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

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.

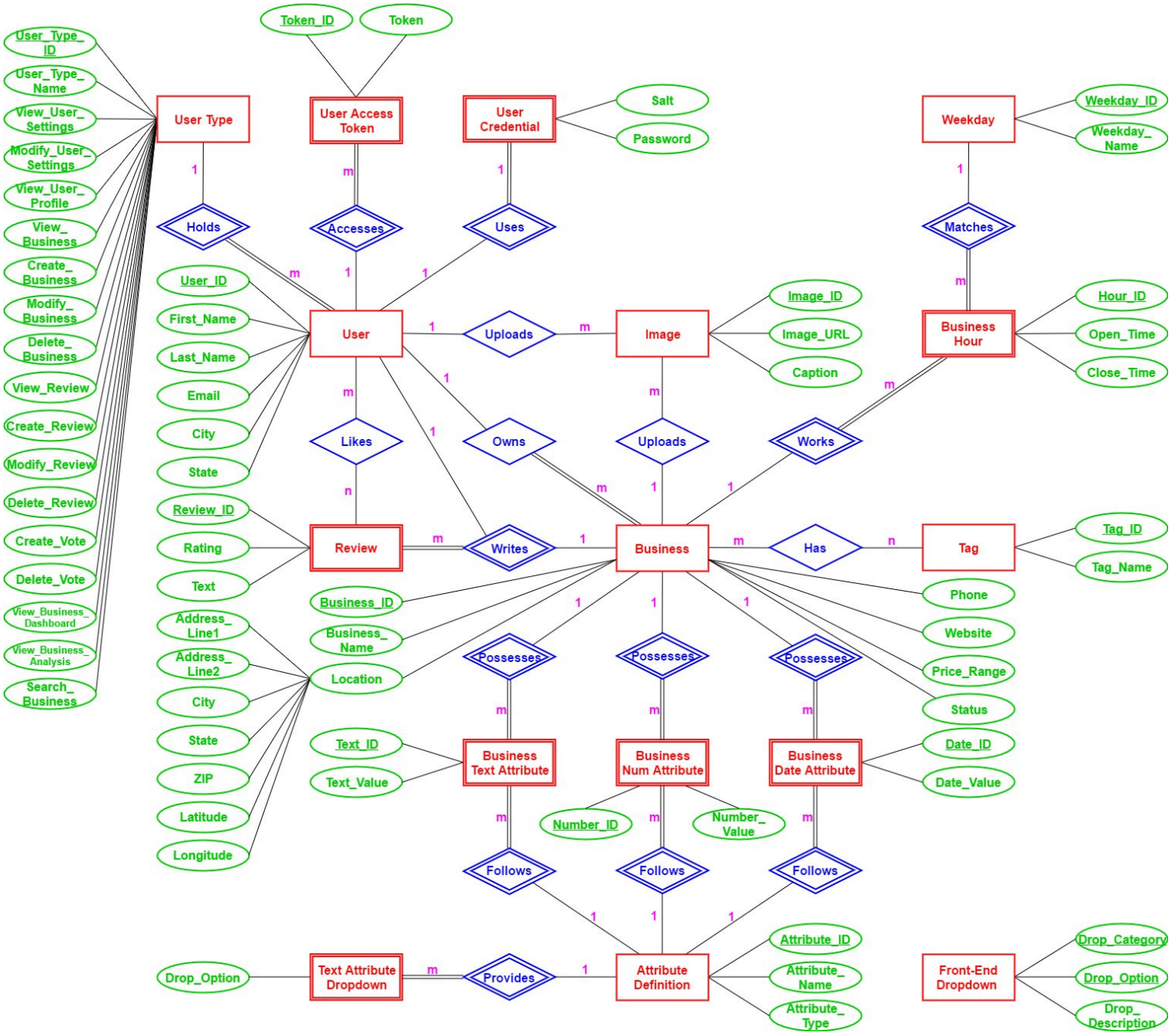
2. Data Model

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.

