

【DB】 インデックスを少 し理解した

ひがき

@higaki_program

DB のインデックスについて勉強したので、誰かの役
に立てばと思いまとめてみる

自己紹介

経歴

ひがき

2020年3月 大学院卒業(愛媛)

2020年4月～総合電機メーカーの設計(兵庫)

2022年4月～自社開発企業の開発(福岡)

開発

業務: Laravel + Vue.js

個人開発: Flutter + Firebase 触ってみたり

趣味

サッカー、バスケ、たまに筋トレ

告知

「エンジニアの輪」

が福岡で開催されます！！

2023年1月28日(土) 13:00 ~ 17:00

長岡ゆ一きさん、ひがき

ぜひ、参加していただきたいです('▽')

対象

- インデックス？？
- 実はよくわかってないんだよね
- 過去・未来の自分

インデックスとは

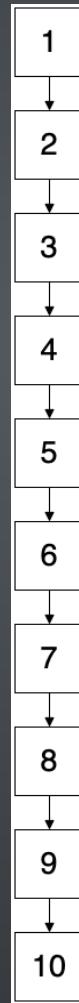
データベースにおけるインデックスとは、目的のレコードを効率よく取得するための「索引」のことです。

インデックスとは？仕組みをわかりやすく解説

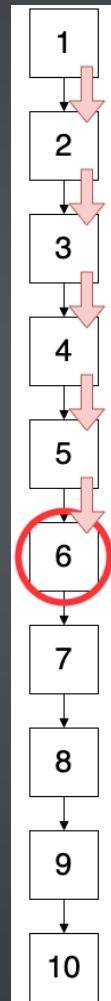
<https://products.sint.co.jp/siob/blog/index>

