CARNEGIE MELLON UNIVERSITY
COMPUTER SCIENCE DEPARTMENT
15-445/645 – DATABASE SYSTEMS (FALL 2025)
PROF. ANDY PAVLO

Homework #2 (by Will)

Due: **Sunday Sept 21, 2025 @ 11:59pm**

IMPORTANT:

- Enter all of your answers into **Gradescope by 11:59pm on Sunday Sept 21, 2025.**
- **Plagiarism**: Homework may be discussed with other students, but all homework is to be completed **individually**.

For your information:

- Graded out of **100** points; **3** questions total
- Rough time estimate: ≈4-6 hours (1-1.5 hours for each question)
- Each part is all or nothing. There is no partial credit.

Revision : 2025/09/10 23:30

Question	Points	Score
Slotted Pages and Log-Structured	30	
Storage Models	35	
Database Compression	35	
Total:	100	

Question 1: Slotted Pages and Log-Structured . . . . . [30 points]

(a) **[10 points]** Which problems are associated with the *log-structured storage* in a database system? Select all that apply.

- ☐ Write Amplification
- ☐ Increased Random Writes
- ☐ Increased Random Reads
- ☐ Fragmentation
- ☐ None of the above

(b) **[10 points]** Which problems are associated with the *slotted-page storage* in a database system? Select all that apply.

- ☐ Write Amplification
- ☐ Increased Random Reads
- ☐ Increased Random Writes
- ☐ Fragmentation
- ☐ None of the above

(c) **[10 points]** You are asked to compare *log-structured storage* to *slotted-page storage* for a new system. Ignore any indexes and overhead from metadata. Select all true statements.

- ☐ Both support variable length tuples.
- ☐ Log-structured storage requires less disk space.
- ☐ For an append-only workload, both achieve comparable performance.
- ☐ Slotted-page requires storing schema metadata with the tuple.
- ☐ After lots of insert/update/deletes, only log-structured benefits from maintenance.
- ☐ None of the above are true.

Question 2: Storage Models . . . . . [35 points]

Consider a database with a single table P(player_id,team_id,G_all,HR), where player_id is the *primary key*, and all attributes are the same fixed width. Suppose P has 20,000 tuples that fit into 200 pages. You should ignore any additional storage overhead for the table (e.g., page headers, tuple headers). Additionally, you should make the following assumptions:

- The DBMS does *not* have any additional meta-data.
- P does *not* have any indexes (including for primary key player_id).
- None of P's pages are already in memory. The DBMS can store an infinite number of pages in memory.
- Content-wise, the tuples of P will always make each query run the longest possible and do the most page accesses.
- The tuples of P can be in any order (keep this in mind when computing *minimum* versus *maximum* number of pages that the DBMS will potentially have to read and think of all possible orderings)

(a) Consider the following query:

```
SELECT MAX(HR) FROM P
    WHERE team_id == 15445 ;
```

i. **[5 points]** Suppose the DBMS uses the N-ary storage model (NSM). How many pages will the DBMS potentially have to read from disk to answer this query?
Be sure to keep in mind the assumption about the contents of P.
☐ 1-50 ☐ 51-100 ☐ 101-150 ☐ 151-200 ☐ ≥ 201 ☐ Not possible to determine

ii. **[5 points]** Suppose the DBMS uses the decomposition storage model (DSM) with implicit offsets. How many pages will the DBMS potentially have to read from disk to answer this query?
Be sure to keep in mind the assumption about the contents of P.
☐ 1-50 ☐ 51-100 ☐ 101-150 ☐ 151-200 ☐ ≥ 201 ☐ Not possible to determine

(b) Now consider the following query:

```
SELECT HR, team_id FROM P
   WHERE player_id = 1 OR player_id = 999
```

i. Suppose the DBMS uses the N-ary storage model (NSM).

α) **[5 points]** What is the *minimum* number of pages that the DBMS will potentially have to read from disk to answer this query?
☐ 1 ☐ 2-3 ☐ 4-6 ☐ 7-9 ☐ 10-100 ☐ 101-200 ☐ ≥ 201
☐ Not possible to determine

β) **[5 points]** What is the *maximum* number of pages that the DBMS will potentially have to read from disk to answer this query?
☐ 1 ☐ 2-3 ☐ 4-6 ☐ 7-9 ☐ 10-100 ☐ 101-200 ☐ ≥ 201
☐ Not possible to determine

ii. Suppose the DBMS uses the decomposition storage model (DSM) with implicit offsets.

α) **[5 points]** What is the *minimum* number of pages that the DBMS will potentially have to read from disk to answer this query?
☐ 1-3 ☐ 4-6 ☐ 7-9 ☐ 10-100 ☐ 101-200 ☐ ≥ 201 ☐ Not possible to determine

β) **[5 points]** What is the *maximum* number of pages that the DBMS will potentially have to read from disk to answer this query?
☐ 1-20 ☐ 21-40 ☐ 41-60 ☐ 61-80 ☐ 81-100 ☐ ≥ 101
☐ Not possible to determine

(c) Finally consider the following query:

```
SELECT MIN(player_id) FROM P
   WHERE HR = (SELECT MIN(HR) FROM P);
```

Suppose the DBMS uses the decomposition storage model (DSM) with implicit offsets.

i. **[5 points]** What is the *minimum* number of pages that the DBMS will potentially have to **read from disk** to answer this query?
☐ 1-50 ☐ 51-100 ☐ 101-150 ☐ 151-200 ☐ ≥ 201 ☐ Not possible to determine

Question 3: Database Compression . . . . . [35 points]

(a) **[5 points]** Suppose that the DBMS has a `VARCHAR` column storing the following values:

```
[Milwaukee Braves, Baltimore Orioles, Buffalo Blues,
Brooklyn Gladiators, Chicago Cubs]
```

Which of the following are valid encodings (`uint32`) for this column under dictionary compression as discussed in lecture that will support both point queries and range queries? Select **all** the valid encodings.

☐ `[1,2,3,4,5]`
☐ `[4,0,2,1,3]`
☐ `[79,12,32,33,31]`
☐ `[50,40,20,10,30]`
☐ `[79,12,33,15,32]`
☐ `[49,9,29,19,39]`

(b) **[15 points]** Suppose the DBMS wants to compresses a table `R(a)` using columnar compression. Which of the following compression schemes **will benefit** (when considering space efficiency) from sorting the table before compressing column `a`? Select **all** that apply.

Hint: "Benefit" means that the efficacy of the compression scheme improves on sorted data. You should not make any assumptions about the column type or its distribution of values.

☐ Bit-packing Encoding
☐ Run-length Encoding
☐ Mostly Encoding
☐ Bitmap Encoding
☐ Dictionary Encoding
☐ Delta Encoding
☐ None of the above will benefit.

(c) **[15 points]** A colleague approaches with a list of true and false statements about run-length encoding, delta encoding, bitmap encoding, and dictionary encoding. The colleague wants your assistance in identifying the true statements. Select **all** that apply.

☐ Delta Encoding is good at compressing large text values.
☐ Inserts and updates on high cardinality columns with bitmap encoding has no overhead.
☐ For *point lookup-only* workload, order-preserving dictionary encoding is required.
☐ For a heavy update workload, dictionary encoding and delta encoding always require re-encoding.
☐ Run-length Encoding is effective for compressing low cardinality integer column.
☐ None of the above.