

Business Recommendation Distributed Database System

Phase II

Yaning Liu (415, yliu176@jhu.edu)

Zhihao Cao (415, zcaos8@jhu.edu)

A web application backed by distributed database system for users to search for the best business like restaurants, bars, shopping and others with attributes they want, as well as reviews given by specific user.

1. Change Since Phase I

We decide to put emphasis on the distributed database design in terms of scalability. The front-end web interface is not longer our focus so it may be a little bit unpolished.

2. Dataset

Our project is developed on Yelp Challenge

Dataset(http://www.yelp.com/dataset_challenge). It is in JSON format originally.

We load the dataset in the following steps:

- Partition the tables horizontally according to id field into 4 fragments; Each tuple belongs to the $\text{hash}(\text{id}) \% 4$ -th server; This procedure is fulfilled by partition.py.
- Load each fragments into its server's MySQL; This is accomplished by jsonToDB.php.

3. Platform

Our system is developed on 9 Amazon EC2 instances(t2.micro):

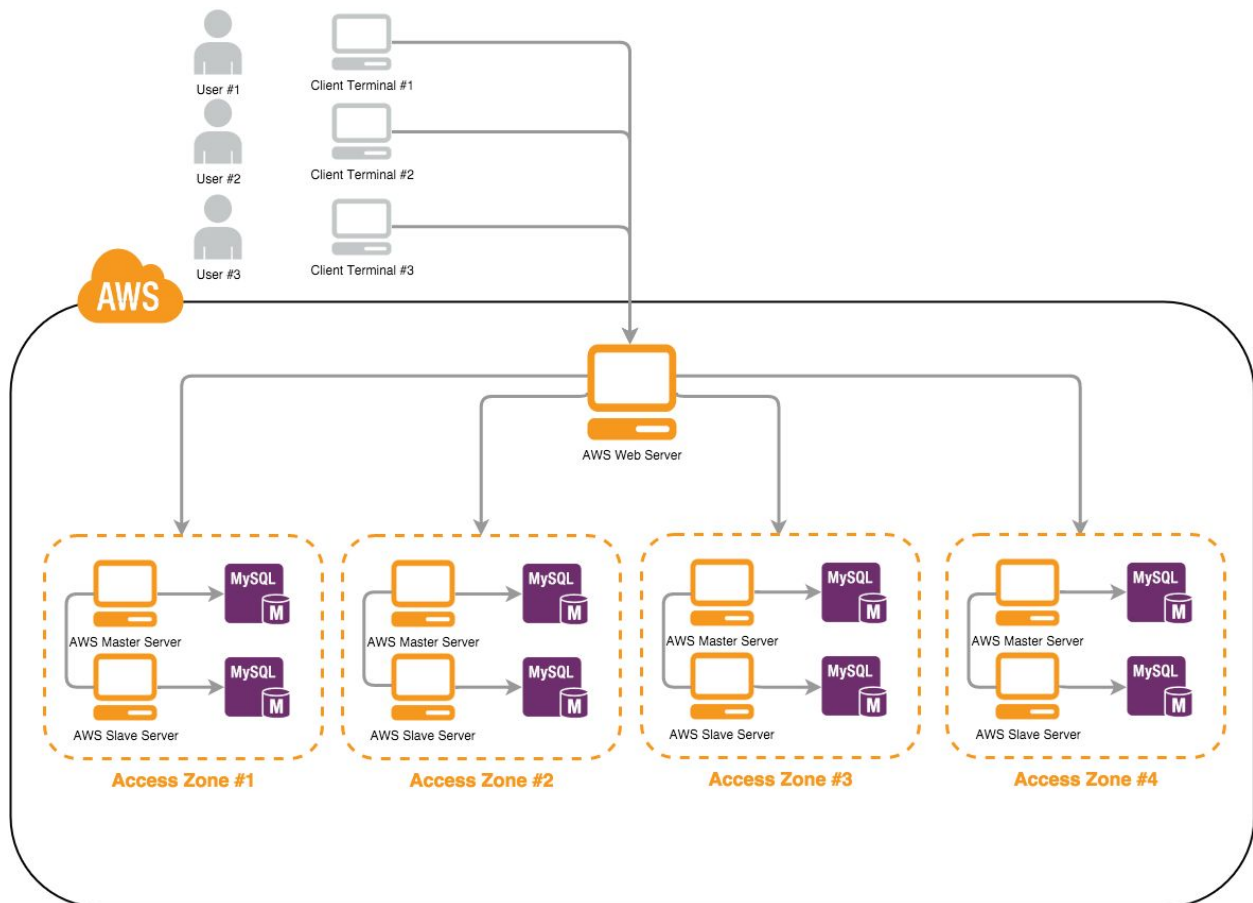


Fig 3.1 Servers Overview

Among them, one AWS server acts as a web server. Users request web pages from it and it grabs data from backend servers.

The other 8 AWS servers are divided by 4 access zones, each of which contains a pair of master/slave setup.

4. How To Run

- `partition.py`: Place it under the same directory with JSON files of the Yelp Challenge Dataset. Then run: `./partition.py`. After that you will get 4 fragments for each JSON file.
- `jsonToDB.php`: Place it under the same directory with the JSON fragments on each server. Run the php code to insert data into MySQL. Note that in order to prevent failure you may want to insert one table at a time.
- Web files(under `web/`): Place them under `/var/www/html/` on each server.
 - (1) `business_recommendation.html`: users can input the values into “Find”, “Name”, “Location” to find the top 20 results.
 - (2) `user_review.html`: users can input the username to find the reviews of the specific user.

5. Specialization

We specialize in distributed database design(major). We also deal with issues regarding security problems(minor) and implement a web application(minor).

6. Limitations and Possible Improvements

- The web server in the diagram can be improved in the future. In practice, it should be replaced by a load balancer that dispatches queries of users to corresponding server. This web server is also a single point of failure, which is not good in terms of scalability.
