This assignment is a collaboration between:
- Megan Fantes (U56999246)
- Efstathios Karatsiolis (U65214507)

PA 4
Design Document

Describe any design decisions you made. If you used something other than a nested-loops join, describe the tradeoffs of the algorithm you chose. Discuss and justify any changes you made to the API. Describe any missing or incomplete elements of your code.

- estimateJoinCost()
    - We implemented an algorithm for a simple nested-loop join. The cost of a nested-loop join amounts to the cost of the cross-product of the two tables. So the final cost is: (cost of reading the left-side table once) + (cost of reading the entire right-side table once for every tuple in the left-side table) + (the cost of writing every tuple in the cross-product).
- estimateJoinCardinality()
    - We actually did not have to write any code for this algorithm, we had to implement the logic of this method in estimateTableJoinCardinality().
    - We focused mostly on the case where we are doing and equijoin.
        - If we are joining the tables on the primary key of the left-side table (table 1), the cardinality of the result will be the cardinality of the right-side table (table 2). This is because the number of matches depends entirely on the number of times a value of the table 1 primary key appears in table 2.
        - Similarly, if we are joining the tables on the primary key of table 2, the cardinality of the result will be the cardinality of table 1.
        - If we are not joining on the primary key of either table, then the cardinality of the result is the cardinality of the larger table.
    - In the event that the join is not an equijoin, then the cardinality of the result is roughly 40% of the size of the cross-product.
        - So we multiply the size of the cross-product (card1*card2) by 0.4.
- orderJoins()
    - Order joins simply iterates through every possible set for every possible set size and then chooses the best plan according to the cost card
    - I have added 2 constructors in CostCard class. One without arguments which initializes to max values and one with arguments for the sake of abstraction and completeness.

Describe how long you spent on the lab, and whether there was anything you found particularly difficult or confusing.

- This assignment took us about 15 hours of coding between 2 people.
- We were confused with the instructions for the assignment. Specifically, we were confused about which tests we needed to be able to pass after Exercise 3. The instructions said we could pass every unit test in JoinOptimizerTest after Exercise 3, but in actuality we were only able to pass all the tests after finishing Exercise 4.