

DSCI 551 – Spring 2026

Homework 2: MySQL InnoDB Storage Engine (100 points)

Due: 11:59pm, February 6, 2025, Friday

NO Late Submissions will be accepted!

1. Background: As discussed in class, MySQL stores data and metadata of a table in a .ibd (InnoDB tablespace) file. In this assignment, you will analyze the **physical structure of .ibd files** by reading and interpreting raw bytes.

Note that this homework is about **storage-engine internals**, not SQL. It connects to the class discussions on storage and file systems.

InnoDB organizes the .ibd file into pages and the default page size is 16KB.

Generic page layout:

File Header (38 bytes)
Page Header (56 bytes)
Page Body
File Trailer (8 bytes)

The following fields in the file header are needed for this homework. Note offset starts from the beginning of the page.

Field	Offset	Purpose
FIL_PAGE_OFFSET	4–7	Page number within the tablespace
FIL_PAGE_TYPE	24–25	Page type identifier

Different page types interpret the *page body* differently, but the **outer structure is always the same**.

InnoDB stores metadata (table name, attributes, data types, indexes, etc.) of the table in a table with type SDI (page type value = 0x45BD). The metadata are stored in records that are linked together. The linked list has two sentinels (infimum and supremum). You can think of them as the first and the last record in the linked list. The sentinels are there so that you know where to start and where the end of the list is.

The layout of SDI page is as follows.

File Header (38 bytes)

Page Header (56 bytes)
Infimum record
Supremum record
User / SDI records
Free space
Page directory
File trailer (8 bytes)

Note that InnoDB stores two sentinel records first (they are physically next to each other).

Note that the Infimum record starts at offset 94. Each record (sentinel, user/sdi) has a record header, which stores the following information:

Record Header (all records, 5 bytes)

Field	Offset (from record start)
info_bits	+0
n_owned	+1–2
next_record	+3–4

Note that next_record field stores the relative offset of next record (relative to the beginning of the current record). Note the next_record field of the supremum field is zero (end of linked list).

SDI Record Fixed Fields

```

record_start
├— record header (5 bytes)
├— object_type  (4 bytes)
├— object_id   (8 bytes)
├— DB_TRX_ID   (6 bytes)
├— DB_ROLL_PTR (7 bytes)
└— payload (inline or external)

```

Note that the four fields required in the assignment are right after the record header with offsets shown above.

2. Learning Objectives: By completing this homework, you should be able to:

1. Explain how an .ibd file is organized into pages
2. Identify page types
3. Parse InnoDB **file headers**
4. Traverse records on SDI using the **record header**
5. Understand the structural role of SDI pages
6. Explain why InnoDB separates **logical order** from **physical layout**

3. Files Provided

- hw2_template.ipynb: A partially completed notebook with:
 - constants
 - helper functions
 - function stubs with **TODOs**
- user.ibd: A small MySQL 8.0 per-table tablespace containing:
 - metadata pages
 - SDI pages
 - index pages
 - etc.

4. Assignment Tasks

Part 1 — Page Type Scan (30 points)

Notebook function:

```
def print_page_types(path):
```

Task:

Complete this function to:

- Open the .ibd file
- Read it **page by page** (16 KB each)

- Print:
 - page number
 - page type (using PAGE_TYPES)

Expected output format

Page 0: FSP_HDR
 Page 1: IBUF_BITMAP
 Page 2: INODE
 Page 3: SDI
 Page 4: INDEX
 ...

Key requirements

- Do **not** hard-code page numbers (instead, read it from the page)
- Use FIL_PAGE_TYPE at offset 24–25
- Handle unknown page types safely

Part 2 — SDI Record Structure (20 points)

Notebook function:

```
def print_sdi_record_fields(page, rec_offset):
```

Task

For each SDI record, print:

Record header

- info_bits
- n_owned
- next_record

Fixed SDI fields

- object_type
- object_id
- DB_TRX_ID
- DB_ROLL_PTR

Do **not** decode or interpret the payload.

Output example

SDI record at offset 420

Record header:

info_bits = 0x00

n_owned = 24

next_record = 65238

Fixed SDI fields:

object_type = 1

object_id = 1316

DB_TRX_ID = 000000000000

DB_ROLL_PTR = 00000000000000

Payload starts at offset 450

Part 3 — Find SDI pages (30 points)

Notebook function:

```
def find_sdi_pages(path):
```

Task:

Return a list of (page_no, page_bytes) for all SDI pages in the .ibd file.

Part 4 — Conceptual Questions (Markdown answers)

Add **Markdown cells** to the notebook and answer the following.

Question 1 — Page Types and Their Purpose (10 points)

Explain the **purpose of each of the following InnoDB page types**:

- FSP_HDR
- INODE
- XDES
- SDI
- INDEX
- IBUF_BITMAP
- ALLOCATED

For each type, describe:

- what kind of information it stores
- why it is necessary in the storage engine

Question 2 — Traversing SDI Records (10 points)

Explain, in your own words, **how InnoDB traverses SDI records inside an SDI page**.

Your answer should address:

- the role of the **infimum record**
- why record length is *not* stored
- how next_record defines the traversal order
- why physical record order is irrelevant
- what is the purpose of this line of code: `rec = (rec + next_rel) & 0xFFFF?`

Notes: you may find the following resource useful.

- <https://blog.jcole.us/2013/01/07/the-physical-structure-of-innodb-index-pages/>
Scroll down to discussions on system records (infimum and supremum) and user records. Note sdi page is structured very similar to index page. Also note that this is for older version of InnoDB. Implementation of pages in modern InnoDB might vary.

5. Submissions

1. Rename hw2_template.ipynb to hw2_ipynb. Prepend your name to the file name, e.g., John_Smith_hw2.ipynb. Complete the missing codes shown in the templates. Show the execution result of:
 - a. `print_page_types('user.ibd')`
 - b. `print_all_sdi_records("user.ibd")`
2. Add two markdown cells and write your answers to two conceptual questions stated above.