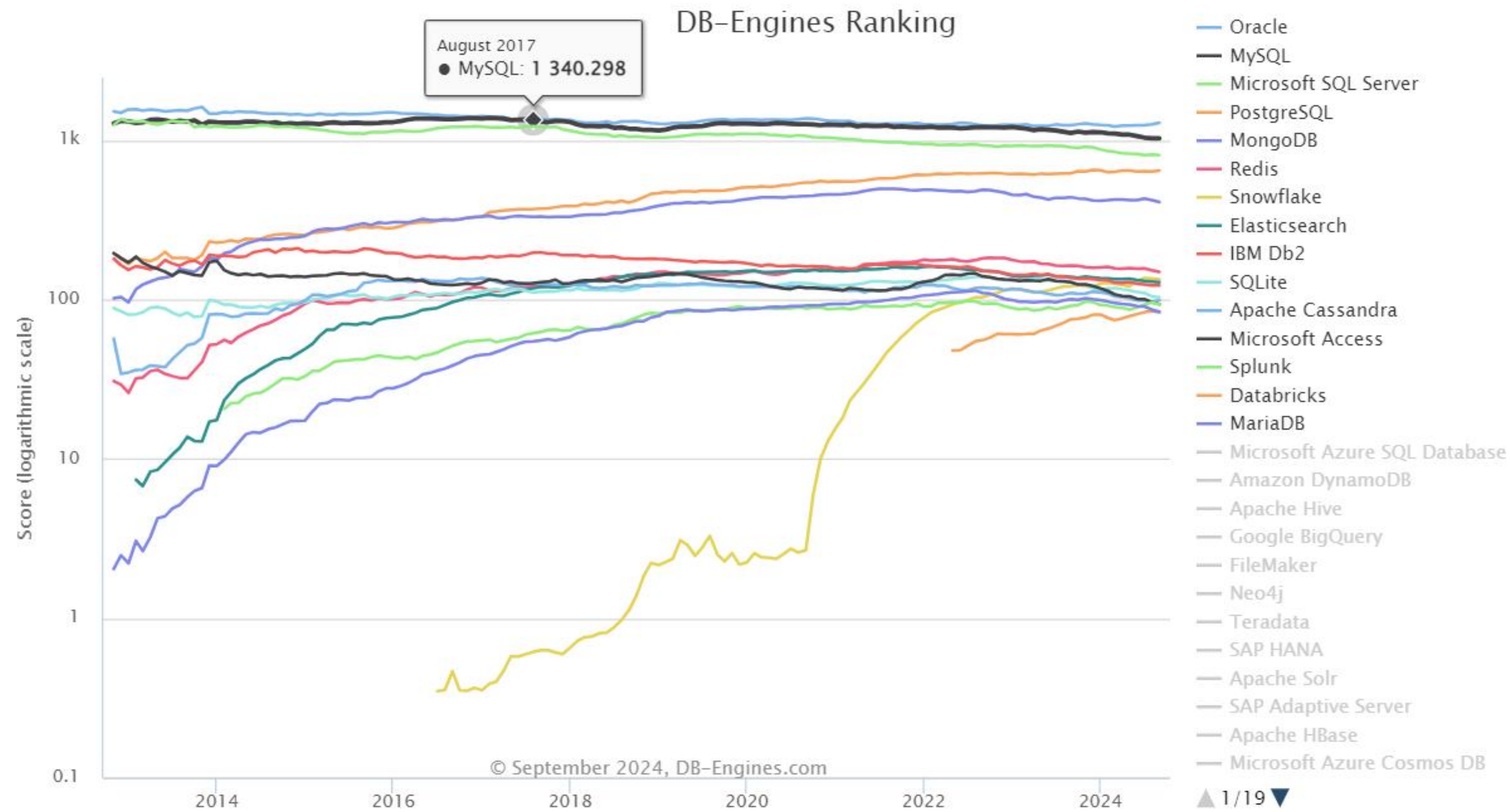


Database week 1

1, 2

발표자: 박지원

MySQL



발표자: 박지원

Architecture

MySQL Connectors

MySQL Shell

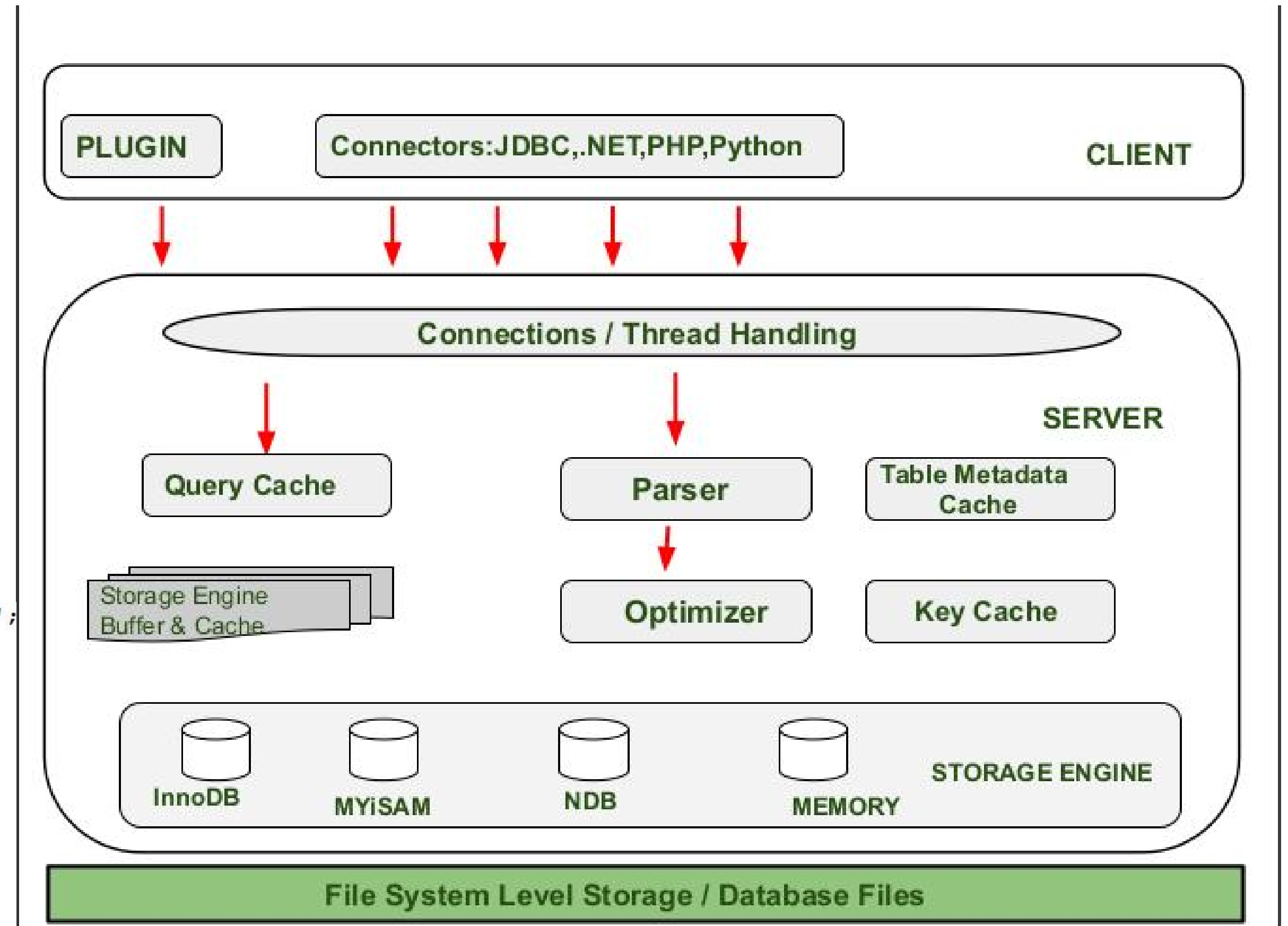
MySQL engine

Storage engine

```
mysql> show global status like '%handler%';
```

Variable_name	Value
Handler_commit	729
Handler_delete	8
Handler_discover	0
Handler_external_lock	7227
Handler_mrr_init	0
Handler_prepare	0
Handler_read_first	53
Handler_read_key	2183
Handler_read_last	0
Handler_read_next	4599
Handler_read_prev	0
Handler_read_rnd	0
Handler_read_rnd_next	4744
Handler_rollback	0
Handler_savepoint	0
Handler_savepoint_rollback	0
Handler_update	331
Handler_write	1303

18 rows in set (0.02 sec)



발표자: 박지원

Architecture

Storage engine

InnoDB: 트랜잭션, 대용량

