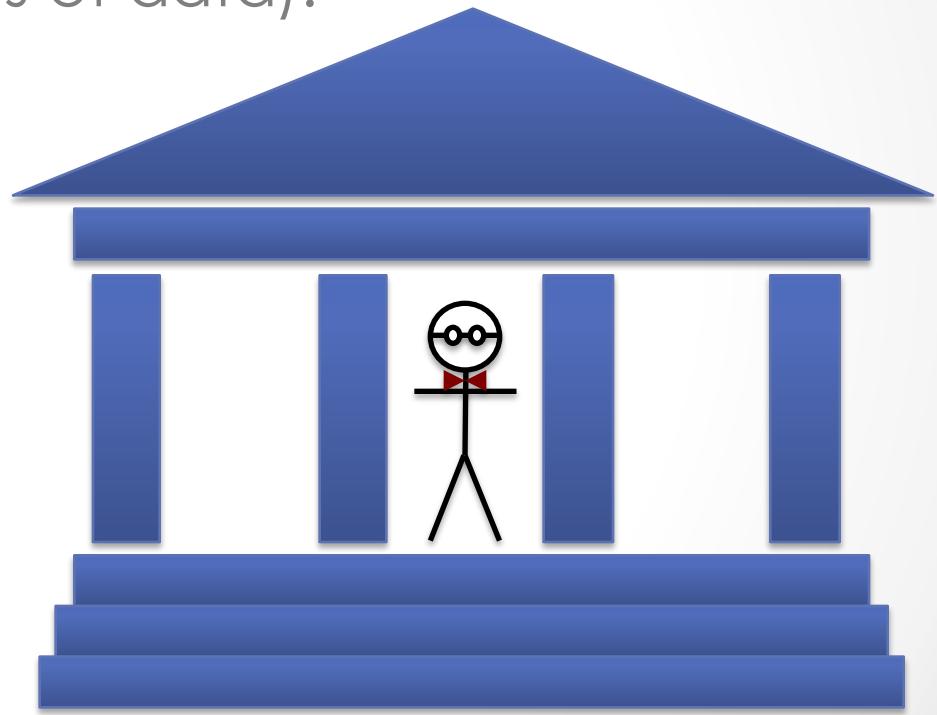
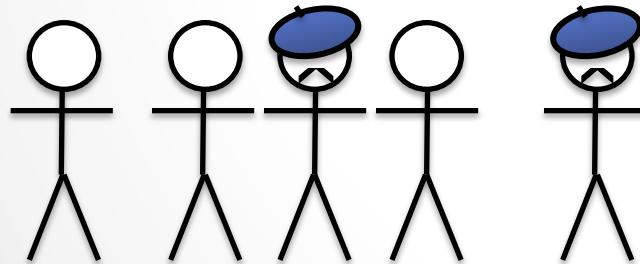


the analogy

...

a museum

- Many people can visit the museum (a database table) to view art (rows of data).

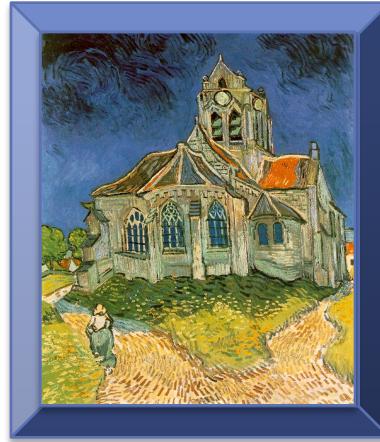
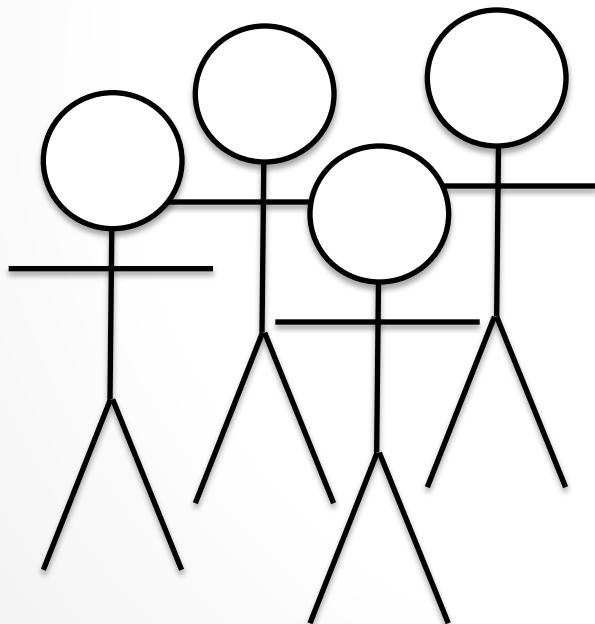


reads and writes

...

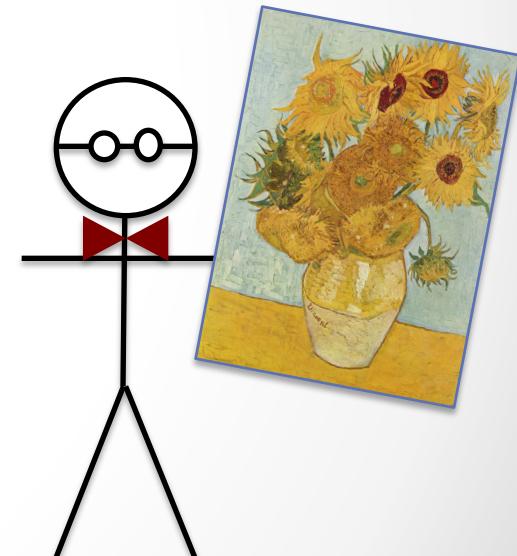
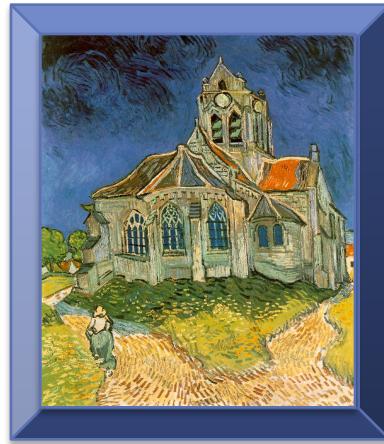
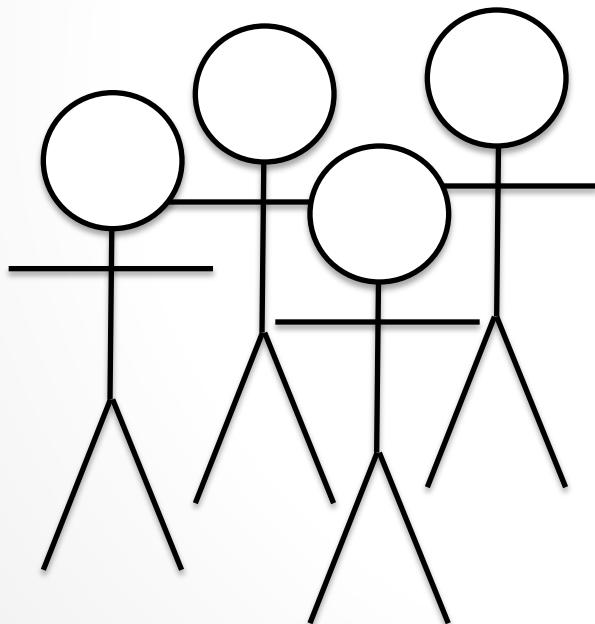
reads

- Many visitors can view paintings at the same time—no locking required.



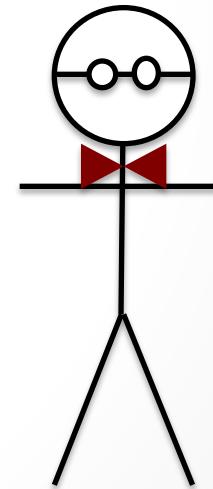
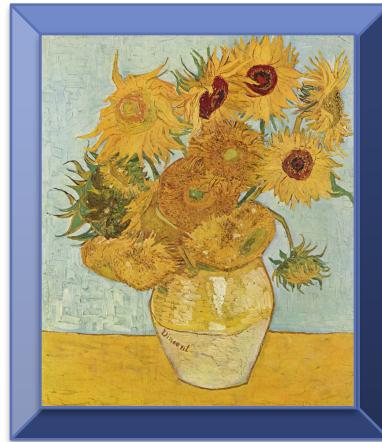
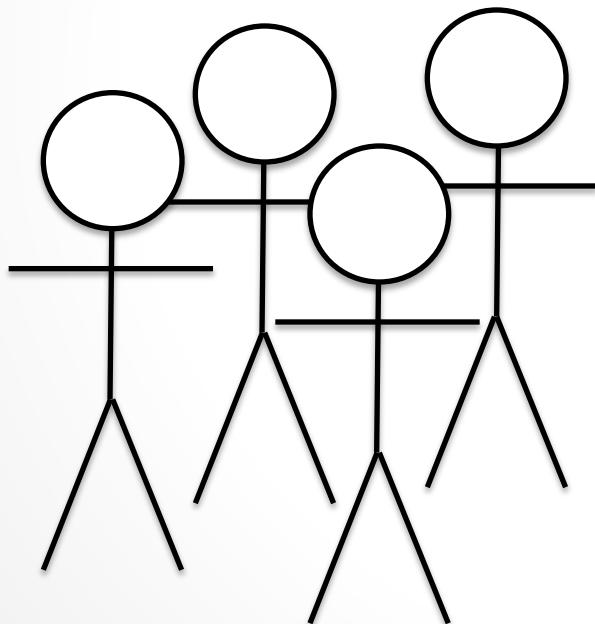
writes

- A curator can change the paintings—while casual visitors are viewing them.



writes

- A curator can change the paintings—while casual visitors are viewing them.



