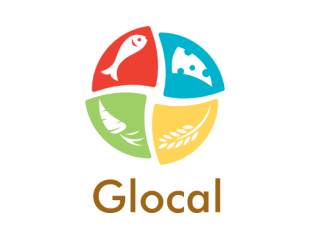
Database Design

Glocal - Business Rating and Review System



Salman Azam Khan <salmanakhan@ufl.edu>

Chang Xu <theoneneo@ufl.edu>

Unmey Mahaddalkar <umahaddalkar@ufl.edu>

Shubham Agiwal <shubhamagiwal92@ufl.edu>

**History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Author** | **Description** |
| 0.1 | 2/23/2017 | Neo | Initial version. |
| 0.2 | 3/7/2017 | Neo | Updated ERD, Data Model, and Table description according to database changes. |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

[**1. Introduction**](#_iylqsh2iabvy) **3**

[**2. Data Model**](#_wk0lhl9tesy5) **4**

[2.1. E-R Diagram](#_ad3hn4rpjqof) 4

[2.2. Relational Model](#_sjyww4mwj8vd) 6

[2.3. Data Flow Diagram](#_6vbe2ua95q0l) 8

[**3. Tables**](#_dq2kig7tl1yc) **10**

[3.1. Configuration Tables](#_lnp403ud3okb) 10

[3.1.1. CONFIG\_DROPDOWN](#_he438kj54u00) 10

[3.1.2. CONFIG\_USER\_TYPE](#_6dhacobd33qf) 11

[3.1.3. CONFIG\_TAG](#_2mljbcw8t2p0) 13

[3.1.4. CONFIG\_WEEKDAY](#_e74cet8dagoa) 13

[3.1.5. CONFIG\_BUSINESS\_ATTR](#_e8rwhts5dzii) 14

[3.1.6. CONFIG\_BUSINESS\_ATTR\_DROP](#_6yalst2qwe59) 15

[3.2. Data Tables](#_gsjyd057tswc) 16

[3.2.1. USER\_INFO](#_ie0cu9oa918y) 16

[3.2.2. USER\_CREDENTIAL](#_mk8d03266ccr) 17

[3.2.3. USER\_ACCESS\_TOKEN](#_a0kkvsijws74) 17

[3.2.4. BUSINESS](#_2g5wxe15hjs) 18

[3.2.5. BUSINESS\_TAG](#_1rx54ifdrcvv) 19

[3.2.6. BUSINESS\_HOUR](#_luwelryk6n6e) 20

[3.2.7. BUSINESS\_OWNERSHIP](#_emm2t967lqyh) 21

[3.2.8. REVIEW](#_2t57swdf8oyh) 21

[3.2.9. VOTE](#_h838owqlbu05) 22

[3.2.10. IMAGE](#_pbgp19wm4ayu) 23

[3.2.11. USER\_IMAGE](#_y6gib1eb10ks) 24

[3.2.12. BUSINESS\_IMAGE](#_gj5dw7emgre0) 24

[3.2.13. BUSINESS\_ATTR\_TEXT](#_wxyphwlkm4wg) 25

[3.2.14. BUSINESS\_ATTR\_NUMBER](#_b1naj7s195kg) 26

[3.2.15. BUSINESS\_ATTR\_DATE](#_44xpubgtrynb) 27

[**4. Other Database Objects**](#_pmz5ykqp6ag3) **28**

[4.1. Sequences](#_6k102tb6n46f) 28

[4.2. Triggers](#_zi0ulpkqld5g) 28

# **1. Introduction**

For Glocal, Oracle 11g database is used to store all information of the system except image binary data. For satisfactory accessibility and reliability, image binary data is stored externally on Amazon Web Services. Only the Image URLs are stored in the Oracle database.

Section 2 Data Model provides overview of the database structure by the form of diagrams. An enhanced E-R Diagram is presented to illustrate the database structure from the conceptual level. Then, a Relational Model describes physical level details of the database. And the third one, Data Flow Diagram, depicts the information input and output through the system from the external level data flow perspective.

Section 3 Tables provides details of all tables in the database via descriptions, column lists, and SQL statements used to create the tables. In Glocal database, tables can be classified into two primary categories: Configuration Tables and Data Tables. Configuration Tables store static reference data for Data Tables. Data in Configuration Tables is part of the system implementation and is maintained by the development team. Data Tables store data generated by Users during the running of the system.

Section 4 Other Database Objects provides description about required Sequences, Triggers, Views, and Procedures. For now, Sequences and Triggers have been defined. Views and Procedures will be added during the development of the system.