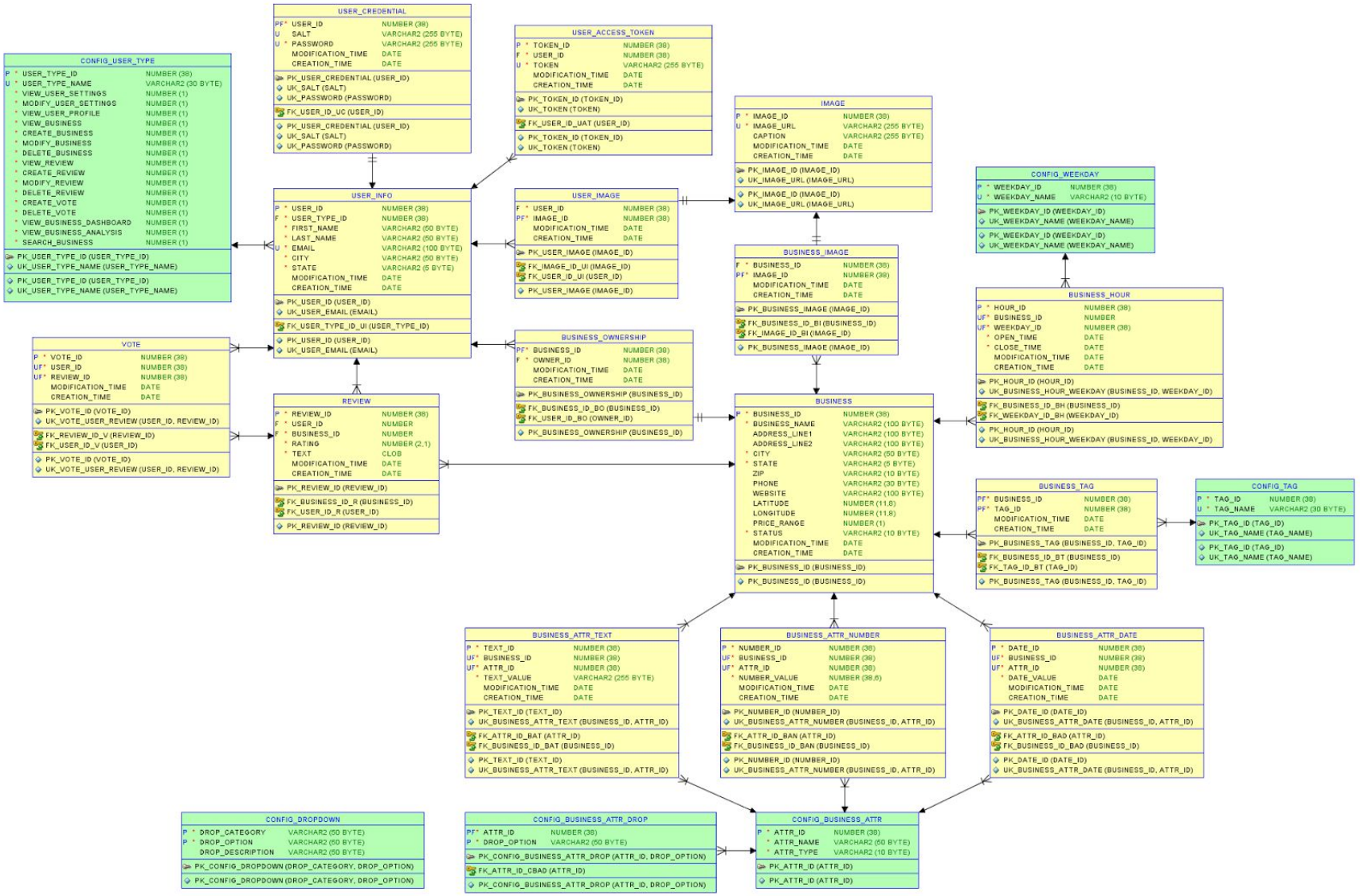
All entity attributes except the modification and creation timestamps are shown in the E-R Diagram. The logical relationships are primarily about User and Business which are exactly the two cynosures of the system.