- Our web interface can be more polished and versatile. However, since our focus is learning the concepts of distributed system and how to deploy it, the interface is good enough to convey our idea.

7. Our Work(“output”)

We spend much time on deployment and configuration. We omit detailed steps in this section and just display our final result, so it contains much more work than it might look.

a. Server Deployment

We deploy our servers on AWS. Each server is t2.micro instance with Amazon Linux AMI 2015.09.1 (HVM), SSD Volume Type. We also install a LAMP web server on each server for development use. To expedite this process, AMI is used in order to spawn multiple similar EC2 instances.

<input type="checkbox"/>	Name	Instance ID	Instance Type	Availability Zone	Instance State	Status Checks	Alarm Status	Public DNS	Public IP
<input type="checkbox"/>	yelp4	i-0e03a8bf	t2.micro	us-east-1c	running	Initializing	None	ec2-52-20-153-168.co...	52.20.153.168
<input type="checkbox"/>	yelp1_slave	i-20289591	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-91-115-127.co...	52.91.115.127
<input type="checkbox"/>	yelp1	i-258c2494	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-20-153-92.com...	52.20.153.92
<input type="checkbox"/>	yelp3_slave	i-26289597	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-90-41-223.com...	52.90.41.223
<input type="checkbox"/>	yelp4_slave	i-27289596	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-91-80-152.com...	52.91.80.152
<input type="checkbox"/>	yelp2_slave	i-54c978e5	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-90-42-214.com...	52.90.42.214
<input type="checkbox"/>	yelp2	i-bb0da60a	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-21-83-20.comp...	52.21.83.20
<input type="checkbox"/>	yelp_web	i-cbdd6e7a	t2.micro	us-east-1c	running	2/2 checks ...	None	ec2-52-3-61-95.comput...	52.3.61.95
<input type="checkbox"/>	yelp3	i-cd0da67c	t2.micro	us-east-1c	running	Initializing	None	ec2-52-21-131-52.com...	52.21.131.52

b. Partition

We partition the original dataset horizontally into 4 pieces according to id(User_ID or Business_ID). Each fragment contains complete attribute set of the data. Given an id, we can instantaneously know that its tuple must show up in server $[\text{hash}(\text{id}) / 4]$. Therefore this approach can effectively spread load across all 4 servers. It also has another advantage: since the dataset are partitioned according to id, all relations associated with a given id(e.g. as a foreign key) must appear in the same site. Therefore we don't have to quest data from other server during join operations.