# **2. Data Model**

An enhanced E-R Diagram is provided in this Section. It illustrates the database structure from the conceptual level. Following E-R Diagram, the Relational Model describes physical level details of the database. And the third one, Data Flow Diagram, depicts the information input and output through the system from the external level data flow perspective.

## **2.1. E-R Diagram**

Entity-Relationship Diagram shows relationships of entity sets of the system. It provides logical structure of the database.

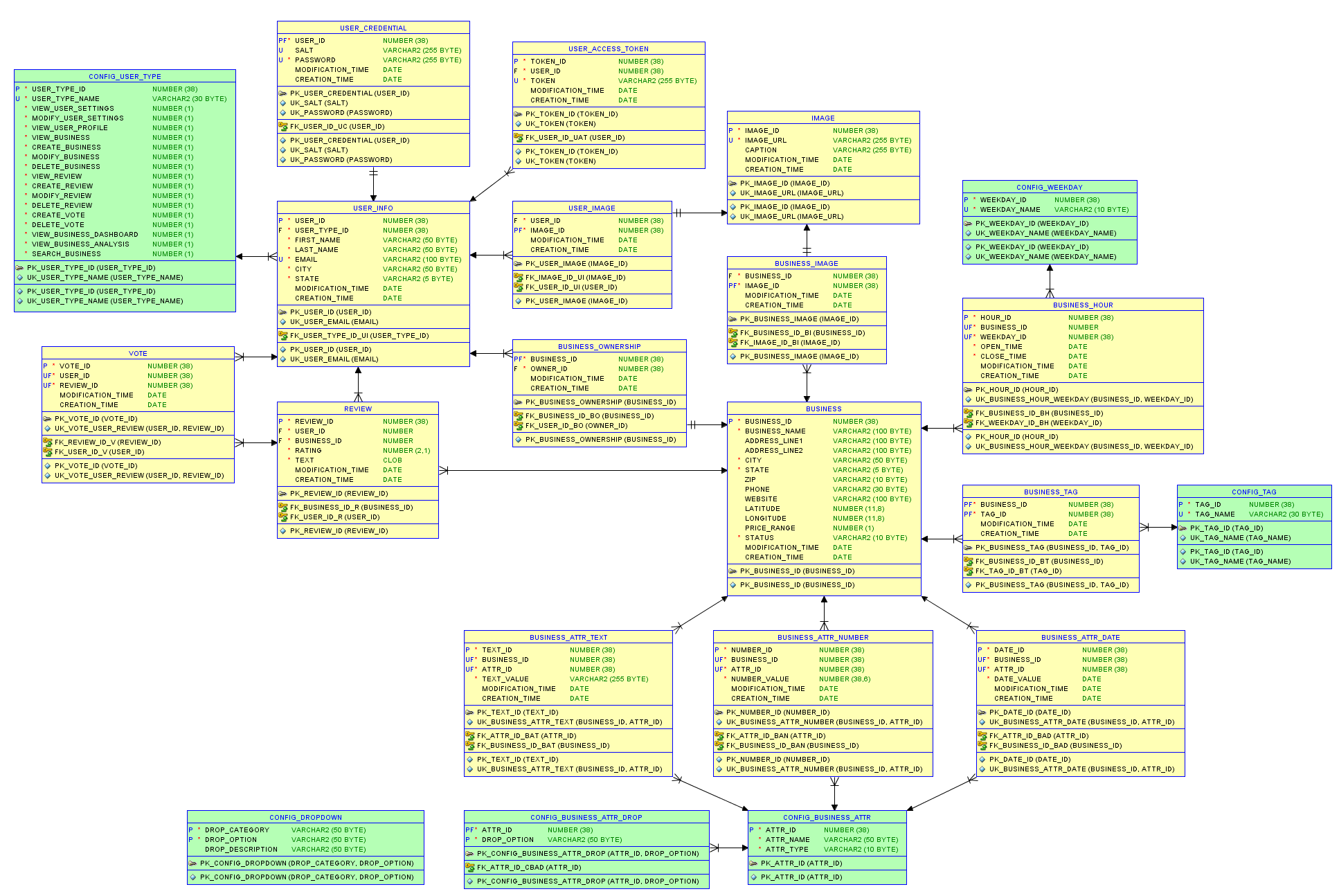
All entity attributes except the modification and creation timestamps are shown in the E-R Diagram. The logical relationships are primarily about User and Business which are exactly the two cynosures of the system.