こんなイメージ

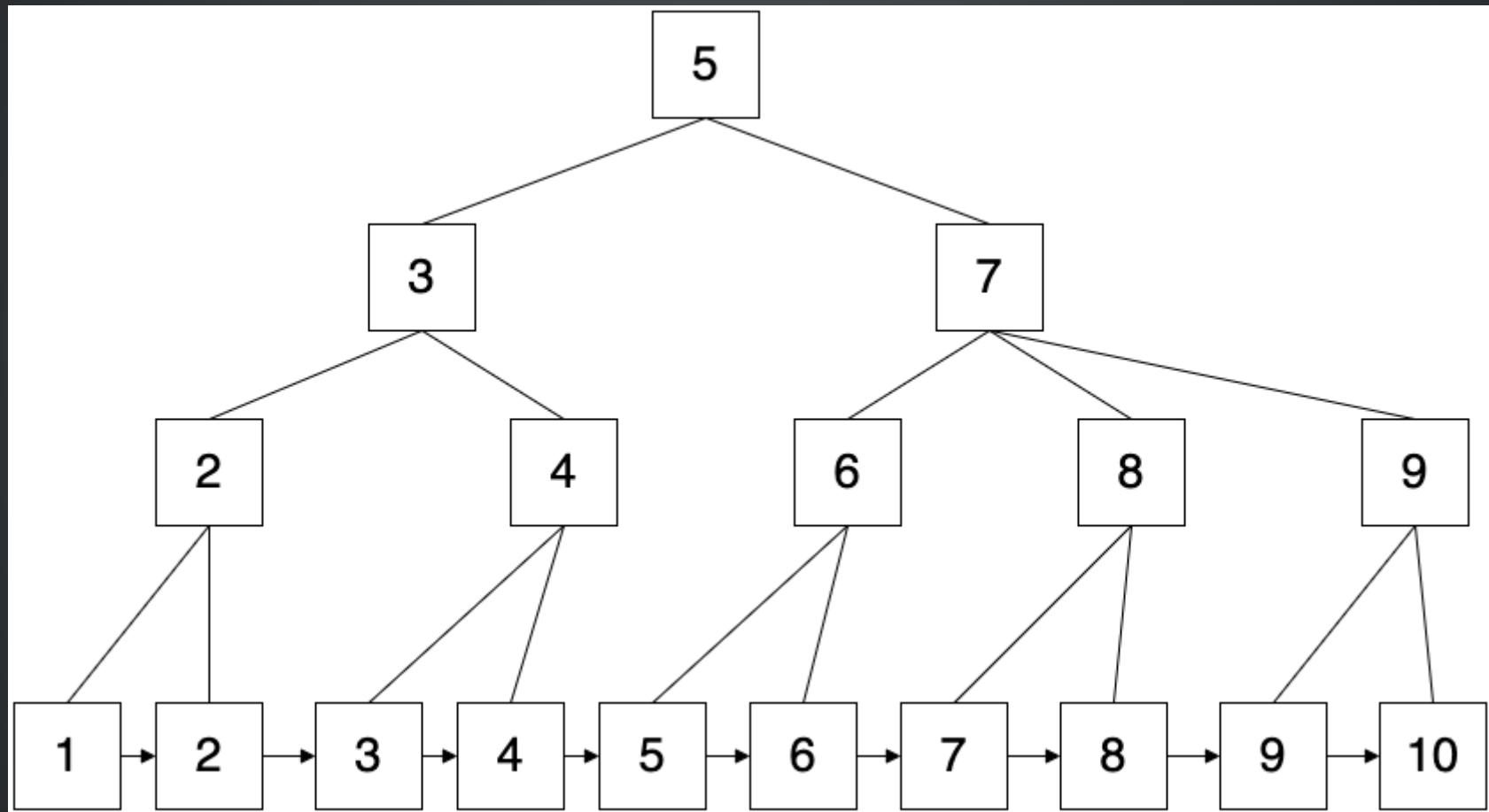
インデックスなし



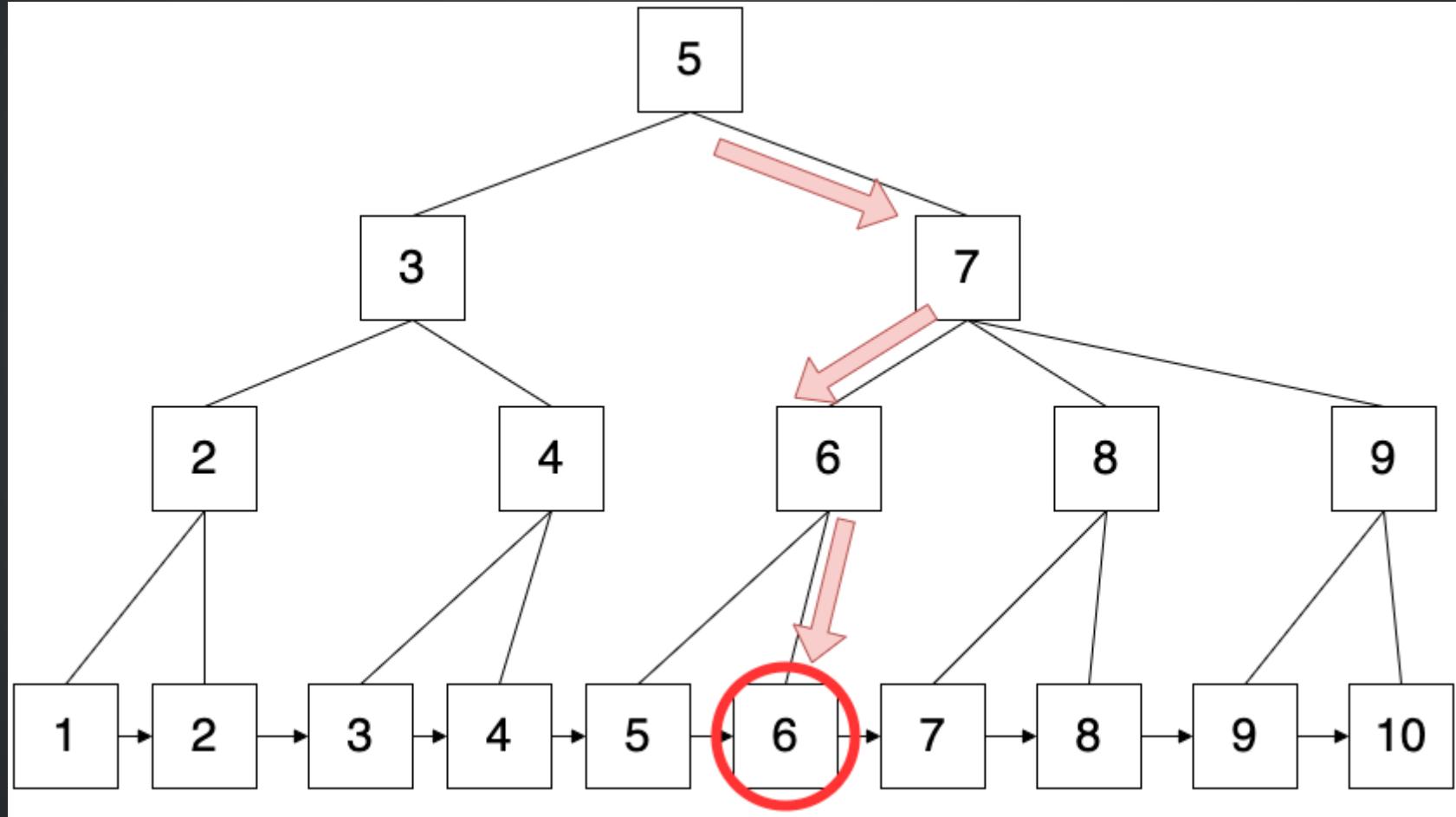
例) 6をインデックスなしで検索する



インデックスあり



例) 6をインデックスありで検索する



確かにインデックスを使うと効率良さそう

インデックスの種類

- クラスインデックス
- セカンダリインデックス

クラスタインデックス

基本的に PRIMARY KEY に貼られるインデックス

最下層には全カラムの情報がある

DB が自動で作成してくれる

