

計算機科学実験及演習4 データベース 課題5

1029-32-6611 山田裕晃

2022 年 10 月 21 日

1 概要

質問および更新を実行する SQL 文を作成する。

2 課題

2.1 関係代数の射影および選択に対応する SQL 文

```
1 eventdb=# SELECT user_id FROM reservation WHERE event_id = 1;
2 user_id
3 -----
4         1
5         3
6 (2 rows)
```

イベント ID1 に対する予約が存在するユーザーの ID を取得する処理である。すなわち京都大学 11 月祭に来場予約をしているユーザーの ID 一覧を取得している。

2.2 関係代数の自然結合に対応する SQL 文

```
1 eventdb=# SELECT user_id, name FROM reservation NATURAL INNER JOIN event_user WHERE event_id =
2         1;
3 user_id | name
4 -----+-----
5         1 | 山田
6         3 | 下田
7 (2 rows)
```

先程の処理に加え、event_user テーブルと reservation テーブルを自然結合させることで、user_id に対応する名前を表示させている。

2.3 UNION を含む SQL 文

```
1 eventdb=#SELECT user_id FROM reservation WHERE event_id = 1 UNION SELECT user_id FROM
2         reservation WHERE event_id = 2;
3 user_id
4 -----
5         1
6         3
7         4
8         2
9 (4 rows)
```

event_id が 1 または 2 のイベントに予約しているユーザーの ID 一覧を取得している。

2.4 EXCEPT を含む SQL 文

```
1 eventdb=#SELECT user_id FROM reservation WHERE event_id = 2 EXCEPT SELECT user_id FROM
      reservation WHERE event_id = 1;
2 user_id
3 -----
4          3
5 (1 row)
```

event_id が 2 のイベントに予約しているユーザーの ID から、event_id が 1 のイベントに予約しているユーザーの ID を除いたものを表示している。すなわち、計算機科学実験及演習 4 に予約し、京都大学 11 月祭に予約していないユーザーを取得している。

2.5 DISTINCT を含む SQL 文

```
1 eventdb=# SELECT DISTINCT name FROM reservation NATURAL INNER JOIN event_user;
2 name
3 -----
4 伊藤
5 山田
6 加藤
7 下田
8 (4 rows)
```

何らかの予約を行なっているユーザーの名前の一覧を表示させる。DISTINCT 句を使うことで、名前の重複をさせないようにしている。

2.6 集合関数 (COUNT, SUM, AVG, MAX, MIN) を用いた SQL 文

```
1 eventdb=# SELECT COUNT(user_id) FROM reservation WHERE event_id = 1;
2 count
3 -----
4          2
5 (1 row)
```

京都大学 11 月祭に来場予約をしているユーザーの数を取得している。

2.7 副質問 (sub query) を含む SQL 文

```
1 eventdb=# SELECT COUNT(user_id) FROM (SELECT user_id FROM reservation WHERE accepted = '1')
      as T;
2 count
3 -----
4          2
5 (1 row)
```

何らかのイベントに対する受付が終わっているユーザーの数を取得している。

2.8 UPDATE を含む SQL 文

```
1 eventdb=# UPDATE reservation SET accepted = '1' WHERE token = 'abcde';
2 UPDATE 1
3 eventdb=# SELECT * from reservation;
4  user_id | event_id | token | accepted
5  -----+-----+-----+-----
6         1 | 2 | fghij | 1
7         2 | 2 | klmno | 1
8         3 | 1 | pqrst | 0
9         4 | 2 | uvwxy | 0
10        1 | 1 | abcde | 1
11 (5 rows)
```

token が”abcde”の予約の受付状態を 1 に、すなわち受付済に変更する処理をしている。

2.9 ORDER BY を含む SQL 文

```
1 eventdb=# SELECT * FROM reservation ORDER BY user_id, event_id;
2  user_id | event_id | token | accepted
3  -----+-----+-----+-----
4         1 | 1 | abcde | 1
5         1 | 2 | fghij | 1
6         2 | 2 | klmno | 1
7         3 | 1 | pqrst | 0
8         4 | 2 | uvwxy | 0
9 (5 rows)
```

reservation テーブルを user_id, event_id の順に並べる処理をしている。

2.10 CREATE VIEW を含む SQL 文

```
1 eventdb=# CREATE VIEW november_festival AS SELECT user_id, event_id, token, accepted FROM
      reservation WHERE event_id = '1';
2 CREATE VIEW
3 eventdb=# SELECT * FROM november_festival;
4  user_id | event_id | token | accepted
5  -----+-----+-----+-----
6         3 | 1 | pqrst | 0
7         1 | 1 | abcde | 1
8 (2 rows)
```

event_id が 1、すなわち京都大学 11 月祭に対する予約のデータのみを表示する view を定義した。