ER Diagram.png

## **2.2. Relational Model**

Relational Model provides overview of all tables with table details and their connections. Table details include column names, data types, nullable settings, primary keys, unique keys, and foreign keys. Table connections are shown by relationships among them.

There are two categories of Tables in the Relational Model: Configuration Tables and Data Tables. Configuration Tables, marked with green background, store static reference data for the system and for other Tables. Data Tables, marked with yellow background, store data generated by actual Users during the running of the system. More information about the two categories can be found in Section 3 Tables.



## **2.3. Data Flow Diagram**

Data Flow Diagram shows how information is processed by the system in terms of inputs and outputs.

There are 3 major external entities for the system: Customer, Business Owner, and Anonymous User. For current release, all functions are provided for only registered Users. Anonymous User must enter the system first via account authentication process which includes Sign Up, Log In and Forgot Password.

Both Customer and Business Owner can edit profile of their own and view profile of any other User. Only Business Owner is allowed to add new Business or modify owned Business; Customer is not allowed to do edit Business. Only Customer is allowed to write Review and vote likeness of Review; Business Owner is not allowed to do that.

Analysis of Business is available to only Business Owner. All other major functions are available to all Users.

Data Flow Diagram.png

# **3. Tables**

All Tables can be separated into two categories: Configuration Tables and Data Tables.

Configuration Tables store static reference data for Data Tables. Data in Configuration Tables is part of the system implementation and hence is maintained by the development team. (Some configuration data may be maintained by system admin in the later release when an admin interface is implemented.)

Data Tables store data generated by actual Users during the running of the system. Including all data related to User, Business, Review and etc. In addition to the entity attributes shown in the ER Diagram, two timestamp attributes (Modification Time and Creation Time) are added to every Data Tables for the purpose of tracking changes.

## **3.1. Configuration Tables**

### **3.1.1. CONFIG\_DROPDOWN**

Table CONFIG\_DROPDOWN stores configuration of system level dropdown options. Data in this table is used by front-end to show various dropdown options on UI pages.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| DROP\_CATEGORY | VARCHAR2(50 BYTE) | Primary Key (DROP\_CATEGORY, DROP\_OPTION) | Not Null |  |
| DROP\_OPTION | VARCHAR2(50 BYTE) | Primary Key (DROP\_CATEGORY, DROP\_OPTION) | Not Null |  |
| DROP\_DESCRIPTION | VARCHAR2(50 BYTE) |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_dropdown(  drop\_category varchar2(50 byte) not null,  drop\_option varchar2(50 byte) not null,  drop\_description varchar2(50 byte),  constraint pk\_config\_dropdown primary key (drop\_category, drop\_option)  ); |

### **3.1.2. CONFIG\_USER\_TYPE**

Table CONFIG\_USER\_TYPE stores configuration of User Type. It contains permission settings for each type. These permission settings are used in backend APIs to validate if user has appropriate permission to perform certain operation. For now, only two types are supported: Customer and Business Owner.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| USER\_TYPE\_ID | NUMBER(38,0) | Primary Key | Not Null | 1  2 |
| USER\_TYPE\_NAME | VARCHAR2(30 BYTE) | Unique | Not Null | Customer  Business Owner |
| VIEW\_USER\_SETTINGS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view user's own settings |
| MODIFY\_USER\_SETTINGS | NUMBER(1,0) |  | Not Null0 | 0 - not allowed  1 - modify user's own settings |
| VIEW\_USER\_PROFILE | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view any profile |
| VIEW\_BUSINESS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view any business |
| CREATE\_BUSINESS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - create business |
| MODIFY\_BUSINESS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - modify user's own business  2 - modify any business |
| DELETE\_BUSINESS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - delete user's own business  2 - delete any business |
| VIEW\_REVIEW | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view any user's review |
| CREATE\_REVIEW | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - create review |
| MODIFY\_REVIEW | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - modify user's own review  2 - modify any review |
| DELETE\_REVIEW | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - delete user's own review  2 - delete any review |
| CREATE\_VOTE | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - create vote |
| DELETE\_VOTE | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - delete user's own vote  2 - delete any vote |
| VIEW\_BUSINESS\_DASHBOARD | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view business dashboard |
| VIEW\_BUSINESS\_ANALYSIS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - view analysis of user's own business  2 - view analysis of any business |
| SEARCH\_BUSINESS | NUMBER(1,0) |  | Not Null | 0 - not allowed  1 - search business |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_user\_type(  user\_type\_id number(38,0) not null,  user\_type\_name varchar2(30 byte) not null,  view\_user\_settings number(1,0) not null,  modify\_user\_settings number(1,0) not null,  view\_user\_profile number(1,0) not null,  view\_business number(1,0) not null,  create\_business number(1,0) not null,  modify\_business number(1,0) not null,  delete\_business number(1,0) not null,  view\_review number(1,0) not null,  create\_review number(1,0) not null,  modify\_review number(1,0) not null,  delete\_review number(1,0) not null,  create\_vote number(1,0) not null,  delete\_vote number(1,0) not null,  view\_business\_dashboard number(1,0) not null,  view\_business\_analysis number(1,0) not null,  search\_business number(1,0) not null,  constraint pk\_user\_type\_id primary key (user\_type\_id),  constraint uk\_user\_type\_name unique (user\_type\_name)  ); |

