INNODB®

Transactional Storage for MySQL
FAST. RELIABLE. PROVEN.

InnoDB Internals: InnoDB File Formats and Source Code Structure

MySQL University, October 2009

Calvin Sun
Principal Engineer
Oracle Corporation

INNOBASE

Today's Topics

- Goals of InnoDB
- Key Functional Characteristics
- InnoDB Design Considerations
- InnoDB Architecture
- InnoDB On Disk Format
- Source Code Structure
- Q & A

Goals of InnoDB

- OLTP oriented
- Performance, Reliability, Scalability
- Data Protection
- Portability

InnoDB Key Functional Characteristics

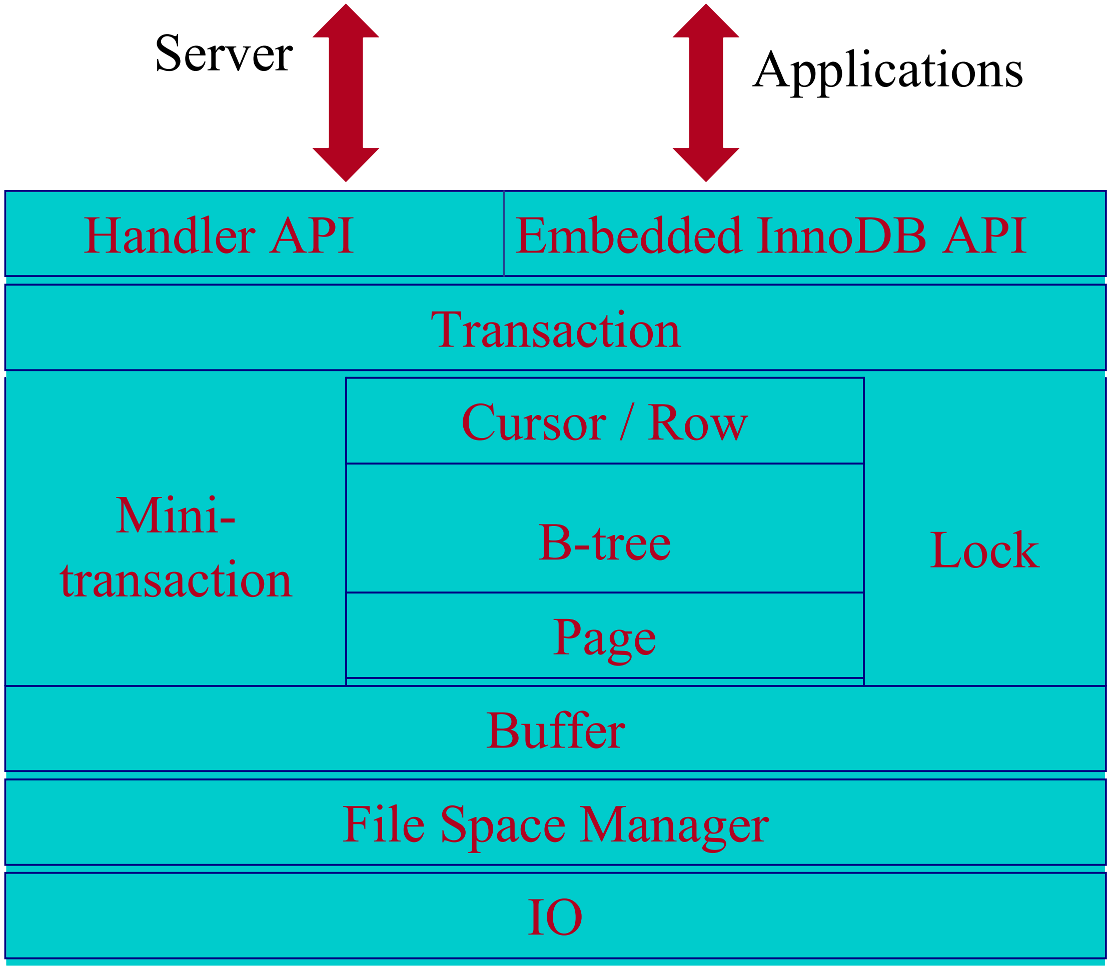
- Full transaction support
- Row-level locking
- MVCC
- Crash recovery
- Efficient IO

INNOBASE

Design Considerations

- Modeled on Gray & Reuter's "*Transactions Processing: Concepts & Techniques*"
- Also emulated the Oracle architecture
- Added unique subsystems
  - Doublewrite
  - Insert buffering
  - Adaptive hash index
- Designed to evolve with changing hardware & requirements

