**Project #1 (First Part)**

1. **Introduction**

**This database is for a website called “Cookzilla” that focuses on cooking and recipes. The purpose for this website is for users to share their recipes and form groups to communicate and hold events. The main functions are posting recipes, posting reviews and suggestions to reviews, forming groups, holding events, reseerving enents and posting meeting reports.**

1. **Explanation**

**To design the database, the first table to create is the "User" table. The "User" table is for storing the information of the users. This table should have a field of unique user id or name as primary key to represent each user. Here we choose using numeric user id as primary key because integer has better performance than string as primary key. Other than primary key, the table also has a field of "uname" which is unique for each id, and a password field, so as for users to use for login. The table also stores the real name and a profile for each user.**

**Each recipe should be posted by one user, therefore the "Recipe" table has a foreign key "uid" reference to the "uid" in the "User" table. The primary key of the "Recipe" table is "rid".**

**Because one recipe could link to one or more other recipes, we create a new table "Link\_Recipe" to store the links between recipes.**

**Each recipe could have more than one ingredient, therefore create a new table "Ingredient". The primary keys are "rid" and "iname", means each recipe can not have ingredients with same name. Each ingredient should have a quantity, so add a field "iquantity" to the table.**

**Each recipe could have one or more pictures. Therefore, create a table "Recipe\_Picture" to store the pictures. The "rid" is a foreign key reference to "Recipe".**

**The reviews are posted by users and are comments to the recipes. So the table "Review" has a foreign key "uid" reference to "User" and a foreign key "rid" reference to "Recipe". The primary key of the table is "r\_id".**

**The "rate" field of "Review" can only have integer value 1 - 5, therefore create a restriction to the values by adding a table "Rate" as code book and set the field "rrate" in "Review" as foreign key.**

**One review could have more than one suggestions and more than one photo, therefore add two individual tables "Review\_suggestion" and "Review\_Photo", and set "r\_id" as foreign keys in each table reference to table "Review".**

**The "GGroup" table stores the records of groups. "gid" is the primary key of the table.**

**Users could join one or more groups. These records are stored in the "Join\_Group" table. The "uid" is a foreign key reference to the "uid" key in the "User" table. The "gid" is a foreign key reference to the "gid" key in the "GGroup" table.**