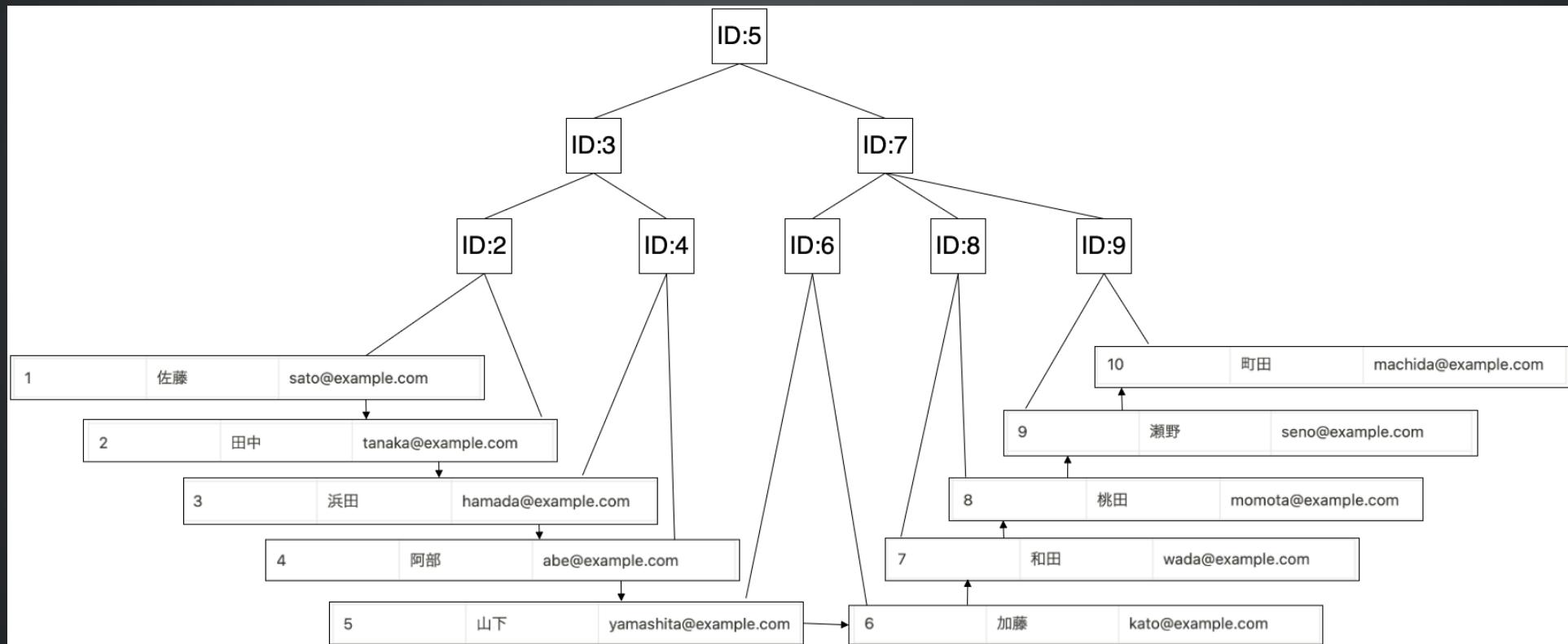
具体例

User テーブル

ID	NAME	EMAIL
1	佐藤	sato@example.com
2	田中	tanaka@example.com
3	浜田	hamada@example.com
4	阿部	abe@example.com
5	山下	yamashita@example.com
6	加藤	kato@example.com
7	和田	wada@example.com
8	桃田	momota@example.com
9	瀬野	seno@example.com
10	町田	machida@example.com

