In Memory Database

Problem

Implement basic SQL-like operations, such as select, where, join, etc., of an in-memory database.

Instructions

Assume the following data is stored in the in-memory database

departments

id	name
0	engineering
1	finance

users

id	department_id	name
0	0	Ian
1	0	John
2	1	Eddie
3	1	Mark

salaries

id	user_id	amount
0	0	100
1	1	150
2	1	200
3	3	200
4	3	300
5	4	400

First step

Please implement `select` and `where` in `Table` class. For the sake of simplicity, you can assume all operations will be passed valid parameters. You do not need to concern yourself with error handling for occurrences like selecting an arbitrary column from a non-existent table.

For example,

```
# Should print something like
# id, department_id, name
# 1, 0, John
print db.table('users')
        .where('id', 1)
        .select(['id', 'department_id', 'name'])
```

Second step

Please implement `inner_join` in `Database` class.

```
# Should print something like
# users.name, departments.name
# Ian, engineering
# John, engineering
print db.inner_join(db.table('users'), 'department_id', db.table('departments'), 'id')
        .where('departments.name', 'engineering')
        .select(['users.name', 'departments.name'])
```

Third step

Please implement `left_join`, `right_join` and `outer_join` in `Database` class.

```
# Should print something like
# users.name, salaries.amount
# Ian, 100
# John, 150
# John, 200
# Mark, 200
# Mark, 300
# Eddie, None
print db.left_join(db.table('users'), 'id', db.table('salaries'), 'user_id')
        .select(['users.name', 'salaries.amount'])

# Should print something like
# users.name, salaries.amount
# Ian, 100
# John, 150
# John, 200
# Mark, 200
# Mark, 300
# None, 400
print db.right_join(db.table('users'), 'id', db.table('salaries'), 'user_id')
          .select(['users.name', 'salaries.amount'])

# should print something like
# users.name, salaries.amount
# Ian, 100
# John, 150
# John, 200
# Mark, 200
# Mark, 300
# Eddie, None
# None, 400
print db.outer_join(db.table('users'), 'id', db.table('salaries'), 'user_id')
          .select(['users.name', 'salaries.amount'])
```

Submission

Upon completion, please follow the instructions described in the website (where you found the instructions to download the project) to submit your solution. You can submit as many times as you prefer. Your last submission will be used for evaluation as well as marking the end of your coding assessment.

Lastly, do not be concerned if you are running a little bit over time (0-10 minutes). We do not penalize moderately tardy submissions.