repeatable reads

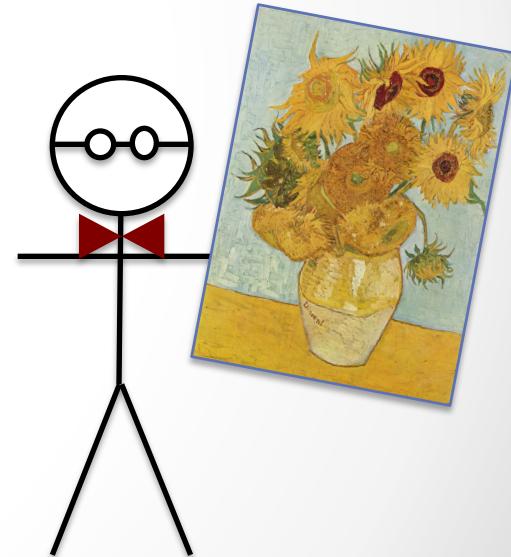
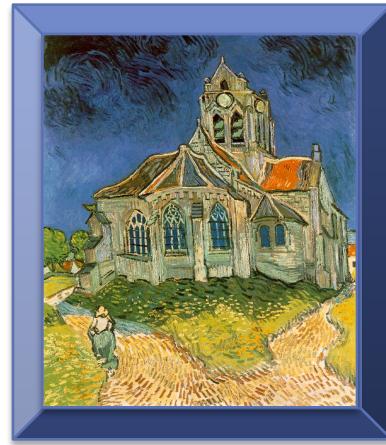
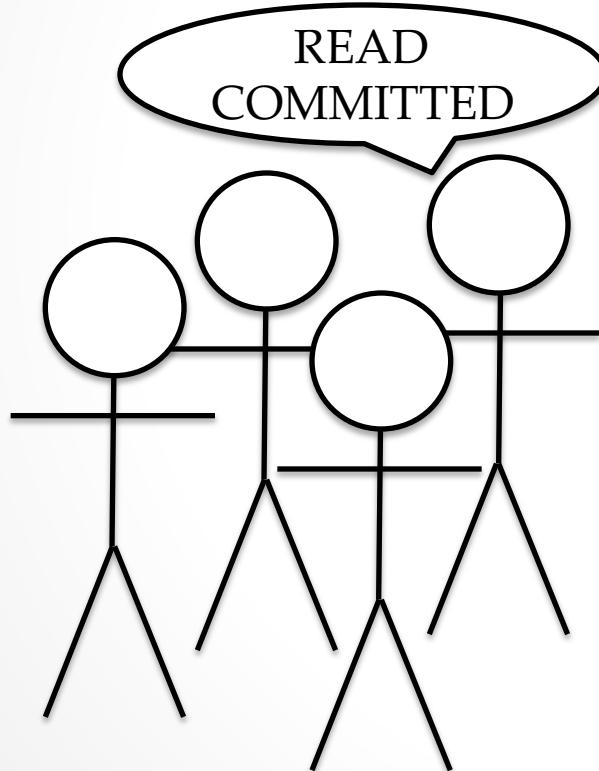
- The viewers still see the prior painting in spite of the change, because they used their tablets to capture the image.*



* Please do not take photographs of the art in a real museum.

read committed

- If the viewers are okay allowing their view to change, they can simply say so.



read committed

- If the viewers are okay allowing their view to change, they can simply say so.

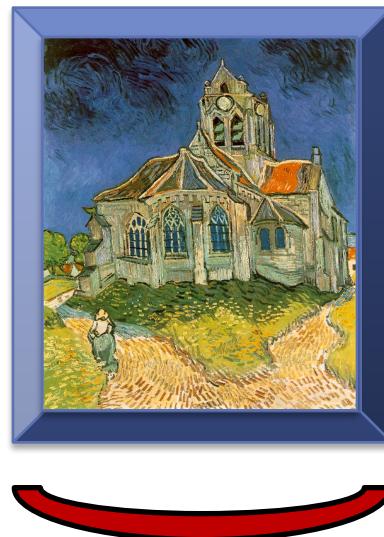
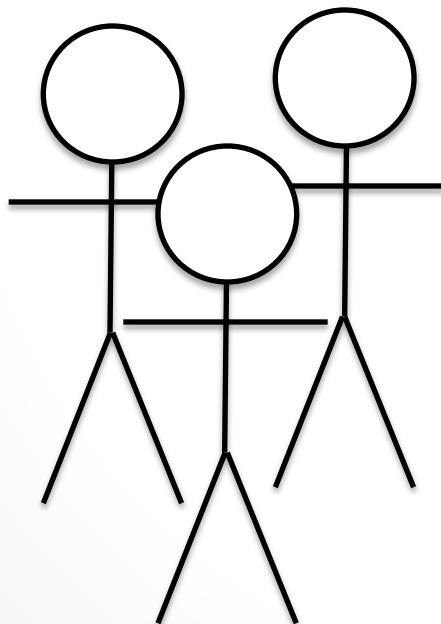


exclusive locks

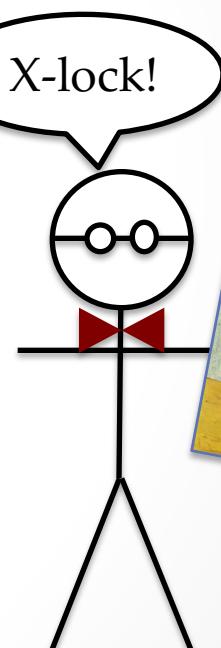
• • •

exclusive locks

- The curator can change the painting if they are the exclusive person working on it.

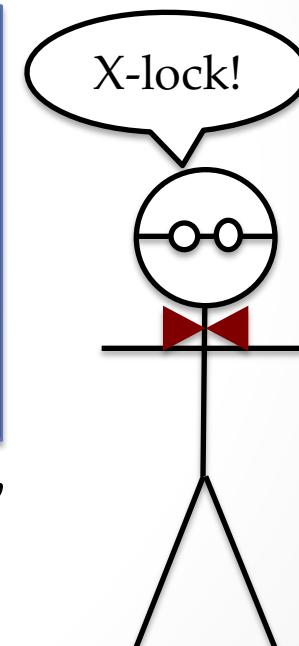
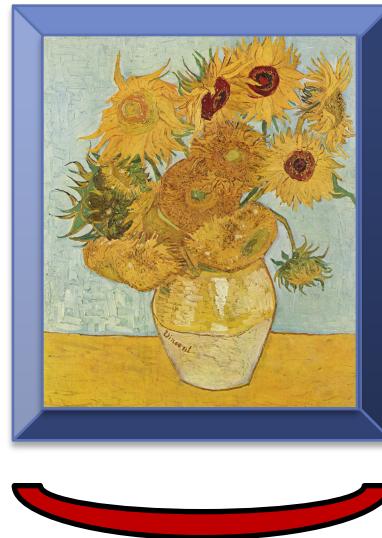
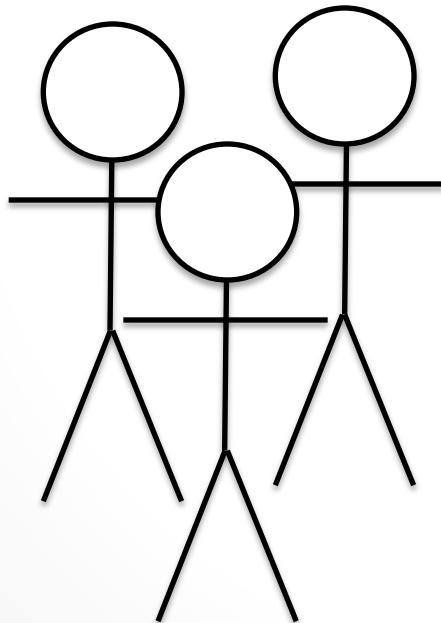


X-lock!



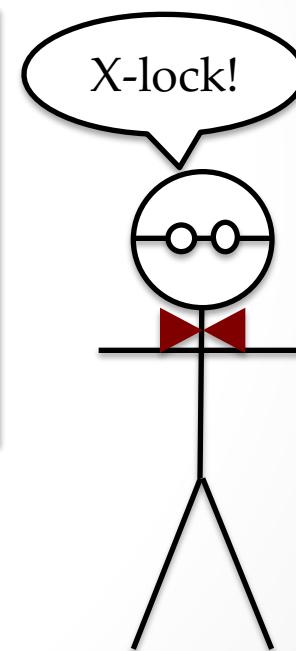
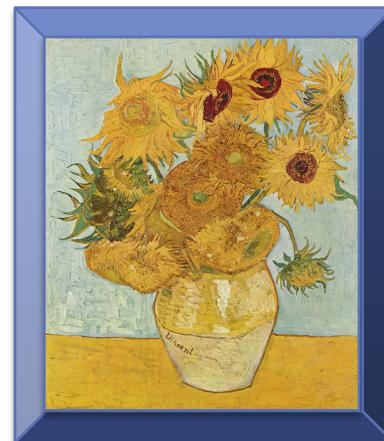
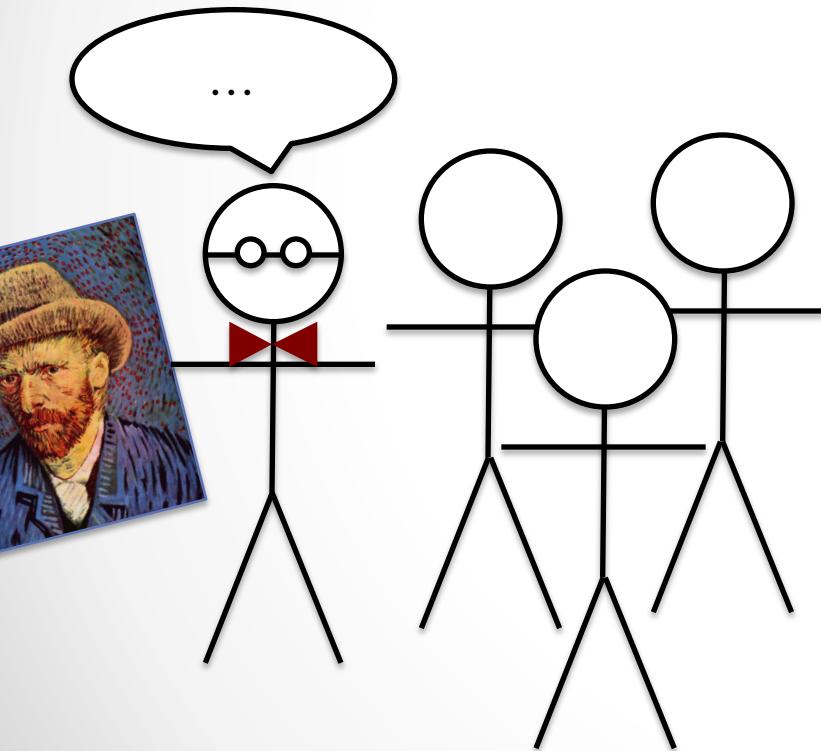
exclusive locks

- The curator can change the painting if they are the exclusive person working on it.



exclusive locks

- The curator can change the painting if they are the exclusive person working on it.



what is an exclusive lock?