クラスタインデックス

最下層に ID, NAME, EMAIL の全ての情報がある



セカンダリインデックス

指定した KEY に貼られるインデックス

最下層には 指定した KEY と PRIMARY KEY の情報が
ある

基本的に自動では作られない

MySQL では外部キー制約により自動で作られる
PostgreSQL では作られない

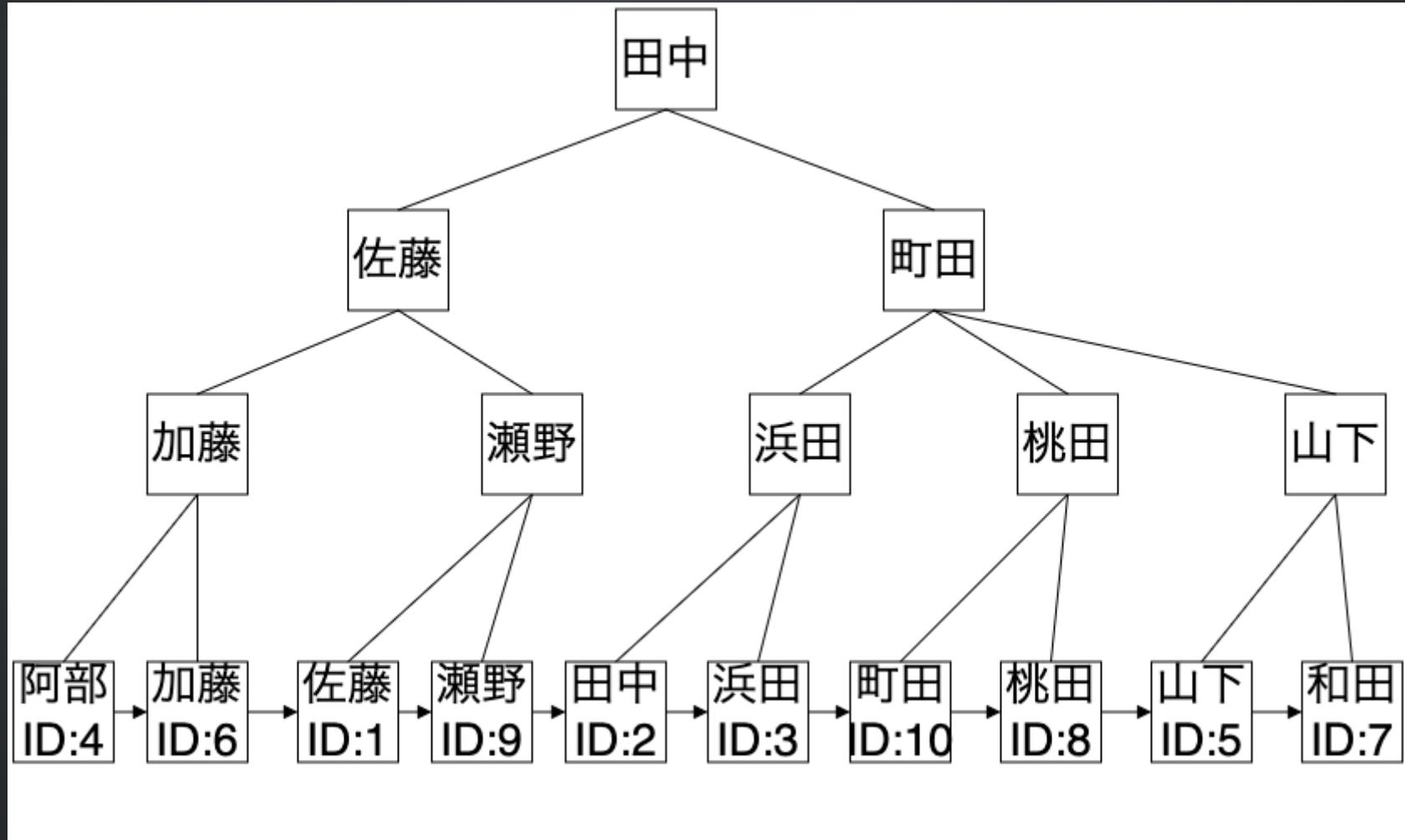
具体例

User テーブル

ID	NAME	EMAIL
1	佐藤	sato@example.com
2	田中	tanaka@example.com
3	浜田	hamada@example.com
4	阿部	abe@example.com
5	山下	yamashita@example.com
6	加藤	kato@example.com
7	和田	wada@example.com
8	桃田	momota@example.com
9	瀬野	seno@example.com
10	町田	machida@example.com

NAME にインデックスを貼った場合

```
CREATE INDEX index_user_name ON user(name);
```



ちなみに、インデックスは複数貼ることもできる

```
CREATE INDEX index_user_name ON user(name, email);
```

なるほど、
でもインデックスってどのように使われるんや？

インデックスでの検索

User テーブル

ID	NAME	EMAIL
1	佐藤	sato@example.com
2	田中	tanaka@example.com
3	浜田	hamada@example.com
4	阿部	abe@example.com
5	山下	yamashita@example.com
6	加藤	kato@example.com
7	和田	wada@example.com
8	桃田	momota@example.com
9	瀬野	seno@example.com
10	町田	machida@example.com