Scripts partition.py and jsonToDB.php are intended for the partition as illustrated in previous section.

c. Replication(Master/Slave)

Replication is a good way to improve availability and parallelism. We decide to use master/slave model here. Therefore at each site we have one master and one corresponding slave. We have set them up properly:

```
mysql> SHOW MASTER STATUS\G
***** 1. row *****
      File: mysql-bin.000009
      Position: 107
      Binlog_Do_DB:
      Binlog_Ignore_DB:
1 row in set (0.00 sec)
```

Fig 7.1 Master 2 status

```

mysql> SHOW SLAVE STATUS\G
***** 1. row *****
      Slave_IO_State: Waiting for master to send event
        Master_Host: 52.21.83.20
        Master_User: slave_user
        Master_Port: 3306
        Connect_Retry: 60
        Master_Log_File: mysql-bin.000009
    Read_Master_Log_Pos: 107
        Relay_Log_File: relay-log.000002
        Relay_Log_Pos: 253
    Relay_Master_Log_File: mysql-bin.000009
        Slave_IO_Running: Yes
        Slave_SQL_Running: Yes
        Replicate_Do_DB:
    Replicate_Ignore_DB:
        Replicate_Do_Table:
    Replicate_Ignore_Table:
    Replicate_Wild_Do_Table:
    Replicate_Wild_Ignore_Table:
            Last_Errno: 0
            Last_Error:
        Skip_Counter: 0
    Exec_Master_Log_Pos: 107
        Relay_Log_Space: 403
        Until_Condition: None
        Until_Log_File:
        Until_Log_Pos: 0
    Master_SSL_Allowed: No
    Master_SSL_CA_File:
    Master_SSL_CA_Path:
        Master_SSL_Cert:
        Master_SSL_Cipher:
        Master_SSL_Key:
    Seconds_Behind_Master: 0
Master_SSL_Verify_Server_Cert: No
            Last_IO_Errno: 0
            Last_IO_Error:
            Last_SQL_Errno: 0
            Last_SQL_Error:
    Replicate_Ignore_Server_Ids:
        Master_Server_Id: 1
1 row in set (0.00 sec)

```

Fig 7.2 Slave 2 status

The master and slave are configured correctly so that whenever the master gets updated, it will record the updates in its bin log, then the slave will grab them to its own relay log. Then the slave will parse the content in the relay log and get updated as well.

```
[mysqld]
server-id    = 1
log-bin=mysql-bin
binlog_format=ROW
```

Fig 7.3 Master 2 /etc/my.cnf

```
[mysqld]
server-id = 2
relay_log=relay-log
skip-slave-start
```

Fig 7.4 Slave 2 /etc/my.cnf

In order to ensure high availability, we also realize **read-write splitting** on the master/slave model. That is, we want users' reads can be spread over all slaves while their writes only happen on master.

mysqlnd_ms plugin is a very useful tool for this purpose. Carefully, we configure as follows:

```
extension=mysqlnd_ms.so
mysqlnd_ms.enable=1
mysqlnd_ms.config_file=/home/ec2-user/mysqlnd_ms_plugin.ini
mysqlnd_ms.force_config_usage=1
```

Fig 7.5 /etc/php.d/mysqlnd_ms.ini

```
{
  "52.21.83.20": {
    "master": {
      "master_0": {
        "host": "52.21.83.20",
        "port": 3306
      }
    },
    "slave": {
      "slave_0": {
        "host": "52.90.42.214",
        "port": 3306
      }
    }
  }
}
```

Fig 7.6 ~/mysqlnd_ms_plugin.ini

To see whether our configuration is correct, we can use strace to monitor whether the query goes:

```
[ec2-user@ip-172-31-62-55 html]$ strace /usr/bin/php test3.php |& grep connect
read(3, "<?php\n$con = mysql_connect(\"52.2...\", 4096) = 465
connect(3, {sa_family=AF_INET, sin_port=htons(3306), sin_addr=inet_addr("52.90.42.214"
)}, 16) = -1 EINPROGRESS (operation now in progress)
```

Fig 7.7 strace result

test3.php only includes a basic contain query. It shows that the query is delivered to 52.80.42.214 which is the slave address, instead of the master address. Therefore we are convinced that the read-write splitting is successful: it dispatches reads across slaves and writes to master. This strategy can guarantee high availability of the system due to distributed read/write.

d. Security Issues

Since our distributed database system design involves several servers and communication among them, we do run into some security issues when we deploy the servers.