### **3.1.3. CONFIG\_TAG**

Table CONFIG\_TAG stores definitions of Business Tags. For now, a flat Tag system is used. The detailed Tag definitions will be completed during the system development phase.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| TAG\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| TAG\_NAME | VARCHAR2(30 BYTE) | Unique | Not Null | Restaurant  Active Life  ... |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_tag(  tag\_id number(38,0) not null,  tag\_name varchar2(30 byte) not null,  constraint pk\_tag\_id primary key (tag\_id),  constraint uk\_tag\_name unique (tag\_name)  ); |

### **3.1.4. CONFIG\_WEEKDAY**

Table CONFIG\_WEEKDAY stores definitions of all 7 weekdays. Table BUSINESS\_HOUR references these weekdays by WEEKDAY\_ID.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| WEEKDAY\_ID | NUMBER(38,0) | Primary Key | Not Null | 1 to 7 |
| WEEKDAY\_NAME | VARCHAR2(10 BYTE) | Unique | Not Null | Monday to Sunday |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_weekday(  weekday\_id number(38,0) not null,  weekday\_name varchar2(10 byte) not null,  constraint pk\_weekday\_id primary key (weekday\_id),  constraint uk\_weekday\_name unique (weekday\_name)  ); |

|  |
| --- |
| **SQL Used to Initialize Table** |
| insert into config\_weekday values (1, 'Monday');  insert into config\_weekday values (2, 'Tuesday');  insert into config\_weekday values (3, 'Wednesday');  insert into config\_weekday values (4, 'Thursday');  insert into config\_weekday values (5, 'Friday');  insert into config\_weekday values (6, 'Saturday');  insert into config\_weekday values (7, 'Sunday');  commit; |

### 

### **3.1.5. CONFIG\_BUSINESS\_ATTR**

Table CONFIG\_BUSINESS\_ATTR stores definitions of configurable attributes for Business. Actual attribute values of Businesses are stored in 3 Business Attribute Tables according to attribute type. There are 4 types of attributes are supported: Text, Dropdown, Number and Date. If an attribute is of type Dropdown, then its possible options are stored in Table CONFIG\_BUSINESS\_ATTR\_DROP.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| ATTR\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| ATTR\_NAME | VARCHAR2(50 BYTE) |  | Not Null | Accepts Credit Cards  Takes Reservation  ... |
| ATTR\_TYPE | VARCHAR2(10 BYTE) |  | Not Null | Text  Dropdown  Number  Date |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_business\_attr(  attr\_id number(38,0) not null,  attr\_name varchar2(50 byte) not null,  attr\_type varchar2(10 byte) not null,  constraint pk\_attr\_id primary key (attr\_id)  ); |

### **3.1.6. CONFIG\_BUSINESS\_ATTR\_DROP**

Table CONFIG\_BUSINESS\_ATTR\_DROP stores dropdown options of Business configurable attributes which have ATTR\_TYPE = ‘Dropdown’. These dropdown options are used by front-end to show on the Business Settings page.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| ATTR\_ID | NUMBER(38,0) | Primary Key (ATTR\_ID, DROP\_OPTION)  CONFIG\_BUSINESS\_ATTR (ATTR\_ID) | Not Null | AMBIENCE |
| DROP\_OPTION | VARCHAR2(50 BYTE) | Primary Key (ATTR\_ID, DROP\_OPTION) | Not Null | (options of AMBIENCE) Upscale Casual Intimate Trendy Classy Romantic ... |

|  |
| --- |
| **SQL Used to Create Table** |
| create table config\_business\_attr\_drop(  attr\_id number(38,0) not null,  drop\_option varchar2(50 byte) not null,  constraint pk\_config\_business\_attr\_drop primary key (attr\_id, drop\_option),  constraint fk\_attr\_id\_cbad foreign key (attr\_id) references config\_business\_attr(attr\_id)  ); |

