Implementation for Exercise 1

1. In **Predicate.java**, the constructor initializes a Predicate object with a field number, comparison operation, and operand. The getField() method returns the field number of the predicate, while getOp() returns the comparison operation, and getOperand() returns the operand. The filter() method compares a tuple to the predicate and returns true if the comparison is true, false otherwise. The toString() method returns a string representation of the predicate.

2. In **JoinPredicate.java**, the constructor creates a JoinPredicate object over two fields of two tuples with a specified operation. The filter() method applies the predicate to two tuples and returns true if the tuples satisfy the predicate. getField1() returns the field index into the first tuple in the predicate, and getField2() returns the field index into the second tuple. getOperator() returns the operation to apply in the join predicate.

3. In **Filter.java**, the constructor initializes a Filter operator with a predicate and a child operator. The getPredicate() method returns the predicate used by the filter, and getTupleDesc() returns the tuple descriptor of the child operator. The open() method opens the filter operator, close() closes it, rewind() rewinds the child operator, and fetchNext() fetches the next tuple from the child operator that satisfies the predicate.

4. In **Join.java**, the constructor creates a Join operator with a join predicate and two child operators. The getJoinPredicate() method returns the join predicate used by the join operator. getJoinField1Name() returns the field name of join field1, and getJoinField2Name() returns the field name of join field2. getTupleDesc() returns the tuple descriptor of the joined tuples. The open() method opens the join operator, close() closes it, rewind() rewinds the child operators, and fetchNext() fetches the next matching tuple generated by the join operator.

Implementation for Exercise 2

IntegerAggregator.java: Constructor: Initializes an IntegerAggregator with parameters like group-by field index, type, aggregate field index, and operator. mergeTupleIntoGroup: Merges a new tuple into the aggregate, updating results based on group-by field and operator. iterator: Creates an iterator for group aggregate results, producing tuples with group value and aggregate value pairs.

StringAggregator.java: Constructor: Initializes a StringAggregator with parameters similar to IntegerAggregator, supporting only the COUNT aggregation operator for string fields. mergeTupleIntoGroup: Merges a new tuple into the aggregate for string fields, updating occurrence counts based on the group-by field.iterator: Creates an iterator for group aggregate results for string fields.

Aggregate.java: Constructor: Constructs an Aggregate operator with parameters for child iterator, aggregate field, group-by field, and operator. groupField: Returns the group-by field index in input tuples or NO_GROUPING if no grouping. groupFieldName: Returns the group-by field name in output tuples or null if no grouping. aggregateField: Returns the aggregate field index. aggregateFieldName: Returns the aggregate field name in output tuples. aggregateOp: Returns the aggregation operator. open: Opens the Aggregate operator, initializing the aggregator and merging tuples into groups. fetchNext: Fetches the next tuple, producing aggregate results until no more tuples. rewind: Rewinds the child operator. getTupleDesc: Returns the TupleDesc of the Aggregate operator, specifying output tuple structure.

close: Closes the Aggregate operator. getChildren: Returns the child operator. setChildren: Sets the child operator.

Implementation for Exercise 3

HeapFile.java: the `insertTuple` method is implemented to add a tuple to the heap file. It iterates through the pages in the file to find a page with an empty slot. If such a page exists, it inserts the tuple into that page and updates the list of modified pages. If no such page exists, it creates a new page, inserts the tuple into it, writes the page to disk, and updates the list of modified pages. The `deleteTuple` method in `HeapFile.java` removes a tuple from the heap file. It first identifies the page that contains the tuple to be deleted, then removes the tuple from the page and marks the corresponding slot as unused. Finally, it updates the list of modified pages.

HeapPage.java: the `insertTuple` method adds the specified tuple to the page by finding an empty slot and marking it as used. If the page is full, it throws a `DbException`. The `deleteTuple` method removes the specified tuple from the page by marking its corresponding slot as unused.

BufferPool.java: the `insertTuple` method inserts a tuple into the specified table using the provided transaction ID. It retrieves the file associated with the table, inserts the tuple into the file, and updates the buffer pool accordingly. Similarly, the `deleteTuple` method deletes a tuple from the specified table using the provided transaction ID. It retrieves the file associated with the table, deletes the tuple from the file, and updates the buffer pool accordingly.