First, every server other than the web server needs to grant select privilege to the web server address:

```
mysql> GRANT SELECT ON *.* TO 'dbuser'@'52.3.61.95' IDENTIFIED BY  
'yelp' WITH GRANT OPTION;  
Query OK, 0 rows affected (0.00 sec)
```

Fig 7.8 grant select

Second, when we set up the master/slave, the master needs to grant replication privilege to the slave address:

```
mysql> GRANT REPLICATION SLAVE ON *.* TO 'slave_user'@'52.90.42.214'  
' IDENTIFIED BY 'slave';  
Query OK, 0 rows affected (0.00 sec)
```

Fig 7.9 grant result

We also run into a problem that using root for remote access will result in access failure. After Googling for a while that it seems to be a problem caused by upgrading from old MySQL version to MySQL 5.5. In the end, we decide to create a new user 'dbuser' to circumvent this problem, since abusing root account is not a good practice after all.

e. Web Interface

We implement the web interface using PHP+HTML+CSS+Bootstrap. Users can search for restaurants, hotels, etc. by their names, or search for top 10 businesses with highest score, etc.

(1) Search without attributes

ec2-52-3-61-95.compute-1.amazonaws.com/bootstrap-3.3.6/docs/examples/starter-template/index.html

Worldwide Business Evaluation System

United States

United Kindom

Germany

Canada

Find Restaurants

Name (Optional) Brand Name

Near Las Vegas

☐ Wi-Fi

☐ Dogs Allowed

☐ Smoking

☐ Open 24 Hours

☐ Parking

☐ Take Reservations

Search

ec2-52-3-61-95.compute-1.amazonaws.com/bootstrap-3.3.6/docs/examples/starter-template/index.html#

Fig 7.10 Search without attributes

ec2-52-3-61-95.compute-1.amazonaws.com/bootstrap-3.3.6/docs/examples/starter-template/dbConn.php

#	Name	Address	Stars
1	Bonjour Bistro	9055 S Eastern Ave Southeast Las Vegas, NV 89123	5
2	Tetsuro's Sayonara,	5020 Spring Mountain Rd Chinatown Las Vegas, NV 89146	5
3	Snow Ono Shave Ice	Las Vegas, NV	5
4	Sleepy Wilson's Barb	101 S Rainbow Blvd Ste 7 Westside Las Vegas, NV 89145	5
5	Egg Works	2025 Village Center Dr Summerlin Las Vegas, NV 89104	5
6	La Tradicion	4371 Stewart Sunrise Las Vegas, NV 89110	5
7	Bonjour Bistro	9055 S Eastern Ave Southeast Las Vegas, NV 89123	5
8	La Tradicion	4371 Stewart Sunrise Las Vegas, NV 89110	5
9	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5
10	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5
11	Tony Marc's Eats	6700 W Charleston Blvd Unit C Westside Las Vegas, NV 89146	5
12	Capriotti's Sandwich	9210 S Eastern Ave Ste 115 Southeast Las Vegas, NV 89123	5
13	Bonjour Bistro	9055 S Eastern Ave Southeast Las Vegas, NV 89123	5
14	Egg Works	2025 Village Center Dr Summerlin Las Vegas, NV 89104	5
15	La Tradicion	4371 Stewart Sunrise Las Vegas, NV 89110	5
16	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5

Fig 7.11 Search without attributes result

(2) Search with attributes

Worldwide Business Evaluation System

[United States](#)
[United kindom](#)
[Germany](#)
[Canada](#)

Find

Name (Optional)