MyISAM: 트랜잭션 X, Read Only

MEMORY

Table 18.1 Storage Engines Feature Summary

Feature	MyISAM	Memory	InnoDB	Archive	NDB
B-tree indexes	Yes	Yes	Yes	No	No
Backup/point-in-time recovery (note 1)	Yes	Yes	Yes	Yes	Yes
Cluster database support	No	No	No	No	Yes
Clustered indexes	No	No	Yes	No	No
Compressed data	Yes (note 2)	No	Yes	Yes	No
Data caches	No	N/A	Yes	No	Yes
Encrypted data	Yes (note 3)	Yes (note 3)	Yes (note 4)	Yes (note 3)	Yes (note 5)
Foreign key support	No	No	Yes	No	Yes
Full-text search indexes	Yes	No	Yes (note 6)	No	No
Geospatial data	Yes	No	Yes	Yes	Yes

발표자: 박지원

Architecture

Mysql threading

foreground thread

innnoDB: buffer, cache, read

MyISAM: + read and write

background thread

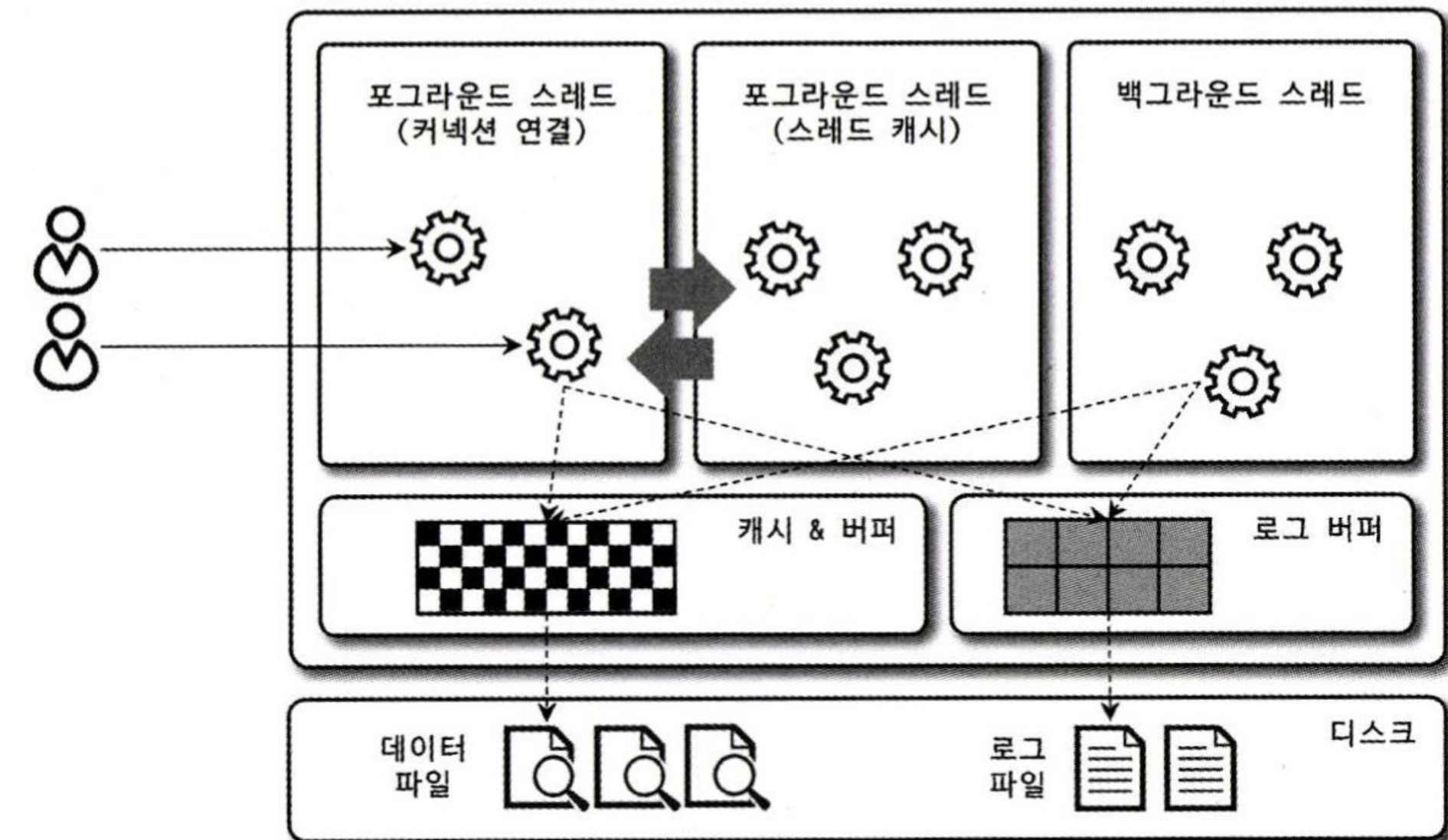
insert buffer merge (change buffer)

