

## 計算機科学実験及演習 4 データベース課題 3

1029259152 田中 勝也

構築したデータベースに対して, 以下の各内容の SQL 文を作成して, 実行してください. レポートでは, 各項目について, SQL 文, その説明および実行結果を示しなさい.

### 1. 関係代数の射影および選択に対応する SQL 文

- 射影

- SQL

- ```
SELECT title from threads;
```

threads テーブルの全ての行の title 属性を取り出す.

- 実行結果

- ```
sqlite> select title from threads;
foo
bar
baz
```

- 選択

- SQL

- ```
SELECT * FROM users WHERE user_id=4 AND password="2bcda11";
```

users テーブルから, user\_id が 4 かつ, password が 2bcda11 の行を全て取り出す.

- 実行結果

- ```
sqlite> select * from users where user_id=4 and password="2bcda11";
4|2bcda11
```

### 2. 関係代数の自然結合に対応する SQL 文

- SQL

- ```
SELECT * FROM create_thread NATURAL JOIN users;
```

create\_thread と users を自然結合したテーブルから全ての行を取り出す.

- 実行結果

```
sqlite> select * from create_thread natural join users;
0|0|1450414722187|100c90b
0|1|1450414722188|100c90b
0|2|1450414722188|100c90b
```

create\_thread(user\_id, thread\_id) に users(user\_id, password) が自然結合されているのがわかる。

### 3.UNION を含む SQL 文

- SQL

```
SELECT user_id FROM create_thread WHERE timestamp > 1450000000000
UNION SELECT user_id FROM post_comment WHERE timestamp > 1450000000000;
```

スレッドに最近関わったユーザの id を取得する。

- 実行結果

```
sqlite> select user_id from create_thread where timestamp > 1450000000000
...> union select user_id from post_comment where timestamp > 1450000000000;
0
1
2
```

### 4.EXCEPT を含む SQL 文

- SQL

```
SELECT user_id FROM post_comment WHERE timestamp > 1450000000000
EXCEPT SELECT user_id FROM create_thread;
```

最近コメントしているユーザのうち、スレッドを作ったことのないユーザの ID を全て抜きだす。

- 実行結果

```
sqlite> SELECT user_id FROM post_comment WHERE timestamp > 1450000000000
...> EXCEPT SELECT user_id FROM create_thread WHERE timestamp > 1450000000000;
1
2
```

## 5. DISTINCT を含む SQL 文

- SQL

```
SELECT DISTINCT user_id FROM post_comment WHERE timestamp > 1450000000000
```

最近コメントしているユーザの ID を重複なしで全て取り出す。

- 実行結果

```
sqlite> SELECT user_id FROM post_comment WHERE timestamp > 1450000000000;
0
0
1
2
sqlite> SELECT DISTINCT user_id FROM post_comment
...> WHERE timestamp > 1450000000000;
0
1
2
```

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

- SQL

```
SELECT COUNT(DISTINCT user_id) FROM post_comment WHERE timestamp > 1450000000000;
```

最近コメントしているユーザ id の総数を求める。

- 実行結果

```
sqlite> SELECT COUNT(DISTINCT user_id) FROM post_comment
...> WHERE timestamp > 1450000000000;
3
```

## 7. 副質問 (sub query) を含む SQL 文

- SQL

```
SELECT DISTINCT user_id FROM create_thread WHERE user_id IN (SELECT user_id FROM post_comment WHERE timestamp > 1450000000000);
```

最近コメントしているユーザの中で、スレッドを作ったことのあるユーザの id を求める。

- 実行結果

```
sqlite> select distinct user_id from create_thread where user_id in
...> (select user_id from post_comment where timestamp > 1450000000000);
0
```

## 8.UPDATE を含む SQL 文

- SQL

```
UPDATE users SET password = 'newPassWord' WHERE user_id = 2;
```

- 実行結果

```
sqlite> select password from users where user_id = 2;
22ee696
sqlite> UPDATE users SET password = 'newPassWord' WHERE user_id = 2;
sqlite> select password from users where user_id = 2;
newPassWord
```

## 9.ORDER BY を含む SQL 文

- SQL

```
SELECT * FROM create_thread ORDER BY timestamp;
```

- 実行結果

```
sqlite> SELECT * FROM create_thread ORDER BY timestamp;
0|0|1450418092209
0|1|1450418092210
0|2|1450418092210
```

## 10.CREATE VIEW を含む SQL 文

- SQL

```
CREATE VIEW creators AS SELECT * FROM users
WHERE user_id IN (SELECT user_id FROM create_thread);
```

- 実行結果

```
sqlite> create view creators as select * from users
...> where user_id in (select user_id from create_thread);
sqlite> select * from creators;
0|5e68bb8
```