## **3.2. Data Tables**

### **3.2.1. USER\_INFO**

Table USER\_INFO stores basic information about Users, including both Customers and Business Owners. User Type is distinguished by USER\_TYPE\_ID which references Table CONFIG\_USER\_TYPE.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| USER\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| USER\_TYPE\_ID | NUMBER(38,0) | CONFIG\_USER\_TYPE (USER\_TYPE\_ID) | Not Null |  |
| FIRST\_NAME | VARCHAR2(50 BYTE) |  | Not Null |  |
| LAST\_NAME | VARCHAR2(50 BYTE) |  | Not Null |  |
| EMAIL | VARCHAR2(100 BYTE) | Unique | Not Null |  |
| CITY | VARCHAR2(50 BYTE) |  | Not Null |  |
| STATE | VARCHAR2(5 BYTE) |  | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table user\_info(  user\_id number(38,0) not null,  user\_type\_id number(38,0) not null,  first\_name varchar2(50 byte) not null,  last\_name varchar2(50 byte) not null,  email varchar2(100 byte) not null,  city varchar2(50 byte) not null,  state varchar2(5 byte) not null,  modification\_time date,  creation\_time date,  constraint pk\_user\_id primary key (user\_id),  constraint uk\_user\_email unique (email),  constraint fk\_user\_type\_id\_ui foreign key (user\_type\_id) references config\_user\_type(user\_type\_id)  ); |

### **3.2.2. USER\_CREDENTIAL**

Table USER\_CREDENTIAL stores credentials of Users. For security consideration, plain passwords are not stored. Instead, the hashed results of passwords along with randomly generated salts are saved. And salt must unique for each User. For now, Salt is used due to backend is using bcrypt function.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| USER\_ID | NUMBER(38,0) | Primary Key  USER\_INFO (USER\_ID) | Not Null |  |
| SALT | VARCHAR2(255 BYTE) | Unique |  |  |
| PASSWORD | VARCHAR2(255 BYTE) | Unique | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table user\_credential(  user\_id number(38,0) not null,  salt varchar2(255 byte),  password varchar2(255 byte) not null,  modification\_time date,  creation\_time date,  constraint pk\_user\_credential primary key (user\_id),  constraint uk\_salt unique (salt),  constraint uk\_password unique (password),  constraint fk\_user\_id\_uc foreign key (user\_id) references user\_info(user\_id)  ); |

### **3.2.3. USER\_ACCESS\_TOKEN**

Table USER\_ACCESS\_TOKEN stores login tokens of Users. Tokens are used for authentication process. One User is allowed to have multiple valid tokens at the same time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| TOKEN\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| USER\_ID | NUMBER(38,0) | USER\_INFO (USER\_ID) | Not Null |  |
| TOKEN | VARCHAR2(255 BYTE) | Unique | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table user\_access\_token(  token\_id number(38,0) not null,  user\_id number(38,0) not null,  token varchar2(255 byte) not null,  modification\_time date,  creation\_time date,  constraint pk\_token\_id primary key (token\_id),  constraint uk\_token unique (token),  constraint fk\_user\_id\_uat foreign key (user\_id) references user\_info(user\_id)  ); |

### **3.2.4. BUSINESS**

Table BUSINESS stores basic information about Businesses. BUSINESS\_NAME is allowed to be duplicated due to chain Businesses use the same name among all stores.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| BUSINESS\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| BUSINESS\_NAME | VARCHAR2(100 BYTE) |  | Not Null |  |
| ADDRESS\_LINE1 | VARCHAR2(100 BYTE) |  |  |  |
| ADDRESS\_LINE2 | VARCHAR2(100 BYTE) |  |  |  |
| CITY | VARCHAR2(50 BYTE) |  | Not Null |  |
| STATE | VARCHAR2(5 BYTE) |  | Not Null |  |
| ZIP | VARCHAR2(10 BYTE) |  |  |  |
| PHONE | VARCHAR2(30 BYTE) |  |  |  |
| WEBSITE | VARCHAR2(100 BYTE) |  |  |  |
| LATITUDE | NUMBER(11,8) |  |  |  |
| LONGITUDE | NUMBER(11,8) |  |  |  |
| PRICE\_RANGE | NUMBER(1,0) |  |  | 1  2  3  4 |
| STATUS | VARCHAR2(10 BYTE) |  | Not Null | Open  Closed  Moved  Banned |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business( business\_id number(38,0) not null, business\_name varchar2(100 byte) not null, address\_line1 varchar2(100 byte), address\_line2 varchar2(100 byte), city varchar2(50 byte) not null, state varchar2(5 byte) not null, zip varchar2(10 byte), phone varchar2(30 byte), website varchar2(100 byte), latitude number(11,8), longitude number(11,8), price\_range number(1,0), status varchar2(10 byte) not null, modification\_time date, creation\_time date, constraint pk\_business\_id primary key (business\_id) ); |