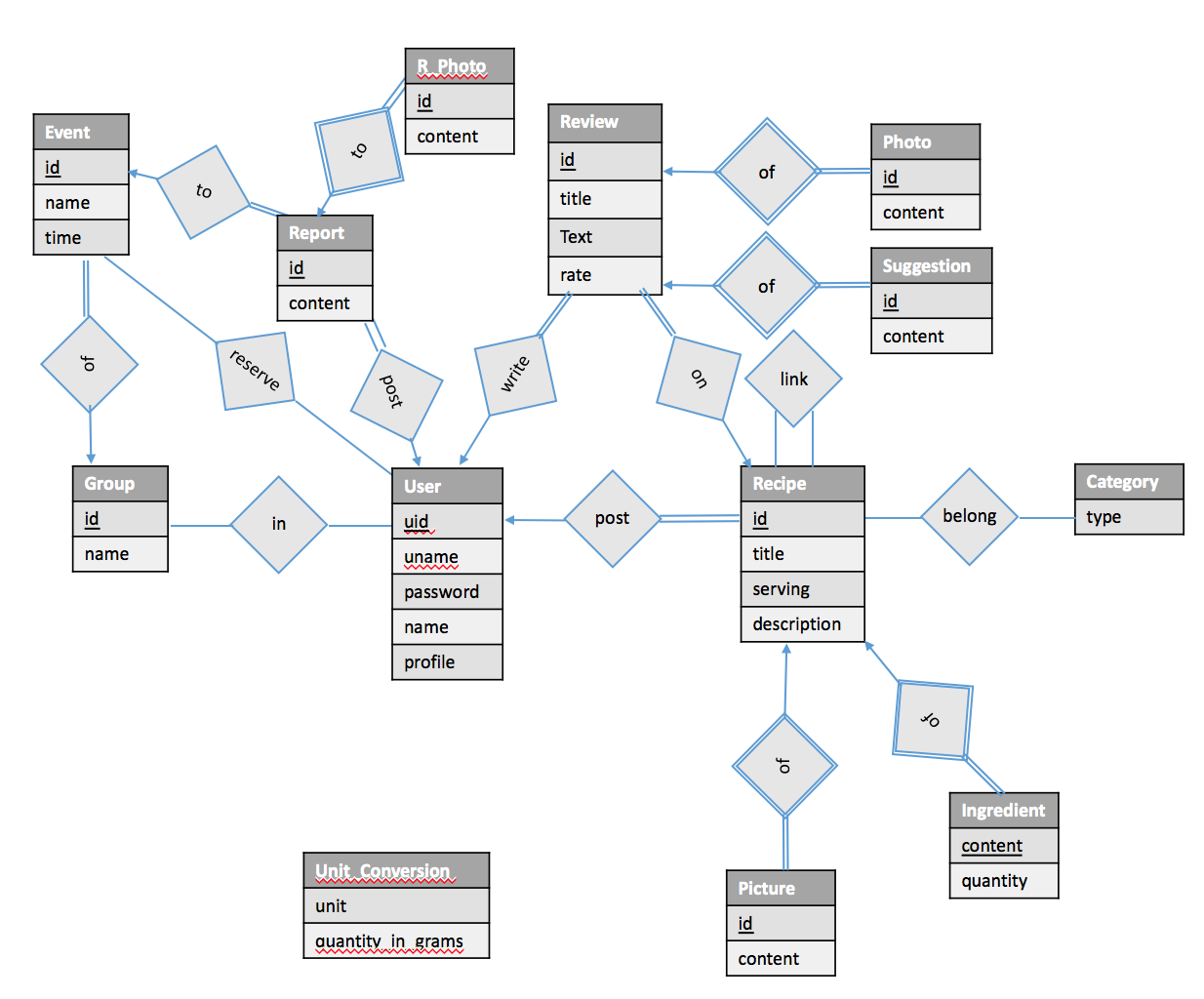
**The "Event" table stores the records of events. Because each event is held by a group, the "Event" table has a foreign key "gid" reference to the "gid" in the "GGroup". It also has fields for the time, location and description of the event.**

**The "Reserve" table stores the records of users' reservations to events.**

**After the events, the user could post reports of the events they attended. So the "Report" table has a foreign key "uid" reference to that in the "User" table and a foreign key "eid" reference to that in the "Event" table.**

**Each report could have one or more photos. These photos are stored in the "Report\_Photo" table.**

1. **Diagram**
2. ER Diagram



1. Relational Schema

User(uid, uname, upassword, name, uprofile)

Recipe(rid, uid, rtitle, rserving, rdescription)

Recipe\_Picture(pid, rid, content)

Recipe\_Tag(rid, tid)  
 Ingredient(rid, iname, iquantity)

Link\_Recipe(rid, rid\_link)

Review(r\_id, rid, uid, rrate, rtext, rtitle)

Review\_Photo(p\_id, r\_id, photo)

Review\_Suggestion(sid, r\_id, content)

Event(eid, gid, etime, elocation, edescription)

GGroup(gid, gname, gnumber, gdescription)

Join\_Group(uid, gid)

Report(id, uid, eid, content)

Report\_Photo(id, rpid, photo)

Rate(rank)

Unit\_Conversion(unit\_name, quantity\_in\_gram)

**Foreign Key References**

Recipe(uid) User(uid)

Recipe\_Picture(rid) Recipe(rid)

Recipe\_Tag(rid) Recipe(rid)

Ingredient(rid) Recipe(rid)

Link\_Recipe(rid, rid\_link) Recipe(rid)

Review(rid, uid) Recipe(rid) User(uid)

Review\_Photo(r\_id) Review(r\_id)

Review\_Suggestion(r\_id) Review(r\_id)

