Info 6210 Final Project

Haoxuan Su

https://github.com/SugerSu/INFO6210Final

contents

1 ·Topic

2 ·Objects

3 ·Implementation

4 ·Report

01

Problem Statement

PART ONE

Problem

The application of cookbook only allow users to check out existing recipes. That means users can’t interact with each other. And the cookbook is only be required by the person who wants to cook.

My goal is to add social feature for the cookbook. That means we can not only use this cookbook app for cooking, but also for social network. That will attract more people to use the app and will make the cooking becomes more interesting.

02

Objects

PART TWO

ER Diagram

User Type

User

Comment

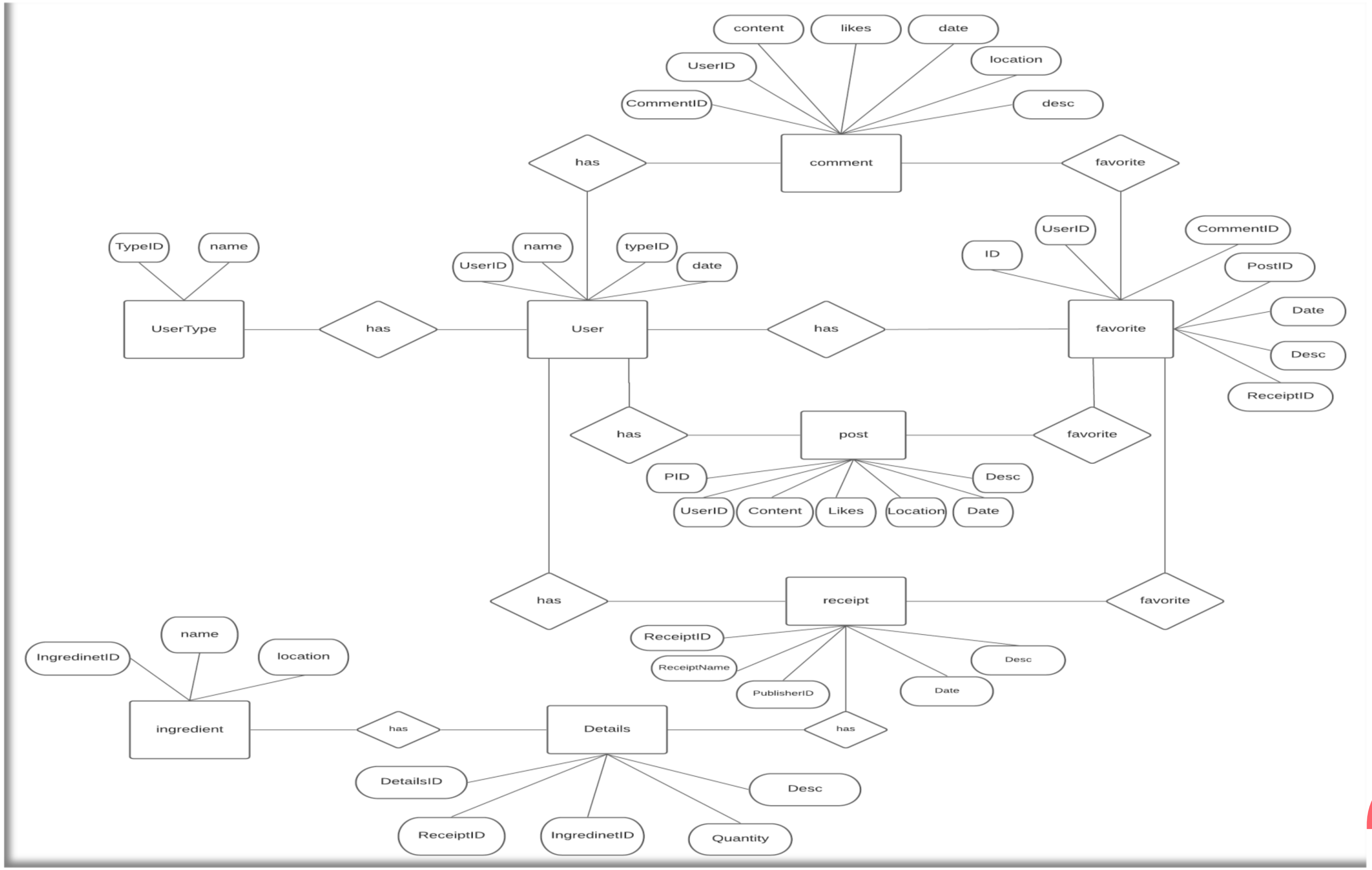
Post

Favorite

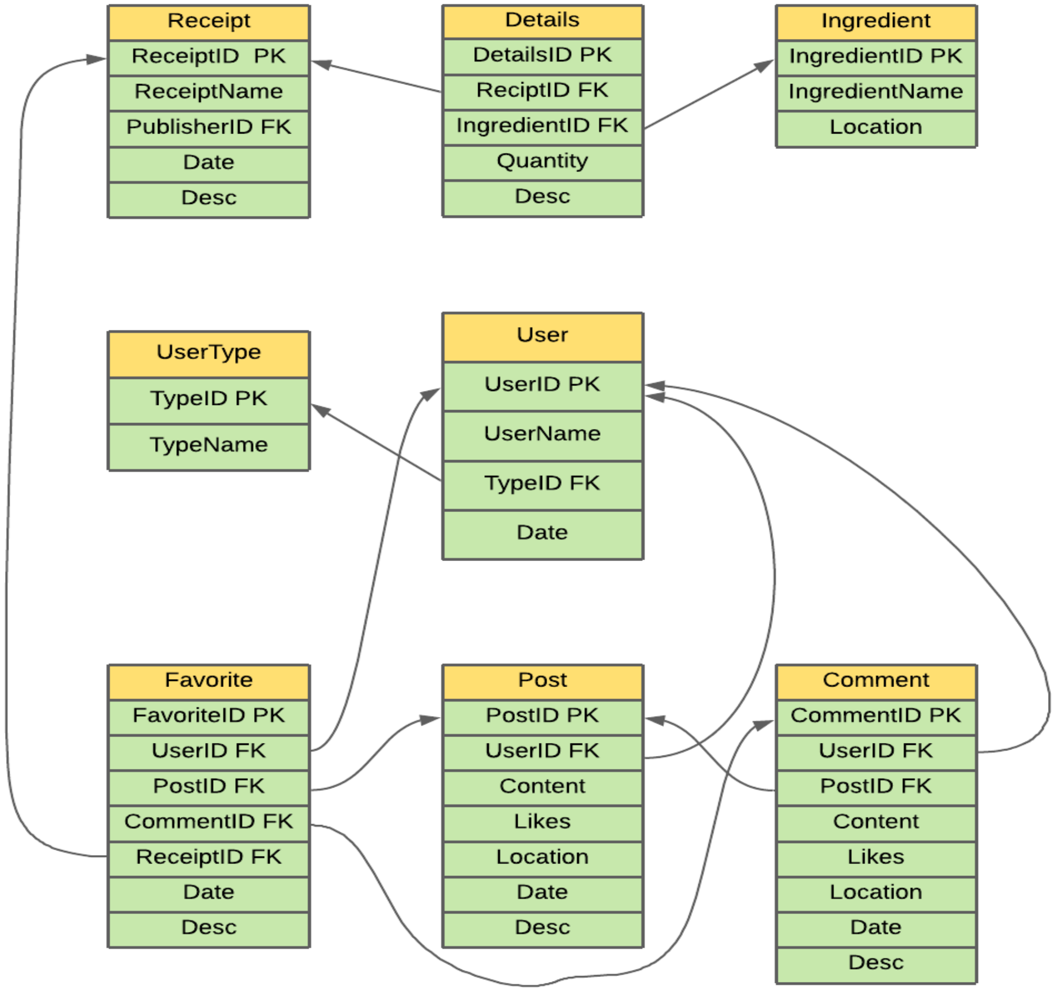
Receipt

Details

Ingredients



Tables



Receipt
ReceiptID PK
ReceiptName
PublisherID FK
Date
Desc

Details
DetailsID PK
ReciptID FK
IngredientID FK
Quantity
Desc

Ingredient
IngredientID PK
IngredientName
Location

UserType
TypeID PK
TypeName

User
UserID PK
UserName
TypeID FK
Date

Favorite
FavoriteID PK
UserID FK
PostID FK
CommentID FK
ReceiptID FK
Date
Desc

Post
PostID PK
UserID FK
Content
Likes
Location
Date
Desc

Comment
CommentID PK
UserID FK
PostID FK
Content
Likes
Location
Date
Desc

User Type, User

