

# Select文考察

---

## 保存済みルートテーブルに対するSelect文

---

ユーザIDが指定されて、そのユーザが保存したルートを返せば良い。

```
SELECT * from SAVED_ROUTE_TABLE
      where user_id = 指定されたid
```

その後、構造体にデータを格納してクライアントに送信する。

## ユーザテーブル

---

ユーザIDが指定されて、そのユーザが持つ情報を返せば良い。ただし、以下の情報に限定する。

- 身長
- 体重
- 年齢
- 性別
- 目標体重
- 目標期間

```
SELECT user_height, user_weight, user_age, user_sex, \
      goal_weight, goal_height from USER_TABLE
      where user_id = 指定されたid
```

この後、取得したデータを構造体に格納して、クライアントに送信する。

## 利用履歴テーブル

---

- 日付、走行開始時刻、走行距離、走行距離、消費カロリーの取得
  - ユーザIDと一致するものを全て返す必要がある。
  - この時、1ヶ月以上前のデータを消してしまう必要もある。

```
SELECT jogging_date, then_time, jogging_distance, \  
       jogging_time, burnced_calorie, route from USEDLOG_TABLE  
where user_id = 指定されたid
```

ただし、この前に該当データがいくつあるのかを調べて、その文構造体の配列を作る必要がある。

```
SELECT count(*) from USEDLOG_TABLE  
where user_id = 指定されたid
```

上記の結果をnとする。

UL型の配列(要素がn個)を作るC言語のコード。アクセスはいつも通り `geted_data[i]` で可能。  
ただし、このままのデータだと、ユーザIDが重複して保存されるので少し冗長。

```
UL *geted_data = (UL *)malloc(sizeof(UL) * n);
```