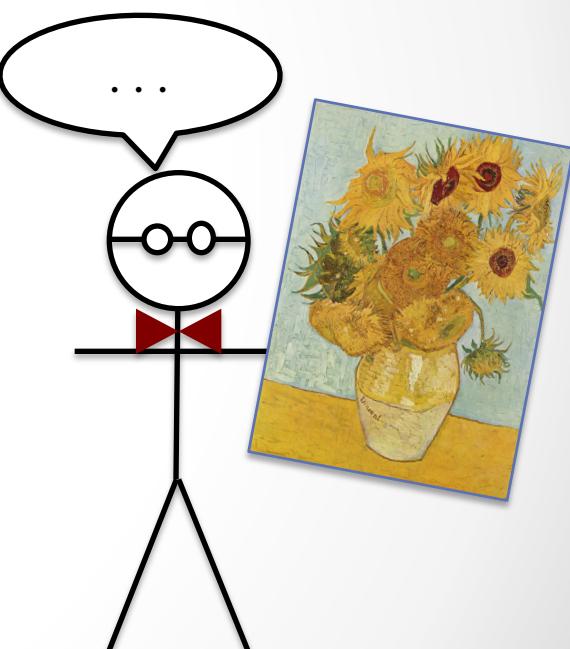
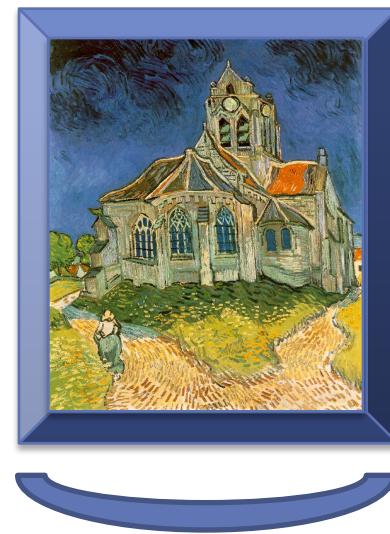
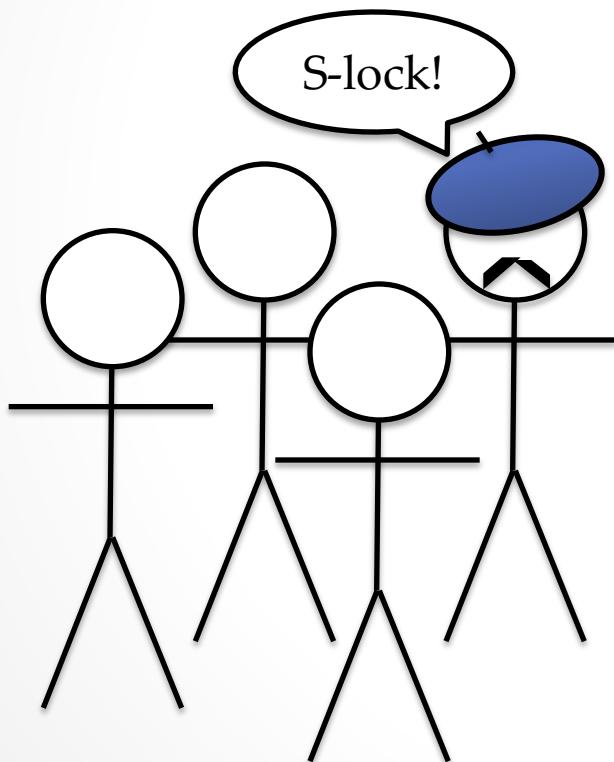
- A lock that does not share.
- It must be the only lock on the resource.
- Request for an exclusive lock waits for the release of any other shared or exclusive lock on that resource.

shared locks

...

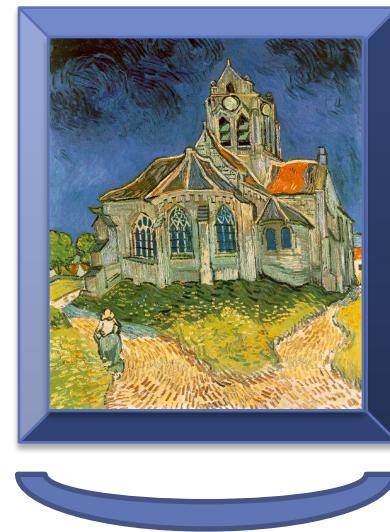
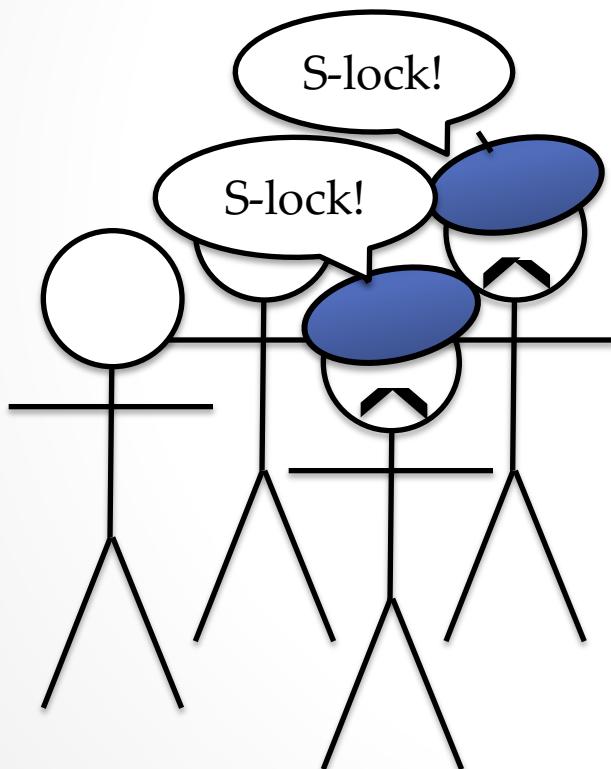
shared locks

- The curator cannot make changes while the art critic is viewing a painting.



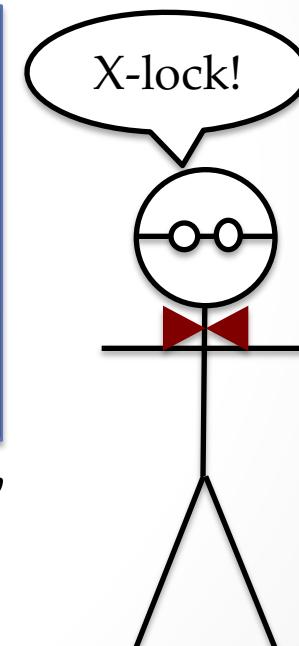
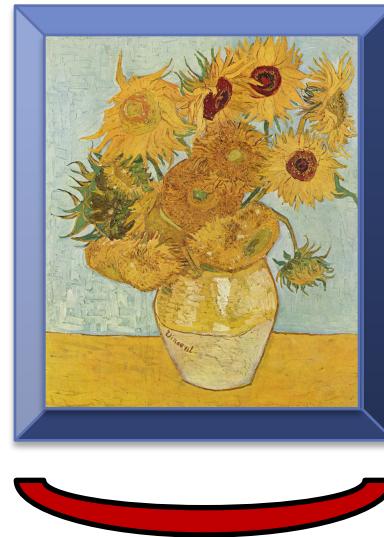
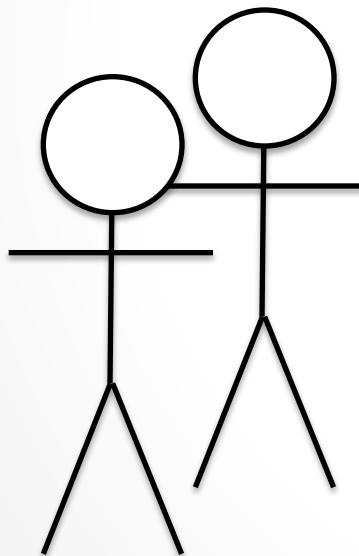
shared locks

- Art critics can share—they do not block each other.



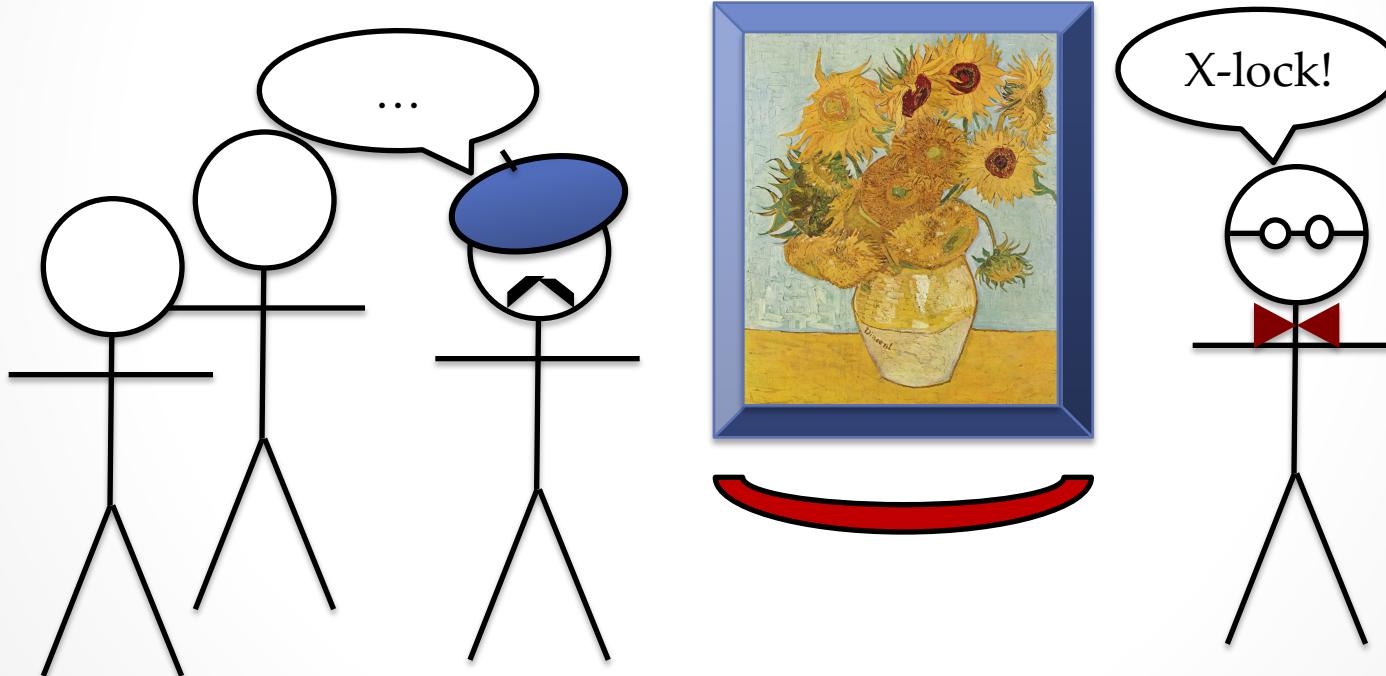
shared locks

- Once the art critics leaves, the curator can proceed to change the art.



shared locks

- A new art critic will not begin his viewing while the curator is still working on changing the painting.



what is a shared lock?

- A lock that allows other shared locks on the same table or rows.
- Shared locks blocks exclusive locks.
- Exclusive locks block shared locks.

table intention locks

...

table locks

- A construction worker needs to remodel the museum, but not while visitors are inside.

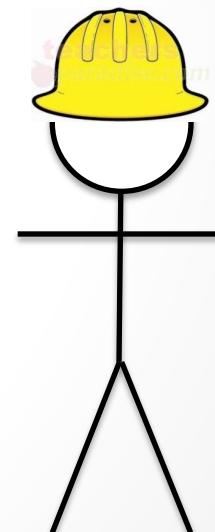
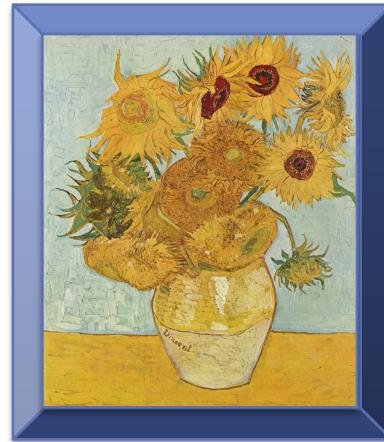
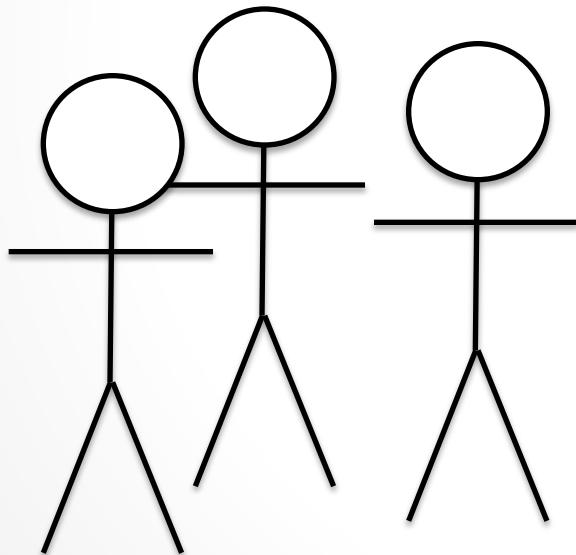


table locks

- Each visitor is given a special visitor's badge as they enter the museum, showing their *intention* to view the paintings.