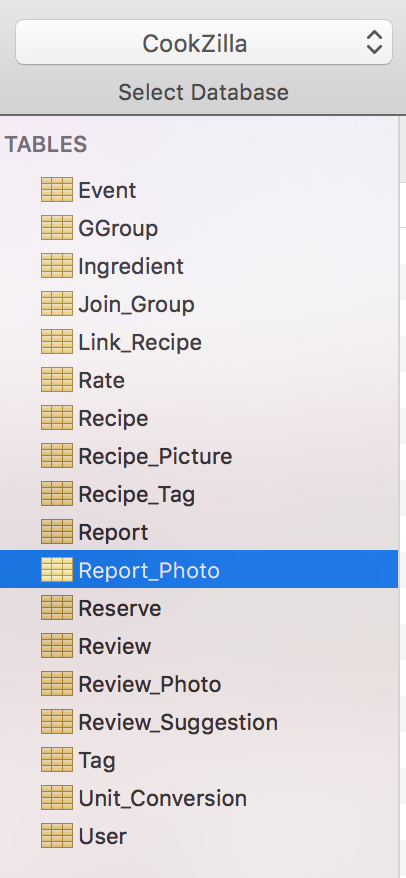
Event(gid) GGroup(gid)

Join\_Group(uid,gid) GGroup(gid), User(uid)

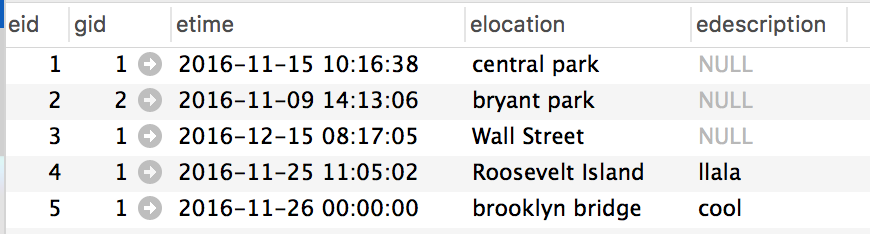
Report(uid, eid) Event(eid), User(uid)

Report\_Photo(rpid) Report(id)

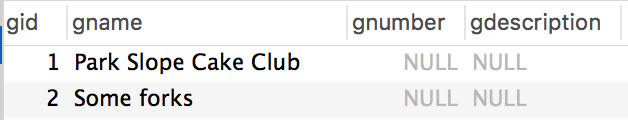
1. **Test Data**
2. **Create Tables and Constraints**