User_Type_ID
User_Type_Name
View_User_Settings
Modify_User_Settings
View_User_Profile
View_Business
Create_Business
Modify_Business
Delete_Business
View_Review
Create_Review
Modify_Review
Delete_Review
Create_Vote
Delete_Vote
View_Business_Dashboard
View_Business_Analysis
Search_Business
User Type
Token_ID
Token
User Access Token
User Credential
Salt
Password
Weekday
Weekday_ID
Weekday_Name
Holds
Accesses
Uses
Matches
User_ID
First_Name
Last_Name
Email
City
State
User
Uploads
Image
Image_ID
Image_URL
Caption
Business Hour
Hour_ID
Open_Time
Close_Time
Likes
Owns
Uploads
Works
Review_ID
Rating
Text
Review
Writes
Business
Has
Tag
Tag_ID
Tag_Name
Business_ID
Business_Name
Address_Line1
Address_Line2
City
State
ZIP
Latitude
Longitude
Location
Phone
Website
Price_Range
Status
Possesses
Possesses
Possesses
Text_ID
Text_Value
Business Text Attribute
Business Num Attribute
Business Date Attribute
Date_ID
Date_Value
Number_ID
Number_Value
Follows
Follows
Follows
Drop_Option
Text Attribute Dropdown
Provides
Attribute Definition
Attribute_ID
Attribute_Name
Attribute_Type
Front-End Dropdown
Drop_Category
Drop_Option
Drop_Description

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.


