

채번 테이블의 한 블록. 경합이 많이 발생하는 Hot Block
여러 프로세스가

테이블명	구분특성	최종값
A	a1	10
A	a2	17
B	b1	8

프로세스 ①

프로세스 ②

프로세스 ③

프로세스 ④

이 테이블에 대해 채번할 경우

⇒ RowLock에 의해

관리되어 양쪽 프로세스가
동시에 Update하지 못하도록...

①과 ②가 RowLock으로 경합

③과 ④가 " "

같은 레코드에 대한

동시 채번은 RowLock이 막아
준다.

프로세스 ①은 A 테이블의 최종값을 읽어옴. } 이때는 RowLock이 작동X
③은 B " " " " }

근데 두 최종값을 저장한 각 레코드는 한 블록에 있음. → ①, ③도 경합을 함.
채번
1차로 해시체인에 대한 래치 Lock
2차로 블록에 대한 버퍼 Lock

Q. 위 블록에 저장된 레코드가 100개 일 때, 10개 일 때
어떤 경우가 더 경합이 많이 발생할까?

A. 같은 테이블에 대한 경합은 어쩔 수 없음.

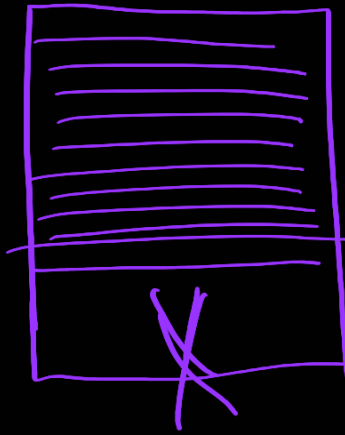
채번의

근데, 서로 다른 테이블을 채번하려는 상황에서도
(채번 테이블 블록상에서 서로 다른 레코드를 변경하려는 상황)
래치락은 물론 버퍼 Lock으로 성능이 떨어짐

해결책! 블록 사이즈 줄여서 수용가능한 레코드 수를 줄여라.
2K, 4K, 8K, 16K, 32K.

대신 더 많은 블록
을 위한 테이블 스페이스 / 버퍼블록을 설정해야 함.

해결책2) PCT free 10% \Rightarrow 90%로 설정.



블록의
10% 공간은 남겨놓는 방식.

원리: 레코드의 길이를 더 큰 값으로

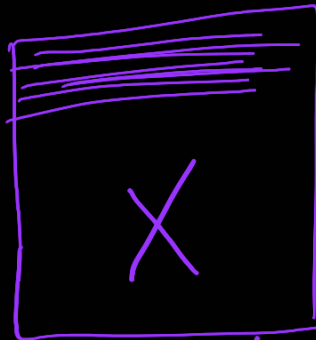
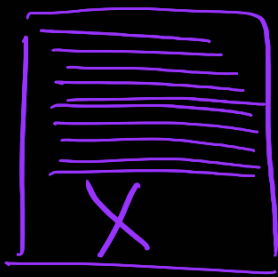
Update할 때 ('A' \rightarrow 'ABCDEF')

블록 새 블록을 만들고 그 쪽으로 Update함.

\Rightarrow Row migration
비효율적.

따라서 PCT free로 빈공간을

확보해놓음으로서 Row migration을 대체함.



90%

가능한 Row 수를 줄일 수
있다

\Rightarrow 공간을 줄일 수 있다.

재번테이بل의
같은
레코드

동일테이블 A에 대한 동시 Insert가 많은 경우
재번테이بل 사용 불가 (Row lock으로 인해 Commit 전까지 다른 트랜잭션
이 기다려야 해서 성능 저하를 일으킴)

다른 테이블 A, B에 대한 (재번테이بل의 다른 레코드)의 경우
PCT free로
해결 가능.