log write (log thread)

buffer pool write (write thread)

buffer read (read thread)

lock monitor



발표자: 박지원

Architecture

Shared Memory (global)

Buffer Pool

Change Buffer

Log Buffer

Key Cache

Table Cache

Local Memory (per thread)

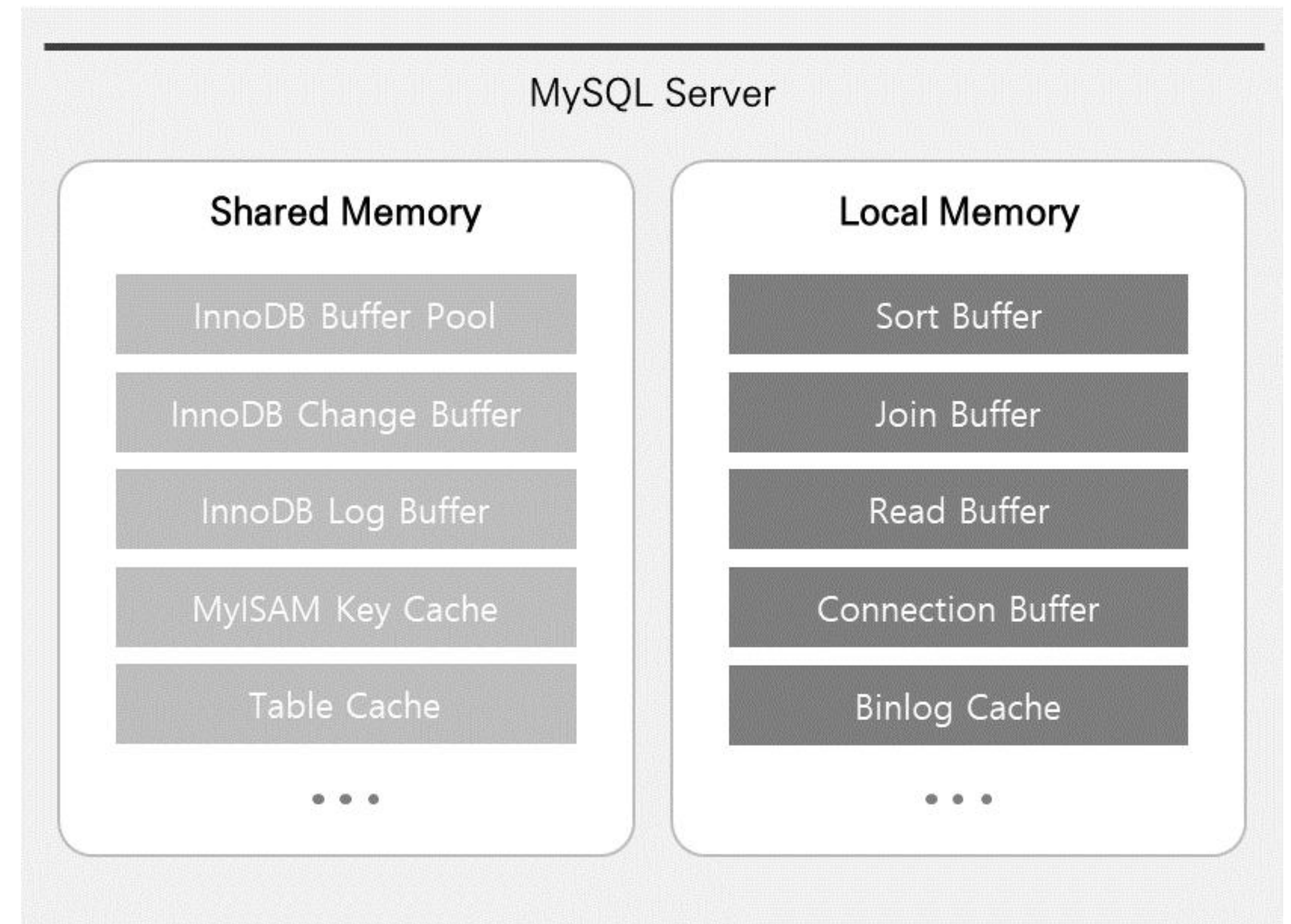
sort buffer

join buffer

read buffer

connection buffer

Binlog Cache(Transaction)