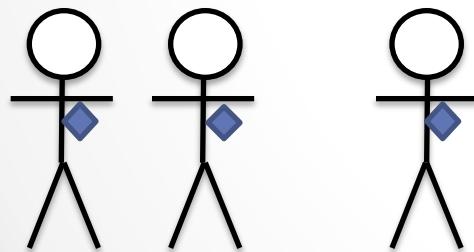


table locks

- A construction worker requests exclusive access to the museum, but can not get it.

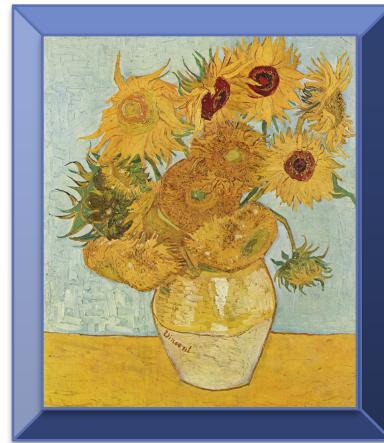
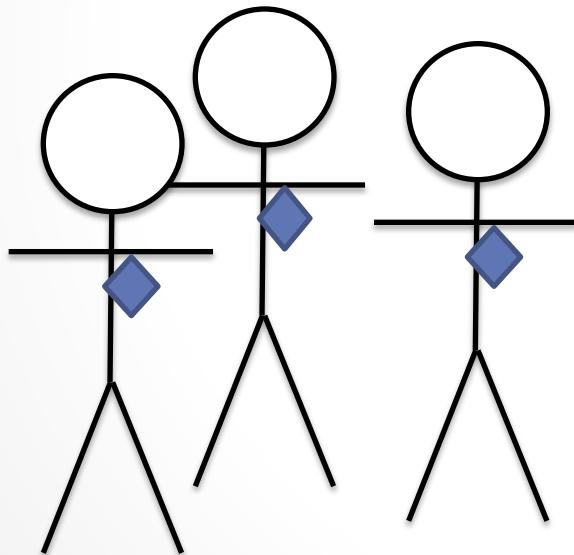


table locks

- When all the visitors have left, the worker can finally get his exclusive access to do his work.

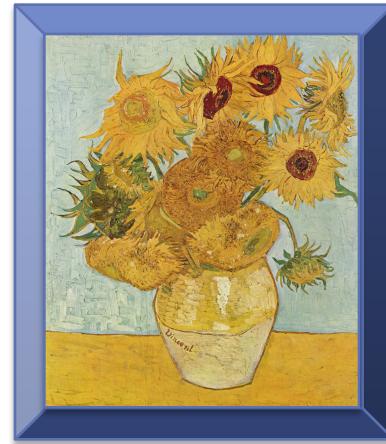
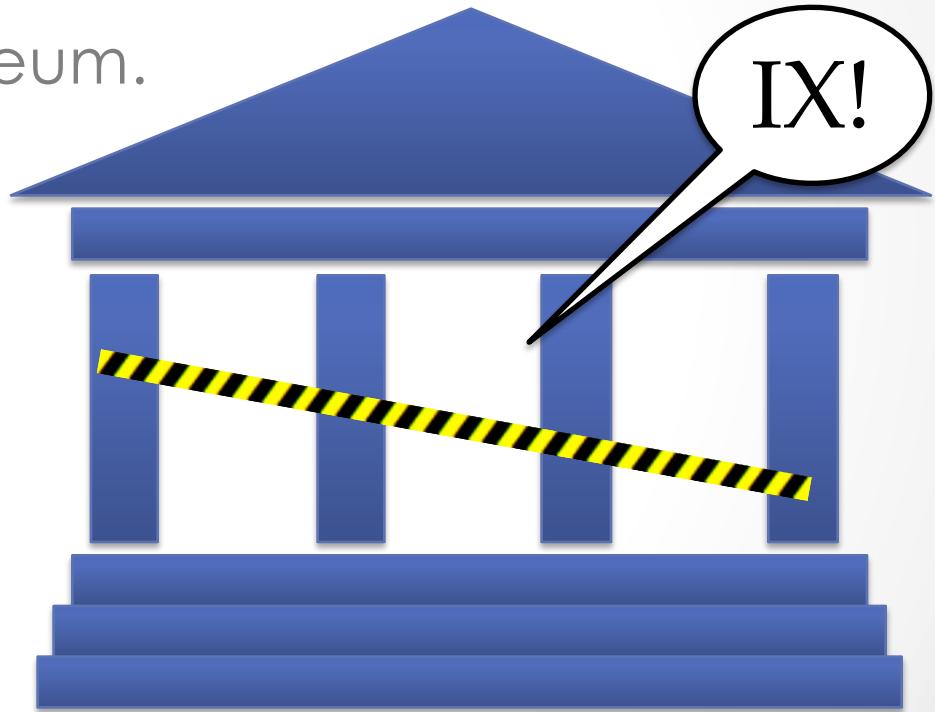
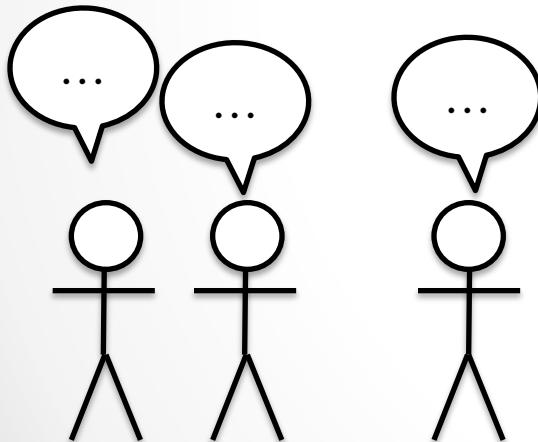


table locks

- While the construction is going on, visitors cannot get their badges, and cannot enter the museum.



what is an intention lock?

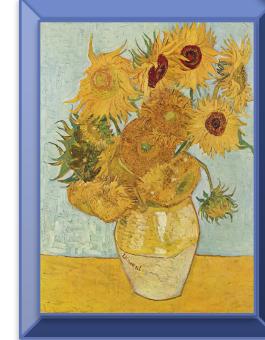
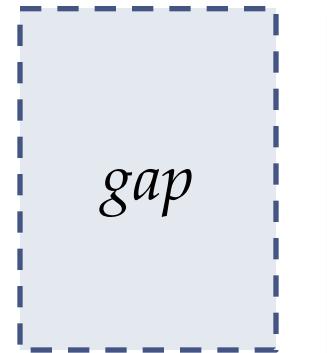
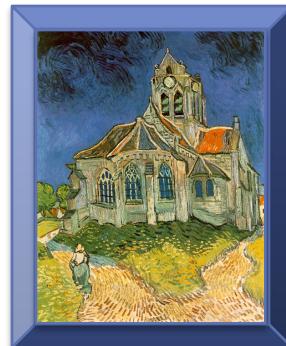
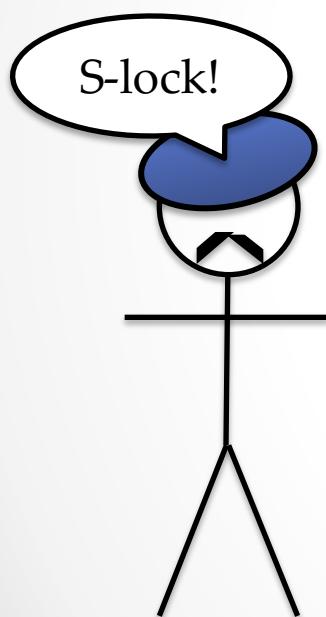
- During SELECT/INSERT/UPDATE/DELETE, a shared intention lock ("IS") allows access by multiple threads to work on the same table, but blocks exclusive table locks.
- During ALTER TABLE or DROP TABLE, an exclusive intention lock ("IX") blocks other table locks, either shared or exclusive. So no one else can query, and no one can run their own ALTER/DROP TABLE.

gap locks

...

gap locks

- The art critic needs to view the exact collection without changes and no new inserts.



what is a gap lock?

- A lock on a painting locks the “space” before the painting.
- The gap lock prevents inserts of new paintings before the locked painting (within the space).
- This happens automatically, to prevent “phantom reads”—i.e. the view of data changes during a transaction.

what is a gap lock?

- Exception: if the art critic is okay with seeing the latest additions among the paintings, he can choose to use the READ COMMITTED transaction isolation level.
- Another exception: no gap lock is needed for a UNIQUE or PRIMARY KEY index.

gap locks

- Optionally be more permissive about inserts.



gap locks

- Optionally be more permissive about inserts.

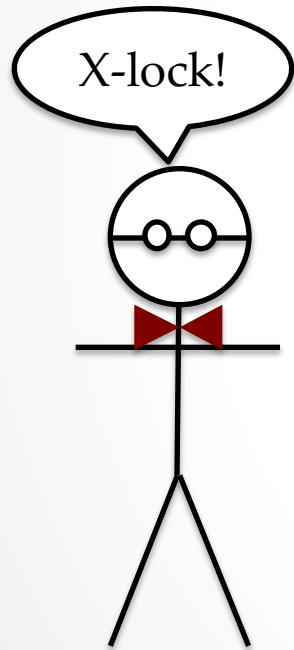


insert intention locks

...