USER_CREDENTIAL
PF* USER_ID NUMBER (38)
U SALT VARCHAR2 (255 BYTE)
U * PASSWORD VARCHAR2 (255 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_USER_CREDENTIAL (USER_ID)
UK_SALT (SALT)
UK_PASSWORD (PASSWORD)
FK_USER_ID_UC (USER_ID)
PK_USER_CREDENTIAL (USER_ID)
UK_SALT (SALT)
UK_PASSWORD (PASSWORD)
USER_ACCESS_TOKEN
P * TOKEN_ID NUMBER (38)
F * USER_ID NUMBER (38)
U * TOKEN VARCHAR2 (255 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_TOKEN_ID (TOKEN_ID)
UK_TOKEN (TOKEN)
FK_USER_ID_UAT (USER_ID)
PK_TOKEN_ID (TOKEN_ID)
UK_TOKEN (TOKEN)
CONFIG_USER_TYPE
P * USER_TYPE_ID NUMBER (38)
U * USER_TYPE_NAME VARCHAR2 (30 BYTE)
* VIEW_USER_SETTINGS NUMBER (1)
* MODIFY_USER_SETTINGS NUMBER (1)
* VIEW_USER_PROFILE NUMBER (1)
* VIEW_BUSINESS NUMBER (1)
* CREATE_BUSINESS NUMBER (1)
* MODIFY_BUSINESS NUMBER (1)
* DELETE_BUSINESS NUMBER (1)
* VIEW_REVIEW NUMBER (1)
* CREATE_REVIEW NUMBER (1)
* MODIFY_REVIEW NUMBER (1)
* DELETE_REVIEW NUMBER (1)
* CREATE_VOTE NUMBER (1)
* DELETE_VOTE NUMBER (1)
* VIEW_BUSINESS_DASHBOARD NUMBER (1)
* VIEW_BUSINESS_ANALYSIS NUMBER (1)
* SEARCH_BUSINESS NUMBER (1)
PK_USER_TYPE_ID (USER_TYPE_ID)
UK_USER_TYPE_NAME (USER_TYPE_NAME)
PK_USER_TYPE_ID (USER_TYPE_ID)
UK_USER_TYPE_NAME (USER_TYPE_NAME)
IMAGE
P * IMAGE_ID NUMBER (38)
U * IMAGE_URL VARCHAR2 (255 BYTE)
CAPTION VARCHAR2 (255 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_IMAGE_ID (IMAGE_ID)
UK_IMAGE_URL (IMAGE_URL)
PK_IMAGE_ID (IMAGE_ID)
UK_IMAGE_URL (IMAGE_URL)
CONFIG_WEEKDAY
P * WEEKDAY_ID NUMBER (38)
U * WEEKDAY_NAME VARCHAR2 (10 BYTE)
PK_WEEKDAY_ID (WEEKDAY_ID)
UK_WEEKDAY_NAME (WEEKDAY_NAME)
PK_WEEKDAY_ID (WEEKDAY_ID)
UK_WEEKDAY_NAME (WEEKDAY_NAME)
USER_INFO
P * USER_ID NUMBER (38)
F * USER_TYPE_ID NUMBER (38)
* FIRST_NAME VARCHAR2 (50 BYTE)
* LAST_NAME VARCHAR2 (50 BYTE)
U * EMAIL VARCHAR2 (100 BYTE)
* CITY VARCHAR2 (50 BYTE)
* STATE VARCHAR2 (5 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_USER_ID (USER_ID)
UK_USER_EMAIL (EMAIL)
FK_USER_TYPE_ID_UI (USER_TYPE_ID)
PK_USER_ID (USER_ID)
UK_USER_EMAIL (EMAIL)
USER_IMAGE
F * USER_ID NUMBER (38)
PF* IMAGE_ID NUMBER (38)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_USER_IMAGE (IMAGE_ID)
FK_IMAGE_ID_UI (IMAGE_ID)
FK_USER_ID_UI (USER_ID)
PK_USER_IMAGE (IMAGE_ID)
BUSINESS_IMAGE
F * BUSINESS_ID NUMBER (38)
PF* IMAGE_ID NUMBER (38)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_BUSINESS_IMAGE (IMAGE_ID)
FK_BUSINESS_ID_BI (BUSINESS_ID)
FK_IMAGE_ID_BI (IMAGE_ID)
PK_BUSINESS_IMAGE (IMAGE_ID)
BUSINESS_HOUR
P * HOUR_ID NUMBER (38)
UF* BUSINESS_ID NUMBER
UF* WEEKDAY_ID NUMBER (38)
* OPEN_TIME DATE
* CLOSE_TIME DATE
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_HOUR_ID (HOUR_ID)
UK_BUSINESS_HOUR_WEEKDAY (BUSINESS_ID, WEEKDAY_ID)
FK_BUSINESS_ID_BH (BUSINESS_ID)
FK_WEEKDAY_ID_BH (WEEKDAY_ID)
PK_HOUR_ID (HOUR_ID)
UK_BUSINESS_HOUR_WEEKDAY (BUSINESS_ID, WEEKDAY_ID)
VOTE
P * VOTE_ID NUMBER (38)
UF* USER_ID NUMBER (38)
UF* REVIEW_ID NUMBER (38)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_VOTE_ID (VOTE_ID)
UK_VOTE_USER_REVIEW (USER_ID, REVIEW_ID)
FK_REVIEW_ID_V (REVIEW_ID)
FK_USER_ID_V (USER_ID)