발표자: 박지원

Architecture

query parser

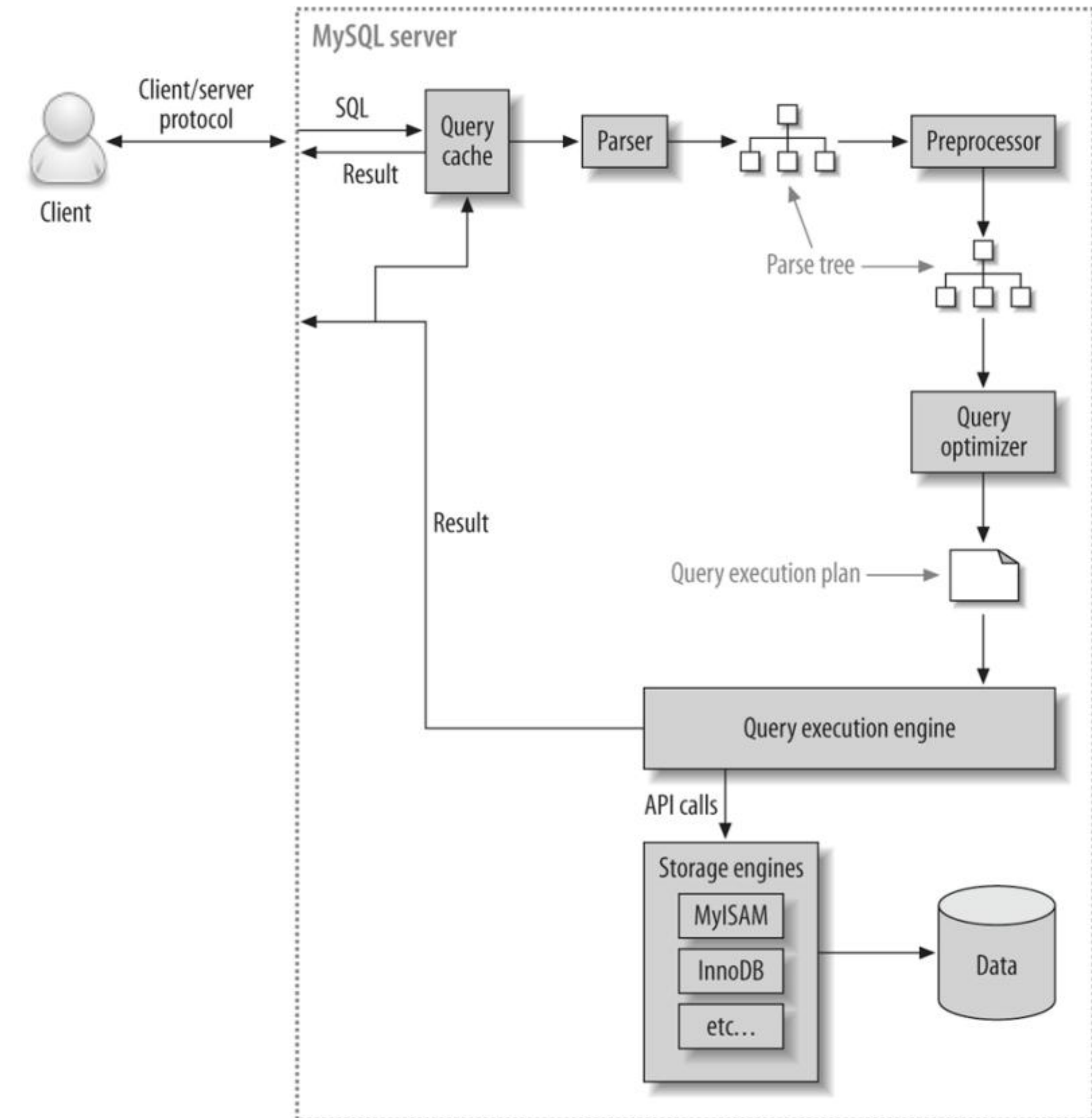
preprocessor

optimizer

execution engine

handler API

Query cache → Deprecated → Delete



발표자: 박지원

Query cache

string match

queries that are a subquery of an outer query

SELECT privilege for all databases

change → removed table relative cache

query cache lock (per table)

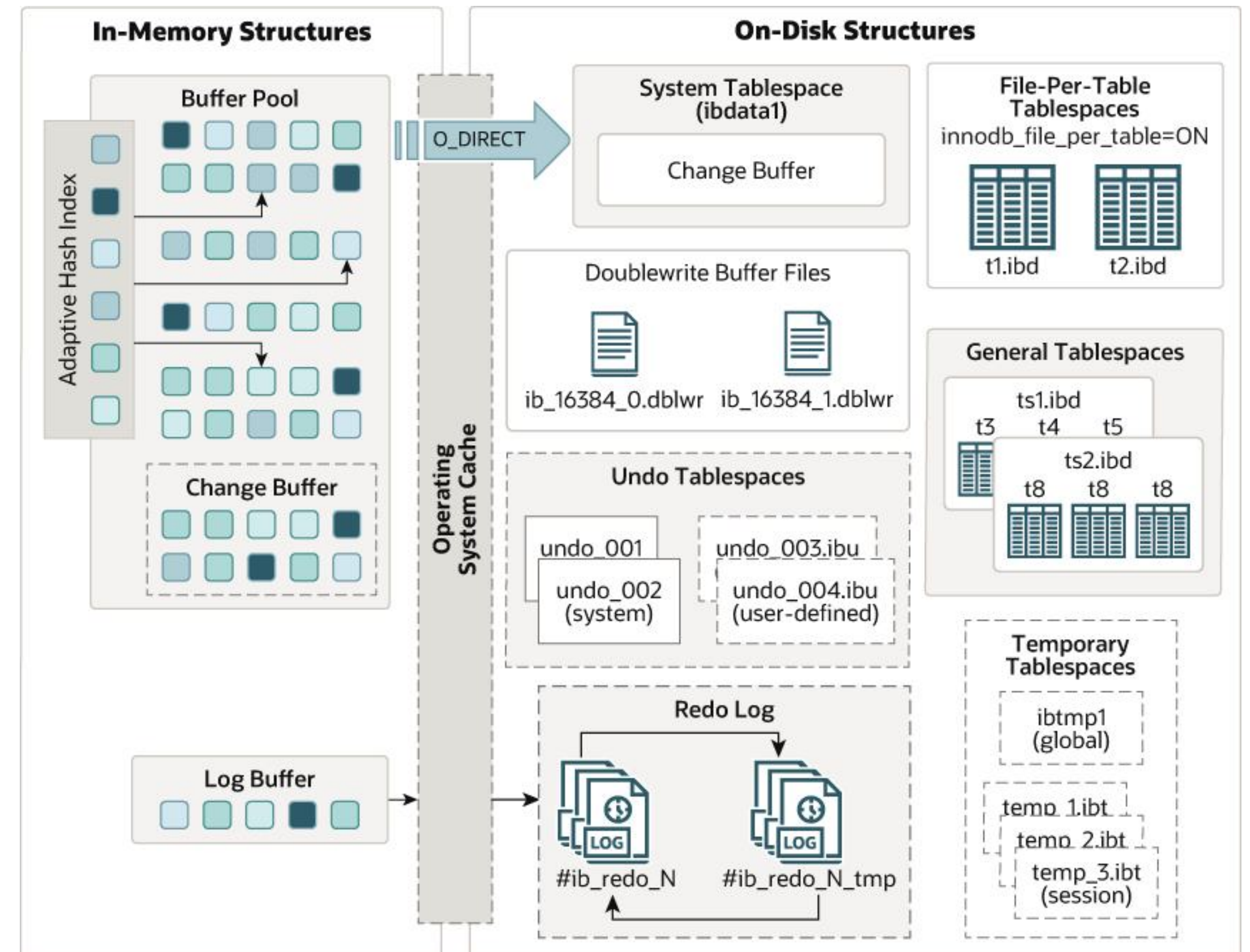


```
SELECT * FROM tbl_name  
Select * from tbl_name
```