メールアドレスが **SATO@EXAMPLE.COM** の 人物の情報を探索せ

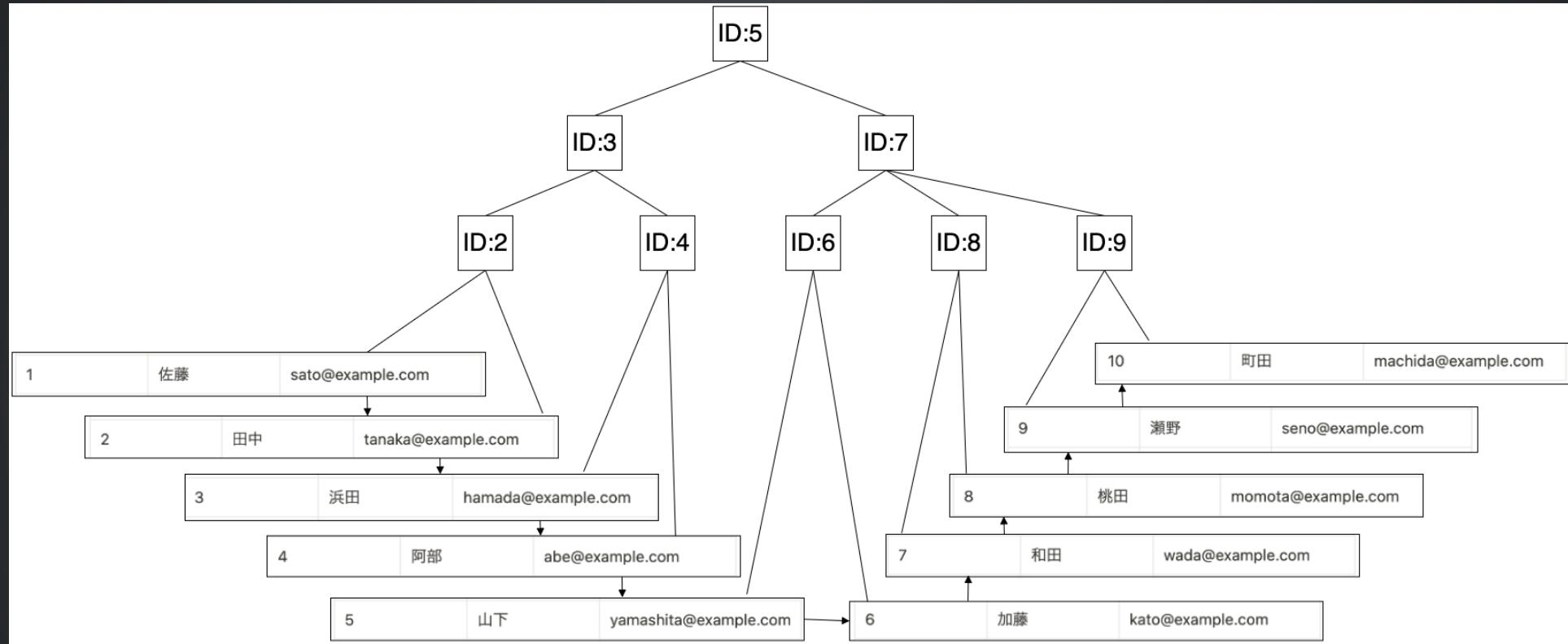
```
SELECT * FROM users WHERE email = 'sato@example.com';
```

この場合、どのように検索されるか？

以下のどれか。

- クラスタインデックスでの検索
- NAME のセカンダリインデックスでの検索
- テーブルをフルスキヤン(インデックスなしで検索)