Near

☒ Wi-Fi
☐ Dogs Allowed
☐ Smoking
☐ Open 24 Hours
☐ Parking
☐ Take Reservations

Fig 7.12 Search with attributes

#	Name	Address	Stars
1	Tony Marc's Eats	6700 W Charleston Blvd Unit C Westside Las Vegas, NV 89146	5
2	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5
3	Snow Ono Shave Ice	Las Vegas, NV	5
4	Snow Ono Shave Ice	Las Vegas, NV	5
5	Egg Works	2025 Village Center Dr Summerlin Las Vegas, NV 89104	5
6	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5
7	Egg Works	2025 Village Center Dr Summerlin Las Vegas, NV 89104	5
8	Capriotti's Sandwich	9210 S Eastern Ave Ste 115 Southeast Las Vegas, NV 89123	5
9	Tony Marc's Eats	6700 W Charleston Blvd Unit C Westside Las Vegas, NV 89146	5
10	Snow Ono Shave Ice	Las Vegas, NV	5
11	Bite Vegan Bakery	440 S Rampart Blvd Westside Las Vegas, NV 89145	5
12	Egg Works	2025 Village Center Dr Summerlin Las Vegas, NV 89104	5
13	Capriotti's Sandwich	9210 S Eastern Ave Ste 115 Southeast Las Vegas, NV 89123	5
14	Capriotti's Sandwich	9210 S Eastern Ave Ste 115 Southeast Las Vegas, NV 89123	5
15	Capriotti's Sandwich	9210 S Eastern Ave Ste 115 Southeast Las Vegas, NV 89123	5
16	Tony Marc's Eats	6700 W Charleston Blvd Unit C Westside Las Vegas, NV 89146	5
17	Tony Marc's Eats	6700 W Charleston Blvd Unit C Westside Las Vegas, NV 89146	5

Fig 7.13 Search with attributes result

(3) Search by Business Name

ec2-52-3-61-95.compute-1.amazonaws.com/business_recommendation.html

Worldwide Business Evaluation System

United States

United kindom

Germany

Canada

Find Restaurants, Shopping, Hotel, Gyms ...

Name (Optional) Dairy Queen

Near State, City

☐ Wi-Fi

☐ Dogs Allowed

☐ Smoking

☐ Open 24 Hours

☐ Parking

☐ Take Reservations

Search

Fig 7.14 Search by Business Name

ec2-52-3-61-95.compute-1.amazonaws.com/business_recommendation.php

#	Name	Address	Stars
1	Dairy Queen	2754 S Alma School Rd Mesa, AZ 85210	4
2	Dairy Queen	3020 Prosperity Church Rd Ste D Highland Creek Charlotte, NC 28269	4
3	Dairy Queen	211 E Florence Blvd Casa Grande, AZ 85222	4
4	Dairy Queen	12456 N 28th Dr Phoenix, AZ 85029	3.5
5	Dairy Queen	1910 Village Ctr Cir Ste 9 Summerlin Las Vegas, NV 89134	3.5
6	Dairy Queen	7510 S Rural Rd Tempe, AZ 85283	3.5
7	Dairy Queen	437 N Gilbert Rd Gilbert, AZ 85234	3.5
8	Dairy Queen	3160 W Sahara Ave Westside Las Vegas, NV 89102	3.5
9	Dairy Queen	2595 S Maryland Pkwy Eastside Las Vegas, NV 89109	3.5
10	Dairy Queen	911 W University Ave Urbana, IL 61801	3.5
11	Dairy Queen	1431 Central Ave Plaza Midwood Charlotte, NC 28205	3
12	Dairy Queen	4036 N Tenaya Way Northwest Las Vegas, NV 89129	3
13	Dairy Queen	3850 Las Vegas Blvd The Strip Las Vegas, NV 89109	3
14	Dairy Queen	268 Sainte-Catherine Rue W Montreal, QC H2X 2A1	3
15	Dairy Queen	800 Settlers Ridge Ctr Dr McKees Rocks, PA 15136	3
16	Dairy Queen	6125 Spring Mountain Road Chinatown Las Vegas, NV 89146	3
17	Dairy Queen	10799 Grand Avenue Sun City, AZ 85351	2

