Final Report

- **DEADLINE: 2020. 12. 20. 23:59**
- You can use any editor program to write the report, however, you must convert the report file into a **pdf file**, named **final_report.pdf**. Then, upload (git push) it into the **final_report** directory of the hconnect project until the deadline.
  (e.g., *your_project_path*/final_report/final_report.pdf)
- You can use either Korean or English, and there is no plus point at all in using English.
- Feel free to use whatever you have written in the Wiki.
- The report should contain the following,

1. Table of Contents

2. Overall Layered Architecture

3. Concurrency Control Implementation

4. Crash-Recovery Implementation

5. (Important) In-depth Analysis
   (This will be the main assessment item, so it must be written very carefully and logically.)

- See the details in the following report templates.

[Template]

Database System 2020-2
Final Report

Class Code (ITE2038-11800 or ITE2038-11801)
Student number
Name

Table of Contents

Overall Layered Architecture

(전체 layered architecture의 구조 및 layer 간의 상호 관계 등에 대하여 자유롭게 기술)

Concurrency Control Implementation

(Concurrency Control 기법이 어떻게, 어느 레이어에 걸쳐져서 구현 되었는지에 대해 세부적으로 기술, 관련 있는 layer의 핵심 component에 대한 설명)

Crash-Recovery Implementation

(Crash-Recovery 기법이 어떻게, 어느 레이어에 걸쳐져서 구현 되었는지에 대해 세부적으로 기술, 관련 있는 layer의 핵심 component에 대한 설명)

In-depth Analysis

1. Workload with many concurrent non-conflicting read-only transactions.
   (많은 수의 non-conflicting read-only transaction이 동시에 수행될 경우 발생할 수 있는 성능 측면에서의 문제점을 설명하고, 이를 해결할 수 있는 디자인을 제시할 것)
   (본인의 최종 프로젝트 코드 및 디자인을 기반으로 설명할 것)
   (non-conflicting: access different records each other)

2. Workload with many concurrent non-conflicting write-only transactions.
   (많은 수의 non-conflicting write-only transaction이 동시에 수행될 경우 발생할 수 있는 **Crash-Recovery 와 관련된** 성능 측면에서의 문제점을 설명하고, 이를 해결할 수 있는 디자인을 제시할 것)
   (본인의 최종 프로젝트 코드 및 디자인을 기반으로 설명할 것)