

Can also chain other transactions using AND CHAIN

What objects should we lock?

Ex

^{part} T1 SELECT S.rating, MIN(S.age)
FROM Sailors S
WHERE S.rating = 8

T2 changes sailors ratings
want to avoid conflicts

What if
someone
updates
while
processing

could get shared lock on whole table (slower, safer)
or on a row level. (Faster, less safe)

DBMS support locking at different granularities

Oracle

Ex LOCK TABLE MY_TABLE
IN SHARE MODE
EXCLUSIVE MODE

supports 4 other modes

By default Oracle automatically does row level locking

More on SQL transactions

Can control access mode, diagnostic size, isolation level

↓
READ ONLY
or not

↓
how many error conditions will be recorded

↑
will talk about

In Oracle ↑

~~ALTER SESSION~~
SET TRANSACTION
READ ONLY
SET TRANSACTION
READ WRITE

↳ can use
alter system
to control
of unlimited in Oracle
or fixed sized

↳ with you can
set what roll back
segment transaction
ends

ISOLATION LEVEL

in SQL-99

(Affects how locks are obtained automatically)

Level	Dirty Read	Unrepeatable Read	Phantom ^{object disappear} _{reads}
READ UNCOMMITTED	Maybe	Maybe	Maybe
READ COMMITTED	No	Maybe	Maybe
REPEATABLE READ	No	No	Maybe
SERIALIZABLE	No	No	No

NO index locking

To set in Oracle:

Different for session versus transaction

Session [ALTER SESSION SET ISOLATION_LEVEL =

[READ COMMITTED | SERIALIZABLE]

Transaction [SET TRANSACTION ISOLATION_LEVEL = [READ COMMITTED | SERIALIZABLE]

Ch 17

More on locking & serializability

Weaker versions of serializability

Defⁿ Two schedules are conflict equivalent if they involve the same set of actions of the same transactions and they order every pair of conflicting operations the same way

<u>Ex)</u>	<u>Ex)</u>	T1	T2		T1	T2
		R(A)				R(A)
			R(A)		R(A)	
		W(A)	W(A)	conflict op	W(A)	W(A)
		C1	C2		C1	C2

Defⁿ A schedule is conflict serializable if it is conflict equivalent to a serial schedule

<u>Ex)</u>	T1	T2	T3
	R(A)		
		W(A)	
	W(A)	C	
	C		W(A)
			C

→ serializable
T1 T2 T3
but not
conflict serializable
b/c writes of T1
and T2 would
be ordered different