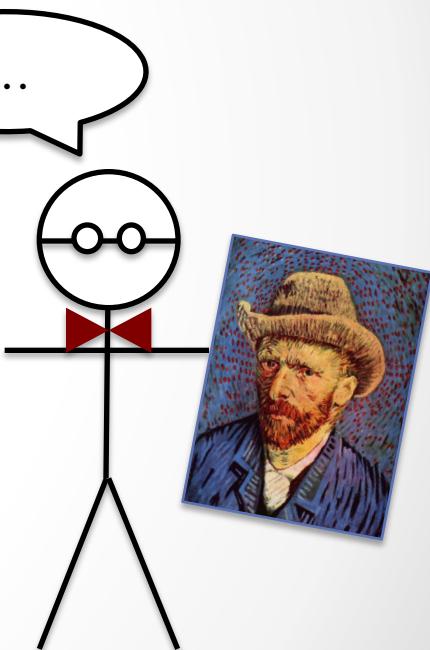
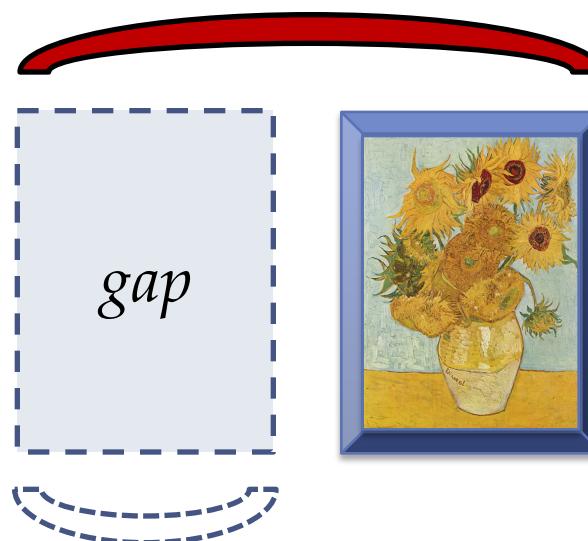
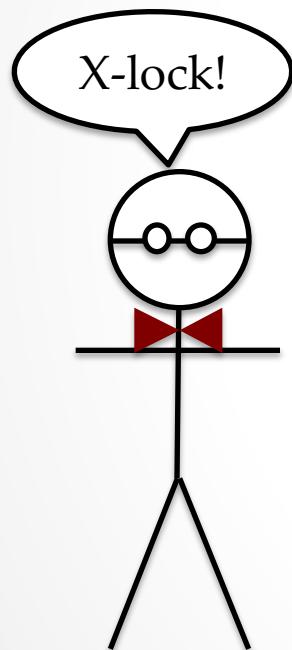
insert intention locks

- One curator wants to update paintings where $\text{year} > 1886$



insert intention locks

- The second curator wants to insert an 1887 painting, but it would fall within the existing gap lock.



insert intention locks

---TRANSACTION 32070411, ACTIVE 6 sec inserting
mysql tables in use 1, locked 1

LOCK WAIT 2 lock struct(s), heap size 360, 1 row lock(s),
undo log entries 1

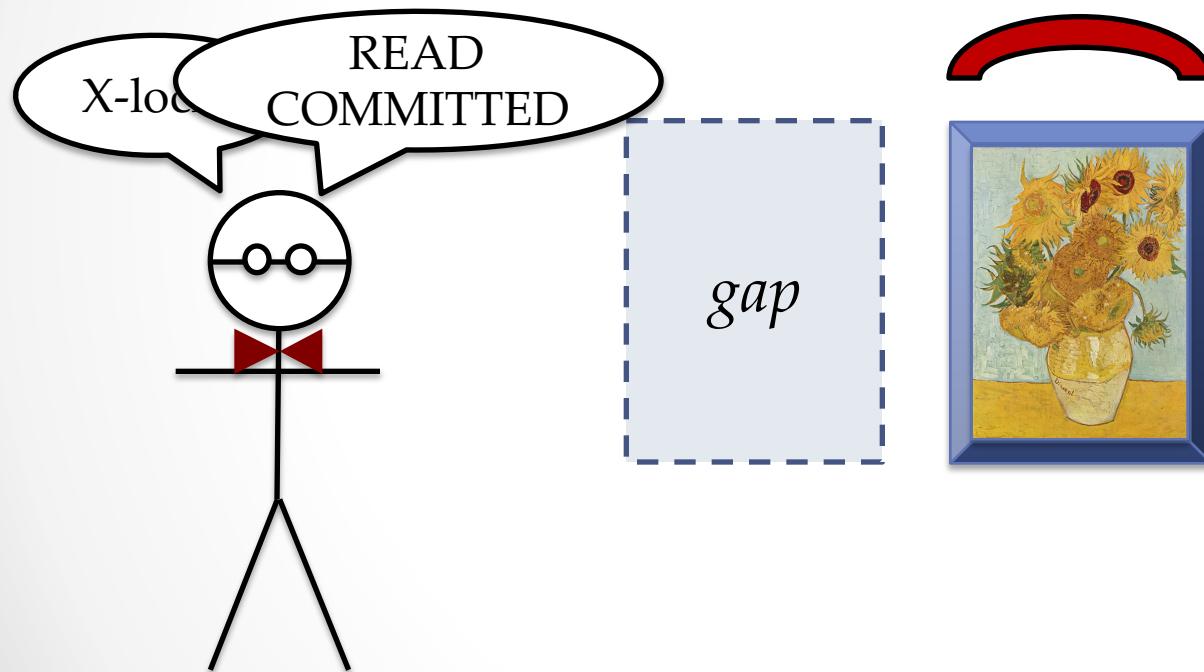
MySQL thread id 26, OS thread handle 0x7f2ba845f700,
query id 1423 192.168.50.1 root update
insert into Museum (year) values (1887)

----- TRX HAS BEEN WAITING 6 SEC FOR THIS LOCK TO BE
GRANTED:

RECORD LOCKS space id 3337 page no 4 n bits 72 index
~~`year` of table `test`.`Museum`~~ trx id 32070411 lock_mode X
insert intention waiting

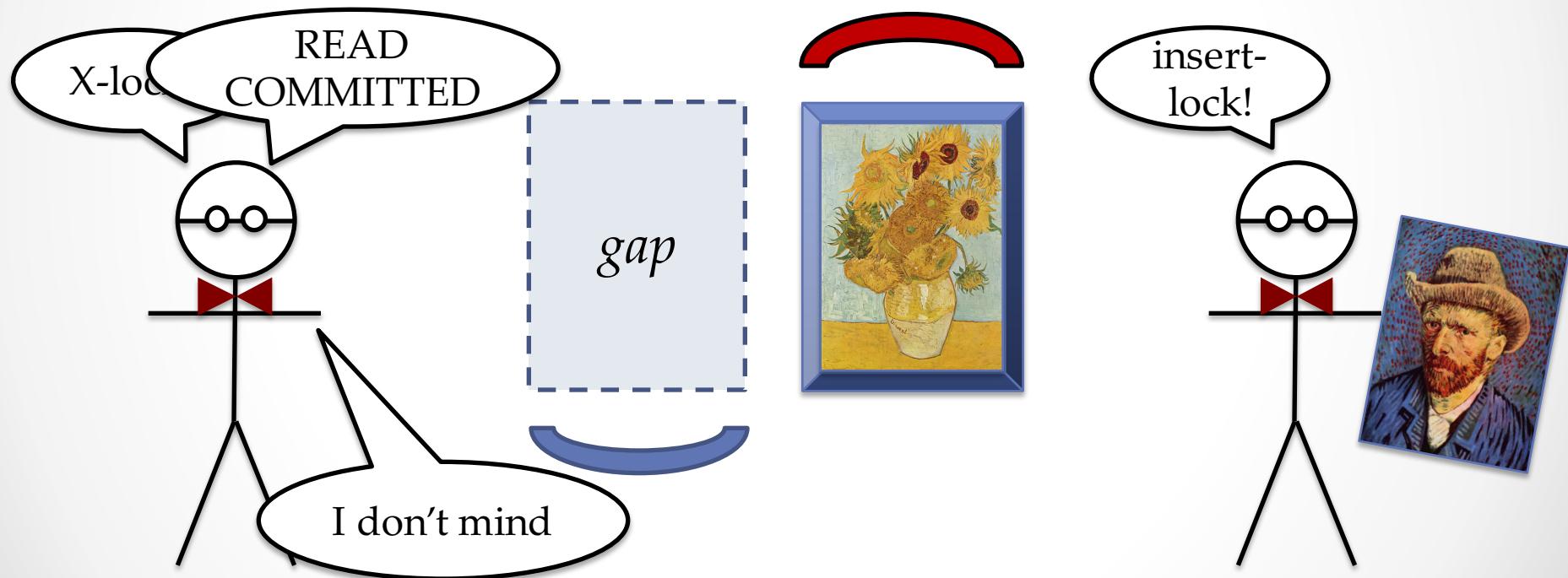
insert intention locks

- But if the first curator doesn't care about new paintings entering his view...



insert intention locks

- Then the second curator gets an insert intention lock, and is then free to insert.



insert intention locks

- Then the second curator gets an insert intention lock, and is then free to insert.



what is an insert intention lock?

- A special kind of gap lock, requested before a client tries to insert a row.
- Insert locks are shared, not exclusive—multiple clients can acquire insert locks on the same gap.
- But insert locks conflict with other exclusive locks.

why is insert intention lock shared?

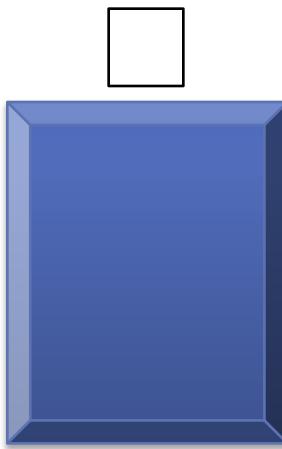
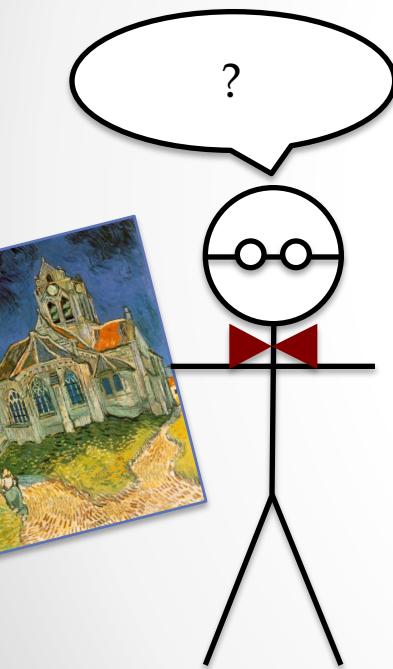
- Multiple clients prepare to insert into the same gap.
- They may be inserting different rows within the same gap, so they don't conflict with each other.
- But the insert intention lock blocks other clients from requesting exclusive locks on the same gap.

auto-inc locks

...

auto-inc locks

- Two curators are installing paintings. They both need to post a unique number for self-guided tours.



