

## Multiple Choice Question (MCQ) from Source Document

[BL-3: Applying]

### Question 1:

A database system is being optimized for a high-frequency OLTP workload where queries typically retrieve only one or two tuples at a time. Which processing model should the developers apply to minimize function call overhead and buffer allocation for intermediate results?

*Options:*

- A) Iterator Model
- B) Materialization Model
- C) Vectorized Model
- D) Pipeline Model

### Correct Answer:

B

[BL-3: Applying]

### Question 2:

Suppose a DBMS is performing a sequential scan on a table and encounters a page where the Zone Map shows a MAX value of 400 and a MIN value of 100 for the 'val' column. If the query is 'SELECT \* FROM table WHERE val > 600', how will the system apply the Zone Map data?

*Options:*

- A) It will scan the page anyway to verify the AVG value.
- B) It will skip the page because the MAX value (400) is less than the predicate value (600).
- C) It will use the SUM aggregate to estimate if the value exists.
- D) It will check the COUNT to determine if it should scan the page.

### Correct Answer:

B

[BL-3: Applying]

### Question 3:

A query 'SELECT \* FROM students WHERE age < 30 AND dept = "CS"' is executed using a Multi-Index Scan. The system has already retrieved a set of Record IDs for 'age < 30' and a separate set for 'dept = "CS"'. What is the next step the system must take to apply this access method correctly?

*Options:*

- A) Perform a union of the two Record ID sets.
- B) Compute the intersection of the Record ID sets using a bitmap or hash table.
- C) Discard the index results and perform a sequential scan.
- D) Retrieve the records for 'age < 30' and then filter for 'CS' manually.

### Correct Answer:

B

[BL-3: Applying]

**Question 4:**

An engineer notices that an 'UPDATE' query on a clustered index is causing a single tuple to be updated multiple times because its physical location changes during the scan. To apply the standard solution for the Halloween Problem, what should the DBMS do?

*Options:*

- A) Switch from an Index Scan to a Sequential Scan.
- B) Use JIT compilation for the update expression.
- C) Track modified record IDs for the duration of the query to prevent re-visiting them.
- D) Materialize the entire table into a temporary buffer before starting the update.

**Correct Answer:**

C

[BL-3: Applying]

**Question 5:**

A DBMS is processing the query: 'SELECT \* FROM products WHERE category = UPPER("electronics")'. Before evaluating this for every tuple in the table, how would the system apply the 'Constant Folding' optimization?

*Options:*

- A) By identifying repeated sub-expressions across multiple queries.
- B) By pre-computing `UPPER("electronics")` to "ELECTRONICS" once and using that result for all comparisons.
- C) By converting the string comparison into a machine code function using LLVM.
- D) By skipping the predicate evaluation if the category index exists.

**Correct Answer:**

B