Fig 7.15 Search by Business Name Result

(4) Search User's Review

Fig 7.16 Search User's Review

#	Review	Stars	Date
1	Husnus.....definitely my "date place" freshman year before I could a) afford anything else and b) get anywhere off campus. On the first block of Street.....this place is a great value. \$12 for salad and an entree. I love the kabobs and anything with lamb in it. The yogurt dressing is really good and the bread is very traditional. Don't miss it, no matter how many much more expensive meals I have, I always return to this favorite. 5 stars.	5	2006-01-13
2	This place is supposed to have the best steak in Madison. It is very good, but I'm not sure it's the best. My date got the \$37 special for the night...a very thick steak, beautiful cut of meat. The only problem is that he ordered it well (I know, we won't be going out again!) which meant that the outside was almost charcoaled but the inside was still pink! I got rabbit (\$18) in a portabella wine sauce which was very good. Portions are definitely generous! The service was fantastic. They seemed to have a girl whose sole purpose in life was to fill water glasses...as soon as I took a drink she was there to fill it up again! Our main waitress was very attentive and they place had a good crowd even on a Wednesday night! I've also been here for drinks before, which was fun...the waitresses are very friendly...which might have been because we came with cute boys, but who knows?	4	2006-01-13
3	Well, if you ignore the meager decor and shitty shitty service, this place rocks!! Lots of food with a great curry sauce. Last time I ate here with my mom, the lady heling us didn't speak a single word and glared at us the entire time. The food is too good to pass up though...and the two of us ate there for about \$18.....3 stars.	3	2006-01-13
4	My favorite Madison mexican restaurant. The decor is vibrant and traditional and the food is very authentic and colorful with lots of fresh veggies. Great chips and salsa. All-in-all, a nice fresh take on a cuisine that can get very predictable. A little more expensive than say, Casa de Lara, but I always look forward to eating here. 5 stars.	5	2006-01-13
5	Great food...a little pricey. Basic pasta is about \$16 and the meat dishes range from \$20-35. I had salmon...it was amazing! When my boyfriend at the time and I ate there while it was still nice out, so we were able to sit outside and people watch. Also, because they share owners, the acts that perform at the Majestic often eat there, so keep an I out for them. Check out my Majestic review for a good story about that! Great food and lots of options, whatever you are in the mood for. 4 stars.	4	2006-01-13
6	Hooray for another place in Madison that satisfies my love of French cuisine!! I ate at Cocoliquot with a group of 6 friends. It was perfect for us! We each ordered a couple of the "little plate" dishes and passed them around. Basically this amounted to us trying almost everything on the menu. The whole philosophy of the restaurant is small, appetizer-sized plates, which I like because you can try so many things! Prices range from \$4 for a tiny cheese platter to \$12 for foie gras. Even though I don't feel like I ate a lot, I was definitely filled up...that's the beauty of French food. I'd recommend the chevre (I'm obsessed) and the mussels. It was about \$20 per person including tip for two items and a dessert per person. I would recommend this for a large group of people or a romantic date where you don't mind using your fingers or feeding each other! 5 Stars.	5	2006-01-13

Fig 7.17 Search User's Review Result

f. SQL

Since our data is partitioned into several fragments, SQL queries becomes a little different. One interesting example is “get top 5 restaurants with “Wi-Fi” with highest score in Las Vegas”.

First we perform the following SQL on each servers:

```
SELECT B.Name, B.Address, B.Stars
FROM Business B, (
    SELECT Business_ID
    FROM Business_Attributes WHERE Attribute = "Wi-Fi"
) B2,
(
    SELECT Business_ID
    FROM Business_Categories
    WHERE Category = "Restaurants") B1
WHERE B.City = "Las Vegas"
    AND B.Business_ID = B1.Business_ID
    AND B.Business_ID = B2.Business_ID
ORDER BY B.Stars DESC
LIMIT 5;
```

Then you can get 20 tuples(5 tuples from a master/slave) in total.

Next, perform merge sort on these tuples to pick 5 with highest scores.