### **3.2.5. BUSINESS\_TAG**

Table BUSINESS\_TAG stores relationships between Businesses and predefined Tags. It references both Table BUSINESS and Table CONFIG\_TAG.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| BUSINESS\_ID | NUMBER(38,0) | Primary Key (BUSINESS\_ID, TAG\_ID)  BUSINESS (BUSINESS\_ID) | Not Null |  |
| TAG\_ID | NUMBER(38,0) | Primary Key (BUSINESS\_ID, TAG\_ID)  CONFIG\_TAG (TAG\_ID) | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_tag(  business\_id number(38,0) not null,  tag\_id number(38,0) not null,  modification\_time date,  creation\_time date,  constraint pk\_business\_tag primary key (business\_id, tag\_id),  constraint fk\_business\_id\_bt foreign key (business\_id) references business(business\_id),  constraint fk\_tag\_id\_bt foreign key (tag\_id) references config\_tag(tag\_id)  ); |

### **3.2.6. BUSINESS\_HOUR**

Table BUSINESS\_HOUR stores hours information of Business. Unique constraint that combines BUSINESS\_ID and WEEKDAY\_ID guarantees one Business can have at most 7 Business Hour records.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| HOUR\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| BUSINESS\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, WEEKDAY\_ID)  BUSINESS (BUSINESS\_ID) | Not Null |  |
| WEEKDAY\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, WEEKDAY\_ID)  CONFIG\_WEEKDAY (WEEKDAY\_ID) | Not Null |  |
| OPEN\_TIME | DATE |  | Not Null |  |
| CLOSE\_TIME | DATE |  | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_hour(  hour\_id number(38,0) not null,  business\_id number(38,0) not null,  weekday\_id number(38,0) not null,  open\_time date not null,  close\_time date not null,  modification\_time date,  creation\_time date,  constraint pk\_hour\_id primary key (hour\_id),  constraint uk\_business\_hour\_weekday unique (business\_id, weekday\_id),  constraint fk\_business\_id\_bh foreign key (business\_id) references business(business\_id),  constraint fk\_weekday\_id\_bh foreign key (weekday\_id) references config\_weekday(weekday\_id)  ); |

### **3.2.7. BUSINESS\_OWNERSHIP**

Table BUSINESS\_OWNERSHIP stores relationships between Businesses and Business Owners. One Business Owner is allowed to own multiple Businesses.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| BUSINESS\_ID | NUMBER(38,0) | Primary Key  BUSINESS (BUSINESS\_ID) | Not Null |  |
| OWNER\_ID | NUMBER(38,0) | USER\_INFO (USER\_ID) | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_ownership(  business\_id number(38,0) not null,  owner\_id number(38,0) not null,  modification\_time date,  creation\_time date,  constraint pk\_business\_ownership primary key (business\_id),  constraint fk\_business\_id\_bo foreign key (business\_id) references business(business\_id),  constraint fk\_user\_id\_bo foreign key (owner\_id) references user\_info(user\_id)  ); |

### **3.2.8. REVIEW**

Table Review stores ratings and text comments given by Customers to Businesses. One User is allowed to give multiple Reviews to the same Business.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| REVIEW\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| USER\_ID | NUMBER(38,0) | USER\_INFO (USER\_ID) | Not Null |  |
| BUSINESS\_ID | NUMBER(38,0) | BUSINESS (BUSINESS\_ID) | Not Null |  |
| RATING | NUMBER(2,1) | Check in (1, 2, 3, 4, 5) | Not Null | 1  2  3  4  5 |
| TEXT | CLOB |  | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table review(  review\_id number(38,0) not null,  user\_id number(38,0) not null,  business\_id number(38,0) not null,  rating number(2,1) not null,  text clob not null,  modification\_time date,  creation\_time date,  constraint pk\_review\_id primary key (review\_id),  constraint fk\_user\_id\_r foreign key (user\_id) references user\_info(user\_id),  constraint fk\_business\_id\_r foreign key (business\_id) references business(business\_id)  ); |

