Assignment2

JOIN 구현

2021038131 장준혁

1. 선택한 알고리즘과 그 이유

Case1. Merge JOIN

Merge Join을 하되, sort는 이미 되어 있어서 sort는 따로 하지 않았음.

Merge Join은 처음에 정렬하는 데 시간이 많이 소요되는 join 방식인데,

두 테이블 모두 이미 정렬이 되어 있기 때문에 Merge Join을 하면 빠르게 바로 Join할 수 있음

Case2. Hash JOIN

만약 Merge Join을 사용한다면 10000개의 데이터를 모두 소팅하는 데에는 시간이 오래걸림.

Nested loop는 더 오래 걸리기 때문에 가능하면 사용하지 않는 것이 좋음.

제시된 코드에서는 한 번에 12개 까지만 출력 스트림을 만들 수 있음.

따라서 약 10개 정도의 버킷을 만들어서 Hash Join하는 것이 가장 빠름.

Case3. Hash Join

2번과 같은 이유로 Hash 조인을 사용함.

2번과 동일하게 Name 기준으로 Hashing함

먼저 name_grade1, name_grade2를 JOIN하고 성적이 항상 된 경우에 한해서 name_number도 조인하였음.

name_grade1, name_grade2를 Join하는 과정에서 성적 항상 여부를 체크하면 만약 성적이 항상 되지 않은 레코드들에 대해서는 name_number 테이블을 조인하지 않아도 돼서 연산 시간을 줄일 수 있음.

2. 트러블슈팅

Window Subsystem Linux를 사용해서 windows에서 과제를 하려고 했는데, wsl이 무한로딩 되면서 실행이 안됐는데 하이퍼바이저 플랫폼 등 가상화 관련 옵션을 컴퓨터 설정에서 활성화하자 해결되었다. 아마 NOX라는 안드로이드 에뮬레이터를 컴퓨터에 설치하면서 컴퓨터 옵션을 조금 건드렸는데 그로 인해 문제가 발생한 것으로 보인다.

Window에 설치된 일반 VSCODE에서 리눅스 터미널은 열 수 있지만, linux 환경에서만 쓸 수 있는 헤더파일이나 함수들은 인식하지 못하기 때문에 자동완성이나 디버깅 기능은 사용할 수 없었다. 확장에서 WSL 관련 확장을 설치하니 리눅스 환경의 VSCODE를 열 수 있어서 자동완성 등의 편의 기능을 사용할 수 있게 되었다.

3번 case를 작업할 때 처음에는 Nested loop join을 사용하려고 했었다. 다양한 조건을 요구하기 때문에 nested loop join을 사용해야 구현이 쉬울 것이라고 생각했다. 하지만 이 경우 disk IO 작업을 최대 10000*10000*10000번 해야 한다. 실제로 실행해 보았을 때 너무 긴 실행시간이 나와서 중간에 중단했다. Hash JOIN으로 먼저 JOIN하고 그 다음에 조건을 체크하는 방식으로 변경했다.