# Project4

Join Table

# Join Table

- Your database systems don’t consider JOIN table operation yet.
- Our goal is to implement a **JOIN operation** with maintaining maximum memory usage.

# Project Specification

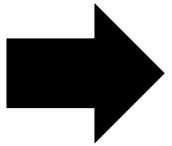
- Join operation takes two table IDs.
- Recall that, each table contains <key (8B integer) , value (120B string)> pairs.
- We will implement natural join operation based on the same key.

Table A

| key | value |
|---|---|
| 1 | ABC |
| 2 | DEF |
| 3 | GHI |
| 4 | JKL |
| 5 | MNO |

Table B

| key | value |
|---|---|
| 1 | PQ |
| 3 | RS |
| 5 | TU |
| 7 | VW |
| 9 | XYZ |



Join Result

| key | value | key | value |
|---|---|---|---|
| 1 | ABC | 1 | PQ |
| 3 | DEF | 3 | RS |
| 5 | GHI | 5 | TU |

# Project Specification

➢Your library (libbpt.a) should provide those API services.

1. **int init_db (int buf_num);**
2. **int open_table (char * pathname);**
3. **int db_insert (int table_id, int64_t key, char * value);**
4. **int db_find (int table_id, int64_t key, char* ret_val);**
5. **int db_delete (int table_id, int64_t key);**
6. **int close_table(int table_id);**
7. **int shutdown_db(void);**
8. **int join_table(int table_id_1, int table_id_2, char * pathname);**
   - **Do natural join with given two tables and write result table to the file using given pathname.**
   - **Return 0 if success, otherwise return non-zero value.**
   - **Two tables should have been opened earlier.**

# Project Specification

- The result of *join_table(a, b)* should be written when the command returns successfully.
- Result file format should contain a line of **“a.key,a.value,b.key,b.value”** where each items are separated by comma.
- a.key == b.key (by join property)
- Each line should be sorted by the same key in increasing order.
- We will check the correctness of join result by this file.

Join Result

| key | value | key | value |
|---|---|---|---|
| 1 | ABC | 1 | PQ |
| 3 | DEF | 3 | RS |
| 5 | GHI | 5 | TU |

Result File Format

```
1,ABC,1,PQ
3,DEF,3,RS
5,GHI,5,TU
```

# Project Specification

- While *join_table()* operates, your system **ONLY** uses a memory region that your buffer manager maintains and should not use another memory area.
  - That means, you should use a memory region that is allocated during *init_db()* for buffer manager and you can't use a dynamic memory allocation (such as *malloc*) while doing *join_table()* operation.
- You can use C / C++ for this project.
- You have to use just a single thread.

# Submission

- Project Deadline : **Nov 14 23:59**
  1. You must check this deadline. The deadline will not be extended.
  2. No points will be given for submissions that are overdue submissions.

- Directory Hierarchy
  - You must submission your project code in your_git_repo/project4

- You should write **Wiki** in your gitlab.