



# Group Project

## Team 3 – Option 3

### Team Members:

Chen Dekun      120090336

Chen Qingyuan   120090747

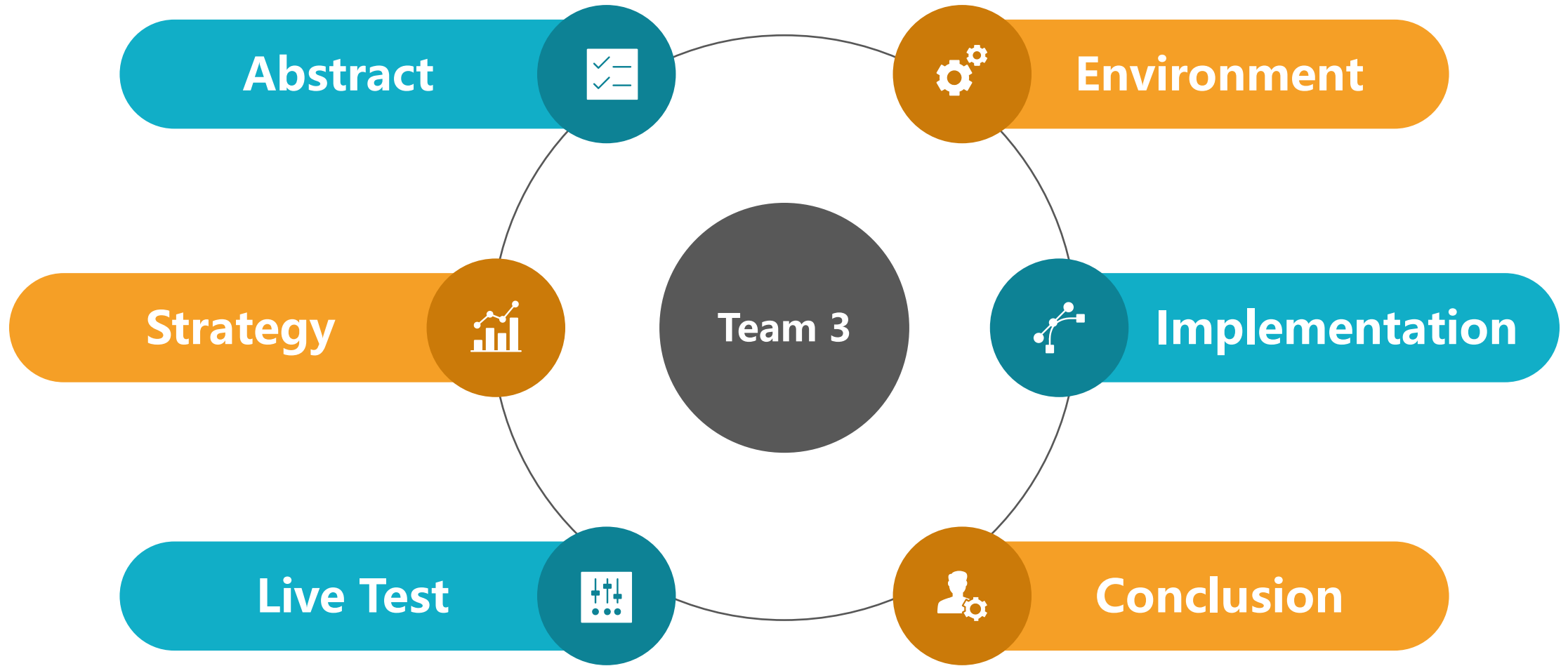
Li Ming          120090675

Nasr Alae-eddine   119010531

Zhang Jiayu        120010027

Zhang Haomin      118010408

# Content



# Abstract

## What The Project Is

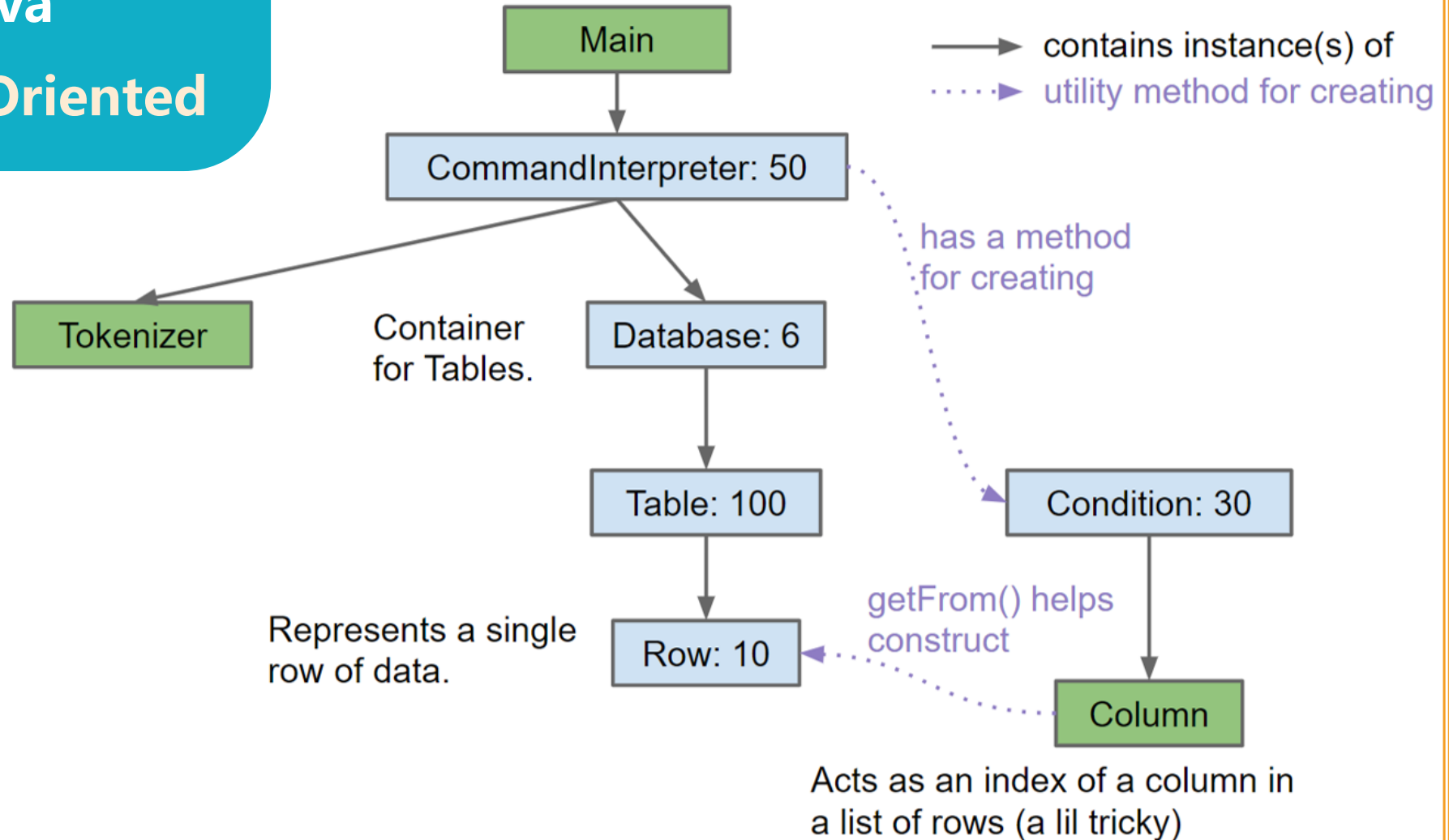
- ✓ **DB61B**
- ✓ **A miniature Relational Database Management System (DBMS)**

## What The Final Goal Is

- ✓ **Basic functions:**  
load, insert, select, ...

# Strategy

## Java Object-Oriented



# Strategy

## Our Checklist

1. Complete the **printing of prompts**
2. Ran **first Test**
3. Implement the **Row class** (except for the constructor)
4. Implement the parts of the Table class: **Create a new Table, Add a Row to it, and Print an entire Table.**
5. Implement the **Database class.**
6. Implement the **Condition class.**
7. Implement **insert and load.**
8. Implement the kind of **select that takes a single table and has no conditions.**
9. Implement the **Row constructor.**
10. Get **single-table select with conditions** to work.
11. Work on the **two-table variety of select.**

# Live Test

**Test Load & Store**

**Test xxx**

**Test Select**

**Test xxx**

**Test Print**

**Test xxx**

# Environment



## Program Language

- Java for functions
- Python for running tests
- ...



## Git Management

- ...
- ...
- ...



## Dependency

- Maven details..
- ...



## Testing

- ...
- ...
- ...

# Implementation

## Selection Clause

Step 1: Create a new table using **selected columns**

```
// without cond
if (conList.size() == 0) {

    // one table
    if (tabList.size() < 2) {
        Table selectTable = tabList.get(index: 0);
        table = selectTable.select(colTitles);

        // more than two table, but currently just up to 2 tables.
        // -need further implementation(if select more than 2 tables together)
    } else {
        Table selectTable = tabList.get(index: 0);
        Table selectTable2 = tabList.get(index: 1);
        table = selectTable.select(selectTable2, colTitles);
    }