Combine these two table will get entire information of a user.

Receipt, Details, Ingredient

Combine these tables will get the entire information of a receipt.

Favorites

This table store all the favorites include receipts, comments and posts

Comment

This table stores all the comments.

Post

This table stores all the posts information of users.

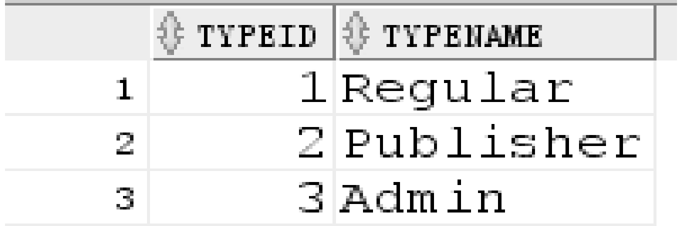
03

Implementation

PART THREE

Tables

User Type Table:



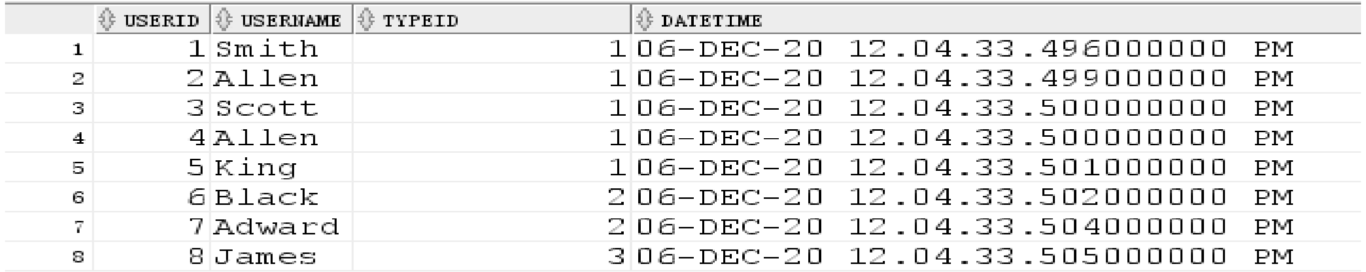
	TYPEID	TYPENAME
1	1	Regular
2	2	Publisher
3	3	Admin

Script:

```
--create table UserType
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where
table_name='COOKBOOK_USERTYPE';
 if(tcnt<=0) then
    EXECUTE IMMEDIATE 'create table cookbook_usertype (
    TypeID number,
    TypeName varchar2(50),
    constraint pk_usertype primary key(TypeID)
    )';
 end if;
end;
/
```

Tables

User Table:



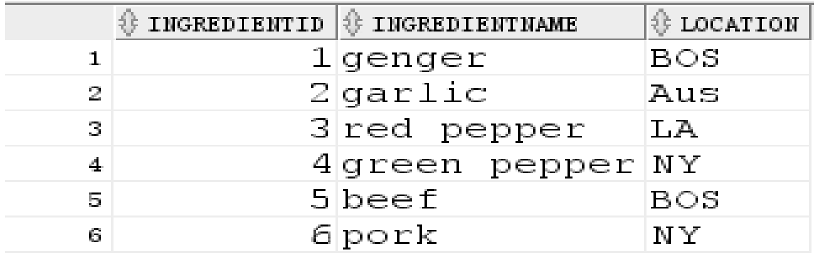
	USERID	USERNAME	TYPEID	DATETIME
1	1	Smith	1	06-DEC-20 12.04.33.496000000 PM
2	2	Allen	1	06-DEC-20 12.04.33.499000000 PM
3	3	Scott	1	06-DEC-20 12.04.33.500000000 PM
4	4	Allen	1	06-DEC-20 12.04.33.500000000 PM
5	5	King	1	06-DEC-20 12.04.33.501000000 PM
6	6	Black	2	06-DEC-20 12.04.33.502000000 PM
7	7	Adward	2	06-DEC-20 12.04.33.504000000 PM
8	8	James	3	06-DEC-20 12.04.33.505000000 PM