InnoDB

In-memory Structure

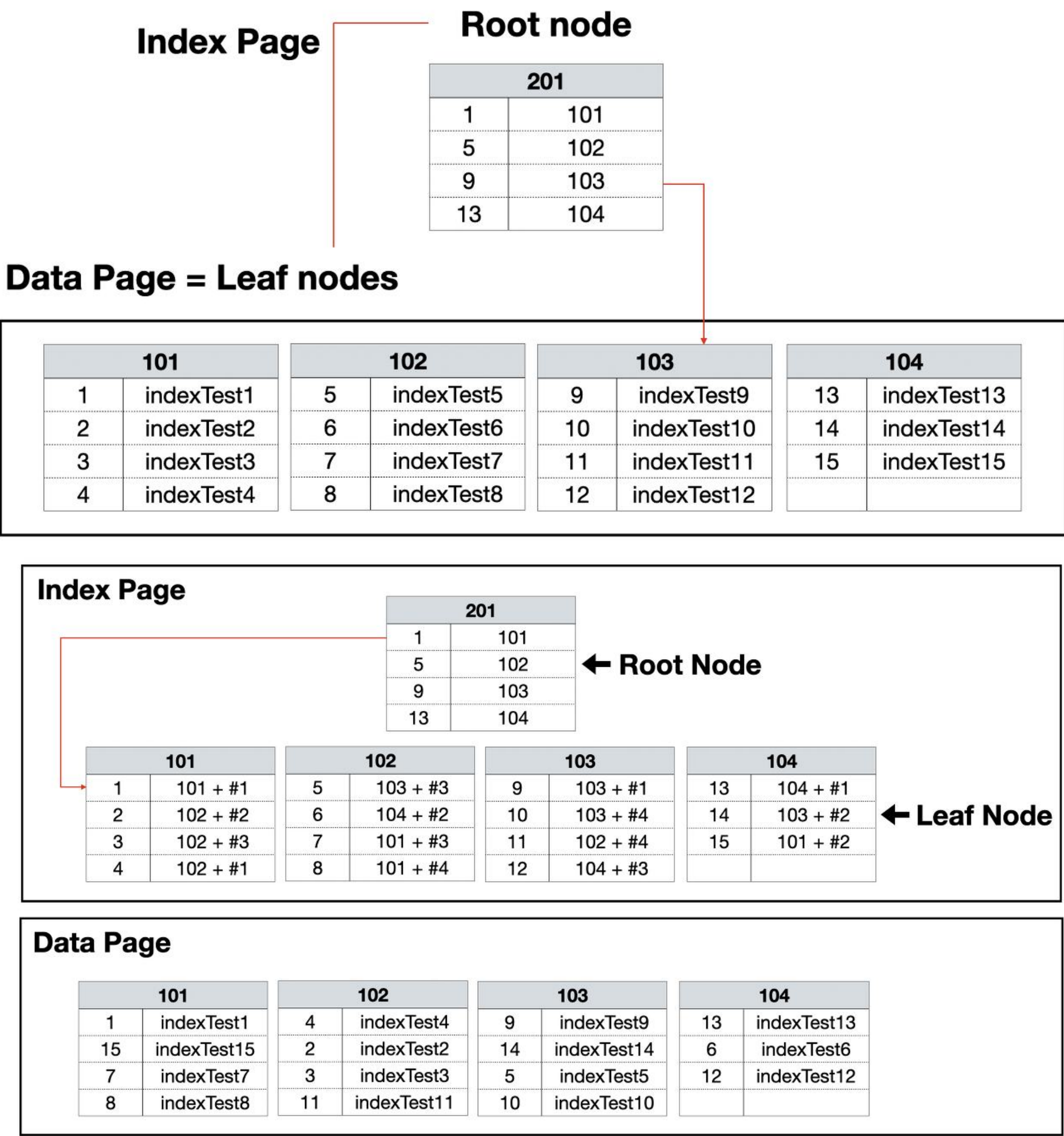
On-Disk Structure



발표자: 박지원

In-memory Structure

클러스터링 인덱스, 논클러스터 인덱스



발표자: 박지원