    // with cond -need further implementation
} else {
    table = table.select(colTitles, conList);
}
return table;
```

Step 3: Execute selection

```
Table selectClause() {
    List<String> colTitles = new ArrayList<String>();
    colTitles.add(columnName());
    while (!_input.nextIf(p: ",")) {
        colTitles.add(columnName());
    }
    _input.next(p: "from");
    List<Table> tabList = new ArrayList<>();

    // initialize a new table to record the selection
    Table table = new Table(colTitles);
```



```
tabList.add(tableName());
while (!_input.nextIf(p: ",")) {
    tabList.add(tableName());
}

List<Condition> conList = new ArrayList<>();

// if next tokenizer is where, check condition.
if (!_input.nextIs(p: "where")) {
    Table[] tabArray = tabList.toArray(new Table[tabList.size()]);
    conList = conditionClause(tabArray);
}
```



Step 2: Create **Arraylist** to store selected tables



# Implementation

## Selection with conditions

```
Table select(List<String> columnNames) {
    Table result = new Table(columnNames);
    List<Integer> columnNum = new ArrayList<>();
    for (String columnName : columnNames) {
        columnNum.add(this.findColumn(columnName));
    }
    for (Row row : _rows) {
        String[] newRow = new String[columnNames.size()];
        for (int i = 0; i < columnNames.size(); i++) {
            newRow[i] = row.get(columnNum.get(i));
        }
        result.add(new Row(newRow));
    }
    // result.print();
    return result;
}
```

```
Table select(List<String> columnNames, List<Condition> conditions) {
    Table result = new Table(columnNames);
    // FILL IN
    List<Integer> columnNum = new ArrayList<>();
    for (String columnName : columnNames) {
        columnNum.add(this.findColumn(columnName));
    }
    for (Row row : _rows) {
        if (test(conditions, row)) {
            String[] newRow = new String[columnNames.size()];
            for (int i = 0; i < columnNames.size(); i++) {
                newRow[i] = row.get(columnNum.get(i));
            }
            result.add(new Row(newRow));
        }
    }
    return result;
}
```

# Conclusion

