

Sharing My Food

The **Sharing My Food** System is designed to help individuals in a community share leftover or excess food with others who might need it. The system aims to reduce food waste, promote sustainability, and assist people in need.

Tables

1. User Table

* **Purpose:** Stores information about users who donate or request food.

* **Attributes:**

- user_id: A unique identifier for each user (Primary Key).
- full_name: The full name of the user.
- email: The email address of the user (unique).
- password: The user's password for authentication.
- phone_number: The user's phone number.
- address: The address where the user is located.

2. Food Table

* **Purpose:** Holds details about food items that users wish to donate.

* **Attributes:**

- food_id: A unique identifier for each food item (Primary Key).
- name: The name of the food item (e.g., Pasta, Pizza).
- quantity: The amount of food available.
- status: Indicates whether the food is available or finished (using a CHECK constraint for 'Available' or 'Finish').
- expire_date: The expiration date of the food item.
- user_id: The ID of the user who donated the food (Foreign Key referencing the user table).

3. Request Table

* **Purpose:** Stores the details of requests made by users for food donations.

* **Attributes:**

- request_id: A unique identifier for each request (Primary Key).
- status: The status of the request (Pending, Accepted, or Cancelled, using a CHECK constraint).
- delivery_address: The address to which the food is to be delivered.
- request_date: The date and time when the request was made.
- delivery_date: The date when the delivery is expected.
- delivery_status: The status of the delivery (In Progress, Out for Delivery, Delivered, or Cancelled).
- user_id: The ID of the user making the request (Foreign Key referencing the user table).
- food_id: The ID of the food item being requested (Foreign Key referencing the food table).

4. Feedback Table

***Purpose:** Allows users to provide feedback on the food they have received..

***Attributes:**

- feedback_id: A unique identifier for each feedback (Primary Key).
- rating: The rating given by the user (between 1 and 5).
- feedback_date: The date when the feedback was provided.
- description: A textual description of the feedback.
- request_id: The ID of the request associated with the feedback (Foreign Key referencing the request table).

SQL Commands

#DDL#

```
create database sharingFood;
```

```
create table user(
    user_id int primary key,
    full_name varchar(20) not null,
    email varchar(20) not null unique,
    password varchar(30) not null,
    phone_number varchar(10) not null,
    address varchar(20)
);
```

```

create table food(
    food_id int primary key,
    name varchar(10) not null,
    quantity int default 0,
    status varchar(15) check ( status= 'Available' or status= 'Finish'),
    expire_date varchar(11),
    user_id int,
    foreign key (user_id) references user(user_id)

);

create table request(
    request_id int primary key,
    status varchar(20) check ( status= 'Pending' or status= 'Accepted' or status= 'Cancelled' ),
    delivery_address varchar(20) not null,
    request_date datetime DEFAULT CURRENT_TIMESTAMP,
    delivery_date varchar(11),
    delivery_status varchar(16) check ( delivery_status= 'In Progress' or delivery_status= 'Out
for Delivery' or delivery_status= 'Delivered' or delivery_status= 'Cancelled' ),
    user_id int,
    food_id int,
    foreign key (user_id) references user(user_id),
    foreign key (food_id) references food(food_id)
);

create table feedback(
    feedback_id int primary key,
    rating int check ( rating between 1 and 5),
    feedback_date varchar(11),
    description varchar(40),
    request_id int,
    foreign key (request_id) references request(request_id)
);

```

```

#####
#####

```

#DML#

```

insert into user (user_id, full_name, email, password, phone_number, address) values
(012, 'Ahmed Nasser', 'ahmed@gmail.com', 'password123', '0501234567', 'Al Narjis, 224'),
(289, 'Deema Fahad', 'deema@gmail.com', 'password456', '0502345678', 'Al Yasmeen, 102'),
(355, 'Omar Abdullah', 'omar@gmail.com', 'password789', '0503456789', 'Al Malqa, 344'),
(441, 'Reem Ali', 'reem@gmail.com', 'password000', '0504567890', 'Hitten, 212'),
(093, 'Mohammed Abdullah', 'mohammad@gmail.com', 'password111', '0505678901', 'Al
wadi, 150');

```

```

insert into food (food_id, name, quantity, status, expire_date, user_id) values

```

```
(08, 'Pasta', 0, 'Finish', '2024-11-28', 441),  
(67, 'Pizza', 20, 'Available', '2024-11-28', 441),  
(49, 'Salad', 10, 'Available', '2024-11-30', 012),  
(02, 'Sandwich', 4, 'Available', '2024-11-29', 289),  
(97, 'Soup', 0, 'Finish', '2024-11-30', 355);
```

```
insert into request (request_id, status, delivery_address, delivery_date, delivery_status,  
user_id, food_id) values  
(300, 'Pending', 'Al Malqa, 344', '2024-11-29', 'Out for Delivery', 12, 49),  
(242, 'Accepted', 'Al Yasmeen, 102', '2024-11-29', 'Delivered', 355, 97),  
(561, 'Pending', 'Hitten, 212', '2024-11-28', 'In Progress', 441, 2),  
(206, 'Cancelled', 'Al Narjis, 224', '2024-11-28', 'Cancelled', 289, 67);
```

```
insert into feedback (feedback_id, rating, feedback_date, description, request_id) values  
(1, 5, '2024-11-30', 'The salad was delicious!', 300),  
(2, 4, '2024-12-01', 'The sandwich was good!', 561),  
(3, 3, '2024-12-01', 'The soup was too cold!', 242);
```

```
select * from user;
```

```
select * from food;
```

```
select * from request;
```

```
select * from feedback;
```

```
update request  
set status= 'Accepted'  
where request_id=300;
```

```
update food  
set quantity= 0, status= 'Finish'  
where food_id= 02;
```

```
update user  
set password= 'password077'  
where user_id= 012;
```

```
update feedback  
set rating= 5  
where feedback_id= 2;
```

```
delete from feedback  
where feedback_id= 2;
```

#To delete from food table, must first delete reference row at request table

delete from request
where request_id= 206;

delete from food
where food_id= 67;

delete from user
where user_id= 093;

Diagram (DataGrip)

