

SQL Server Locks

SQL Server uses **locks** to:

- protect data consistency (ACID)
- allow **concurrent access**
- prevent corruption

Locks are taken at different **granularities**:

- DATABASE
- OBJECT (table)
- PAGE
- KEY (row)
- RID (heap row)

And they have different **modes** (S, X, IX, ...).

1 Core Lock Modes (Most Important)

● S – Shared Lock

Purpose: Read data safely

- Taken during SELECT (READ COMMITTED)
- Multiple S locks **can coexist**
- Blocks writers (X locks)

Example

```
SELECT * FROM Orders WHERE OrderID = 100;
```

Blocks

- **✗** X (Exclusive)

Allows

- **✓** Other S
-

🔴 X – Exclusive Lock

Purpose: Modify data

- Taken during INSERT, UPDATE, DELETE
- Only **one X lock allowed**
- Blocks everyone else

Example

```
UPDATE Orders SET Amount = 500 WHERE OrderID = 100;
```

Blocks

- ✗ S, X, IS, IX
-

🟡 U – Update Lock

Purpose: Prevent deadlocks during read-then-write

- Used internally during UPDATE
- Later converts to X
- Only one U allowed per resource

Why it exists

Prevents S → X deadlock scenarios.

2 Intent Locks (IX, IS, SIX) — VERY IMPORTANT FOR DBAs

Intent locks are **signals**, not actual data locks.

They tell SQL Server “*I plan to lock something lower down.*”

◆ IS – Intent Shared

Meaning:

“I’m going to take S locks at a lower level.”

- Taken at table/page when rows are read
- Allows concurrent readers

Example

```
SELECT * FROM Orders WHERE OrderID = 100;
```

◆ IX – Intent Exclusive

Meaning:

“I’m going to take X locks at a lower level.”

- Common during UPDATE/DELETE
- Very frequently seen in blocking

Example

```
UPDATE Orders SET Amount = 500 WHERE OrderID = 100;
```

◆ SIX – Shared + Intent Exclusive

Meaning:

“I’m reading the table AND updating some rows.”

- Rare but important
- Blocks other writers
- Allows some readers

3 Schema Locks (DDL-related)

Sch-S – Schema Stability

Purpose:

Protects schema during queries.

- Taken by SELECT, INSERT, UPDATE
- Allows concurrent queries
- Blocks schema changes

Example

```
SELECT * FROM Orders;
```

Sch-M – Schema Modification

Purpose:

Change table structure.

- Taken by ALTER TABLE, CREATE INDEX
- Blocks EVERYTHING

Example

```
ALTER TABLE Orders ADD NewColumn INT;
```

4 Bulk & Special Locks

BU – Bulk Update

- Used during bulk operations
 - Optimized for minimal logging
-

Range Locks (Serializable)

Used to prevent phantom reads:

- RangeS-S
- RangeX-X

Seen under **SERIALIZABLE isolation**.



Lock Compatibility (Simplified)

Requested \ Held	S	X	IS	IX
S	✓	✗	✓	✗
X	✗	✗	✗	✗
IS	✓	✗	✓	✓
IX	✗	✗	✓	✓



Lock Wait Types (What DBAs see)

These appear in:

```
sys.dm_exec_requests  
sys.dm_os_waiting_tasks
```



LCK_M_S

Meaning:

Session is **waiting to acquire a Shared lock**

Cause

- Someone else holds X lock

Typical scenario

```
SELECT waiting while UPDATE is running
```



LCK_M_X

Meaning:

Session is **waiting for Exclusive lock**

Cause

- Another session has S or X lock

Typical scenario

UPDATE blocked by long SELECT



LCK_M_IX

Meaning:

Waiting for Intent Exclusive lock

Often seen

- High-contention OLTP tables
-



LCK_M_U

Waiting for Update lock



LCK_M_SCH_M

Waiting for schema modification lock

Classic example

ALTER TABLE blocked by long SELECT



Example DMV Output Explained

You earlier showed something like:

```
70  KEY      X  GRANT
70  OBJECT   IX GRANT
70  PAGE     IX GRANT
```

Interpretation

- Session 70:
 - Has **X lock on a KEY (row)** → updating data
 - Has **IX on PAGE & OBJECT** → intent to update rows

This is **normal behavior**.



DBA Key Takeaways (Very Important)

1. **IX locks are not bad** — they are normal signals
 2. **Blocking happens when X blocks S or vice versa**
 3. Most blocking problems are:
 - o long transactions
 - o missing indexes (large scans holding locks)
 - o wrong isolation level
 4. Schema locks (Sch-M) are the most disruptive
 5. **RCSI/Snapshot** reduce S vs X blocking drastically
-



How DBAs Analyze Locking

Typical flow:

```
-- Who is blocked?  
sys.dm_exec_requests  
  
-- Who is blocking?  
blocking_session_id  
  
-- What locks?  
sys.dm_tran_locks  
  
-- What SQL?  
sys.dm_exec_sql_text
```



Interview-Ready One-Liner

“SQL Server uses intent locks (IS, IX) to coordinate hierarchical locking, while shared (S) and exclusive (X) locks protect data. Blocking occurs when incompatible locks are requested, commonly seen as LCK_M_* waits. Proper indexing and row-versioning isolation reduce lock contention.”