Implementation for Exercise 4

Insert.java: This class is responsible for inserting tuples into a specified table. The Insert operator takes a TransactionId as an argument in its constructor. This allows the insert operation to be part of a transaction, ensuring atomicity and durability of the operation. Inserts are passed through the BufferPool, which helps manage the buffer pages efficiently. The insert operation is performed through the buffer pool to ensure that the inserted tuples are written to disk correctly .The result of the insert operation is a single tuple containing the count of inserted records. This design choice simplifies the interface and allows users to know how many records were inserted

Delete.java: The Delete operator is responsible for removing tuples from a table, as read from its child operator. Upon opening, the operator iterates through its child tuples, deleting each tuple one by one using the buffer pool provided by Database.getBufferPool(). The deletion process is encapsulated within a transaction, ensuring atomicity and durability. Once all tuples are processed, the operator returns a single tuple containing the count of deleted records.

Implementation for Exercise 5

BufferPool.java: focuses on implementing the flushPage() method and associated helper methods in the BufferPool.java class within the src/java/simpledb/storage package. The objective is to enable page eviction, which is essential for managing memory efficiently in a database system. The flushPage() method is responsible for flushing a specific page to disk. It checks if the page is dirty, meaning it has been modified since it was last written to disk. If the page is dirty, it needs to be written back to disk to ensure data consistency. This method ensures that all changes made to the page are persisted on disk.

Appendix:

Tests Run:

Exercise 1

Predicate Test

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=PredicateTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.PredicateTest
    [junit] Testsuite: simpledb.PredicateTest
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.014 sec
    [junit]
    [junit] Testcase: filter took 0.009 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

JoinPredicateTest

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=JoinPredicateTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.JoinPredicateTest
    [junit] Testsuite: simpledb.JoinPredicateTest
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.018 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.018 sec
    [junit]
    [junit] Testcase: filterVaryingVals took 0.012 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

Filter Test

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=FilterTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.FilterTest
    [junit] Testsuite: simpledb.FilterTest
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 sec
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 sec
    [junit]
    [junit] Testcase: getTupleDesc took 0.008 sec
    [junit] Testcase: filterSomeLessThan took 0.001 sec
    [junit] Testcase: rewind took 0.001 sec
    [junit] Testcase: filterEqual took 0 sec
    [junit] Testcase: filterEqualNoTuples took 0 sec
    [junit] Testcase: filterAllLessThan took 0.001 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

Join test:

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=JoinTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.JoinTest
    [junit] Testsuite: simpledb.JoinTest
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 sec
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.017 sec
    [junit]
    [junit] Testcase: getTupleDesc took 0.008 sec
    [junit] Testcase: eqJoin took 0.001 sec
    [junit] Testcase: gtJoin took 0.001 sec
    [junit] Testcase: rewind took 0.001 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

System Test FilterTest

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=FilterTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.FilterTest
    [junit] Testsuite: simpledb.systemtest.FilterTest
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.161 sec
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.161 sec
    [junit]
    [junit] Testcase: testLessThan took 0.099 sec
    [junit] Testcase: testLessThanOrEq took 0.016 sec
    [junit] Testcase: testGreaterThanOrEq took 0.013 sec
    [junit] Testcase: testGreaterThan took 0.013 sec
    [junit] Testcase: testEquals took 0.011 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

t

System Test JoinTest

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=JoinTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.JoinTest
    [junit] Testsuite: simpledb.systemtest.JoinTest
    [junit] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.118 sec
    [junit] Tests run: 3, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.118 sec
    [junit]
    [junit] Testcase: testMultipleMatch took 0.102 sec
    [junit] Testcase: testNoMatch took 0.005 sec
    [junit] Testcase: testSingleMatch took 0.004 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

Exercise 2

IntegerAggregatorTest

```
PS C:ant runtest -Dtest=HeapFileWriteTest^CBASE SYSTEMS\DB_LAB\project_24-main>
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=IntegerAggregatorTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.IntegerAggregatorTest
    [junit] Testsuite: simpledb.IntegerAggregatorTest
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.025 sec
    [junit] Tests run: 5, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.025 sec
    [junit]
    [junit] Testcase: mergeAvg took 0.013 sec
    [junit] Testcase: mergeMax took 0.001 sec
    [junit] Testcase: mergeMin took 0.001 sec
    [junit] Testcase: mergeSum took 0 sec
    [junit] Testcase: testIterator took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

StringAggregatorTest

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=StringAggregatorTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.StringAggregatorTest
    [junit] Testsuite: simpledb.StringAggregatorTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.019 sec
    [junit]
    [junit] Testcase: mergeCount took 0.012 sec
    [junit] Testcase: testIterator took 0.001 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

AggregateTest

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runtest -Dtest=AggregateTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.AggregateTest
    [junit] Testsuite: simpledb.AggregateTest
    [junit] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 sec
    [junit] Tests run: 8, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.031 sec
    [junit]
    [junit] Testcase: getTupleDesc took 0.019 sec
    [junit] Testcase: maxAggregate took 0.001 sec
    [junit] Testcase: rewind took 0.001 sec
    [junit] Testcase: minAggregate took 0 sec
    [junit] Testcase: sumStringGroupBy took 0 sec
    [junit] Testcase: sumAggregate took 0.001 sec
    [junit] Testcase: countStringAggregate took 0.001 sec
    [junit] Testcase: avgAggregate took 0 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

AggregateTest System Test