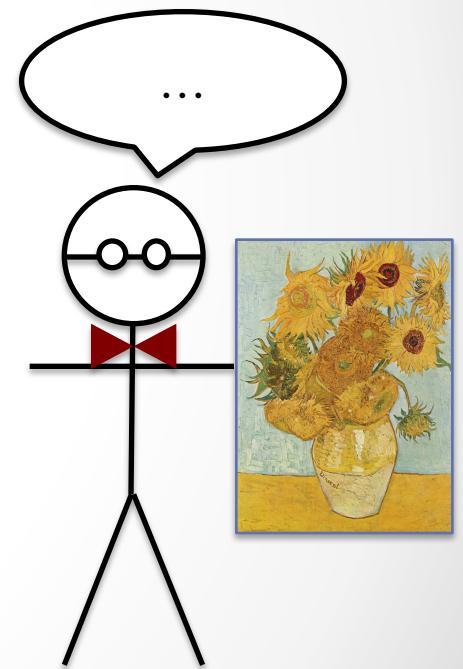
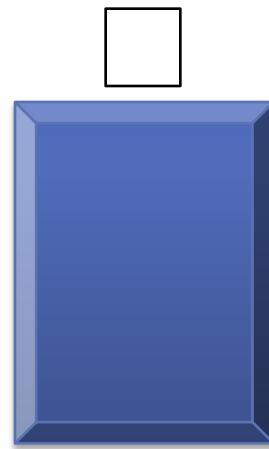
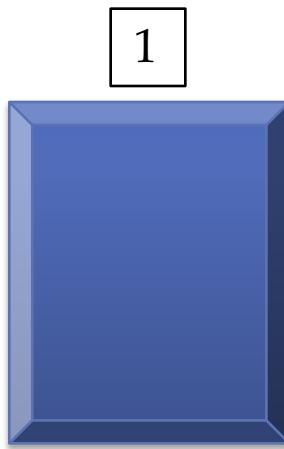
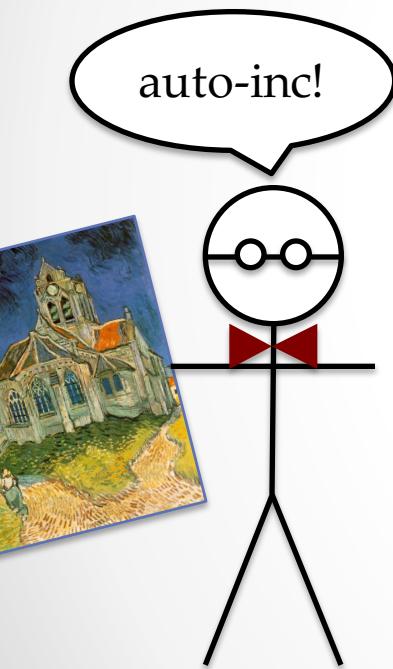
auto-inc locks

- There must be one number generator per table.
- One curator at a time can request the next value.



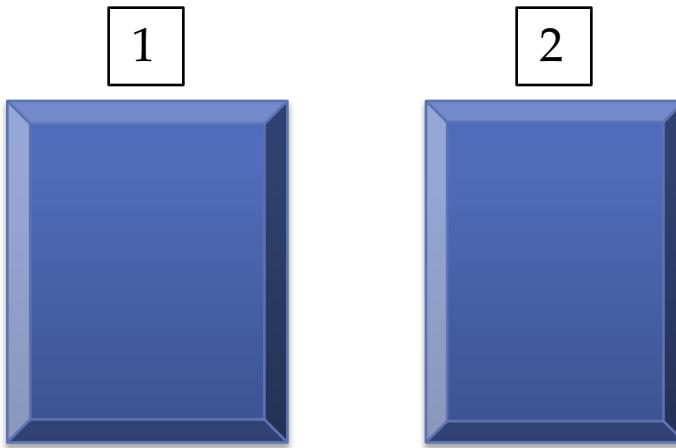
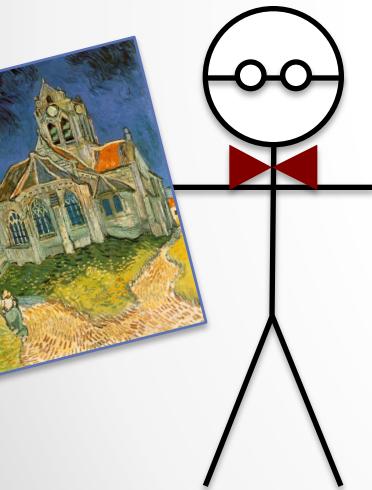
auto-inc locks

- The first curator gets a number.



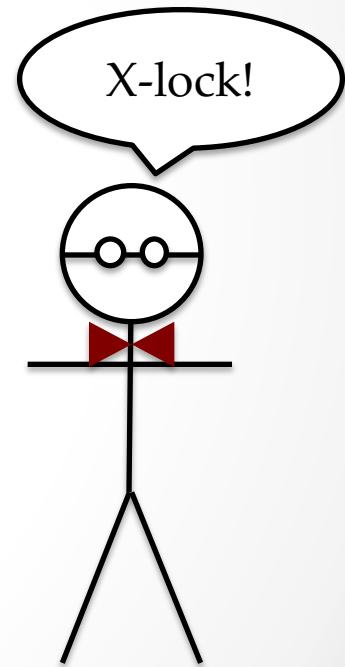
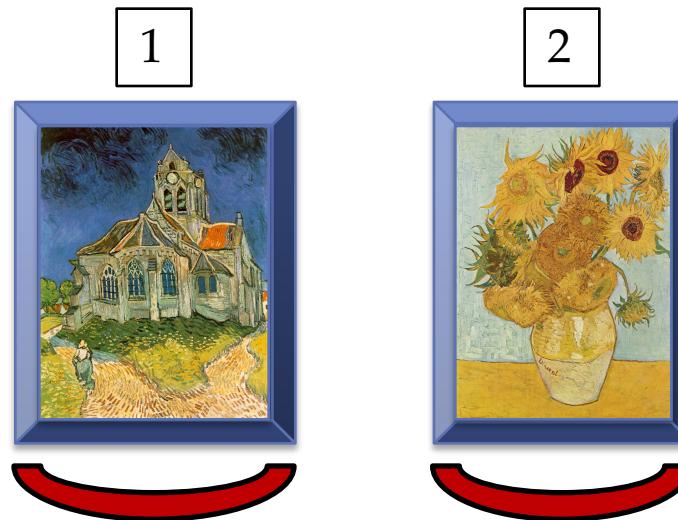
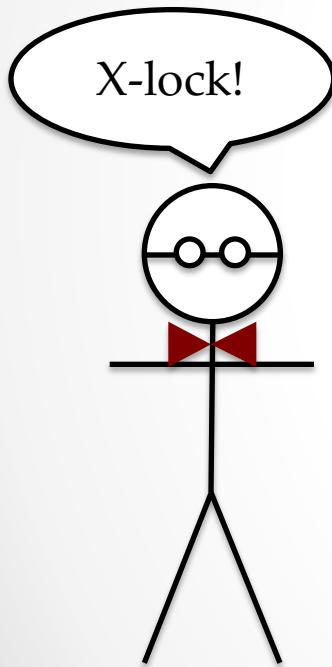
auto-inc locks

- As soon as the first curator gets his number, the second curator can proceed.



auto-inc locks

- They may both keep locks on the paintings, but they're done allocating numbers.



what is an auto-inc lock?

- A table lock, used when a client requests the next unique id for a given table.
- Ensures that each id is given to one client.
- Brief—it is released as soon as the id is generated, instead of lasting to the end of the transaction like other locks.
- Because the lock is so brief, neither client can “undo”— i.e. return their id to the stack for someone else to use.

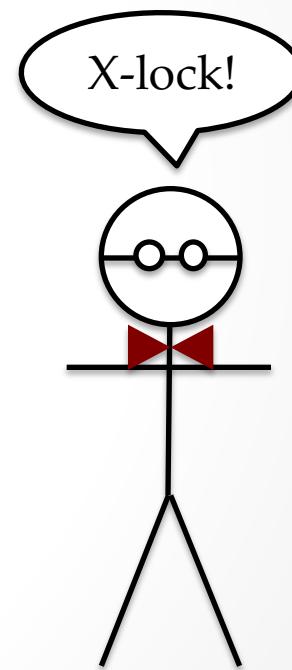
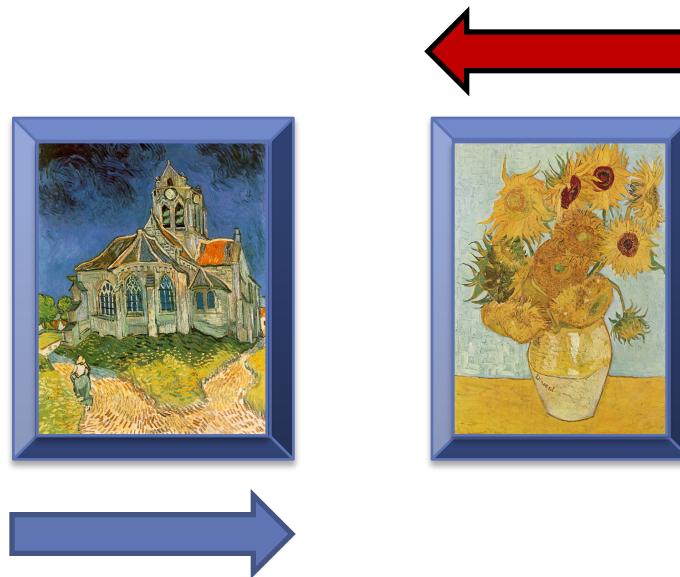
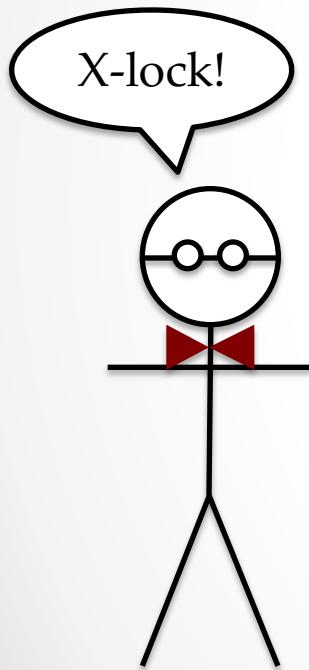


deadlocks

...

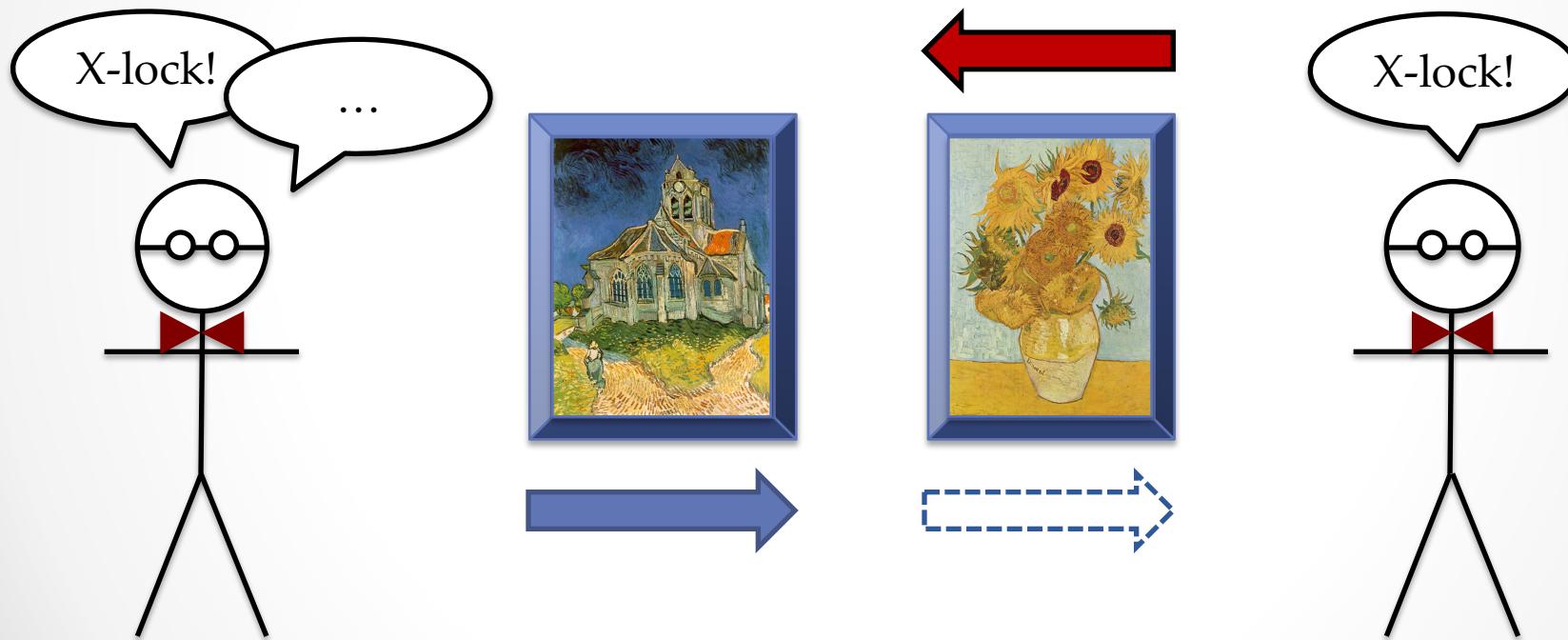
deadlocks

- Two curators are updating the art, but they start from different ends of the collection.