InnoDB Architecture



Server

Applications

Handler API

Embedded InnoDB API

Transaction

Mini-transaction

Cursor / Row

B-tree

Page

Lock

Buffer

File Space Manager

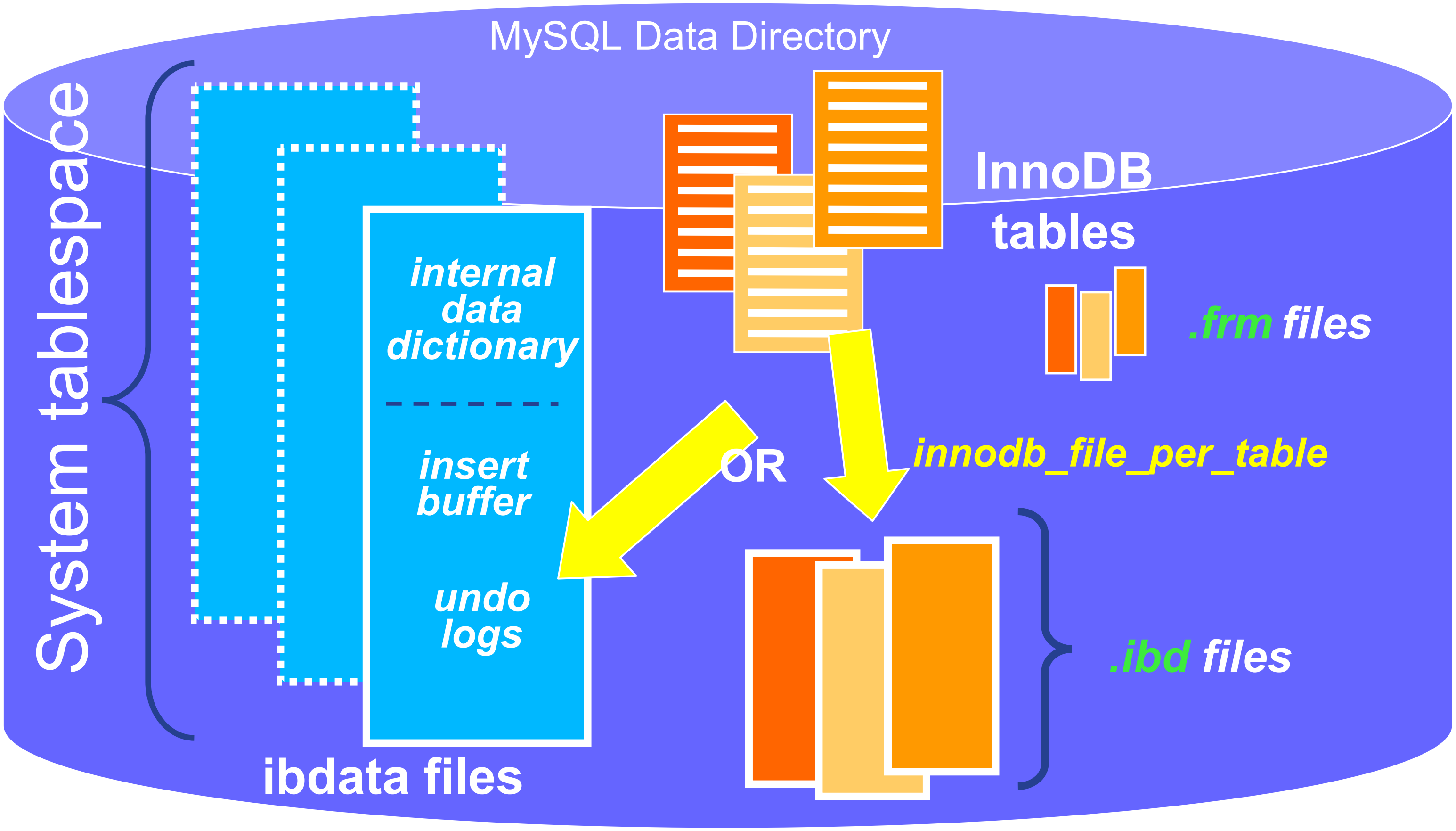
IO

INNOBASE

InnoDB On Disk Format

- InnoDB Database Files
- InnoDB Tablespaces
- InnoDB Pages / Extents
- InnoDB Rows
- InnoDB Indexes
- InnoDB Logs
- File Format Design Considerations