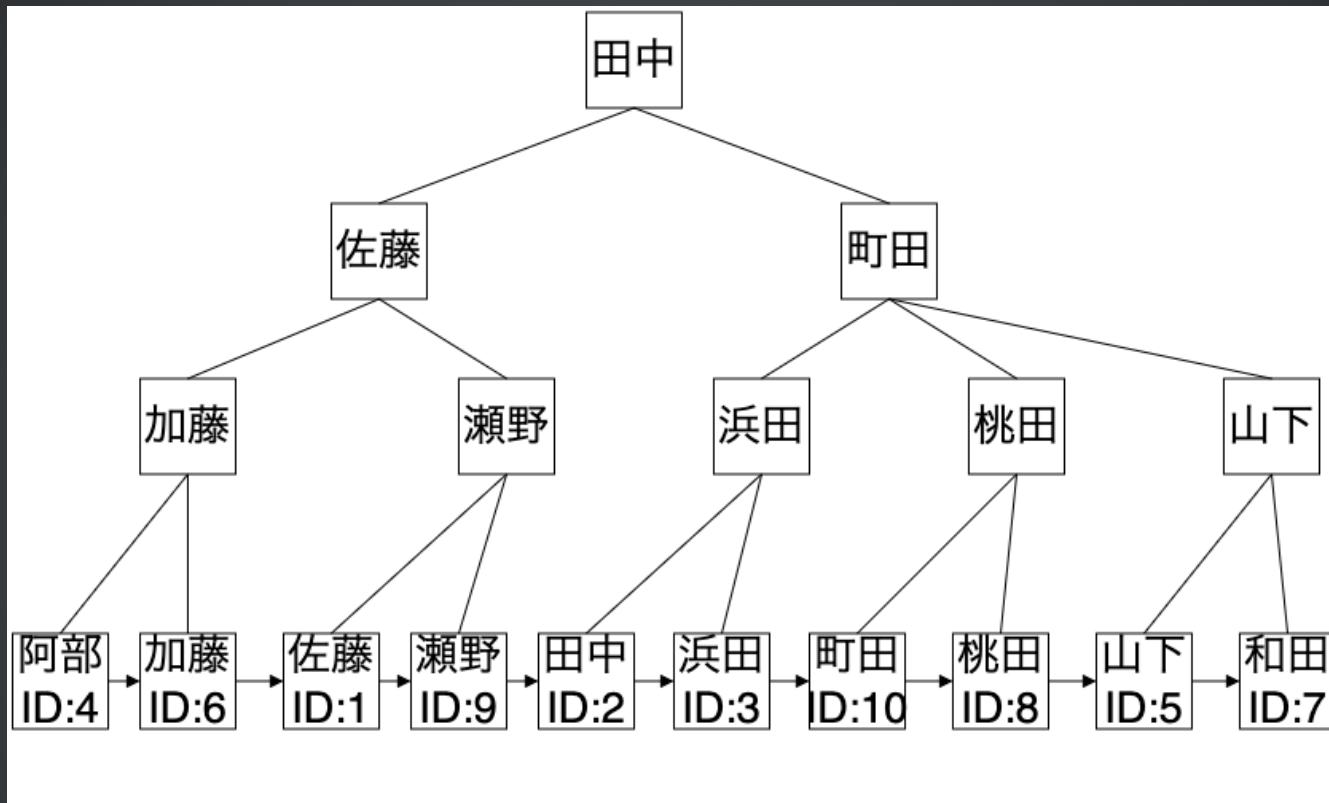
クラスタインデックスで検索できるか？



ID での検索はできるが

メールアドレスで検索は無理そう

NAME のセカンダリインデックスで検索できるか？



NAME での検索はできるが
メールアドレスでの検索は無理そう

テーブルをフルスキャン(インデックスなしで検索)

ID	NAME	EMAIL
1	佐藤	sato@example.com
2	田中	tanaka@example.com
3	浜田	hamada@example.com
4	阿部	abe@example.com
5	山下	yamashita@example.com
6	加藤	kato@example.com
7	和田	wada@example.com
8	桃田	momota@example.com
9	瀬野	seno@example.com
10	町田	machida@example.com