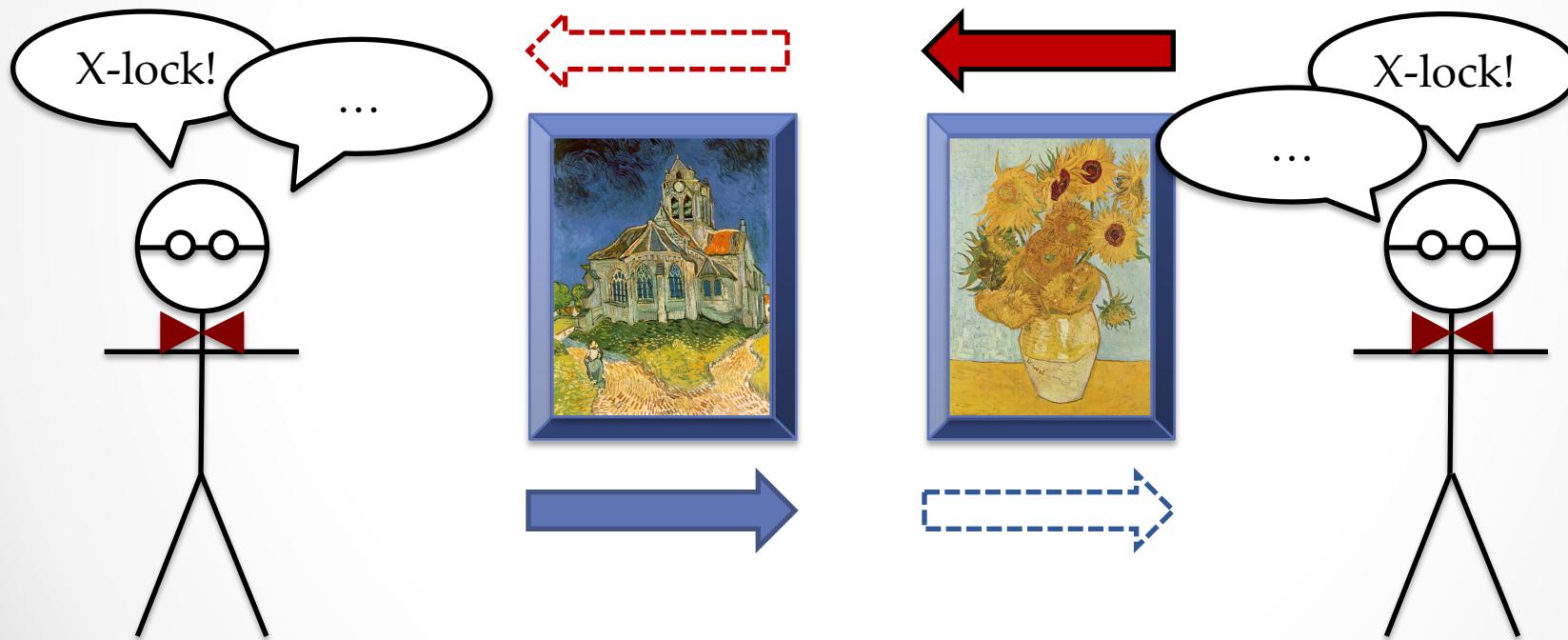
deadlocks

- A curator requests a lock on the second painting, which is already locked. He waits.



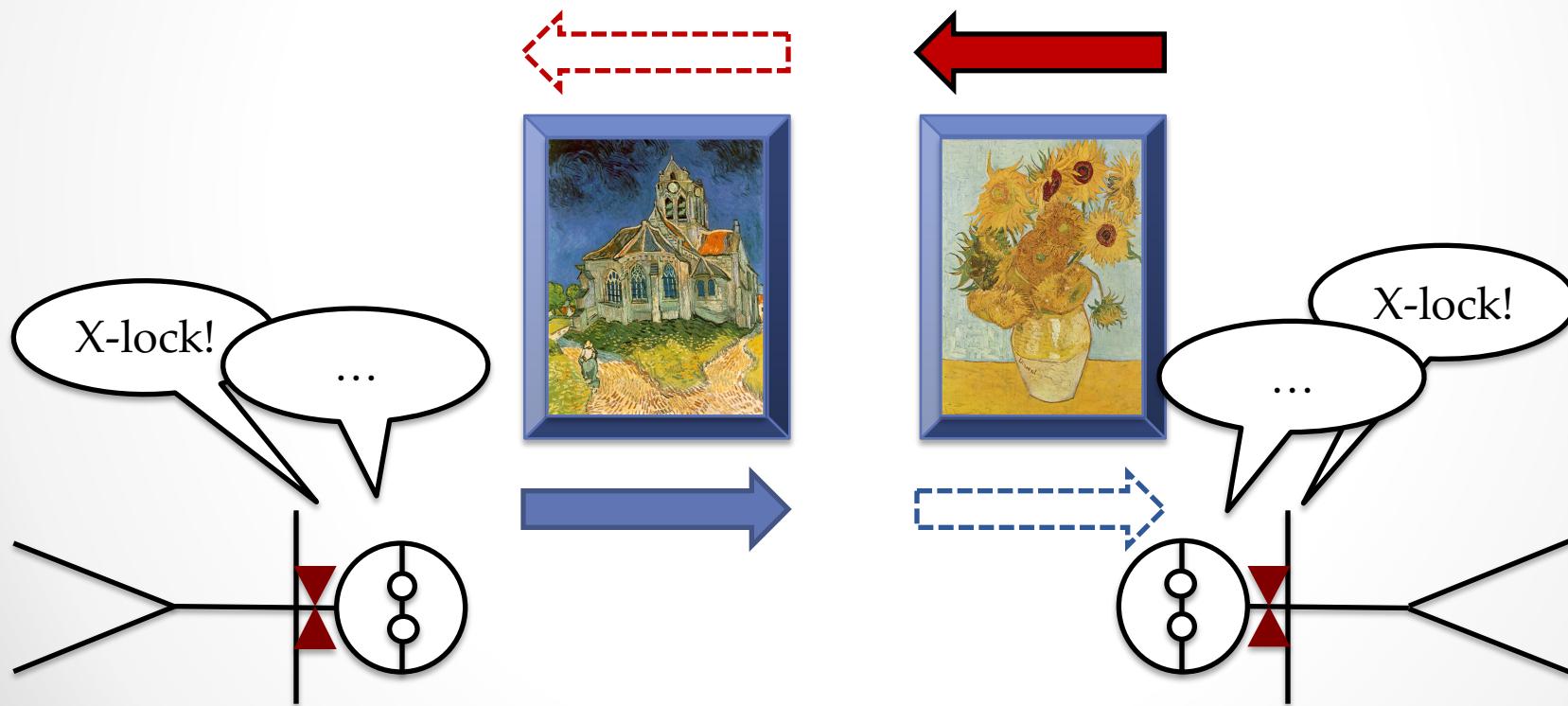
deadlocks

- The other curator requests a lock on the first painting, which is already locked. He also waits.



deadlocks

- Neither will give up the lock they have, so they are doomed to wait until one or both of them dies.

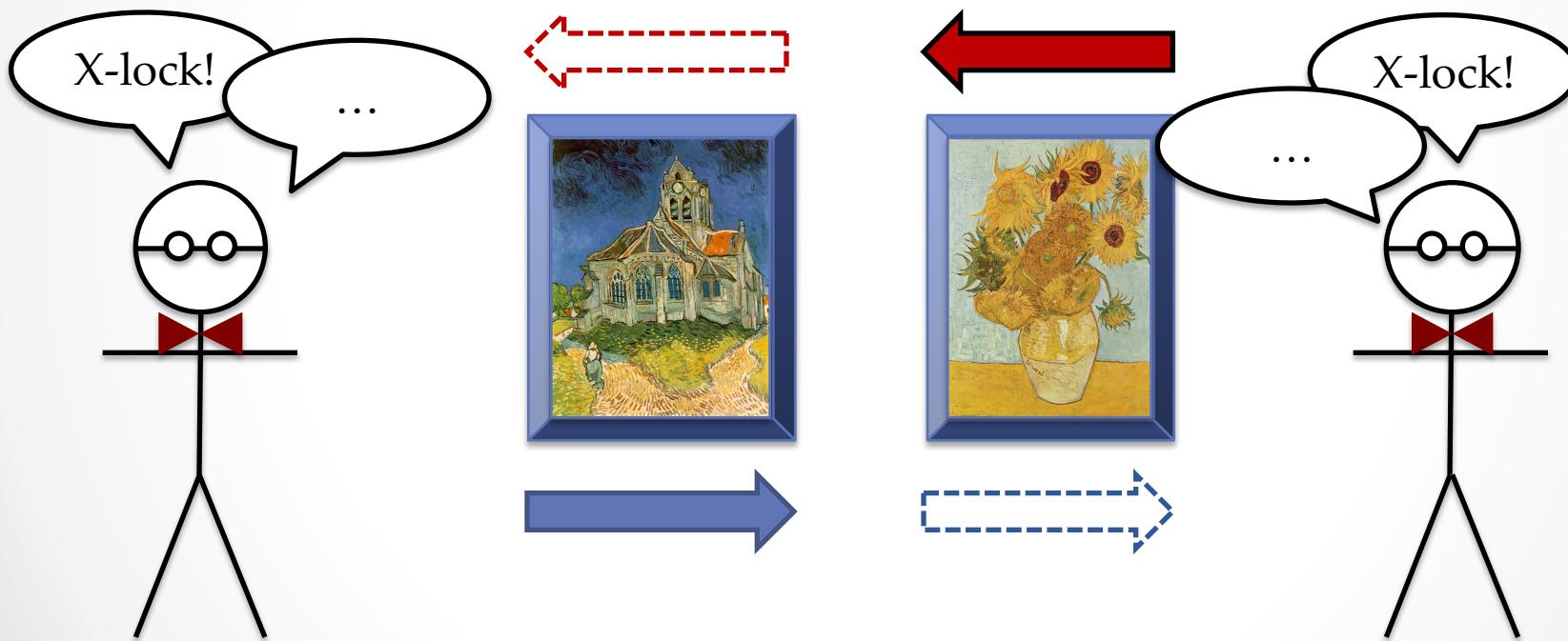


what is a deadlock?