InnoDB Database Files



INNOBASE

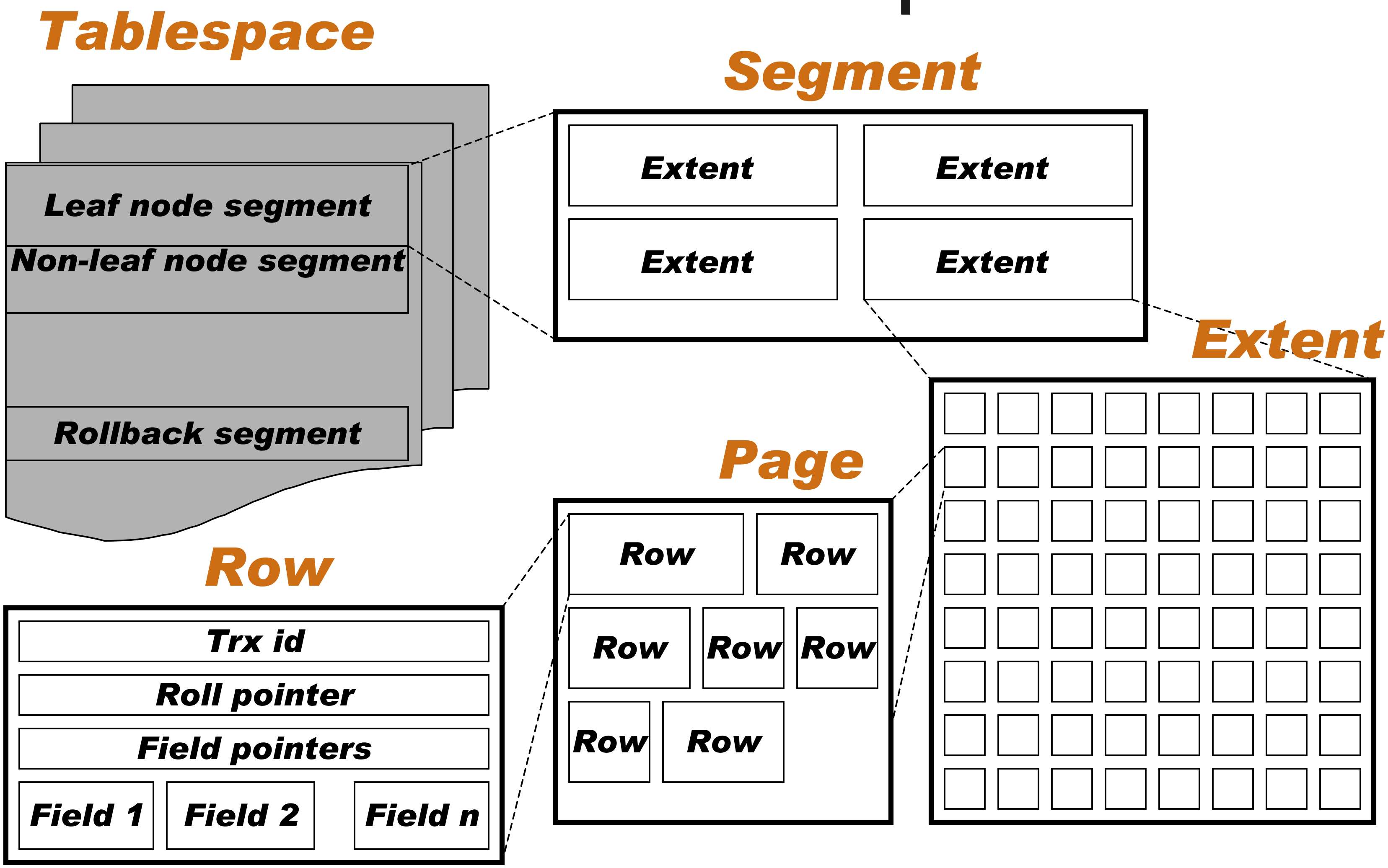
InnoDB Tablespaces

- A tablespace consists of multiple files and/or raw disk partitions.
  file_name:*file_size*[:autoextend[:max:*max_file_size*]]
- A file/partition is a collection of segments.
- A segment consists of fixed-length pages.
- The page size is always 16KB in uncompressed tablespaces, and 1KB-16KB in compressed tablespaces (for both data and index).

System Tablespace

- Internal Data Dictionary
- Undo
- Insert Buffer
- Doublewrite Buffer
- MySQL Replication Info