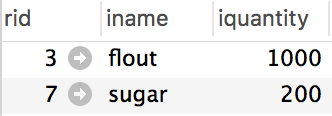
**Event**



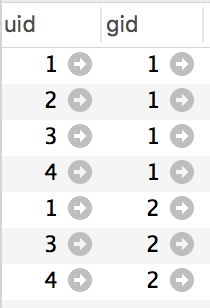
**GGroup**



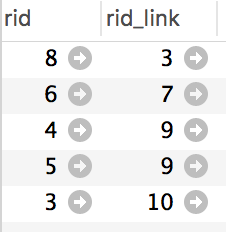
**Ingredient**

****

**Join\_Group**

****

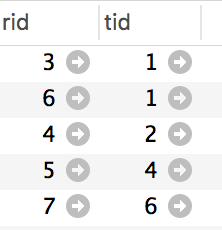
**Link\_Recipe**

****

**Rate**

****

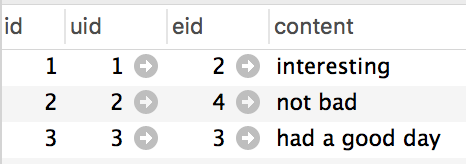
**Recipe\_Tag**

****

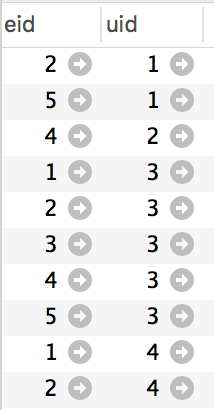
**Recipe**



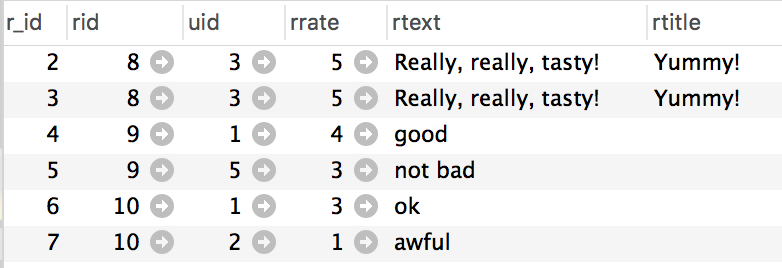
**Report**



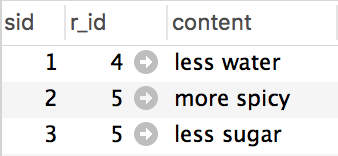
**Reserve**



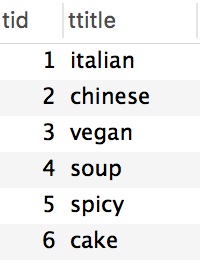
**Review**



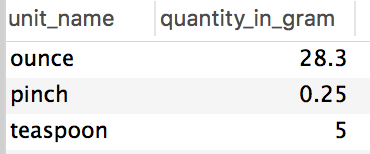
**Review\_Suggestion**



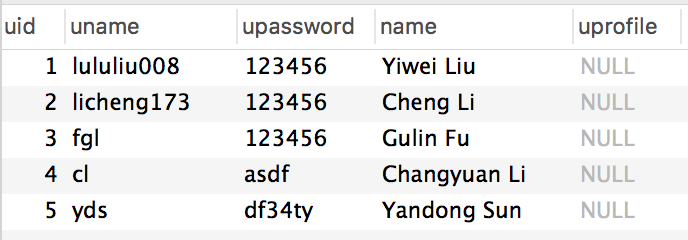
**Tag**



**Unit\_Conversion**



**User**



1. **SQL Queries**
2. *Create a record for a new user account, with a name, a login name, and a password.*

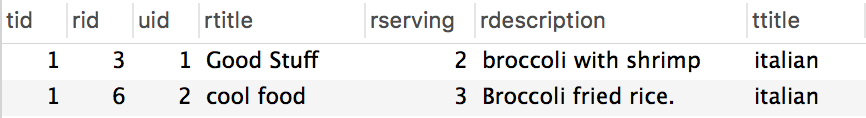
**Insert into** User (uname, upassword, name) **values** ('lululiu008', 123456, 'Yiwei Liu');

1. *List all recipes with tag “italian” that contain the keyword ``broccoli’’.*

**Select** \*

**from** Recipe R **natural** **join** Recipe\_Tag RT **natural** **join** Tag T

**where** T.ttitle = 'italian' **and** R.rdescription **like** '%broccoli%';



1. *List all members of the group “Park Slope Cake Club” that have given a positive RSVP to more that three events of the group.*

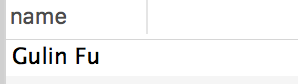
**Select** name

**From** user **natural** **join** reserve **natural** **join** event

**group** **by** uid, gid

**having** gid = (**select** gid **from** ggroup

**where** gname = 'Park Slope Cake Club') **and** **count**(event.eid) > 3

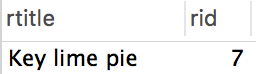


1. *List all recipes with tag “cake” that contain more than 50 grams of sugar per serving.*

**Select** r.rtitle, r.rid

**From** recipe r **natural** **join** recipe\_tag rt **natural** **join** ingredient i **natural** **join** tag t

**where** t.ttitle = 'cake' **and** i.iname = 'sugar' **and** i.iquantity / r.rserving > 50;



1. *Add a review with title “Yummy!”, text “Really, really, tasty!”, and a rating of 5 stars to the recipe for “Grandma’s Fettuccini Alfredo”*

**Insert** **into** review (rid, uid, rrate, rtext, rvtitle)

**values** (8, 3, 5, 'Really, really, tasty!', 'Yummy!');

1. *List all recipes containing the word “tuna”, sorted from highest to lowest average rating.*

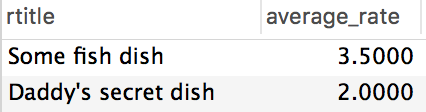
**select** r.rtitle, (**sum**(rv.rrate) / **count**(\*)) **as** average\_rate

**from** recipe r, review rv

**where** r.rid = rv.rid **and** r.rdescription **like** '%tuna%'

**group** **by** r.rtitle

**order** **by** average\_rate desc;



1. *List all recipes that are related to a recipe that contains the word “tuna”.*

**select** r1.rtitle, r1.rid

**from** recipe r1, link\_recipe l, recipe r2

**where** r1.rid = l.rid **and** l.rid\_link = r2.rid **and** r2.rdescription **like** '%tuna%';

