

# InnoDB locks

S	SELECT
X	UPDATE / DELETE
IS	SELECT FOR SHARE
IX	SELECT FOR UPDATE

## • SHARED LOCK / EXCLUSIVE LOCK (row-level)

T1rST2r  
T2S  
T2X

T1rXT2rT1r

## • INTENTION LOCK (table-level)

InnoDB

**InnoDB**

S/X

- intention shared lock TtS  
*select ... lock in share mode*
- intention exclusive lock TtX U/D  
*select ... for update*
  
- SISstronger
- XIXstronger

	X	IX	S	IS
X	x	x	x	x
IX	x		x	
S	x	x		
IS	x			

## • RECORD LOCK

Record lock

Record locksInnoDBrecord locking

## • GAP LOCK

index recordsindex recordindex record

```
SELECT c1 FROM t where c1 BETWEEN 10 and 20 FOR UPDATE ; SQLc115
```

**index record**

**10 20 index record 10-20**

**10 20 index record 10index record - 20 index record**

**RR**

```
SELECT * FROM child WHERE id = 100 ;
```

idid100record lock

id 50 100 50100]

id

**gap**

**Bgap lockgap X-lockAgap lockgap S-lock**

**indexpurgegapgapmerged**

Gap lockgap lockgap

gapgap locksgap x-lock gap s-lock

gap lock

[read committed innodb\\_locks\\_unsafe\\_for\\_binlog](#)

gap locking

## • NEXT-KEY LOCK

next-key lockrecord lockrecord lockgap lock

InnoDBInnoDBindex recordSX

row-levelindex-record

index recordnext-keyindex recordgap

sessionRSXsessionRgaprecord

gap lock

10111320next-key locks

( - 10]

(10 , 11]

(11 , 13]

(13 , 20]  
(20 , +)  
PS(13 , 20] 13 < x <= 20

InnoDBrepeatable readInnoDBnext-key locks

## • INSERT INTENTION LOCK

insert intention lockINSERT

index gap

4756X47

## • AUTO-INC LOCK table-level

Auto-inc

innodb\_autoinc\_lock\_modeauto-increment

## • predicate locks for spatial indexes

InnoDBSPATIAL index

SPATIAL indexesnext-keyREPEATABLE READSERIALIZABLE“next-key“

SPATIAL indexesInnoDBpredicate locks

SPATIAL index minimum bounding rectangle(MBR)InnoDBMBRpredicate locksession