InnoDB Tablespaces



an extent = 64 pages

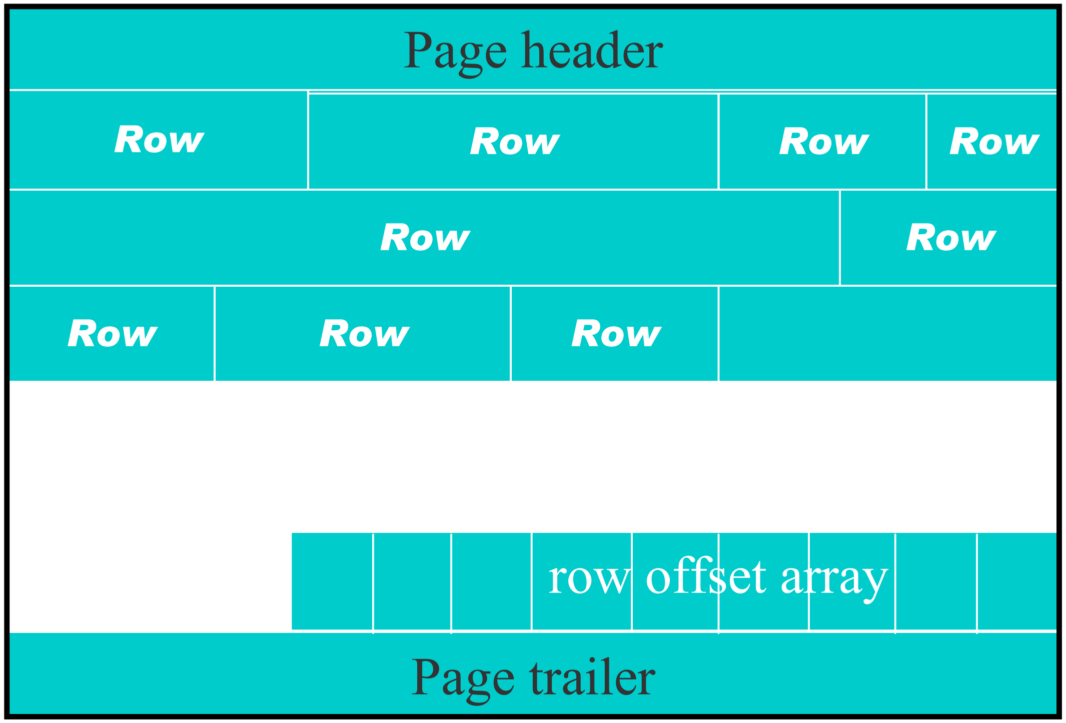
INNOBASE

InnoDB Pages

InnoDB Page Types		
Symbol	Value	Notes
FIL_PAGE_INODE	3	File segment inode
FIL_PAGE_INDEX	17855	B-tree node
FIL_PAGE_TYPE_BLOB	10	Uncompressed BLOB page
FIL_PAGE_TYPE_ZBLOB	11	1st compressed BLOB page
FIL_PAGE_TYPE_ZBLOB2	12	Subsequent compressed BLOB page
FIL_PAGE_TYPE_SYS	6	System page
FIL_PAGE_TYPE_TRX_SYS	7	Transaction system page
others		i-buf bitmap, I-buf free list, file space header, extent desp page, new allocated page

InnoDB Pages

A page consists of: a page header, a page trailer, and a page body (rows or other contents).



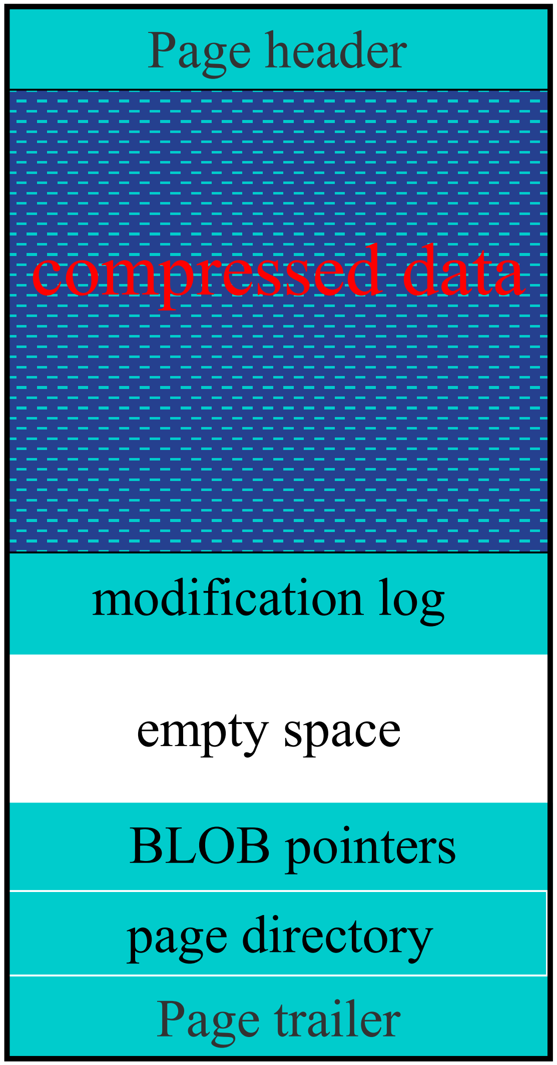
Page Declares

```
typedef struct                   /* a space address */
   {
     ulint     pageno;           /* page number within the file */
     ulint     boffset;          /* byte offset within the page */
   }  fil_addr_t;

typedef struct
  {
   ulint      checksum;    /* checksum of the page (since 4.0.14) */
   ulint      page_offset; /* page offset inside space */
   fil_addr_t previous;    /* offset or fil_addr_t */
   fil_addr_t next;        /* offset or fil_addr_t */
   dulint     page_lsn;    /* lsn of the end of the newest
                              modification log record to the page */
   PAGE_TYPE  page type;   /* file page type */
   dulint     file_flush_lsn;/* the file has been flushed to disk
                             at least up to this lsn */
   int        space_id;   /* space id of the page */
   char       data[];     /* will grow */
   ulint      page_lsn;   /* the last 4 bytes of page_lsn */
   ulint      checksum;   /* page checksum, or checksum magic, or 0 */
   } PAGE, *PAGE;
```

