# Storage Models and Compression

Consider a database with a single table R(*q_id, txns, total, failed*), where q_id is the *primary key*, and all attributes are the same fixed width. Suppose R has 20,000 tuples that fit into 100 pages, 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 (e.g., sort order, zone maps).
- R does *not* have any indexes (including for primary key q_id)
- None of the pages are in the buffer pool.

Consider the following query:

```
SELECT total - failed FROM R
    WHERE q_id = 96 AND txns > 420;
```

(a) Suppose the DBMS uses the decomposition storage model (DSM):

i. What is the *minimum* number of pages that the DBMS will potentially have to read from disk to answer this query?

- 1
- 2 – 10
- 11 – 50
- 51 – 100
- > 100

ii. What is the *maximum* number of pages that the DBMS will potentially have to read from disk to answer this query?

- 1
- 2 – 10
- 11 – 50
- 51 – 100
- > 100

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

i. What is the *minimum* number of pages that the DBMS will potentially have to read from disk to answer this query?

- 1
- 2 – 10

- 11 – 50
- 51 – 100
- > 100

ii. What is the *maximum* number of pages that the DBMS will potentiallyhave to read from disk to answer this query?
    - 1
    - 2 – 10
    - 11 – 50
    - 51 – 100
    - > 100