Script:

```
--create table cookbook_user
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where table_name='COOKBOOK_USER';
 if(tcnt<=0) then
EXECUTE IMMEDIATE 'create table cookbook_user (
    UserID number generated always as identity,
    UserName varchar2(50),
    TypeID number default 1,
    DateTime timestamp default systimestamp,
    constraint pk_user primary key(UserID),
    constraint fk_usertype foreign key(TypeID) references COOKBOOK_USERTYPE(typeID)
on delete set null
    )'; end if;
end;
/
```

Tables

Ingredient Table:



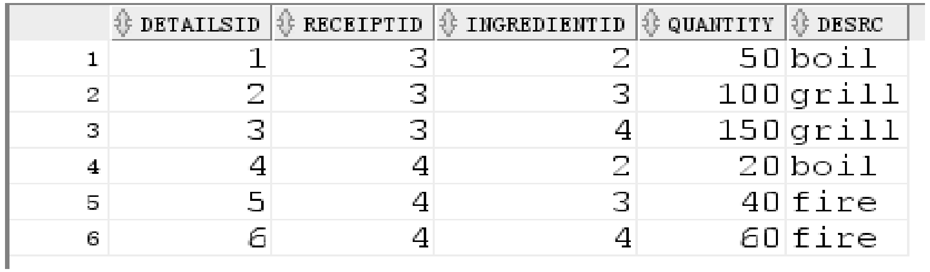
	INGREDIENTID	INGREDIENTNAME	LOCATION
1	1	genger	BOS
2	2	garlic	Aus
3	3	red pepper	LA
4	4	green pepper	NY
5	5	beef	BOS
6	6	pork	NY

Script:

```
--create table ingredient
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where
table_name='COOKBOOK_INGREDIENT';
 if(tcnt<=0) then
    EXECUTE IMMEDIATE 'create table cookbook_ingredient (
    IngredientID number generated always as identity,
    IngredientName varchar2(20),
    Location varchar(50),
    constraint pk_ingredient primary key(IngredientID)
    )’;
 end if;
end;
/
```

Tables

Receipt Details Table:



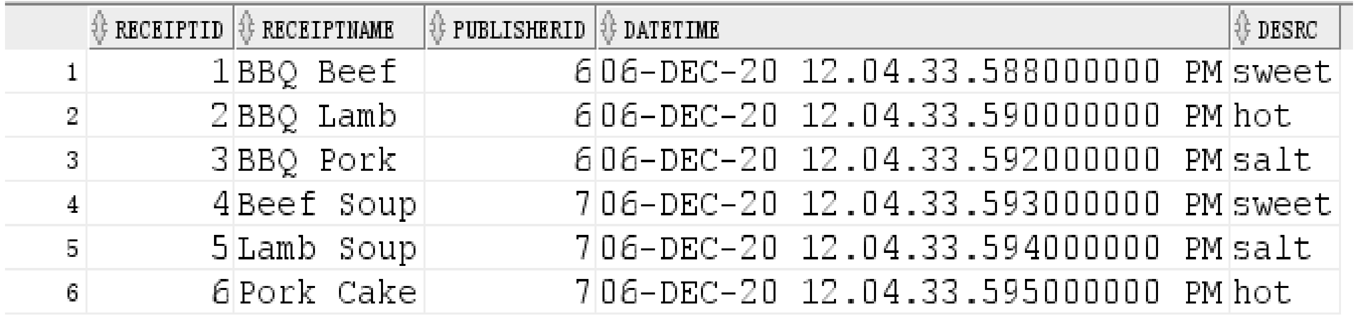
	DETAILSID	RECEIPTID	INGREDIENTID	QUANTITY	DESRC
1	1	3	2	50	boil
2	2	3	3	100	grill
3	3	3	4	150	grill
4	4	4	2	20	boil
5	5	4	3	40	fire
6	6	4	4	60	fire

Script:

```
--create table ReceiptDetails
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where
table_name='COOKBOOK_DETAILS';
 if(tcnt<=0) then
    EXECUTE IMMEDIATE 'create table cookbook_details (
    DetailsID number generated always as identity,
    ReceiptID number,
    IngredientID number,
    Quantity number,
    Desrc varchar(50),
    constraint pk_detials primary key(DetailsID),
    constraint fk_ingredient foreign key(IngredientID) references
cookbook_ingredient(IngredientID) on delete cascade
    )';
 end if;
end;
/
```

Tables

Receipt Table:



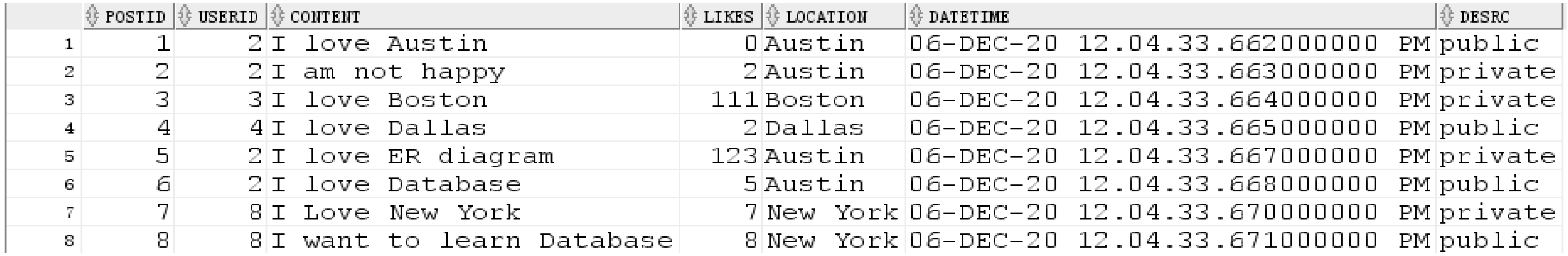
	RECEIPTID	RECEIPTNAME	PUBLISHERID	DATETIME	DESRC
1	1	BBQ Beef	6	06-DEC-20 12.04.33.588000000 PM	sweet
2	2	BBQ Lamb	6	06-DEC-20 12.04.33.590000000 PM	hot
3	3	BBQ Pork	6	06-DEC-20 12.04.33.592000000 PM	salt
4	4	Beef Soup	7	06-DEC-20 12.04.33.593000000 PM	sweet
5	5	Lamb Soup	7	06-DEC-20 12.04.33.594000000 PM	salt
6	6	Pork Cake	7	06-DEC-20 12.04.33.595000000 PM	hot