### **3.2.9. VOTE**

Table VOTE stores likeness of Reviews from Users. One User is allowed to like a certain Review at most one time.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| VOTE\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| USER\_ID | NUMBER(38,0) | Unique (USER\_ID, REVIEW\_ID)  USER\_INFO (USER\_ID) | Not Null |  |
| REVIEW\_ID | NUMBER(38,0) | Unique (USER\_ID, REVIEW\_ID)  REVIEW (REVIEW\_ID) | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table vote(  vote\_id number(38,0) not null,  user\_id number(38,0) not null,  review\_id number(38,0) not null,  modification\_time date,  creation\_time date,  constraint pk\_vote\_id primary key (vote\_id),  constraint uk\_vote\_user\_review unique (user\_id, review\_id),  constraint fk\_user\_id\_v foreign key (user\_id) references user\_info(user\_id),  constraint fk\_review\_id\_v foreign key (review\_id) references review(review\_id)  ); |

### **3.2.10. IMAGE**

Table IMAGE stores URLs and text description of uploaded Images, including both User Images and Business Images. The actual data of Images is stored externally on Amazon Web Services.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| IMAGE\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| IMAGE\_URL | VARCHAR2(255 BYTE) | Unique | Not Null |  |
| CAPTION | VARCHAR2(255 BYTE) |  |  |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table image(  image\_id number(38,0) not null,  image\_url varchar2(255 byte) not null,  caption varchar2(255 byte),  modification\_time date,  creation\_time date,  constraint pk\_image\_id primary key (image\_id),  constraint uk\_image\_url unique (image\_url)  ); |

### **3.2.11. USER\_IMAGE**

Table USER\_IMAGE stores relationships between Users and uploaded Images. One User is allowed to have multiple Images.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| USER\_ID | NUMBER(38,0) | USER\_INFO (USER\_ID) | Not Null |  |
| IMAGE\_ID | NUMBER(38,0) | Primary Key  IMAGE (IMAGE\_ID) | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table user\_image(  user\_id number(38,0) not null,  image\_id number(38,0) not null,  modification\_time date,  creation\_time date,  constraint pk\_user\_image primary key (image\_id),  constraint fk\_user\_id\_ui foreign key (user\_id) references user\_info(user\_id),  constraint fk\_image\_id\_ui foreign key (image\_id) references image(image\_id)  ); |

### **3.2.12. BUSINESS\_IMAGE**

Table BUSINESS\_IMAGE stores relationships between Businesses and uploaded Images. One Business is allowed to have multiple Images.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| BUSINESS\_ID | NUMBER(38,0) | BUSINESS (BUSINESS\_ID) | Not Null |  |
| IMAGE\_ID | NUMBER(38,0) | Primary Key  IMAGE (IMAGE\_ID) | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_image(  business\_id number(38,0) not null,  image\_id number(38,0) not null,  modification\_time date,  creation\_time date,  constraint pk\_business\_image primary key (image\_id),  constraint fk\_business\_id\_bi foreign key (business\_id) references business(business\_id),  constraint fk\_image\_id\_bi foreign key (image\_id) references image(image\_id)  ); |

### **3.2.13. BUSINESS\_ATTR\_TEXT**

Table BUSINESS\_ATTR\_TEXT stores values of configurable text attributes of Business. It references Table BUSINESS\_ATTR to determine the attribute. If a Business has no certain text attribute, there should be no record referencing the Business and the attribute.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| TEXT\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| BUSINESS\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  BUSINESS (BUSINESS\_ID) | Not Null |  |
| ATTR\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  CONFIG\_BUSINESS\_ATTR (ATTR\_ID) | Not Null |  |
| TEXT\_VALUE | VARCHAR2(255 BYTE) |  | Not Null | Casual |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_attr\_text(  text\_id number(38,0) not null,  business\_id number(38,0) not null,  attr\_id number(38,0) not null,  text\_value varchar2(255 byte) not null,  modification\_time date,  creation\_time date,  constraint pk\_text\_id primary key (text\_id),  constraint uk\_business\_attr\_text unique (business\_id, attr\_id),  constraint fk\_business\_id\_bat foreign key (business\_id) references business(business\_id),  constraint fk\_attr\_id\_bat foreign key (attr\_id) references config\_business\_attr(attr\_id)  ); |