PK_VOTE_ID (VOTE_ID)
UK_VOTE_USER_REVIEW (USER_ID, REVIEW_ID)
BUSINESS_OWNERSHIP
PF* BUSINESS_ID NUMBER (38)
F * OWNER_ID NUMBER (38)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_BUSINESS_OWNERSHIP (BUSINESS_ID)
FK_BUSINESS_ID_BO (BUSINESS_ID)
FK_USER_ID_BO (OWNER_ID)
PK_BUSINESS_OWNERSHIP (BUSINESS_ID)
REVIEW
P * REVIEW_ID NUMBER (38)
F * USER_ID NUMBER
F * BUSINESS_ID NUMBER
* RATING NUMBER (2,1)
* TEXT CLOB
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_REVIEW_ID (REVIEW_ID)
FK_BUSINESS_ID_R (BUSINESS_ID)
FK_USER_ID_R (USER_ID)
PK_REVIEW_ID (REVIEW_ID)
BUSINESS
P * BUSINESS_ID NUMBER (38)
* BUSINESS_NAME VARCHAR2 (100 BYTE)
ADDRESS_LINE1 VARCHAR2 (100 BYTE)
ADDRESS_LINE2 VARCHAR2 (100 BYTE)
* CITY VARCHAR2 (50 BYTE)
* STATE VARCHAR2 (5 BYTE)
ZIP VARCHAR2 (10 BYTE)
PHONE VARCHAR2 (30 BYTE)
WEBSITE VARCHAR2 (100 BYTE)
LATITUDE NUMBER (11,8)
LONGITUDE NUMBER (11,8)
PRICE_RANGE NUMBER (1)
* STATUS VARCHAR2 (10 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_BUSINESS_ID (BUSINESS_ID)
PK_BUSINESS_ID (BUSINESS_ID)
BUSINESS_TAG
PF* BUSINESS_ID NUMBER (38)
PF* TAG_ID NUMBER (38)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_BUSINESS_TAG (BUSINESS_ID, TAG_ID)
FK_BUSINESS_ID_BT (BUSINESS_ID)
FK_TAG_ID_BT (TAG_ID)
PK_BUSINESS_TAG (BUSINESS_ID, TAG_ID)
CONFIG_TAG
P * TAG_ID NUMBER (38)
U * TAG_NAME VARCHAR2 (30 BYTE)
PK_TAG_ID (TAG_ID)
UK_TAG_NAME (TAG_NAME)
PK_TAG_ID (TAG_ID)
UK_TAG_NAME (TAG_NAME)
BUSINESS_ATTR_TEXT
P * TEXT_ID NUMBER (38)
UF* BUSINESS_ID NUMBER (38)
UF* ATTR_ID NUMBER (38)
* TEXT_VALUE VARCHAR2 (255 BYTE)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_TEXT_ID (TEXT_ID)
UK_BUSINESS_ATTR_TEXT (BUSINESS_ID, ATTR_ID)
FK_ATTR_ID_BAT (ATTR_ID)
FK_BUSINESS_ID_BAT (BUSINESS_ID)
PK_TEXT_ID (TEXT_ID)
UK_BUSINESS_ATTR_TEXT (BUSINESS_ID, ATTR_ID)
BUSINESS_ATTR_NUMBER
P * NUMBER_ID NUMBER (38)
UF* BUSINESS_ID NUMBER (38)
UF* ATTR_ID NUMBER (38)
* NUMBER_VALUE NUMBER (38,6)
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_NUMBER_ID (NUMBER_ID)
UK_BUSINESS_ATTR_NUMBER (BUSINESS_ID, ATTR_ID)
FK_ATTR_ID_BAN (ATTR_ID)
FK_BUSINESS_ID_BAN (BUSINESS_ID)
PK_NUMBER_ID (NUMBER_ID)
UK_BUSINESS_ATTR_NUMBER (BUSINESS_ID, ATTR_ID)
BUSINESS_ATTR_DATE
P * DATE_ID NUMBER (38)
UF* BUSINESS_ID NUMBER (38)
UF* ATTR_ID NUMBER (38)
* DATE_VALUE DATE
MODIFICATION_TIME DATE
CREATION_TIME DATE
PK_DATE_ID (DATE_ID)
UK_BUSINESS_ATTR_DATE (BUSINESS_ID, ATTR_ID)
FK_ATTR_ID_BAD (ATTR_ID)
FK_BUSINESS_ID_BAD (BUSINESS_ID)
PK_DATE_ID (DATE_ID)
UK_BUSINESS_ATTR_DATE (BUSINESS_ID, ATTR_ID)
CONFIG_DROPDOWN
P * DROP_CATEGORY VARCHAR2 (50 BYTE)
P * DROP_OPTION VARCHAR2 (50 BYTE)
DROP_DESCRIPTION VARCHAR2 (50 BYTE)
PK_CONFIG_DROPDOWN (DROP_CATEGORY, DROP_OPTION)
PK_CONFIG_DROPDOWN (DROP_CATEGORY, DROP_OPTION)
CONFIG_BUSINESS_ATTR_DROP
PF* ATTR_ID NUMBER (38)
P * DROP_OPTION VARCHAR2 (50 BYTE)
PK_CONFIG_BUSINESS_ATTR_DROP (ATTR_ID, DROP_OPTION)
FK_ATTR_ID_CBAD (ATTR_ID)
PK_CONFIG_BUSINESS_ATTR_DROP (ATTR_ID, DROP_OPTION)
CONFIG_BUSINESS_ATTR
P * ATTR_ID NUMBER (38)
* ATTR_NAME VARCHAR2 (50 BYTE)
* ATTR_TYPE VARCHAR2 (10 BYTE)
PK_ATTR_ID (ATTR_ID)
PK_ATTR_ID (ATTR_ID)

