**Report - Ankit Joshi(Manager)**

1. In project 1 each team member implemented different methods and then I integrated source code.
2. I helped in code integration and integration testing.
3. Each team member was also responsible for writing unit test for their methods
4. Each team member has equal contribution (meet expectation :0).
5. Java Doc for methods are created in project1 -> doc folder.
6. UML diagram of classes is included as a SVG image which can we viewed in any browser.

Below are individual’s reports with their strategy and assumption while implementing methods-

**Report - Chirag Jain**

Implemented public Table minus (Table table2)

1. Initialized a variable Boolean "matched" to check whether a record from first table matches any record from another table.
2. Loop over tables: for each tuple
3. Outer loop for first table
4. Inner loop for second table
5. If Tuples from both tables match set "matched" to true
6. Adding the unmatched tuples from first table to new table -- result of minus operator.

**Report - Shubhi Jain**

1. Implemented function "public Table join (String attributes1, String attributes2, Table table2)":
   1. Found out the index value of each attribute of Table and Table2 using function "match (String [] column)" and "match (String [] column, Table2)" respectively. With the help of index, the value of each attribute in tuple is determined which helped in fulfilling the equality condition. Taken an empty row and add in all the tuples on condition where value(attribute1) matches with value(attribute2). The new returned table contains the tuples which satisfies condition value(attribute1) = value(attribute2). The String attribute1 and attribute2 can take multiple columns separated by space so it is kept in mind the tuples in resulting table satisfy equality between each respective attribute.
2. Implemented function "public Table join (Table table2)":
   1. The common attributes from the two tables are identified then the index value of each attribute of Table and Table2 using function "match (String [] column)" and "match (String [] column, Table2)" respectively. Here the common attributes can be more than one so the tuple in resulting table satisfies the equality between the values of all the found common attribute. Taken an empty row and add in all the tuples on condition where value of common attribute in both the table are equal. Also, the common attribute is eliminated from the resulting table. If in case no common attribute exists then CROSS Join or CARTESIAN PRODUCT is implemented between the two-given table where each row of table one is turn by turn concatenated with every row of table2.

**Report - Vinay Kumar Bingi**

* 1. Implemented "public Table project (String attributes)" function:
     1. Taken an empty list of "rows". Filled with values from each tuple (of original table) from columns mentioned in "attributes".
     2. return the new Table with rows as tuples.
  2. Implemented "private boolean typeCheck (Comparable [] t)" function:
     1. This method is used in insert function to check if the new tuple satisfies the column definitions of the table.
     2. i.e. checks if the number of attributes are same and if domains of all columns and the new tuple are same.
     3. For cases where Float is expected and Double is given, typecheck function still accepts the input. Same with the case where Long is expected and Integer is given
  3. Added a function "private static Class findClass (String className)":
     1. There's a findClass function which accepts String array as attribute. But needed a findClass function which accepts String as attribute.
     2. Used this in typeCheck function.
  4. Updated "public boolean equals(Table table2)" function in Table.java:
     1. Extended comparisons from table values to attribute names, key and domain

**Report - Pranay Reddy Armoor**

* 1. Implemented "public Table union (Table table2)" function:
     1. Created an empty array list. Added rows from table1 and table 2 if both the table meet the constrains of “Union”.
     2. returns a new table with rows as tuples.
     3. Constraint was checked with the help of function “private boolean compatible (Table table2)” which was provided with the initial code.
     4. Key for the Union function is all the attributes of table #1 because union has duplicate for project #1.
  2. Implemented "public Table select (KeyType keyVal)" function:
     1. Created an empty array list. Based on the parameter “keyVal” received in the function “select”,retrieved all the tuples from “index”.
     2. returns a new table with rows as tuples.