StreamDB

Let's implement a relational database management system! In this system, we manage *tables* consisting of several organizing *columns* and *rows* containing the entries for those columns.

titles	SID	Lastname	Firstname	Major
rows	101	Yao	Alan	Food Systems
	102	Chen	Antares	Literature
	103	Nguyen	Daniel	Parrot Biology
	104	Lee	Maurice	Chemical Engineering
	105	Jian	Lisa	Cosmology
	106	Kim	Sarah	Parrot Biology

```
class Table {
    String name;
    String[] titles;
    Set<Row> rows;

    Table(String name, String... titles) {
        this.name = name;
        this.titles = titles;
        this.rows = new HashSet<>();
    }

    class Row {
        String[] data;

        Row(String... data) {
            this.data = data;
        }

        /** Return a new row with the entries specified by COLUMNS. */
        Row getRow(String... columns) {
            String[] newData = new String[columns.length];
            for (int i = 0; i < columns.length; i++) {
                int rowIndex = getColumnIndex(columns[i]);
                newData[i] = data[rowIndex];
            }
            return new Row(newData);
        }
    }

    /** Add multiple ROWSTOADD to this table and return this table. */
    Table addAll(Stream<Row> rowsToAdd) {
        rowsToAdd.forEach(rows::add);
        return this;
    }
```

```
    /** Return a table containing new rows where PRED is true. */
    Table select(Predicate<Row> pred) {
        return new Table(name, titles).
    }

    /** Return a table of rows consisting of COLUMNS from this table. */
    Table select(String... columns) {
        return
    }

    /** Return a table containing new rows from first applying PRED and
     *  then selecting the remaining COLUMNS. */
    Table select(Predicate<Row> pred, String... columns) {
        return new Table(name, columns).

    }

    /** Return a stream of rows applying PRED then SELECTOR to each. */
    Stream<Row> select(Predicate<Row> pred, Function<Row,Row> selector) {
        return rows.stream().

    }

    /** Return a list of strings by applying GETTER to each row. */
    List<String> export(Function<Row,String> getter) {
        return
    }

    /** Return a list of strings from rows applying PRED then GETTER. */
    List<String> export(Predicate<Row> pred,Function<Row,String> getter) {
        return rows.stream().



    }

    /** Return a map of strings to results grouped according to GROUPER
     *  and with values selected by GETTER. */
    Map<String, List<String>> exportBy(
            Function<Row,String> grouper, Function<Row,String> getter) {
        return
    }

    /** Return a map of strings to results where the row satisfies PRED,
     *  grouped according to GROUPER, and values selected by GETTER. */
    Map<String, List<String>> exportBy(Predicate<Row> pred,
            Function<Row,String> grouper, Function<Row,String> getter) {
        return rows.stream().



    }
}
```