Db2 for z/OS

Change version

11

☑ Show full table of contents

Other resources on this topic

DB2 11 for z/OS Application Programming Workshop
Course

DB2 11 for z/OS Application Performance and Tuning
Course

DB2 11 for z/OS Database Administration Workshop Part 1
Course

z/OS and DB2 Basics for DB2 for z/OS DBA Beginners
Course

DB2 11 for zOS Implementation Workshop
Course

Db2 for z/OS / 11 /

Feedback Product list

Lock modes and compatibility of locks

The mode of a lock tells what access to the locked object is permitted to the lock owner and to any concurrent processes.

PSPI Db2 uses of the lock modes to determine whether one lock is compatible with another. Some lock modes do not exclude all other users. For example, assume that application process A holds a lock on a table space that process B also wants to access. Db2 requests, on behalf of process B, a lock of some particular mode. If the mode of the lock for process A permits the lock requested by process B, the modes of the two locks are said to be compatible. However, if the two locks are not compatible, process B cannot proceed. It must wait until process A releases its lock, and until all other existing incompatible locks are released.

3

Page and row lock modes

The modes and their effects are listed in the order of increasing control over resources.

S lock (share)

The lock owner and any concurrent processes can read, but not change, the locked page or row. Concurrent processes can acquire S or U locks on the page or row or might read data without acquiring a page or row lock. Only one U lock can be acquired concurrently with one or more S locks.

U lock (update)

The lock owner can read, but not change, the locked page or row. Concurrent processes can acquire S locks or might read data without acquiring a page or row lock, but no concurrent process can acquire a U lock.

U locks reduce the chance of deadlocks when the lock owner is reading a page or row to determine whether to change it. The owner can start with the U lock and then promote the lock to an X lock to change the page or row.

X lock (exclusive)

The lock owner can read or change the locked page or row. A concurrent process cannot acquire S, U, or X locks on the page or row. However, concurrent processes, such as those processes bound with the CURRENTDATA(NO) or ISOLATION(UR) bind options or running with YES specified for the EVALUNC subsystem parameter, can read the data without acquiring a page or row lock.

Partition, table space, and table lock modes

When a page or row is locked, the table, partition, or table space that contains it is also locked. This intent lock indicates the plan that the application process has for accessing the data. In that case, the table, partition, or table space lock has one of the intent modes: either IS for intent share, IX for intent exclusive, or SIX for share with intent exclusive.

The modes S, U, and X of table, partition, and table space locks are sometimes called gross lock modes. In the context of reading, SIX is a gross mode lock because you do not get page or row locks; in this sense, it is like an S lock.

The modes and their effects are listed in the order of increasing control over resources.

IS lock (intent share)

The lock owner can read data in the table, partition, or table space, but not change it. Concurrent processes can both read and change the data. The lock owner might acquire a page or row lock on any data it reads.

IX lock (intent exclusive)

The lock owner and concurrent processes can read and change data in the table, partition, or table space. The lock owner might acquire a page or row lock on any data it reads; it must acquire one on any data it changes.

S lock (share)

The lock owner and any concurrent processes can read, but not change, data in the table, partition, or table space. The lock owner does not need page or row locks on data it reads.

U lock (update)

The lock owner can read, but not change, the locked data; however, the owner can promote the lock to an X lock and then can change the data. Processes concurrent with the U lock can acquire S locks and read the data, but no concurrent process can acquire a U lock. The lock owner does not need page or row locks.

U locks reduce the chance of deadlocks when the lock owner is reading data to determine whether to change it. U locks are acquired on a table space when the lock size is TABLESPACE and the statement is a SELECT with a FOR UPDATE clause. Similarly, U locks are acquired on a table when lock size is TABLE and the statement is a SELECT with a FOR UPDATE clause.

SIX lock (share with intent exclusive)

The lock owner can read and change data in the table, partition, or table space. Concurrent processes can read data in the table, partition, or table space, but not change it. Only when the lock owner changes data does it acquire page or row locks.

X lock (exclusive)

The lock owner can read or change data in the table, partition, or table space. A concurrent process can access the data if the process runs with UR isolation or if data in a partitioned table space is running with CS isolation and CURRENTDATA((NO). The lock owner does not need page or row locks.