In-memory Structure

Buffer pool

midpoint insert

New Sublist → LRU

Old Sublist → MRU

Change buffer

Adaptive Hash Index

Log buffer

Figure 17.2 Buffer Pool List

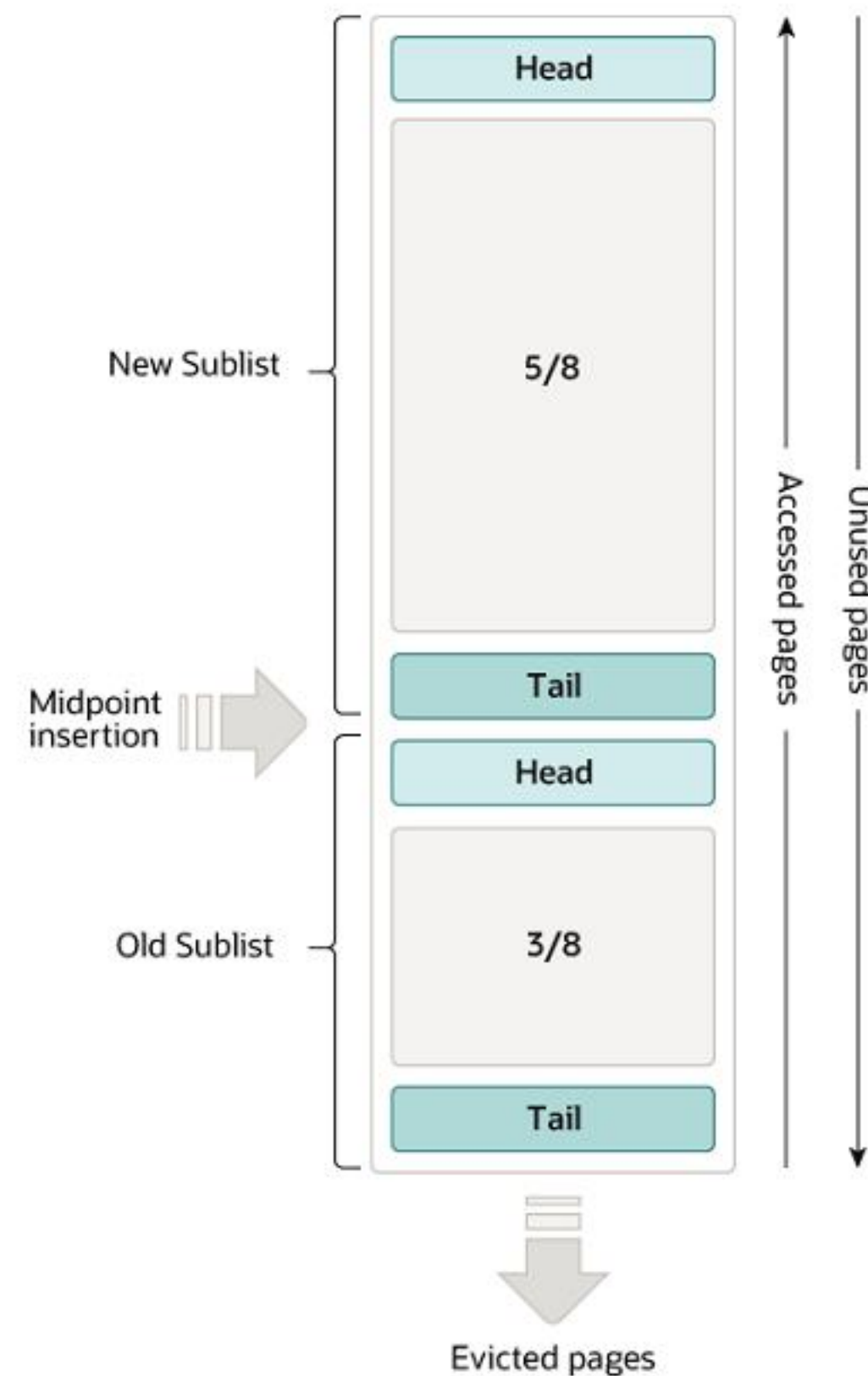
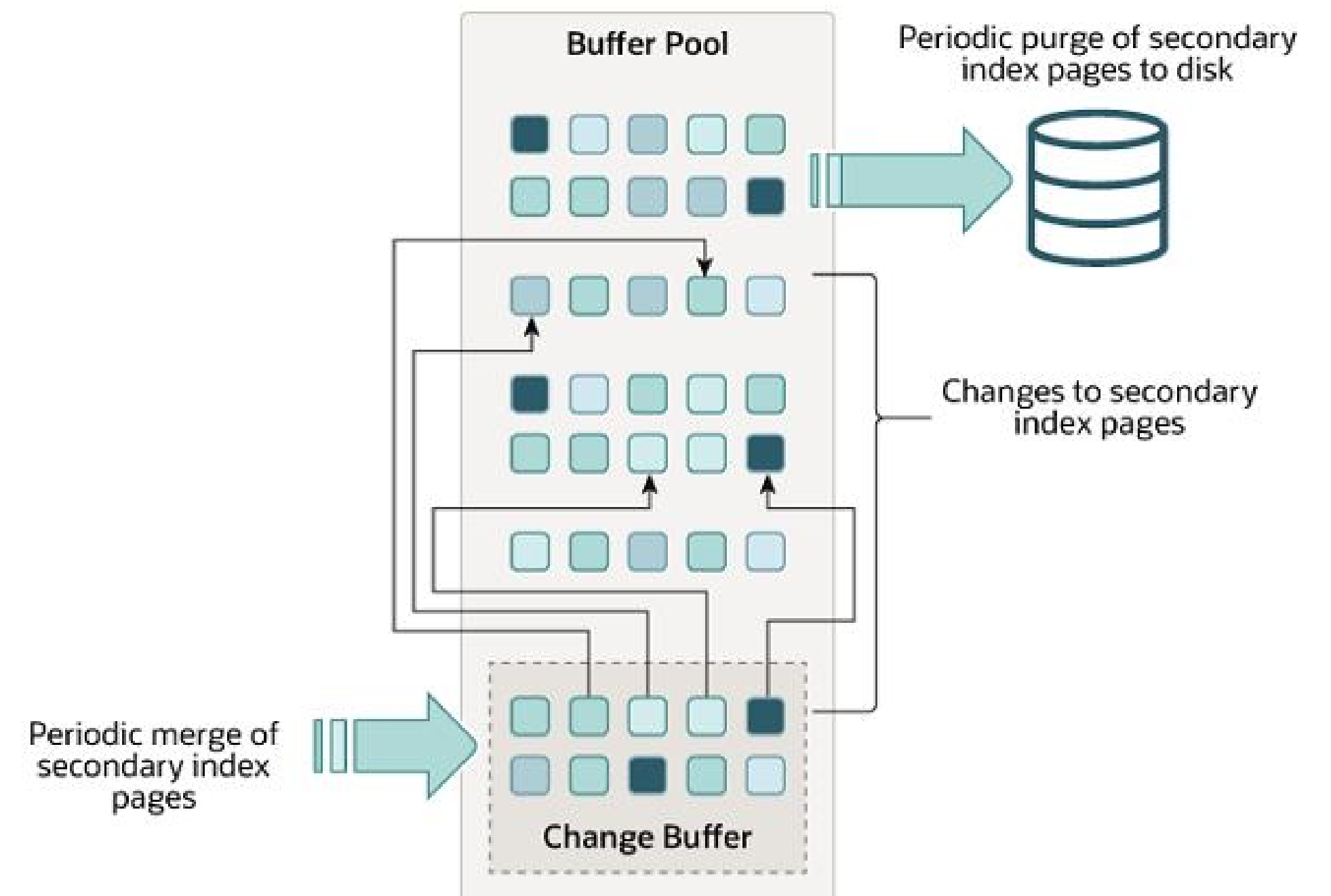
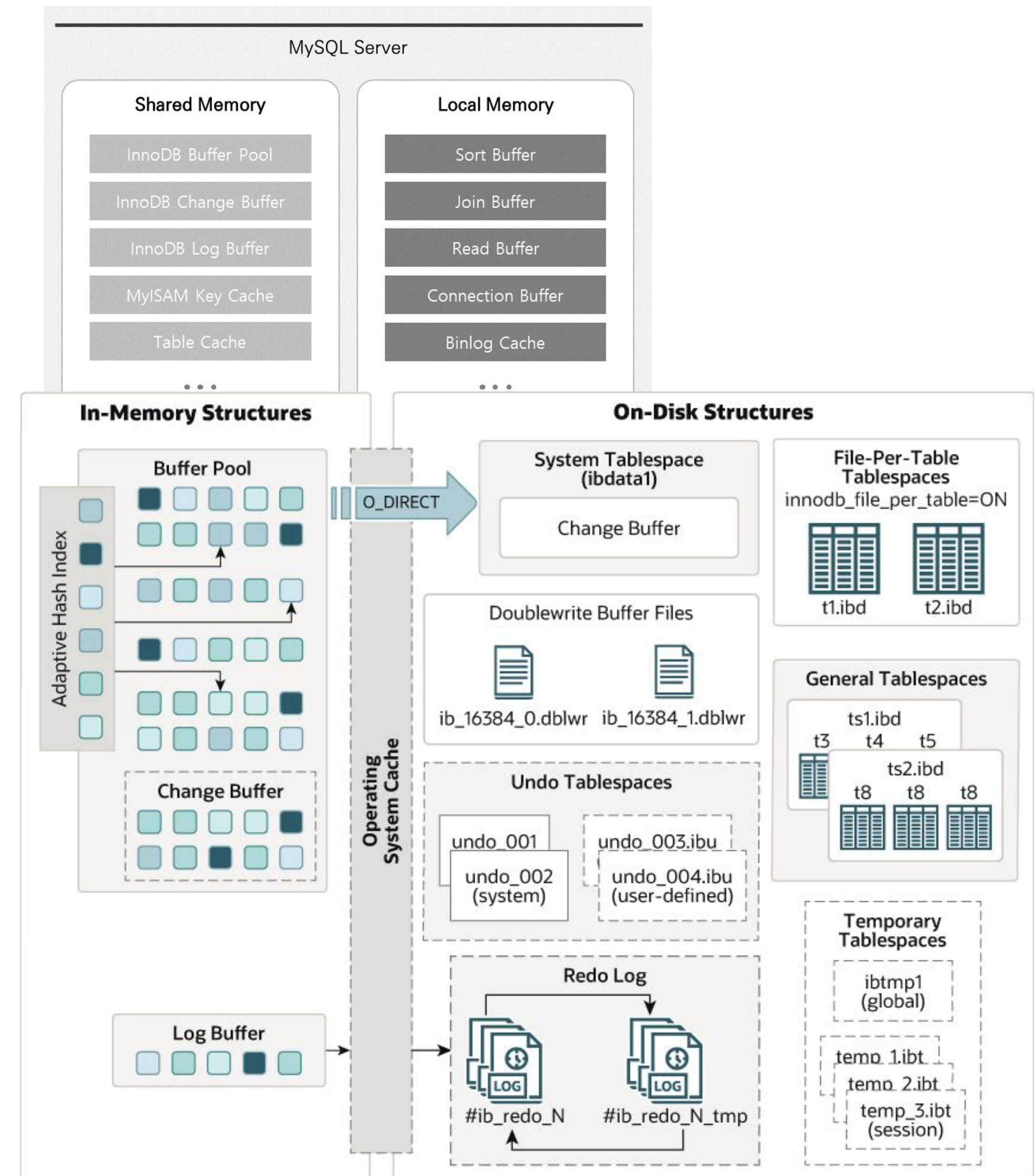
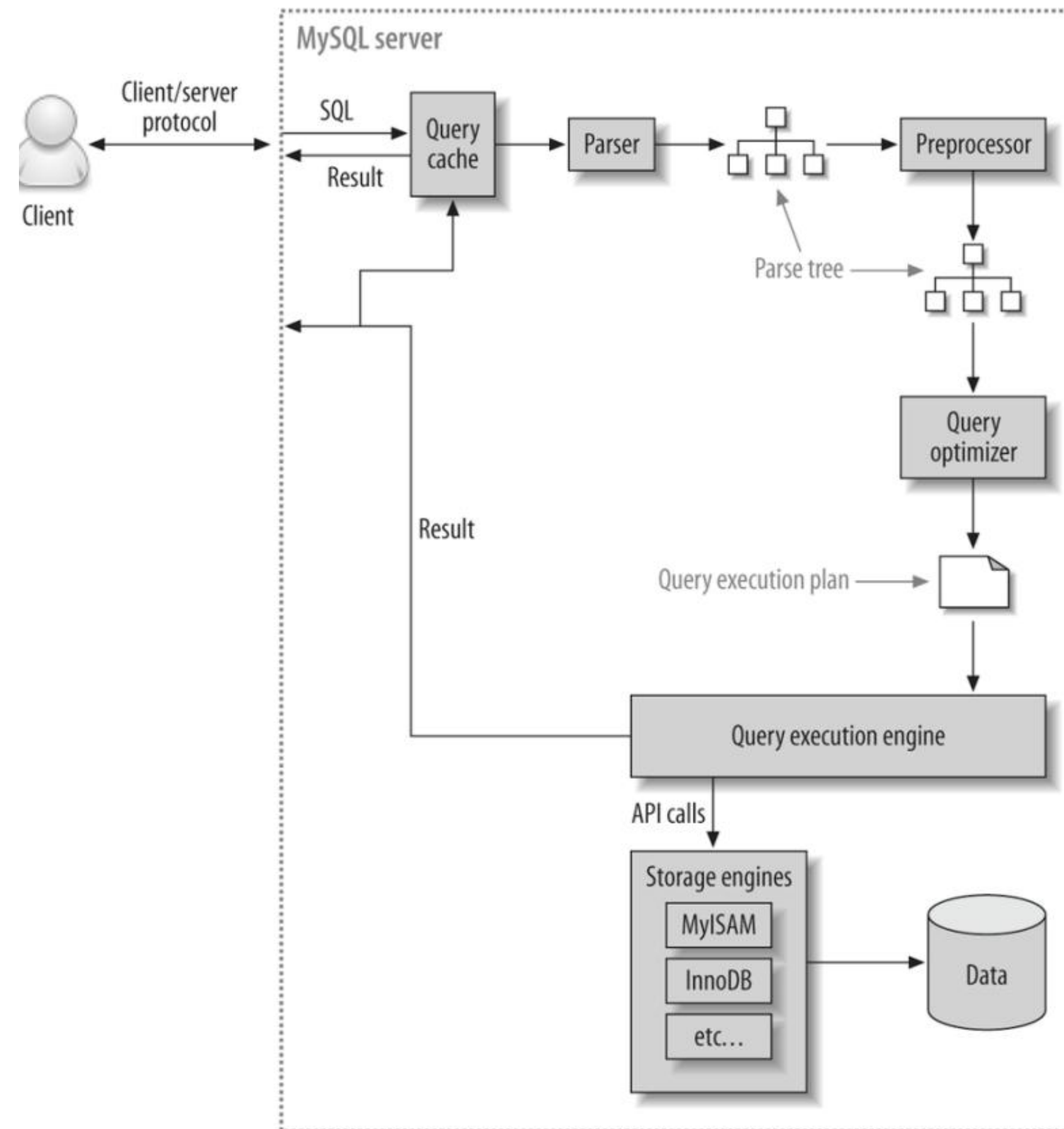