This efficient and simple paradigm is very helpful when your data is spread across different sites. Since each master/slave contains much fewer tuples than a centralized server and merge sort’s complexity is $O(N\log N)$, this query can be still executed efficiently under distributed setup.

8. Database Definition Language

```
create table Business (
    Business_ID    VARCHAR(25) NOT NULL,
    Name           VARCHAR(20) NOT NULL,
    Address        VARCHAR(100),
    City           VARCHAR(15),
    State          VARCHAR(5),
    Latitude       DOUBLE,
    Longitude      DOUBLE,
    Stars          DOUBLE,
    PRIMARY KEY    (Business_ID)
);
```

<u>Business_ID</u>	<u>Name</u>	<u>Address</u>	<u>City</u>	<u>State</u>	<u>Latitude</u>	<u>Longitude</u>	<u>Stars</u>
"JpYn_HdxQNZQSIWWv4P6lw"	"Niva"	"3743 S Las Vegas Blvd The Strip Las Vegas, NV 89109"	"Las Vegas"	"NV "	45.53390139999998	-73.659655799999996	4.3

```

create table Business_Attributes(
  Business_ID      VARCHAR(25) NOT NULL,
  Attribute        VARCHAR(50),
  Value           VARCHAR(10),
  PRIMARY KEY      (Business_ID, Attribute),
  FOREIGN KEY      (Business_ID)
    REFERENCES Business (Business_ID)
    ON DELETE CASCADE
);

```

<u>Business_ID</u>	<u>Attribute</u>	<u>Value</u>
"JpYn_HdxQNZQSIWWv4P6lw"	"Wi-Fi"	"Free"

```

create table Business_Categories(
  Business_ID      VARCHAR(25) NOT NULL,
  Category         VARCHAR(100),
  PRIMARY KEY      (Business_ID, Category),
  FOREIGN KEY      (Business_ID)
    REFERENCES Business (Business_ID)
    ON DELETE CASCADE
);

```

<u>Business_ID</u>	<u>Category</u>
"JpYn_HdxQNZQSIWWv4P6lw"	"Restaurant"

```

create table User(
  User_ID          VARCHAR(25) NOT NULL,
  Name             VARCHAR(15) NOT NULL,
  Review_Count     INTEGER,
  Average_Stars    DOUBLE,
  Yelping_Since    DATE NOT NULL,
  Fans             INTEGER,

```

PRIMARY KEY (User_ID)
);

<u>User_ID</u>	Name	Review _Count	Average_ Stars	Yelping_Since	Fans
"3ctGjRh7NaXZ-Gr34 A5b6A"	"Kathy"	68	3.71	"2008-02-00"	23

```
create table Business_Review(
  Business_ID    VARCHAR(25) NOT NULL,
  User_ID        VARCHAR(25) NOT NULL,
  Stars          DOUBLE,
  Text           TEXT,
  Date           DATE,
  PRIMARY KEY    (Business_ID, User_ID),
  FOREIGN KEY    (User_ID)
    REFERENCES   User (User_ID)
    ON DELETE CASCADE
);
```

Business_ID	User_ID	Stars	Text	Date
"JpYn_HdxQNZ QSIWWv4P6lw"	"FKosz9ExVLhs rITQk5xl4g"	5	"As far as Starbucks go, this is a pretty nice one. The baristas are friendly and while I was here, a lot of regulars must have come in, because they bantered away with almost everyone. The bathroom was clean and well maintained and the trash wasn't overflowing in the canisters around the store. The pastries looked fresh, but I didn't partake."	2007-11-20

```
create table Check_In(
  Business_ID    VARCHAR(25) NOT NULL,
  Monday         INTEGER,
```

Tuesday INTEGER,
 Wednesday INTEGER,
 Thursday INTEGER,
 Friday INTEGER,
 Saturday INTEGER,
 Sunday INTEGER,
 PRIMARY KEY (Business_ID),
 FOREIGN KEY (Business_ID)
 REFERENCES Business (Business_ID)
 ON DELETE CASCADE

);

<u>Business_ID</u>	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
"JpYn_HdxQ NZQSIWWv4 P6lw"	5	3	5	1	9	8	13