INNOBASE

InnoDB Compressed Pages



Page header

compressed data

modification log

empty space

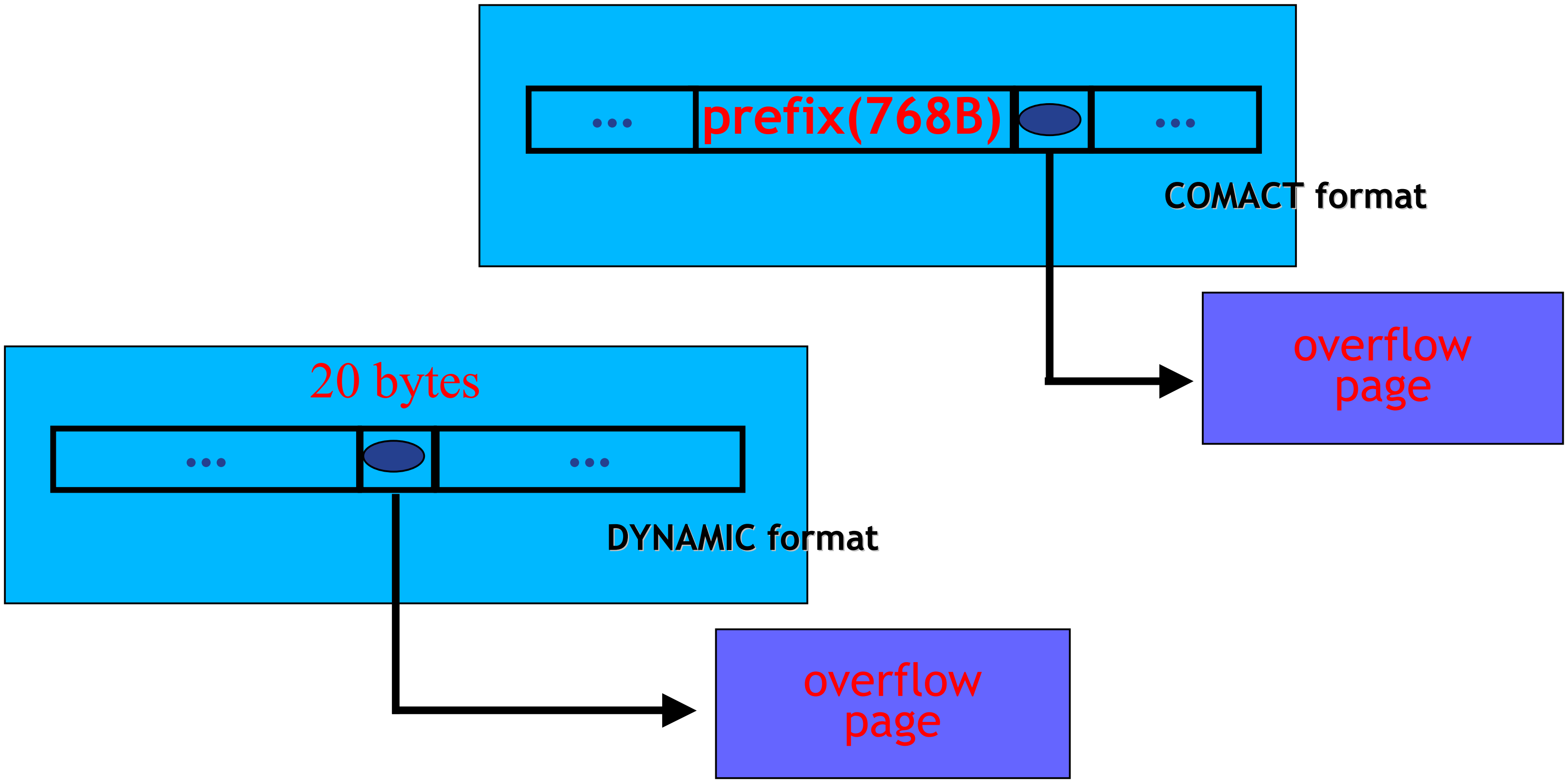
BLOB pointers

page directory

Page trailer

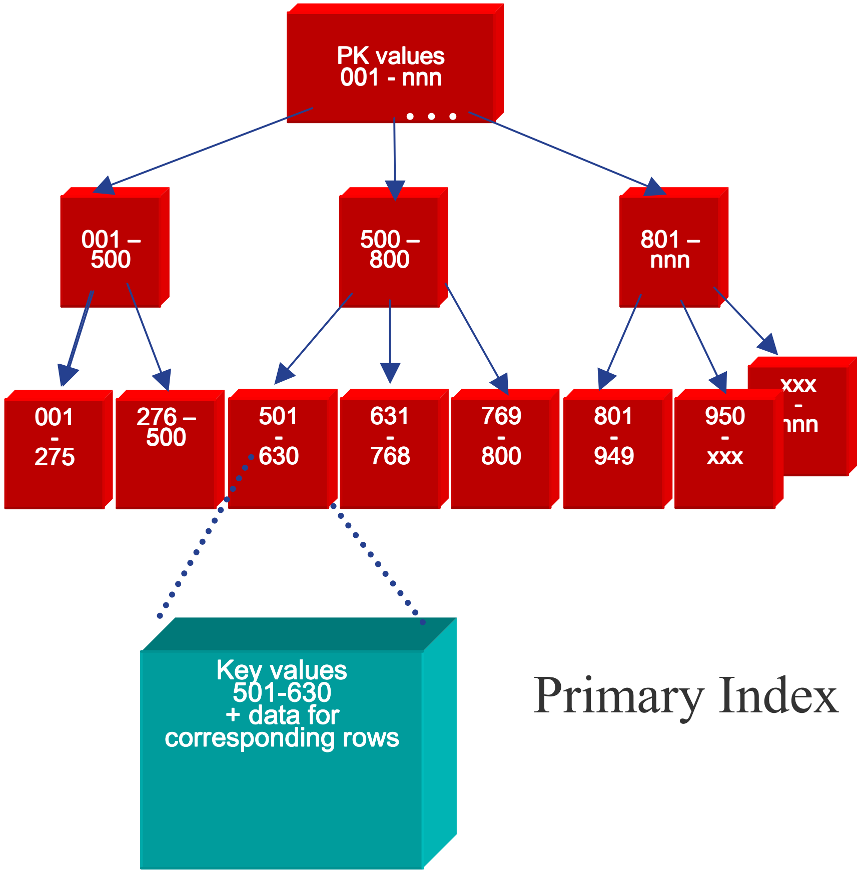
- InnoDB keeps a “modification log” in each page
- Updates & inserts of small records are written to the log w/o page reconstruction; deletes don’t even require uncompression
- Log also tells InnoDB if the page will compress to fit page size