### **3.2.14. BUSINESS\_ATTR\_NUMBER**

Table BUSINESS\_ATTR\_NUMBER stores values of configurable number attributes of Business. It references Table BUSINESS\_ATTR to determine the attribute. If a Business has no certain number attribute, there should be no record referencing the Business and the attribute.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| NUMBER\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| BUSINESS\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  BUSINESS (BUSINESS\_ID) | Not Null |  |
| ATTR\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  CONFIG\_BUSINESS\_ATTR (ATTR\_ID) | Not Null | CAPACITY |
| NUMBER\_VALUE | NUMBER(38,6) |  | Not Null | 80 |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_attr\_number(  number\_id number(38,0) not null,  business\_id number(38,0) not null,  attr\_id number(38,0) not null,  number\_value number(38,6) not null,  modification\_time date,  creation\_time date,  constraint pk\_number\_id primary key (number\_id),  constraint uk\_business\_attr\_number unique (business\_id, attr\_id),  constraint fk\_business\_id\_ban foreign key (business\_id) references business(business\_id),  constraint fk\_attr\_id\_ban foreign key (attr\_id) references config\_business\_attr(attr\_id)  ); |

### **3.2.15. BUSINESS\_ATTR\_DATE**

Table BUSINESS\_ATTR\_DATE stores values of configurable date attributes of Business. It references Table BUSINESS\_ATTR to determine the attribute. If a Business has no certain date attribute, there should be no record referencing the Business and the attribute.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column** | **Data Type** | **Constraint** | **Nullable** | **Sample Value** |
| DATE\_ID | NUMBER(38,0) | Primary Key | Not Null |  |
| BUSINESS\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  BUSINESS (BUSINESS\_ID) | Not Null |  |
| ATTR\_ID | NUMBER(38,0) | Unique (BUSINESS\_ID, ATTR\_ID)  CONFIG\_BUSINESS\_ATTR (ATTR\_ID) | Not Null |  |
| DATE\_VALUE | DATE |  | Not Null |  |
| MODIFICATION\_TIME | DATE |  |  |  |
| CREATION\_TIME | DATE |  |  |  |

|  |
| --- |
| **SQL Used to Create Table** |
| create table business\_attr\_date(  date\_id number(38,0) not null,  business\_id number(38,0) not null,  attr\_id number(38,0) not null,  date\_value date not null,  modification\_time date,  creation\_time date,  constraint pk\_date\_id primary key (date\_id),  constraint uk\_business\_attr\_date unique (business\_id, attr\_id),  constraint fk\_business\_id\_bad foreign key (business\_id) references business(business\_id),  constraint fk\_attr\_id\_bad foreign key (attr\_id) references config\_business\_attr(attr\_id)  ); |

# **4. Other Database Objects**

Database objects other than Tables are described in this Section. For now, Sequences and Triggers have been defined. Required Views and Procedures will be added during the development of the system.

## **4.1. Sequences**

Following Sequences are needed to work with corresponding Triggers to have the primary keys of all Data Tables to increment autonomously. In order to accommodate imported data, the start values of all Sequences are set to 11000001 instead of 1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sequence** | **Table** | **Column** | **Start Value** | **Increment** |
| SEQ\_USER\_ID | USER\_INFO | USER\_ID | 11000001 | 1 |
| SEQ\_TOKEN\_ID | USER\_ACCESS\_TOKEN | TOKEN\_ID | 11000001 | 1 |
| SEQ\_BUSINESS\_ID | BUSINESS | BUSINESS\_ID | 11000001 | 1 |
| SEQ\_HOUR\_ID | BUSINESS\_HOUR | HOUR\_ID | 11000001 | 1 |
| SEQ\_REVIEW\_ID | REVIEW | REVIEW\_ID | 11000001 | 1 |
| SEQ\_VOTE\_ID | VOTE | VOTE\_ID | 11000001 | 1 |
| SEQ\_IMAGE\_ID | IMAGE | IMAGE\_ID | 11000001 | 1 |
| SEQ\_TEXT\_ID | BUSINESS\_ATTR\_TEXT | TEXT\_ID | 11000001 | 1 |
| SEQ\_NUMBER\_ID | BUSINESS\_ATTR\_NUMBER | NUMBER\_ID | 11000001 | 1 |
| SEQ\_DATE\_ID | BUSINESS\_ATTR\_DATE | DATE\_ID | 11000001 | 1 |

## **4.2. Triggers**

For autonomous increment of primary key, and autonomous setting of Creation and Modification timestamps, two Triggers are needed for each Data Table.

Trigger On Insert:

* Set Primary Key Column to next value of corresponding Sequence.
* Set CREATION\_TIME to current time.

Trigger On Update:

* Set MODIFICATION\_TIME to current time.