```
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main> ant runsystest -Dtest=AggregateTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\build.xml

compile:

testcompile:

runsystest:
    [junit] Running simpledb.systemtest.AggregateTest
    [junit] Testsuite: simpledb.systemtest.AggregateTest
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.154 sec
    [junit] Tests run: 6, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.154 sec
    [junit]
    [junit] Testcase: testAverageNoGroup took 0.107 sec
    [junit] Testcase: testMax took 0.009 sec
    [junit] Testcase: testMin took 0.008 sec
    [junit] Testcase: testSum took 0.006 sec
    [junit] Testcase: testCount took 0.007 sec
    [junit] Testcase: testAverage took 0.007 sec

BUILD SUCCESSFUL
Total time: 0 seconds
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main>
```

Exercise 3

HeapPageWriteTest

```
    [javac] C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\src\java\simpledb\storage\HeapPage.java:109: warning: [synchronization] at
tempt to synchronize on an instance of a value-based class
    [javac]             synchronized (oldDataLock) {
    [javac]             ^
    [javac] C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-main\src\java\simpledb\storage\HeapPage.java:122: warning: [synchronization] at
tempt to synchronize on an instance of a value-based class
    [javac]         synchronized (oldDataLock) {
    [javac]         ^
    [javac] 10 warnings

testcompile:

runtest:
    [junit] Running simpledb.HeapPageWriteTest
    [junit] Testsuite: simpledb.HeapPageWriteTest
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.122 sec
    [junit] Tests run: 4, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.122 sec
    [junit]
    [junit] Testcase: addTuple took 0.108 sec
    [junit] Testcase: testDirty took 0.002 sec
    [junit] Testcase: deleteTuple took 0.002 sec
    [junit] Testcase: deleteNonexistentTuple took 0.002 sec

BUILD SUCCESSFUL
```

HeapFileWriteTest

```
ain> ant runtest -Dtest=HeapFileWriteTest
Buildfile: C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\proj
ect_24-main\build.xml

compile:

testcompile:

runtest:
    [junit] Running simpledb.HeapFileWriteTest
    [junit] Testsuite: simpledb.HeapFileWriteTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:
 0.151 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed:
 0.151 sec
    [junit]
    [junit] Testcase: addTuple took 0.13 sec
    [junit] Testcase: testAlternateEmptyAndFullPagesThenIterate took 0.015
sec

BUILD SUCCESSFUL
Total time: 1 second
PS C:\Users\Sree Devi Rajavelu\Desktop\DATABASE SYSTEMS\DB_LAB\project_24-m
ain>
```

Exercise 4
InsertTest

```
runtest:
    [junit] Running simpledb.InsertTest
    [junit] Testsuite: simpledb.InsertTest
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.1 sec
    [junit] Tests run: 2, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 0.1 sec
    [junit]
    [junit] Testcase: getTupleDesc took 0.082 sec
    [junit] Testcase: getNext took 0.009 sec

BUILD SUCCESSFUL
Total time: 3 seconds
```

Exercise 4
System test EvictionTest

```
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.35 sec
    [junit] Tests run: 1, Failures: 0, Errors: 0, Skipped: 0, Time elapsed: 1.35 sec
    [junit] ------------- Standard Output ---------------
    [junit] EvictionTest creating large table
    [junit] EvictionTest scanning large table
    [junit] EvictionTest scan complete, testing memory usage of scan
    [junit] ------------- ---------------- ---------------
    [junit]
    [junit] Testcase: testHeapFileScanWithManyPages took 1.34 sec

BUILD SUCCESSFUL
Total time: 3 seconds
```