- When log space runs out, InnoDB uncompresses the page, applies the changes and recompresses the page

InnoDB Rows



Record hdr	***Trx ID***	***Roll ptr***	***Fld ptrs***	***overflow-page ptr ..***	***Field values***

InnoDB Indexes - Primary



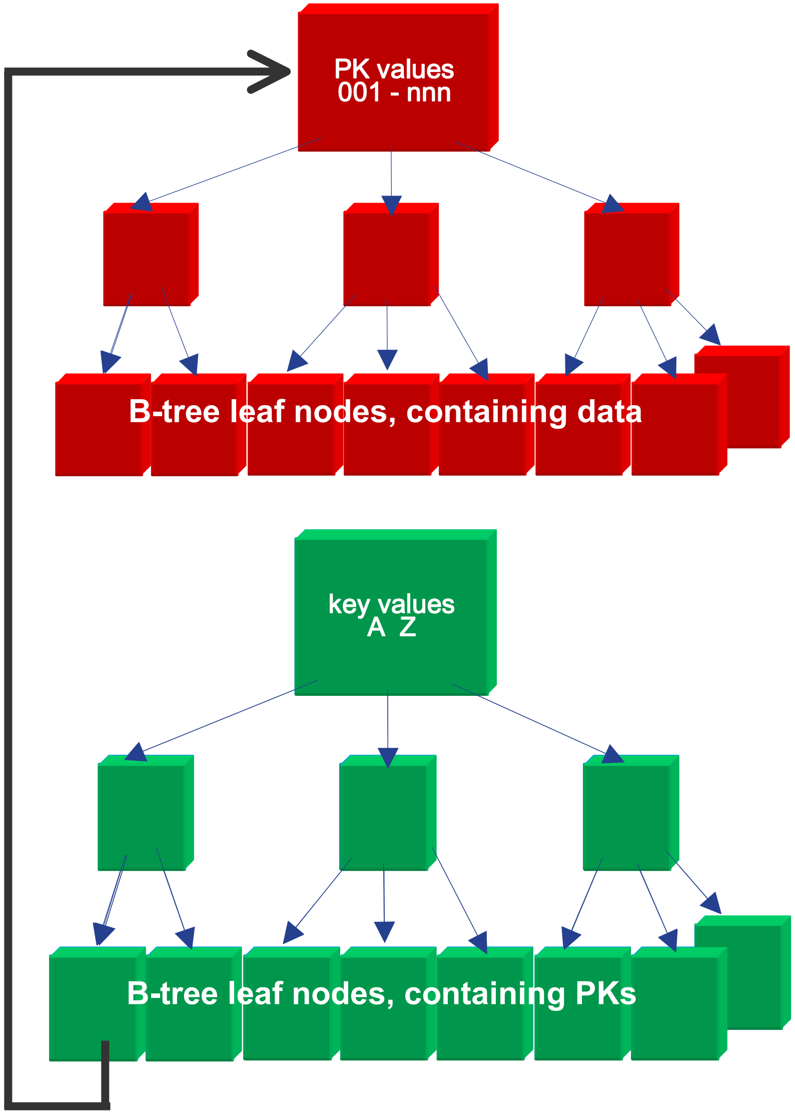
Primary Index

- Data rows are stored in the B-tree leaf nodes of a clustered index
  - B-tree is organized by primary key or non-null unique key of table, if defined; else, an internal column with 6-byte ROW_ID is added.