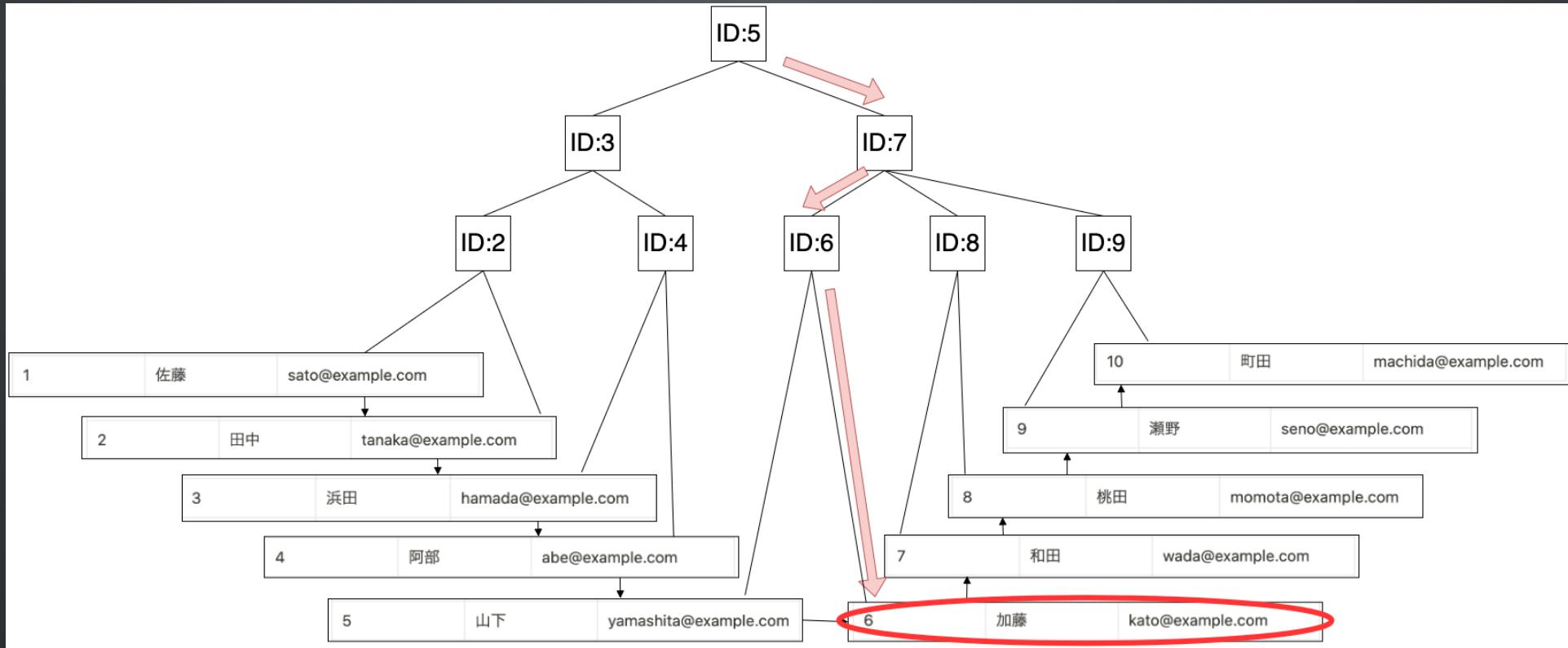
メールアドレスで検索できそう！

これで検索

ユーザーIDが6の人物の情報を探せ

```
SELECT * FROM users WHERE id = 6;
```

idがPRIMARY KEYだから
クラスタインデックス使えそう！！



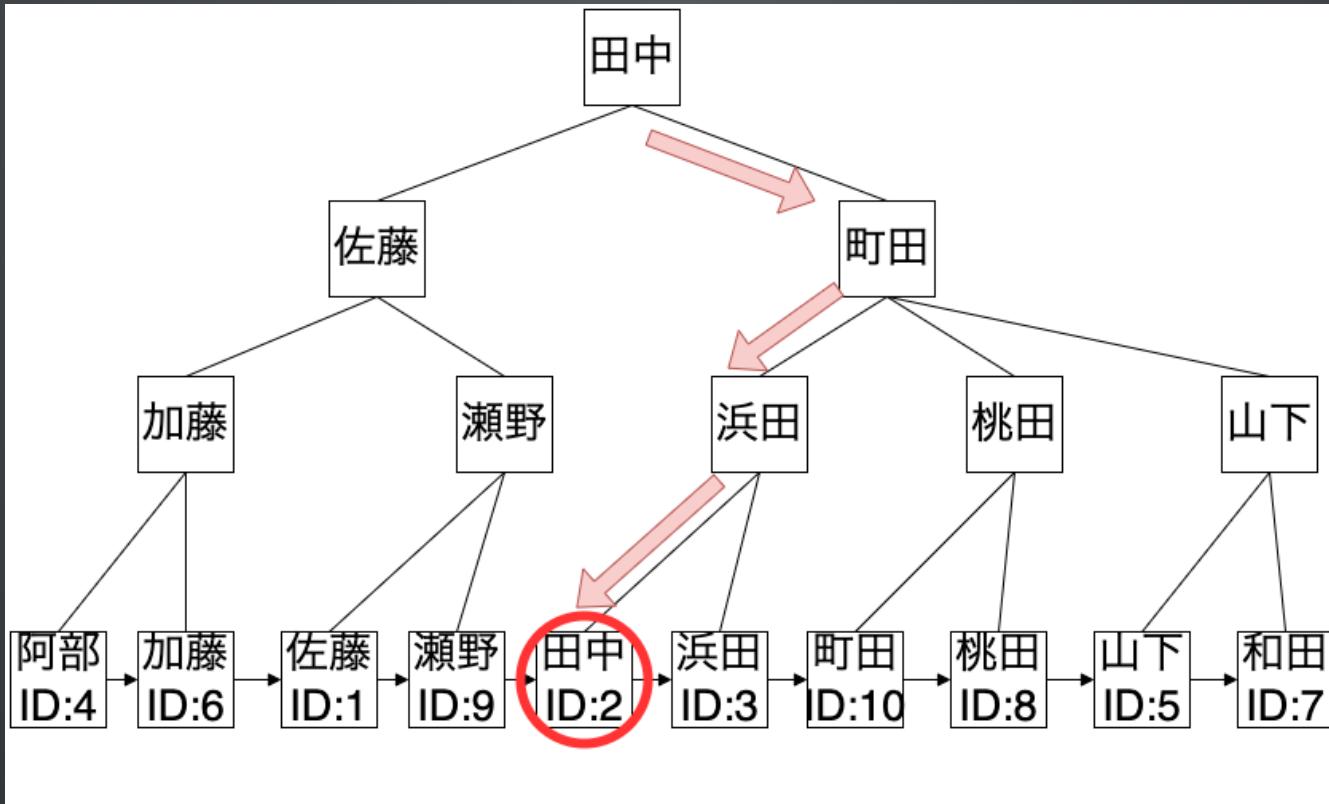
必要な情報を GET !