Script:

```
--create table Receipt
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where table_name='COOKBOOK_RECEIPT';
 if(tcnt<=0) then
EXECUTE IMMEDIATE 'create table cookbook_receipt (
    ReceiptID number generated always as identity,
    ReceiptName varchar2(50) not null,
    PublisherID number not null,--userid
    DateTime timestamp default systimestamp,
    Desrc varchar(50),
    constraint pk_receipt primary key(ReceiptID),
    constraint fk_user foreign key(PublisherID) references cookbook_user(UserID) on delete
cascade
    )’;
 end if;
end;
/
```

Tables

Post Table:



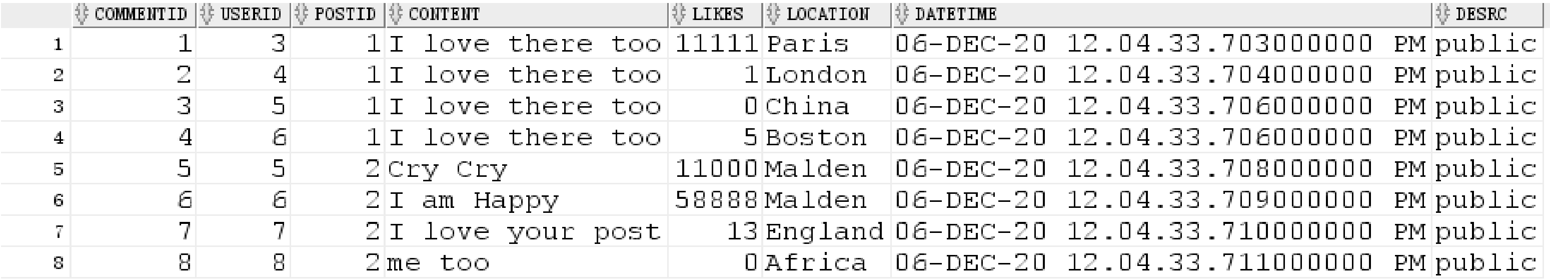
	POSTID	USERID	CONTENT	LIKES	LOCATION	DATETIME	DESRC
1	1	2	I love Austin	0	Austin	06-DEC-20 12.04.33.662000000 PM	public
2	2	2	I am not happy	2	Austin	06-DEC-20 12.04.33.663000000 PM	private
3	3	3	I love Boston	111	Boston	06-DEC-20 12.04.33.664000000 PM	private
4	4	4	I love Dallas	2	Dallas	06-DEC-20 12.04.33.665000000 PM	public
5	5	2	I love ER diagram	123	Austin	06-DEC-20 12.04.33.667000000 PM	private
6	6	2	I love Database	5	Austin	06-DEC-20 12.04.33.668000000 PM	public
7	7	8	I Love New York	7	New York	06-DEC-20 12.04.33.670000000 PM	private
8	8	8	I want to learn Database	8	New York	06-DEC-20 12.04.33.671000000 PM	public

Script:

```
--create table cookbook_post
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where table_name='COOKBOOK_POST';
 if(tcnt<=0) then
     EXECUTE IMMEDIATE 'create table cookbook_post (
    PostID number generated always as identity,
    UserID number not null,
    Content varchar2(50),
    Likes number,
    Location varchar2(50),
    DateTime timestamp default systimestamp,
    Desrc varchar2(20),
    constraint pk_post primary key(PostID),
    constraint fk_user_post foreign key(UserID) references COOKBOOK_USER(UserID) on delete
cascade
    )’;
 end if;
end;
/
```

Tables

Comment Table:



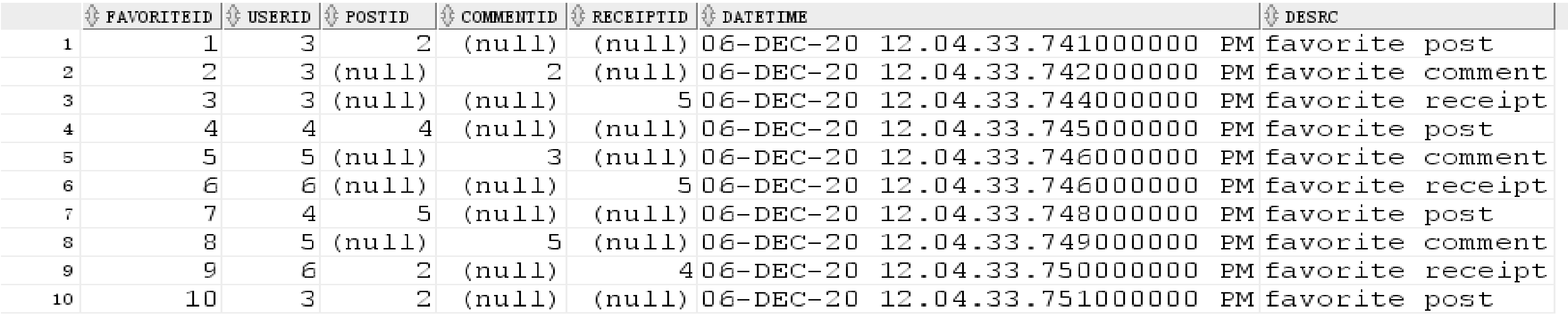
	COMMENTID	USERID	POSTID	CONTENT	LIKES	LOCATION	DATETIME	DESRC
1	1	3	1	I love there too	11111	Paris	06-DEC-20 12.04.33.703000000 PM	public
2	2	4	1	I love there too	1	London	06-DEC-20 12.04.33.704000000 PM	public
3	3	5	1	I love there too	0	China	06-DEC-20 12.04.33.706000000 PM	public
4	4	6	1	I love there too	5	Boston	06-DEC-20 12.04.33.706000000 PM	public
5	5	5	2	Cry Cry	11000	Malden	06-DEC-20 12.04.33.708000000 PM	public
6	6	6	2	I am Happy	58888	Malden	06-DEC-20 12.04.33.709000000 PM	public
7	7	7	2	I love your post	13	England	06-DEC-20 12.04.33.710000000 PM	public
8	8	8	2	me too	0	Africa	06-DEC-20 12.04.33.711000000 PM	public

Script:

```
--create table cookbook_comment
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where table_name='COOKBOOK_COMMENT';
 if(tcnt<=0) then
    EXECUTE IMMEDIATE 'create table cookbook_comment (
    CommentID number generated always as identity,
    UserID number not null,
    PostID number not null,
    Content varchar2(50),
    Likes number,
    Location varchar2(50),
    DateTime timestamp default systimestamp,
    Desrc varchar2(20),
    constraint pk_comment primary key(CommentID),
    constraint fk_user_comment foreign key(UserID) references cookbook_user(UserID) on delete cascade,
    constraint fk_post_comment foreign key(PostID) references cookbook_post(PostID) on delete cascade
    )’;
 end if;
end;
/
```