INNOBASE

InnoDB Indexes - Secondary

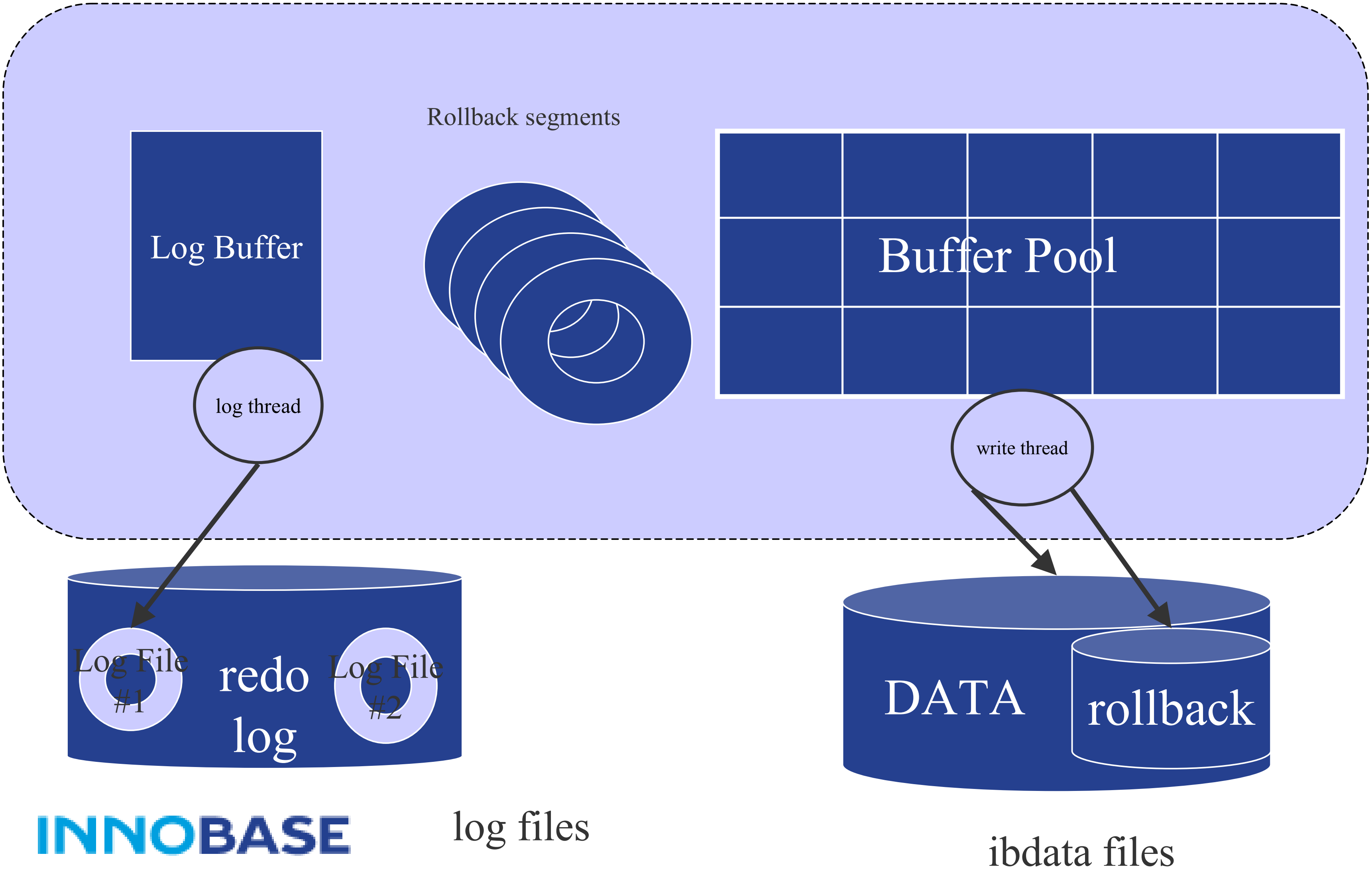
- Secondary index B-tree leaf nodes contain, for each key value, the primary keys of the corresponding rows, used to access clustering index to obtain the data


PK values
001 - nnn
B-tree leaf nodes, containing data
key values
A Z
B-tree leaf nodes, containing PKs

