In-Memory Database

Think about the various functionality exposed by relational databases like Oracle, Mysql etc.
Databases are used in projects for reliable data persistence.
Commonly performed database operations are CRUD i,e (Create, Read, Update, Delete)

Database's also offer additional functionality for GET calls, like

- "ORDER BY" for ordering rows based on some columns
- "GROUP BY" for aggregating rows based on specific columns.

Problem Statement

Create an in-memory database to support following operation

1. Create a database.
2. Create a table

   > Hint:
   > createTable(String tableName, Array of (columnName, columnDataType))

3. Insert row

   > Hint:
   > InsertRow(Array_of(ColumnName, Value))

4. Delete Row

   > Hint:
   > DeleteRow(RowKey) (RowKey: Same as primary_key as in relational DB)

5. Update Row

   > Hint:
   > UpdateRow(RowKey, Object(ColumnName, Value)) (RowKey: Same as primary_key as in relational DB)

6. Get All rows from a table

   > Get all rows from a specific table. This is similar to
   > Select * from tableName;

7. Get specific count of rows from a table. (Same as limit function in relational DB)

```
Select * from tableName limit count;
```

8. SORT by a column, and GET specific count of rows from table

```
Select * from tableName order by column1 limit count;
Where column1 = PrimaryKey
count=row count

Select * from tableName order by column2 limit count;
Where column1 = NOT a PrimaryKey
count=row count
```

9. GroupBy a column, and GET specific count of rows from table. This is the aggregate function.

```
select aggregate_key, count(*) from table group by column1;
Where column1 = PrimaryKey
count=row count
```

Instructions

1. In memory, do not use any IO, storage (file etc).
2. Consider all operations to happen serially, no concurrency handling required.
3. Focus on operation functionality.
4. Focus on data structures for storing data, optimize operations.
5. Explain time & space complexity for each operation.
6. Write unit tests.

Please check-in your code in GIT repo, and send the git link.

Go extra mile

1. Handle concurrent read/writes.