Tables

Favorite Table:



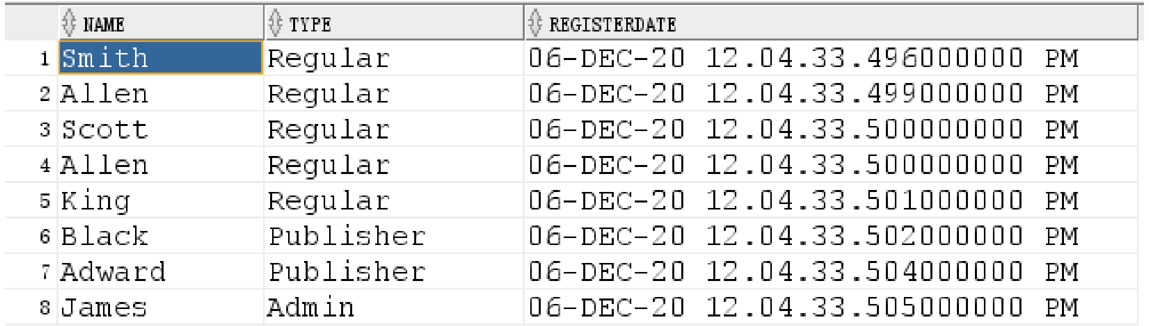
	FAVORITEID	USERID	POSTID	COMMENTID	RECEIPTID	DATETIME	DESRC
1	1	3	2	(null)	(null)	06-DEC-20 12.04.33.741000000 PM	favorite post
2	2	3	(null)	2	(null)	06-DEC-20 12.04.33.742000000 PM	favorite comment
3	3	3	(null)	(null)	5	06-DEC-20 12.04.33.744000000 PM	favorite receipt
4	4	4	4	(null)	(null)	06-DEC-20 12.04.33.745000000 PM	favorite post
5	5	5	(null)	3	(null)	06-DEC-20 12.04.33.746000000 PM	favorite comment
6	6	6	(null)	(null)	5	06-DEC-20 12.04.33.746000000 PM	favorite receipt
7	7	4	5	(null)	(null)	06-DEC-20 12.04.33.748000000 PM	favorite post
8	8	5	(null)	5	(null)	06-DEC-20 12.04.33.749000000 PM	favorite comment
9	9	6	2	(null)	4	06-DEC-20 12.04.33.750000000 PM	favorite receipt
10	10	3	2	(null)	(null)	06-DEC-20 12.04.33.751000000 PM	favorite post

Script:

```
declare
    tcnt number;
begin
 select count(*) into tcnt from user_tables where table_name='COOKBOOK_FAVORITE';
 if(tcnt<=0) then
    EXECUTE IMMEDIATE 'create table cookbook_favorite (
    FavoriteID number generated always as identity,
    UserID number not null,
    PostID number,
    CommentID number,
    ReceiptID number,
    DateTime timestamp default systimestamp,
    Desrc varchar2(20),
    constraint pk_favorite primary key(FavoriteID),
    constraint fk_user_favorite foreign key(UserID) references cookbook_user(UserID) on delete cascade,
    constraint fk_post_favorite foreign key(PostID) references cookbook_post(PostID) on delete cascade,
    constraint fk_comment_favorite foreign key(CommentID) references cookbook_comment(CommentID) on delete cascade,
    constraint fk_receipt_favorite foreign key(ReceiptID) references cookbook_receipt(ReceiptID) on delete cascade
    )’;
 end if;
end;
/
```

Views

User Information View:



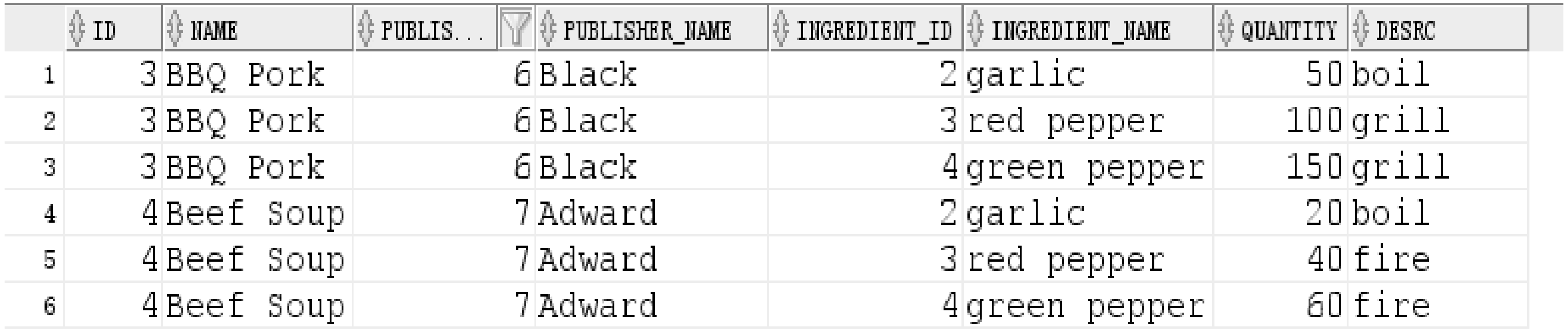
	NAME	TYPE	REGISTERDATE
1	Smith	Regular	06-DEC-20 12.04.33.496000000 PM
2	Allen	Regular	06-DEC-20 12.04.33.499000000 PM
3	Scott	Regular	06-DEC-20 12.04.33.500000000 PM
4	Allen	Regular	06-DEC-20 12.04.33.500000000 PM
5	King	Regular	06-DEC-20 12.04.33.501000000 PM
6	Black	Publisher	06-DEC-20 12.04.33.502000000 PM
7	Adward	Publisher	06-DEC-20 12.04.33.504000000 PM
8	James	Admin	06-DEC-20 12.04.33.505000000 PM

Script:

```
Create or replace view v_user_detials as
select u.username name, t.typename type, u.datetime registerdate
from cookbook_user u join cookbook_usertype t
on u.typeid = t.typeid;
```

Views

Receipt Details View:



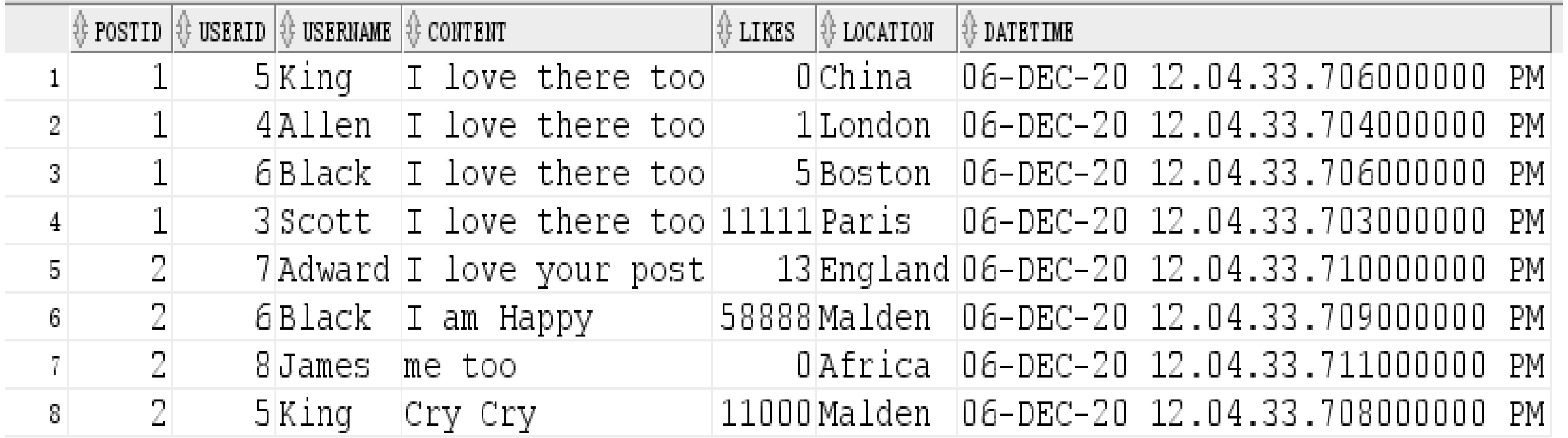
	ID	NAME	PUBLIS...	PUBLISHER_NAME	INGREDIENT_ID	INGREDIENT_NAME	QUANTITY	DESRC
1	3	BBQ Pork	6	Black	2	garlic	50	boil
2	3	BBQ Pork	6	Black	3	red pepper	100	grill
3	3	BBQ Pork	6	Black	4	green pepper	150	grill
4	4	Beef Soup	7	Adward	2	garlic	20	boil
5	4	Beef Soup	7	Adward	3	red pepper	40	fire
6	4	Beef Soup	7	Adward	4	green pepper	60	fire

Script:

```
create or replace view v_receipt_details as
select a.receiptid id,a.receiptname name,a.publisherid publisher_id,d.username
        publisher_name,b.ingredientid ingredient_id,c.ingredientname
        ingredient_name,b.quantity,b.desrc
from cookbook_receipt a ,cookbook_details b ,cookbook_ingredient c,
cookbook_user d
where a.receiptid=b.receiptid and b.ingredientid=c.ingredientid and
a.publisherid=d.userid;
```

Views

Comment Details View:



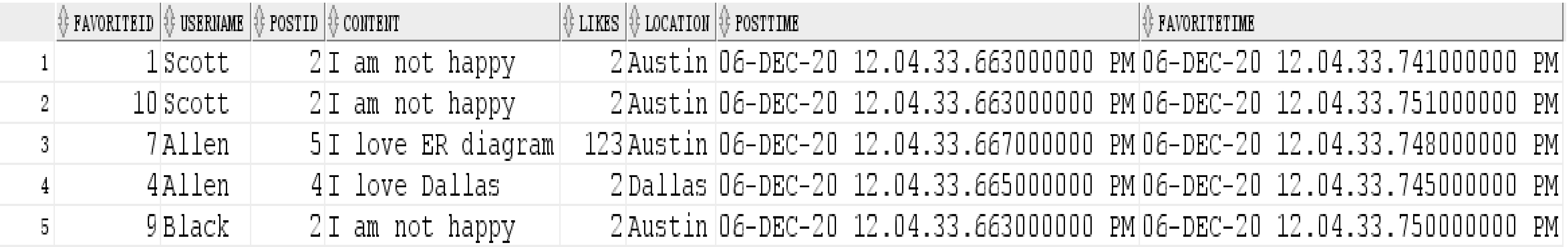
	POSTID	USERID	USERNAME	CONTENT	LIKES	LOCATION	DATETIME
1	1	5	King	I love there too	0	China	06-DEC-20 12.04.33.706000000 PM
2	1	4	Allen	I love there too	1	London	06-DEC-20 12.04.33.704000000 PM
3	1	6	Black	I love there too	5	Boston	06-DEC-20 12.04.33.706000000 PM
4	1	3	Scott	I love there too	11111	Paris	06-DEC-20 12.04.33.703000000 PM
5	2	7	Adward	I love your post	13	England	06-DEC-20 12.04.33.710000000 PM
6	2	6	Black	I am Happy	58888	Malden	06-DEC-20 12.04.33.709000000 PM
7	2	8	James	me too	0	Africa	06-DEC-20 12.04.33.711000000 PM
8	2	5	King	Cry Cry	11000	Malden	06-DEC-20 12.04.33.708000000 PM

Script:

```
create or replace view v_comment_detials as
select c.postid,u.userid,u.username,c.content,c.likes,c.location,c.datetime
from cookbook_comment c, cookbook_user u
where c.userid=u.userid
order by c.postid;
```

Views

Favorite Post View:



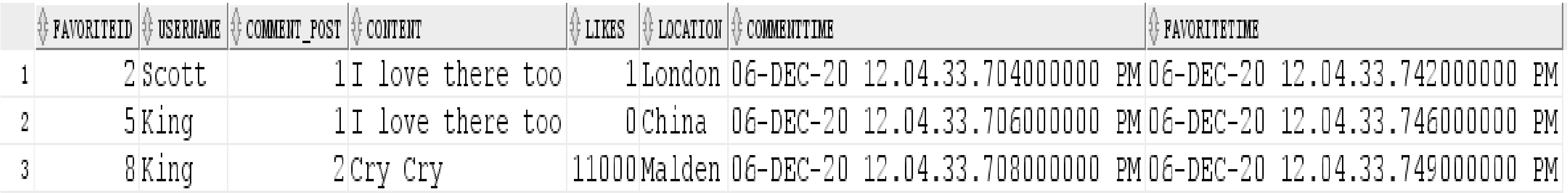
	FAVORITEID	USERNAME	POSTID	CONTENT	LIKES	LOCATION	POSTTIME	FAVORITETIME
1	1	Scott	2	I am not happy	2	Austin	06-DEC-20 12.04.33.663000000 PM	06-DEC-20 12.04.33.741000000 PM
2	10	Scott	2	I am not happy	2	Austin	06-DEC-20 12.04.33.663000000 PM	06-DEC-20 12.04.33.751000000 PM
3	7	Allen	5	I love ER diagram	123	Austin	06-DEC-20 12.04.33.667000000 PM	06-DEC-20 12.04.33.748000000 PM
4	4	Allen	4	I love Dallas	2	Dallas	06-DEC-20 12.04.33.665000000 PM	06-DEC-20 12.04.33.745000000 PM
5	9	Black	2	I am not happy	2	Austin	06-DEC-20 12.04.33.663000000 PM	06-DEC-20 12.04.33.750000000 PM

Script:

```
create or replace view v_favorite_posts as
select fa.favoriteid,u.username,p.postid,p.content,p.likes,p.location,p.datetime
        posttime,fa.datetime favoritetime from cookbook_favorite fa,
cookbook_user  u,cookbook_post p
where fa.userid=u.userid and fa.postid is not null and fa.postid=p.postid
order by fa.userid;
```

Views

Favorite Comment View:



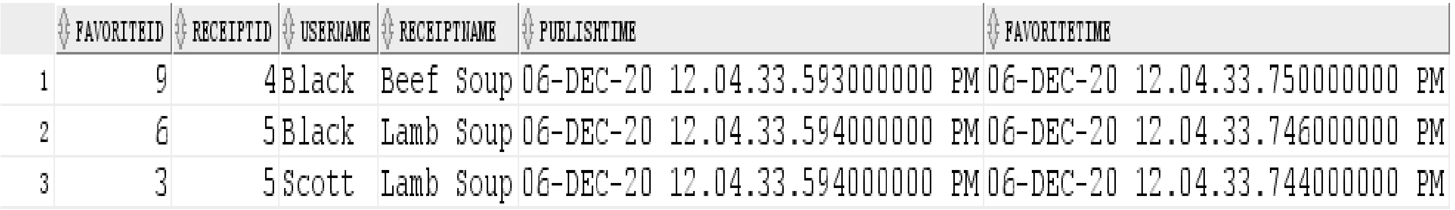
	FAVORITEID	USERNAME	COMMENT_POST	CONTENT	LIKES	LOCATION	COMMENTTIME	FAVORITETIME
1	2	Scott	1	I love there too	1	London	06-DEC-20 12.04.33.704000000 PM	06-DEC-20 12.04.33.742000000 PM
2	5	King	1	I love there too	0	China	06-DEC-20 12.04.33.706000000 PM	06-DEC-20 12.04.33.746000000 PM
3	8	King	2	Cry Cry	11000	Malden	06-DEC-20 12.04.33.708000000 PM	06-DEC-20 12.04.33.749000000 PM

Script:

```
create or replace view v_favorite_comment as
select fa.favoriteid,u.username,c.postid Comment_post,c.content,c.likes,c.location,c.datetime
        commenttime,fa.datetime favoritetime from cookbook_favorite fa, cookbook_user
        u,cookbook_comment c
where fa.userid=u.userid and fa.commentid is not null and fa.commentid=c.commentid
order by c.postid;
```

Views

Favorite Receipt View:



	FAVORITEID	RECEIPTID	USERNAME	RECEIPTNAME	PUBLISHTIME	FAVORITETIME
1	9	4	Black	Beef Soup	06-DEC-20 12.04.33.593000000 PM	06-DEC-20 12.04.33.750000000 PM
2	6	5	Black	Lamb Soup	06-DEC-20 12.04.33.594000000 PM	06-DEC-20 12.04.33.746000000 PM
3	3	5	Scott	Lamb Soup	06-DEC-20 12.04.33.594000000 PM	06-DEC-20 12.04.33.744000000 PM

Script:

```
create or replace view v_favorite_receipt as
select fa.favoriteid,r.receiptid ,u.username,r.receiptname,r.datetime publishtime,fa.datetime favoritetime
from cookbook_favorite fa, cookbook_user u,cookbook_receipt r
where fa.userid=u.userid and fa.receiptid is not null and fa.receiptid=r.receiptid
order by r.receiptid,u.username;
```

Indexes

```
create index username_idx on cookbook_user (upper(username));
select  upper(username) from cookbook_user  where upper(username) like upper('%al%');

create index receipt_name_idx on cookbook_receipt (upper(receiptname));
select * from cookbook_receipt where upper(receiptname) like upper(‘%bbq%’);

create index ingredient_name_idx on cookbook_ingredient (upper(ingredientname));
select * from cookbook_ingredient where upper(ingredientname) like upper('%beef%’);

create index ingredient_location_idx on cookbook_ingredient (upper(location));
select * from cookbook_ingredient where upper(location) like upper(‘%bos%');
```

These four indexes created because the user will query these information more frequently by fuzzy query. And we modify these columns rarely. So, it is good to use indexes to improve query speed.

Triggers

```
create or replace trigger update_usertypeid_trigger
after delete
on cookbook_usertype
begin
    update cookbook_user set typeid=1 where typeid is null;
end;
/
```

After deleting the user type from cookbook_usertype talbe, the typeid in cookbook_user will be null. So, I use this trigger to update the null value to 1 which is the basic user type.

