

## 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

1. 表を正規化していく中で、注目するキーや関数従属性で結果がどのように変化するかを考察しなさい。

例えば 関係  $R(A, B, C, D)$  において、関数従属性

$$\text{FD1} : A, B \rightarrow C$$

$$\text{FD2} : C \rightarrow B$$

$$\text{FD3} : C \rightarrow D$$

が成り立っているとする。

$R$  の候補キーには  $\{A, B\}, \{A, C\}$  がある。

このとき、FD1 に注目して、分解法で関係  $R$  を分解すると、

$$\begin{aligned} &R_{1,1}(A, B, D)\{\} \\ &R_{1,2}(A, B, C)\{\text{FD1}, \text{FD2}\} \end{aligned}$$

となり、関数従属性 FD3 は失われる。また、 $R_{1,1}$  の候補キーは  $\{A, B\}$ 、 $R_{1,2}$  の候補キーは  $\{A, C\}$  となる。

一方、FD2 に注目して、分解法で関係  $R$  を分解すると、

$$\begin{aligned} &R_{2,1}(A, C, D)\{\text{FD3}\} \\ &R_{2,2}(B, C)\{\text{FD2}\} \end{aligned}$$

となり、関数従属性 FD1 は失われる。また、 $R_{2,1}$  の候補キーは  $\{A, C\}$  であるが、 $R_{2,2}$  の候補キーは  $\{B, C\}$  となってしまう。

このように、どの従属性に基づいて表を分解するのかによって、保持できる従属性やキーは変化するため、目的に応じたキーの選び方、分解の仕方をする必要がある。

2. 課題3 で設計した関係スキーマに基づいて関係表を定義しなさい。また定義するためのSQL 文を示しなさい。

今回は DBMS に SQLite を利用して関係表を定義した。

また、課題3 で設計した関係スキーマに加え、users(user\_id, password) という関係スキーマを追加した。

```
CREATE TABLE users (user_id int primary key, password text);
CREATE TABLE threads (thread_id int primary key, title text);
CREATE TABLE requests (request_id int primary key, _i int);
CREATE TABLE questions (questions_id int primary key, question text, _i int,
                           is_request int);
CREATE TABLE question_info (question primary key, answer text);
CREATE TABLE comments (comment_id int primary key, _i int, content text);
CREATE TABLE create_thread (user_id int, thread_id int, timestamp int,
                              primary key(user_id, thread_id),
                              foreign key(user_id) references users(user_id),
                              foreign key(thread_id) references threads(thread_id));
CREATE TABLE post_comment (user_id int, comment_id int, timestamp int,
                             primary key(user_id, comment_id),
                             foreign key(user_id) references users(user_id),
                             foreign key(comment_id) references comments(comment_id));
CREATE TABLE request_comment (request_id int, comment_id int,
                                primary key(request_id, comment_id),
                                foreign key(request_id) references requests(request_id),
                                foreign key(comment_id) references comments(comment_id));
CREATE TABLE thread_request (thread_id int, request_id int,
                               primary key(thread_id, request_id),
                               foreign key(thread_id) references threads(thread_id),
                               foreign key(request_id) references requests(request_id));
CREATE TABLE thread_question (thread_id int, question_id int,
                                primary key(thread_id, question_id),
                                foreign key(thread_id) references threads(thread_id),
                                foreign key(question_id) references questions(question_id));
CREATE TABLE send_request (user_id int, request_id int,
                             primary key(user_id, request_id),
                             foreign key(user_id) references users(user_id),
                             foreign key(request_id) references requests(request_id));
```

関係表を定義するに際して各関数従属性は保持されている。

3. データを作成して上記の表に挿入しなさい。またデータを挿入するための SQL 文を示しなさい。またデータを挿入した表の出力を示しなさい。

今回は以下のような users テーブルに以下のようなデータを挿入する。

(0, 204bd36) (1, 459c895) (2, 338d0e2) (3, 118a909) (4, 25d2432)  
(5, 1e00569) (6, 55d1c19) (7, 49f6db5) (8, 360e552) (9, 3fff8a9)

以下のような SQL を発行し、データを挿入した。

```
INSERT INTO users VALUES(0, 204bd36)
INSERT INTO users VALUES(1, 459c895)
INSERT INTO users VALUES(2, 338d0e2)
INSERT INTO users VALUES(3, 118a909)
INSERT INTO users VALUES(4, 25d2432)
INSERT INTO users VALUES(5, 1e00569)
INSERT INTO users VALUES(6, 55d1c19)
INSERT INTO users VALUES(7, 49f6db5)
INSERT INTO users VALUES(8, 360e552)
INSERT INTO users VALUES(9, 3fff8a9)
```

結果の一部は以下のようになっており、正しくデータを挿入できていることがわかる。

```
% sqlite3 task_4.sqlite3
SQLite version 3.8.5 2014-08-15 22:37:57
Enter ".help" for usage hints.
sqlite> select * from users where user_id < 5;
0|204bd36
1|459c895
2|338d0e2
3|118a909
4|25d2432
```