田中の人数を数えよ

```
SELECT count(name) FROM users WHERE name = '田中';
```

name だから

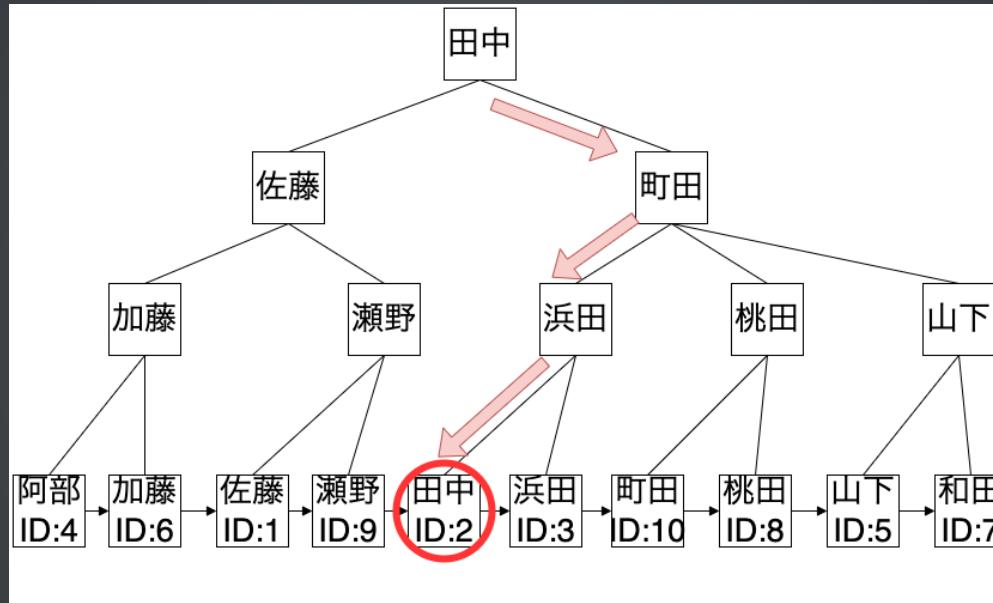
NAME のセカンダリインデックスが使えそう



必要な情報を GET !

田中のメールアドレスを探せ

```
SELECT name, email FROM users WHERE name = '田中';
```

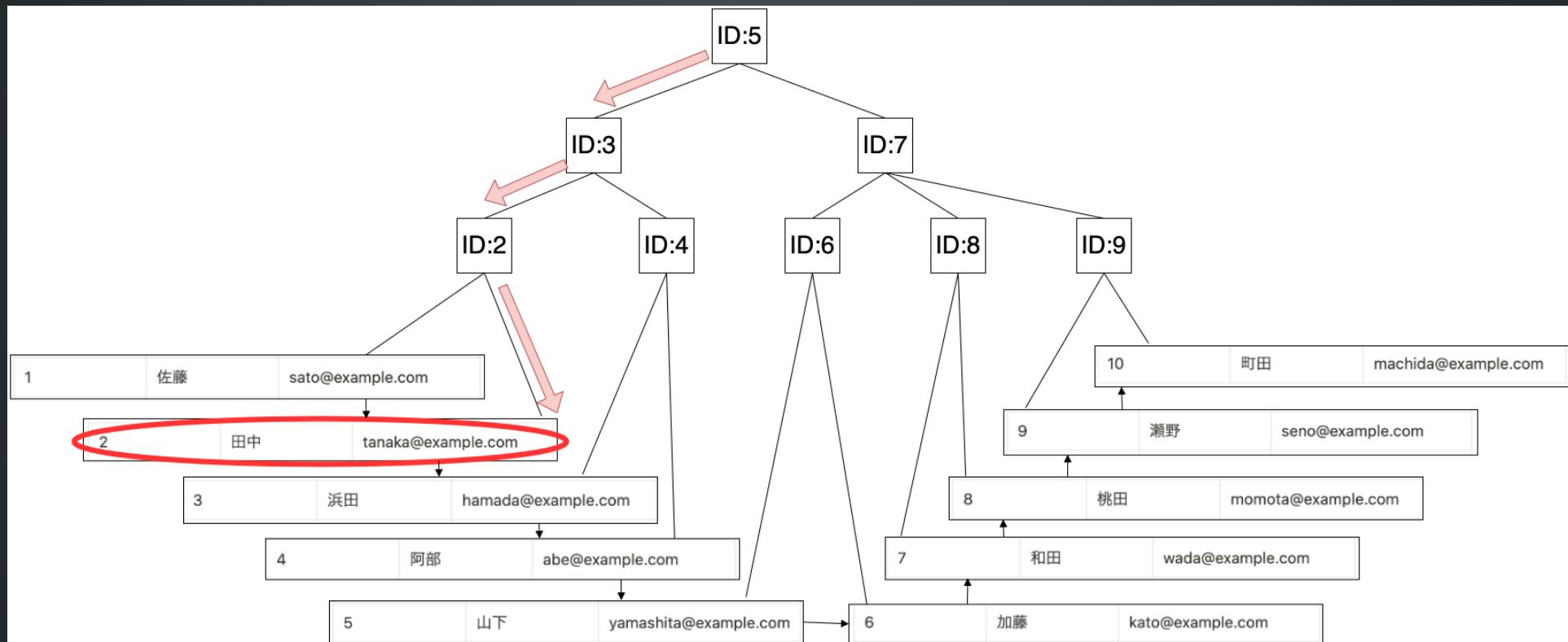


メールアドレスの情報がない！！

クラスタインデックスには全カラムの情報があるはず

田中のメールアドレスを探せ

ID: 2 を使ってクラスタインデックスで検索



メールアドレスの情報を GET !

なるほど、インデックスってこんな感じで使うんや

ちょっと待ってΣ(‘□’)

ユーザーIDが6の人物の情報を探
せ

「テーブルをフルスキャン」でもよくないか？

確かに(・▽・)

誰がインデックスを使うって決めてんの？

オプティマイザー

統計情報を元にデータへのアクセス方法を決定します