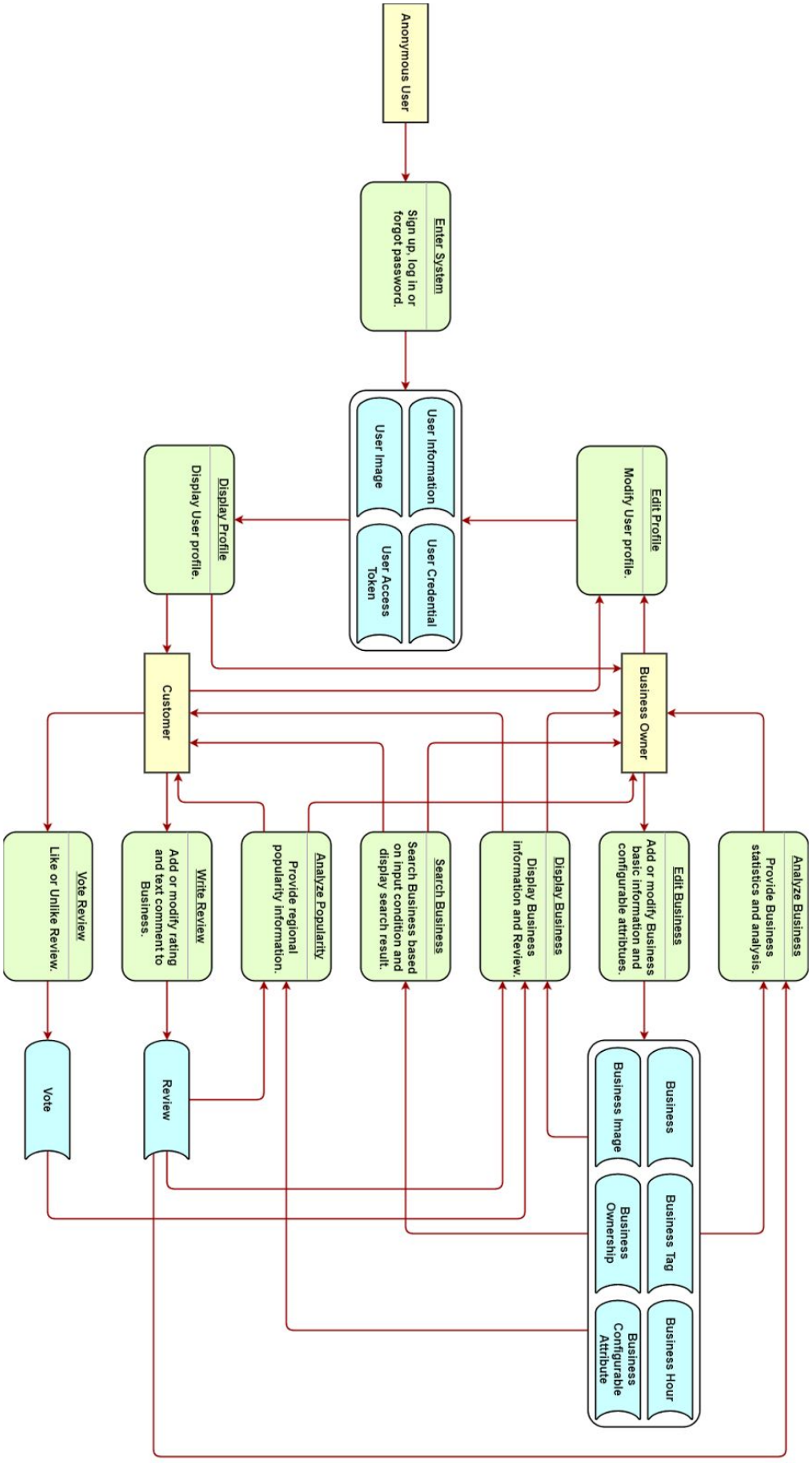
2.3. Data Flow Diagram

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.


Anonymous User
Enter System
Sign up, log in or forgot password.
User Information
User Credential
User Image
User Access Token
Edit Profile
Modify User profile.
Display Profile
Display User profile.
Customer
Business Owner
Vote Review
Like or Unlike Review.
Write Review
Add or modify rating and text comment to Business.
Analyze Popularity
Provide regional popularity information.
Search Business
Search Business based on input condition and display search result.
Display Business
Display Business information and Review.
Edit Business
Add or modify Business basic information and configurable attributes.
Analyze Business
Provide Business statistics and analysis.
Vote
Review
Business
Business Image
Business Tag
Business Ownership
Business Hour
Business Configurable Attribute

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)	Not Null	

		CONFIG_TAG (TAG_ID)		
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.