Figure 17.3 Change Buffer



Summary



발표자: 박지원

용어 정리

필드, 레코드, 튜플, ROW, COLUMN, 속성

슈퍼키: 유일성을 보장하는 속성들의 집합, 최소성은 만족하지 않는다.

후보키: 유일성과 최소성을 보장하는 집합

기본키: 후보키 중 하나

대체키: 기본키를 제외한 후보키

Result Grid					Filter Rows:	Export:	Wrap Cell Content:
	actor_id	first_name	last_name	last_update			
▶	1	PENELOPE	GUINNESS	2006-02-15 04:34:33			
	2	NICK	WAHLBERG	2006-02-15 04:34:33			
	3	ED	CHASE	2006-02-15 04:34:33			
	4	JENNIFER	DAVIS	2006-02-15 04:34:33			
	5	JOHNNY	LOLLOBRIGIDA	2006-02-15 04:34:33			
	6	BETTE	NICHOLSON	2006-02-15 04:34:33			
	7	GRACE	MOSTEL	2006-02-15 04:34:33			
	8	MATTHEW	JOHANSSON	2006-02-15 04:34:33			
	9	JOE	SWANK	2006-02-15 04:34:33			
	10	CHRISTIAN	GABLE	2006-02-15 04:34:33			
	11	ZERO	CAGE	2006-02-15 04:34:33			
	12	KARL	BERRY	2006-02-15 04:34:33			
	13	UMA	WOOD	2006-02-15 04:34:33			
	14	VIVIEN	BERGEN	2006-02-15 04:34:33			
	15	CUBA	OLIVIER	2006-02-15 04:34:33			
	16	FRED	COSTNER	2006-02-15 04:34:33			
	17	HELEN	VOIGHT	2006-02-15 04:34:33			

PK

기본키는 수정이 가능할까요?

사실 MySQL의 경우, 기본키를 설정하지 않아도 테이블이 만들어집니다. 어떻게 이게 가능한 걸까요?

외래키 값은 NULL이 들어올 수 있나요?

어떤 칼럼의 정의에 UNIQUE 키워드가 붙는다고 가정해 봅시다. 이 칼럼을 활용한 쿼리의 성능은 그렇지 않은 것과 비교해서 어떻게 다를까요?

mysql에서 auto increment는 어떻게 처리될까요? (inno db)

발표자: 박지원

PK

수정이 가능한 하나 위험하다.

기본키를 설정하지 않는 경우에 innoDB에서 유니크 인덱스를 찾아 클러스터링 키로 생성하거나, 내부 컬럼을 추가한다.

가능하다.

unique 에 대한 인덱스를 찾으려면 되기 때문에 빠르다고 생각한다. (중복 체크에서 추가비용이 발생할 수 있다.)

AI

mysql에서 auto increment는 어떻게 처리될까요?

	id	name
▶	1	박지원
	2	강대훈
●	NULL	NULL

```
UPDATE member  
SET id = 3  
WHERE name = '박지원';
```

	id	name
▶	2	강대훈
	3	박지원
●	NULL	NULL

```
insert into member (name) values ("홍승택");
```

발표자: 박지원

AI

mysql에서 auto increment는 어떻게 처리될까요?

	id	name
▶	2	강대훈
	3	박지원
	4	홍승택
★	NULL	NULL



```
insert into member (name) values ("김동민");
```

발표자: 박지원

NoSQL

RDB와 NoSQL의 차이에 대해 설명해 주세요.

NoSQL의 강점과, 약점이 무엇인가요?

RDB의 어떠한 특징 때문에 NoSQL에 비해 부하가 많이 걸릴 "수" 있을까요? (주의: 무조건 NoSQL이 RDB 보다 빠르다라고 생각하면 큰일 납니다!)

NoSQL

RDB는 table 형태로 데이터를 저장하는 것과 달리 NoSQL은 json, key-value 형식으로 데이터를 저장하고 데이터간의 관계를 정의하지 않는다, ~~RDB는 엄격하게 ACID를 지키는 반면 NoSQL은 그렇지 않다.~~

Nosql은 비정형 데이터에 대한 처리에 적합하다. 특정 로직에 대해서 좋은 효과를 낼 수 있다.
(read-only)

JOIN연산에 대해서 Nosql이 더 좋은 성능이점을 얻을 수 있다.

출처

<https://dev.mysql.com/doc/refman/8.4/en/innodb-architecture.html>

발표자: 박지원