- When two or more concurrent clients wait for each other to release their locks, but since they are both waiting, they will never give up the lock they have.
 - In other words, a catch-22 of lock-waits.
 - Many people use the term “deadlock” incorrectly—when they are describing a simple one-way lock wait.
-
-

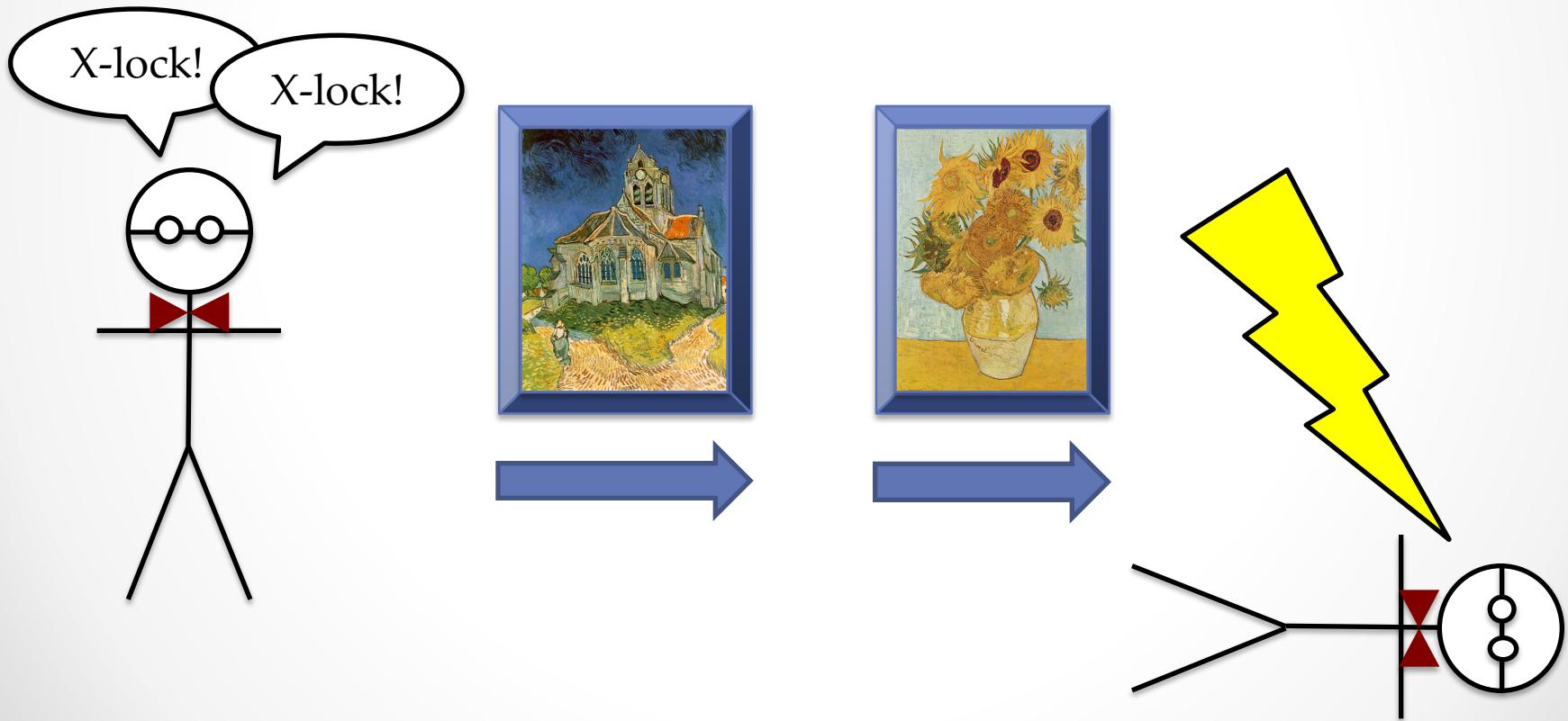
resolving deadlocks

- MySQL detects cycles in lock waits, and kills one of the transactions immediately.



resolving deadlocks

- MySQL detects cycles in lock waits, and kills one of the transactions immediately.

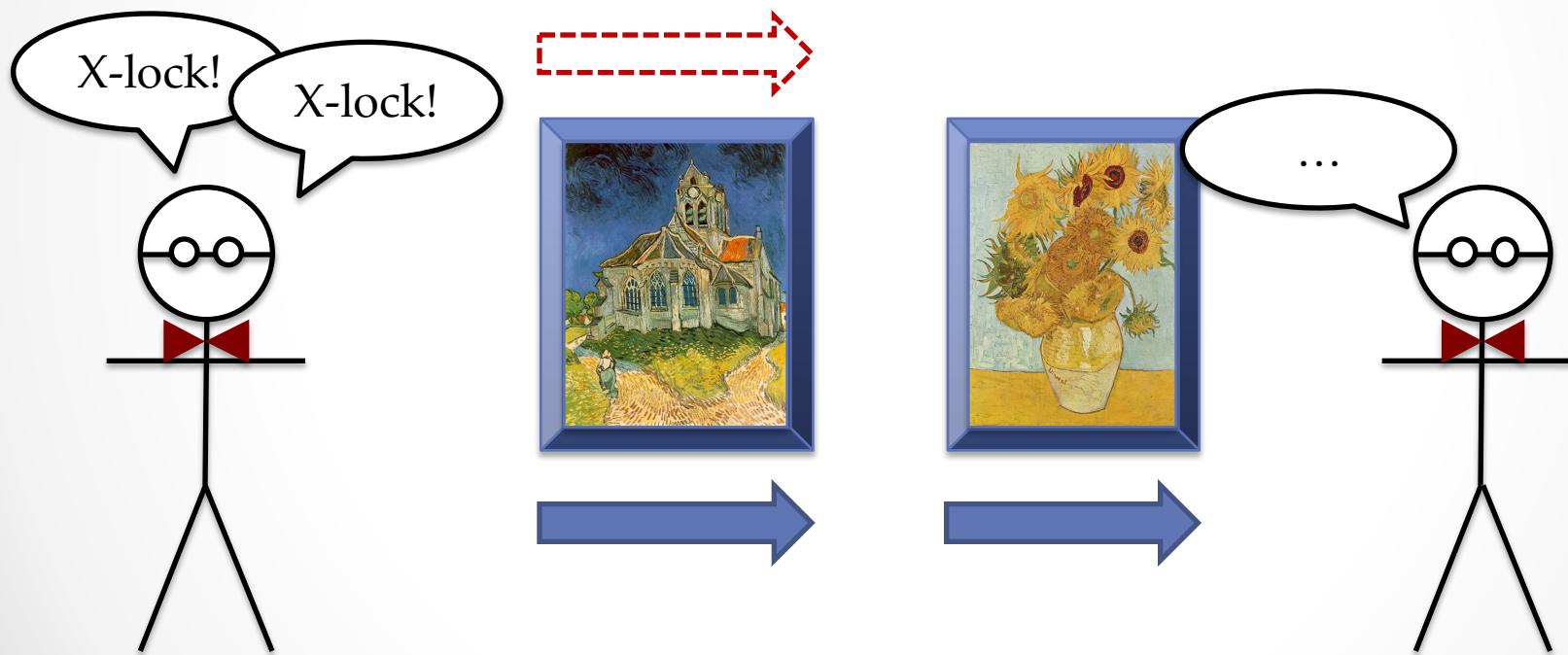


avoiding deadlocks

- Each client locks everything they need in one atomic request.
- All clients request locks in the same order.

avoiding deadlocks

- All clients request locks in the same order.

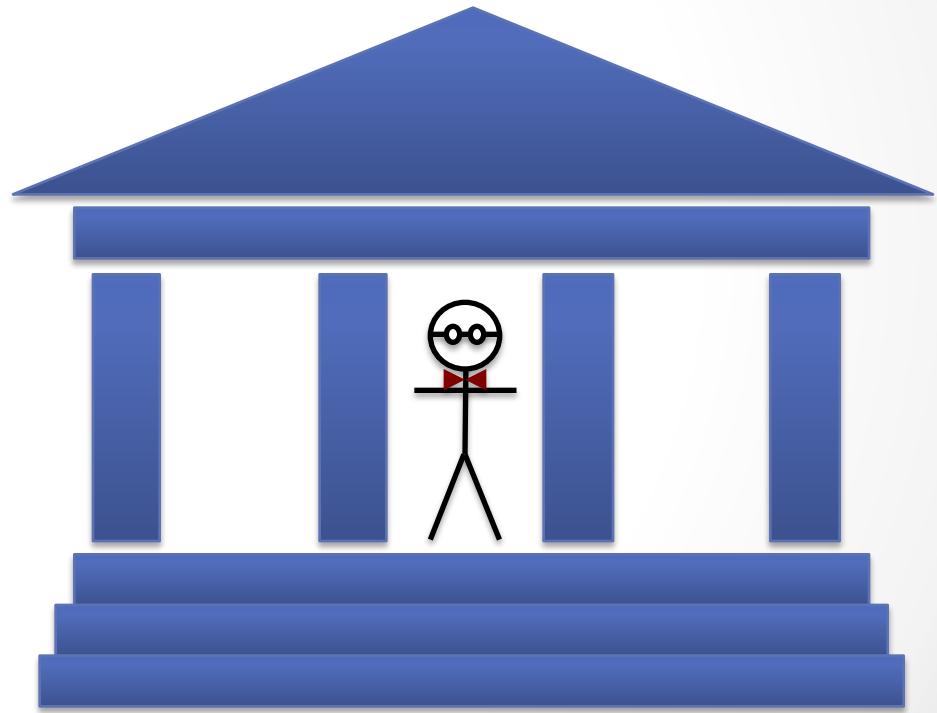
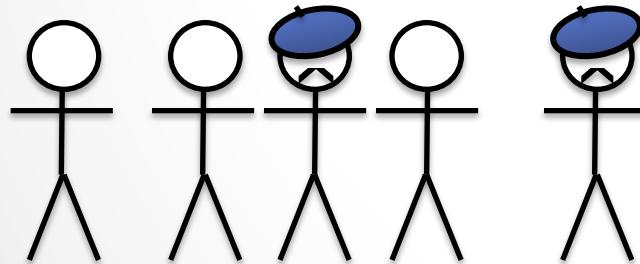


conclusion

...

conclusion

- I hope your trip to the museum was educational.



license and copyright

Copyright 2015 Bill Karwin

<http://www.slideshare.net/billkarwin>

Released under a Creative Commons 3.0 License:

<http://creativecommons.org/licenses/by-nc-nd/3.0/>

You are free to share—to copy, distribute, and transmit this work, under the following conditions:



Attribution.

You must attribute this work to Bill Karwin.

Noncommercial.

You may not use this work for commercial purposes.

No Derivative Works.

You may not alter, transform, or build upon this work.