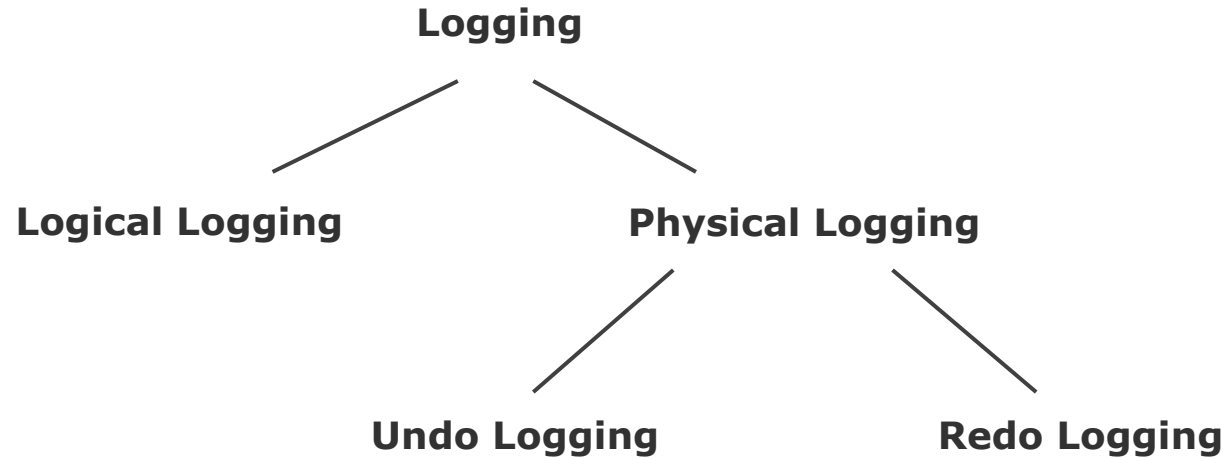


Classification of Logging Approaches



**Logging
Approaches**

Classification of Logging Approaches

Logical vs. Physical Logging

Logical Logging:

- Record the high-level operations executed by transactions
- Example: The UPDATE, DELETE, and INSERT queries invoked by a transaction
 - Writes less data for set operations (faster logging)
 - Queries are reexecuted during recovery
 - Not universal (concurrent queries, non deterministic operations)

Physical Logging:

- Record the changes made to a specific record in the database
- Example: Store the original and new value for an attribute that is changed by a query
 - Logs compressed data
 - Common and general purpose approach

Classification of Logging Approaches

Undo vs. Redo Logging

Undo Logging:

- Record changes to repair the database state by undoing effects of transactions
- Only usable to recover persistent databases

Redo Logging:

- Record changes to repair the database state by redoing effects of transactions
- Applicable for in-memory databases

Undo/Redo Logging:

- Combines both logging styles

**Logging
Approaches**

Classification of Logging Approaches

Undo vs. Redo Logging

- Log entries are stored in a log file using append-only
- Types of log entries:
 - <START T> - Transaction T has started
 - <COMMIT T> - Transaction T is finished
 - <ABORT T> - Transaction T is aborted
 - <T, X, v> - Transaction changes X
 - Undo logging: old value of X is v
 - Redo logging: new value of X is v
 - Undo/redo logging: v is a tuple of X's old and new value

Classification of Logging Approaches

Undo Logging - Example

Transaction T42: INSERT INTO WORLD_POPULATION VALUES("Max", "Meyer", "male", "Germany", "Potsdam", "1986/05/25")

Operation order:

1. Write log entry <START T42>
2. Write log entry <T42, new_entry, WP, ("Max", "Meyer", ...) >
3. Flush log
4. Write changed element
5. Write log entry <COMMIT T42>
6. Flush log

It is sufficient to log the tuple identifier (key or row ID), because there is no old value.

**Logging
Approaches**

Classification of Logging Approaches

Undo Logging – Example Recovery

Transaction T42: INSERT INTO WORLD_POPULATION VALUES("Max", "Meyer", "male", "Germany", "Potsdam", "1986/05/25")

Operation order:

1. Write log entry <START T42>
2. Write log entry <T42, new_entry, WP, ("Max", "Meyer", ...)>
3. Flush log
4. Write changed element
5. Write log entry <COMMIT T42>
6. Flush log



**Logging
Approaches**

Classification of Logging Approaches

Undo Logging – Example Recovery

Transaction T42: INSERT INTO WORLD_POPULATION VALUES("Max", "Meyer", "male", "Germany", "Potsdam", "1986/05/25")

Operation order:

1. Write log entry <START T42>
2. Write log entry <T42, new_entry, WP, ("Max", "Meyer", ...)>
3. Flush log
4. Write changed element
5. Write log entry <COMMIT T42>
6. Flush log

T42 is committed:
Nothing to do



**Logging
Approaches**

Classification of Logging Approaches

Undo Logging – Example Recovery

Transaction T42: INSERT INTO WORLD_POPULATION VALUES("Max", "Meyer", "male", "Germany", "Potsdam", "1986/05/25")

Operation order:

1. Write log entry <START T42>
2. Write log entry <T42, new_entry, WP, ("Max", "Meyer", ...)>
3. Flush log
4. Write changed element
5. Write log entry <COMMIT T42>
6. Flush log

T42 is not committed:
Undo all changes <T42, ...>

Recovery ↑

Logging Approaches

Classification of Logging Approaches

Undo Logging – Example Recovery

Transaction T42: INSERT INTO WORLD_POPULATION VALUES("Max", "Meyer", "male", "Germany", "Potsdam", "1986/05/25")

Operation order:

1. Write log entry <START T42>
2. Write log entry <T42, new_entry, ...>
3. Flush log
4. Write changed element
5. Write log entry <COMMIT T42>
6. Flush log

T42 is not committed:

But no changes in log yet



**Logging
Approaches**