

[제출물 기능 설명 및 사용 방법]

1. 매 실행마다 동일한 환경에서 프로그램이 실행되도록 데이터베이스 초기화 작업 진행
(Drop -> Create -> Insert)

2. 프로그램의 모든 기능은 로그인 된 상태로 실행

2-1. 로그인 : 이미 존재하는 회원 정보로 로그인할 수 있습니다.

ID : Mid1, PW : @lydFWn72(G

2-2. 로그인이 실패한 경우 회원가입 진행 : 포맷에 맞게 정보를 입력하고 성공한 경우 자동으로 로그인 상태가 됨.

2-3. 회원정보 수정 : 아이디는 변경할 수 없고, 콘솔에서 이미지를 입력할 수 없으므로 이미지 수정도 불가능. 바꾸고 싶은 항목만 포맷에 맞게 입력하여 변경 가능.

2-4. 회원 삭제 : 삭제 시 연관된 정보들이 함께 삭제됨

3. 기능 선택 전 여행할 국가 또는 조회하고 싶은 국가를 선택

4. 원하는 기능 선택

4-1. 게시글 작성하기

4-2. '동행 찾기' 게시글에 동행 신청하기

4-3. 내가 쓴 '동행 찾기' 게시글의 신청 정보 조회 및 수정하기

4-4. '일정 소개' 게시글에 댓글 작성하기

4-5. 다른 다양한 기능 더보기 (phase 2의 퀴리문과 관련된 기능)

[유의사항 및 Application 제작 환경]

- 유의 사항

: 'SQL 오류 : 지원되지 않는 문자 집합(클래스 경로에 orai18n.jar 추가)' 오류가 계속 발생.

(lms 문의 게시판에 문의한(11/21일 작성) 오류)

오류 해결을 위해 orai18n.jar 라이브러리를 추가하였습니다. 해당 라이브러리가 없는 경우 오류가 발생할 수 있습니다. **ojdbc8-full.tar.gz** 파일을 제출물과 함께 제출했습니다. 해당 파일을 압축을 풀면 orai18n.jar 파일이 있습니다.

추가적으로 <https://jsp-making.tistory.com/512> 해당 블로그를 참고하셔도 좋습니다.

- Application 제작 환경

: Eclipse IDE, Oracle DBMS 19c, Oracle JDBC Driver ojdbc10.jar, Java Compiler JDK 11

[쿼리 수정사항]

phase2의 1번 쿼리

: 2005년 1월 1일 이후로 태어난 회원들의 닉네임,이메일 그리고 성별을 조회

-> 사용자의 입력을 받아 특정 시점 이후에 태어난 정보를 조회하도록 변경

```
select m.nickname, m.e_mail, m.gender
from member m
where m.birth > to_date('"+date+"','YYYYMMDD');
```

phase2의 3번 쿼리

: 관련 글 2개 이상인 국가 정렬

-> 사용자의 입력을 받아 특정 개수 이상의 글을 가진 경우를 조회하도록 변경

```
select l.nation, count(*) as number_of_post
from travel_introduction_post p, itr_contain c, location l
where c.post_id = p.post_id
and c.location_id = l.location_id
group by l.nation
having count(*) >=num
order by count(*) asc;
```

phase2의 4번 쿼리

: 댓글을 가장 많이 남긴 사람이 쓴 일정글 제목

-> 사용자의 입력을 받아 남긴 댓글의 개수가 상위 n번째 내에 드는 사람이 쓴 일정글 제목을 조회하도록 변경

```
select title
from travel_introduction_post
where member_id IN (select r.member_id
                    from (select member_id
                          from reply
                          group by member_id
                          order by count(*) desc) r
                    where ROWNUM<=num);
```

phase2의 5번 쿼리

: 생일이 같은 사람이 있는 사람

-> 사용자의 입력을 받아 생일이 같은 사람이 몇명 이상인 경우 조회하도록 변경

```
select distinct a.nickname,a.birth
from member a
where exists(select member_id,to_char(a.Birth,'MM-DD')
             from member b
             where to_char(a.Birth,'MM-DD') = to_char(b.Birth,'MM-DD')
             and a.member_id != b.member_id
             group by to_char(a.Birth,'MM-DD')
             having count(*)>=+num)
order by to_char(a.Birth,'MM-DD');
```

phase2의 7번 쿼리

: 여행 소개글 중 아직 스크랩 되지 못한 대륙과 국가를 모두 조회

-> 사용자의 입력을 받아 대륙을 입력 받아 조회하도록 변경 continent

```
select l.Continent, l.nation
from Location l
Where l.Continent = continentInput
And not exists (select * from ITR_CONTAIN I WHERE i.location_id= l.location_id)
```

phase2의 9번 쿼리

: 동행을 구하는 게시글 중 진행 중인 게시물의 국가 아이디를 조회하기

->동행을 구하는 게시글 중 게시물 상태를 사용자의 입력을 받아 게시물의 국가 아이디를 조회하기

```
select c.post_id, c.location_id
from cpn_contain c
Where c.post_id IN ( select p.post_id
                    from travel_companion_post p
                    where c.post_id = p.post_id
                    AND p.state = postState)
order by c.location_id ASC
```

phase2의 11번 쿼리

: 채팅 횟수가 가장 많은 회원의 닉네임과 그 횟수를 조회

-> 사용자의 입력을 받아 채팅 횟수가 많은 회원 상위 n명의 닉네임과 횟수를 조회

```
Select nickname, num
from ( select nickname,count(*) as num
      from member NATURAL JOIN real_time chat
      Group by nickname
      Order by count(*) DESC, nickname ASC)
Where ROWNUM <= num
```

phase2의 12번 쿼리

: 좋아요 수가 4개 이상인 동행글에 신청 거절 수

-> 사용자 입력을 받아 좋아요 n개 이상인 동행글에 신청 거절 수 조회하도록 변경

```
select ap.postid, count(*) as refused
from application_info ap
where ap.request_state = '거절'
and ap.post_id in
( select post_id
  from like_post
  group by post_id
  having count(*) >= like )
group by ap.post_id
```

phase2의 13번 쿼리

: 여행 동행 게시글에 국적 조건은 무관, 구인 상태는 진행 그리고 성별은 남성인 게시글 조회

-> 사용자의 입력을 받아 국적, 성별 조건에 따라 동행 구인 '진행'상태의 글 조회하도록 변경

```
select post_id, title
from travel_companion_post
where nationality_condition = nation and state = '진행' and gender_condition = gender
```

phase2의 18번 쿼리

: '2020-01-01 00:00:00' 이전에 쓰여진 글의 개수를 나라별 정렬

-> 사용자의 입력을 받아 특정 시점 이전에 쓰여진 글의 개수를 조회하도록 변경

```
select l.nation, p.post_id, p.title
from travel_companion_post p, cpn_contain c, location l
where c.post_id = p.post_id
and c.location_id = l.location_id
and p.creation_time < TIMESTAMP 'ans'
```

[테이블 생성 스키마 수정사항]

- 모든 일대다 관계에서 참조하는 테이블은 참조당하는 테이블이 삭제되었을 때 함께 삭제.
(ex. Member가 삭제되었을 때 해당 Member가 작성한 Post가 삭제되어야한다.)
- '열에 대한 값이 너무 큼'이라는 오류 발생. 따라서 varchar의 최대값을 수정.