Procedures

Insert into user type table: before insert check type id and username.

```
create or replace procedure insert_usertype_proc(
id in cookbook_usertype.typeid%type,
name in cookbook_usertype.typename%type,
rescode out number
) as
cnt number :=0;
begin
select count(*) into cnt from cookbook_usertype where typeid=id or typename=name;
if cnt>0 then
    DBMS_OUTPUT.PUT_LINE('Faild:id/name exist!');
    rescode:=-1;
else
    DBMS_OUTPUT.PUT_LINE('success');
    insert into cookbook_usertype values(id,name);
    rescode:=1;
end if;
end insert_usertype_proc;
/
```

Procedures

Delete user type:

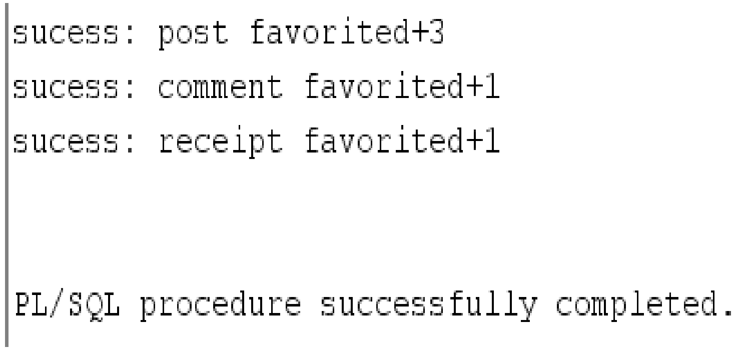
```
create or replace procedure delete_usertype_proc(
        name in cookbook_usertype.typename%type
) as
begin
        delete from cookbook_usertype where typename=name;
end delete_usertype_proc;
/

exec delete_usertype_proc('SuperAdmin');
```

Procedures

```
create or replace procedure insert_favorite_proc
(uid in cookbook_favorite.userid%type, pid in cookbook_favorite.postid%type, cid in cookbook_favorite.commentid %type, rid in cookbook_favorite.receiptid%type
)as cnt number; input_exception exception; pragma exception_init(input_exception,-32767);
begin
   if uid is null then
      DBMS_OUTPUT.PUT_LINE('Failed: UserID is be null.');
      raise input_exception;
   end if;
   if pid is null and cid is null and rid is null then
         DBMS_OUTPUT.PUT_LINE('Failed: nothing favorited.');
         raise input_exception;
   end if;
   if pid is not null then--favorite post
   select count(*) into cnt from cookbook_favorite where userid=userid and postid=pid;
      if cnt>0 then DBMS_OUTPUT.PUT_LINE('sucess: post favorited+'||cnt);
      else insert into cookbook_favorite(userID,postID,commentID,receiptID,desrc) values(uid,pid,null,null,'favorite post');
         DBMS_OUTPUT.PUT_LINE('sucess: new post favorited');
      end if;
   end if;
   if cid is not null then--facorite comment
   select count(*) into cnt from cookbook_favorite where userid=userid and commentid=cid;
      if cnt>0 then DBMS_OUTPUT.PUT_LINE('sucess: comment favorited+'||cnt);
      else insert into cookbook_favorite(userID,postID,commentID,receiptID,desrc) values(uid,null,cid,null,'favorite comment');
         DBMS_OUTPUT.PUT_LINE('sucess: new comment favorited');
      end if;
   end if;
   if rid is not null then --favorite receipt
   select count(*) into cnt from cookbook_favorite where userid=userid and receiptid=rid;
      if cnt>0 then DBMS_OUTPUT.PUT_LINE('sucess: receipt favorited+'||cnt);
      else insert into cookbook_favorite(userID,postID,commentID,receiptID,desrc) values(uid,null,null,rid,'favorite comment');
         DBMS_OUTPUT.PUT_LINE('sucess: new receipt favorited');
      end if;
   end if;
exception
   when input_exception then raise;
end insert_favorite_proc;
/
```

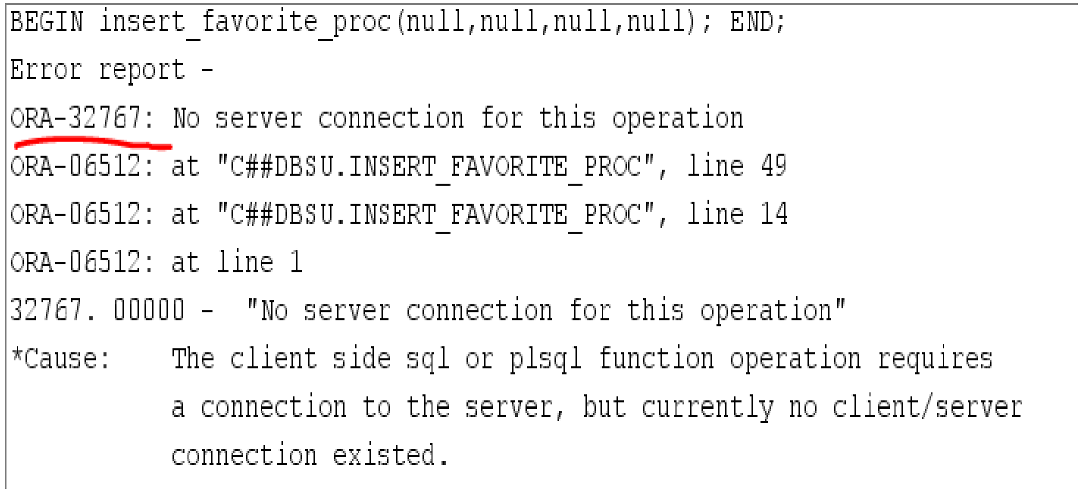
Procedures



```
sucess: post favorited+3
sucess: comment favorited+1
sucess: receipt favorited+1


PL/SQL procedure successfully completed.
```

exec insert_favorite_proc(3,2,2,2);



```
BEGIN insert_favorite_proc(null,null,null,null); END;
Error report -
ORA-32767: No server connection for this operation
ORA-06512: at "C##DBSU.INSERT_FAVORITE_PROC", line 49
ORA-06512: at "C##DBSU.INSERT_FAVORITE_PROC", line 14
ORA-06512: at line 1
32767. 00000 -  "No server connection for this operation"
*Cause:    The client side sql or plsql function operation requires
           a connection to the server, but currently no client/server
           connection existed.
```

exec insert_favorite_proc(null,null,null,null);

04

Reprot

PART FOUR

Report

- The application has a social feature by adding 4 more tables.
- The social feature attracts more people to use this app for making friends or learning cooking.
- The function of sharing a post or adding comment will extend the user online time. When they finished cooking, they are just offline in the past. But now they can share the result of cooking with others and they will gain more sense of success.

Report

Old Cookbook

The old cookbook only have one user type. Everyone can publish a receipt, and everyone can learn a receipt.

New Cookbook

The new cookbook have more user type. This will improve the facility of user management. And different user type have different permission. So, this will improve the quality of the receipt.

Thanks

2020-12-19