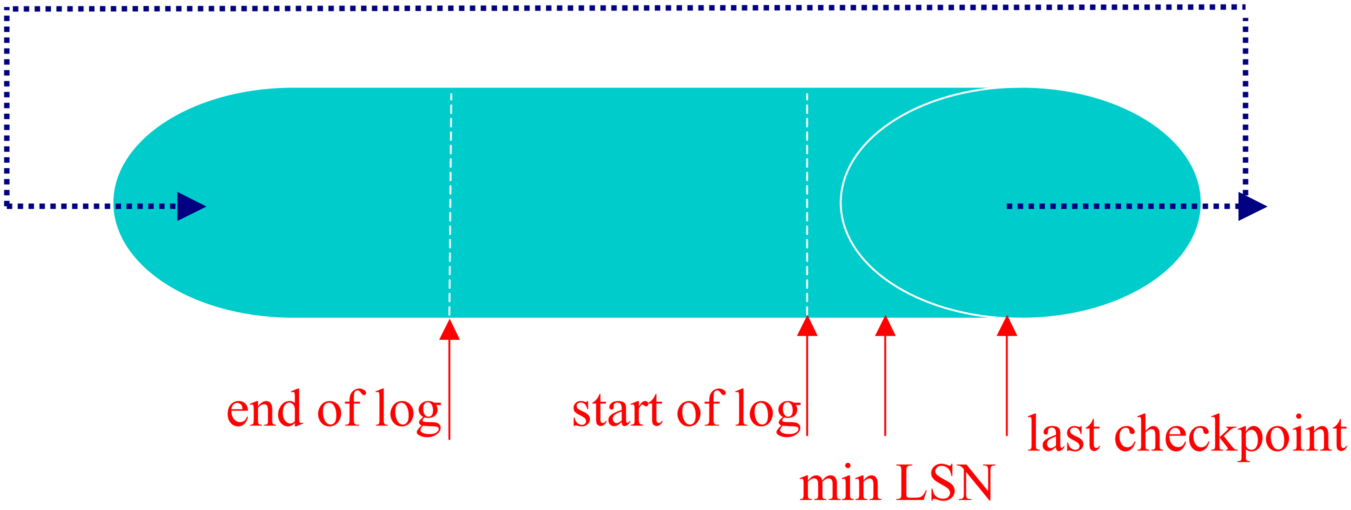
Secondary Index

INNOBASE

InnoDB Logging



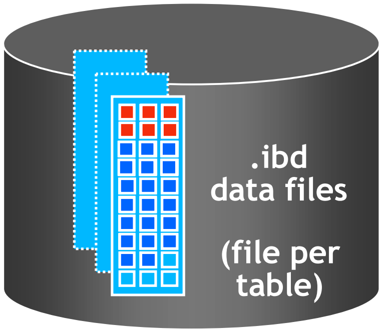
InnoDB Redo Log



Redo log structure:

Space id	*PageNo*	*OpCode*	*Data*

File Format Management



- Builtin InnoDB format: “Antelope”
- New “Barracuda” format enables compression,ROW_FORMAT=DYNAMIC
  - Fast index creation, other features do not require Barracuda file format
- Builtin InnoDB can access “Antelope” databases, but not “Barracuda” databases
  - Check file format tag in system tablespace on startup
- Enable a file format with new dynamic parameter innodb_file_format
- Preserves ability to downgrade easily

INNOBASE

InnoDB File Format Design Considerations

- Durability
  - Logging, doublewrite, checksum;
- Performance
  - Insert buffering, table compression
- Efficiency
  - Dynamic row format, table compression
- Compatibility
  - File format management

Source Code Structure

- 31 subdirectories
- Relevant InnoDB source files on file formats
  - Tablespace: fsp0fsp {.c, .ic, .h}
  - Page: page0page, page0zip {.c, .ic, .h}
  - Log: log0log {.c, .ic, .h}

Source Code Subdirectories

- buf
- data
- db
- dict
- dyn
- eval
- fil
- fsp
- fut
- ha
- handler
- ibuf
- include
- lock
- log
- math
- mem
- mtr
- os
- page
- pars
- que
- read
- rem
- row
- srv
- sync
- thr
- trx
- usr
- ut

Summary:
Durability, Performance, Compatibility & Efficiency

- InnoDB is the leading transactional storage engine for MySQL
- InnoDB's architecture is well-suited to modern, on-line transactional applications; as well as embedded applications.
- InnoDB's file format is designed for high durability, better performance, and easy to manage

Q&A

QUESTIONS

ANSWERS

INNOBASE

InnoDB Size Limits

- Max # of tables: ***4 G***
- Max size of a table: ***32TB***
- Columns per table: ***1000***
- Max row size: n****4 GB***
  - 8 kB if stored on the same page
  - n*4 GB with n BLOBs
- Max key length: ***3500***
- Maximum tablespace size: ***64 TB***
- Max # of concurrent trxs: ***1023***

INNOBASE