

Relational DB Creation

First we analyze data on the following document. [Copia de Diccionario de datos](#)
The structure of This Project and files delivered to us are:

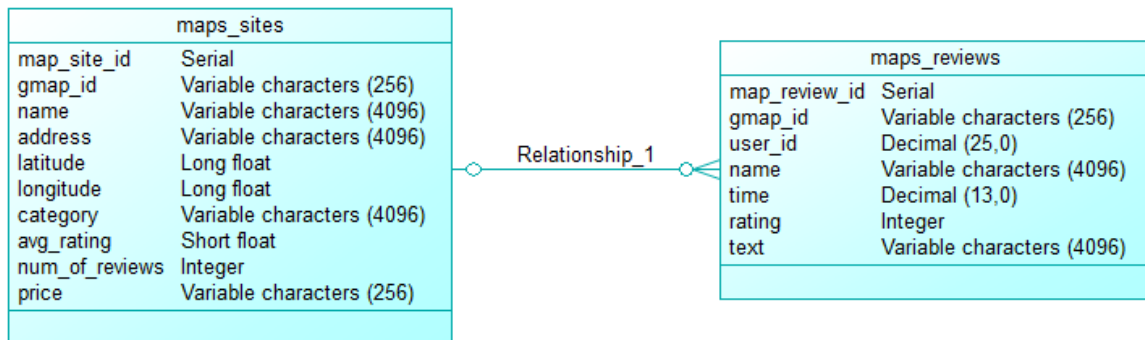
Google Maps:

| metadata_sitios | | |
|-----------------|----------------------------|---|
| key | data type | Example |
| name | String | 'Walgreens Pharmacy' |
| address | String | 'Walgreens Pharmacy, 124 E North St, Kendallville, IN 46755' |
| gmap_id | String | '0x881614ce7c13acbb:0x5c7b18bbf6ec4f7e' |
| description | String | 'Department of the Walgreens chain providing prescription medications & other health-related items.', |
| latitude | Float | 41.451859999999996 |
| longitude | Float | -85.2666757 |
| category | Array of Strings | ['Pharmacy'] |
| avg_rating | Float | 4.2 |
| num_of_reviews | Integer | 5 |
| price | Integer, Nullable | '\$\$' |
| hours | Array of Arrays of Strings | [[['Thursday', '8AM–1:30PM'], ['Friday', '8AM–1:30PM'], ['Saturday', '9AM–1:30PM'], ['Sunday', '10AM–1:30PM'], ['Monday', '8AM–1:30PM'], ['Tuesday', '8AM–1:30PM'], ['Wednesday', '8AM–1:30PM']], |
| misc | Object, String | 'Service options': ['Curbside pickup', 'Drive-through', 'In-store pickup', 'In-store shopping'], 'Health & safety': ['Mask required', 'Staff wear masks', 'Staff get temperature checks'], |

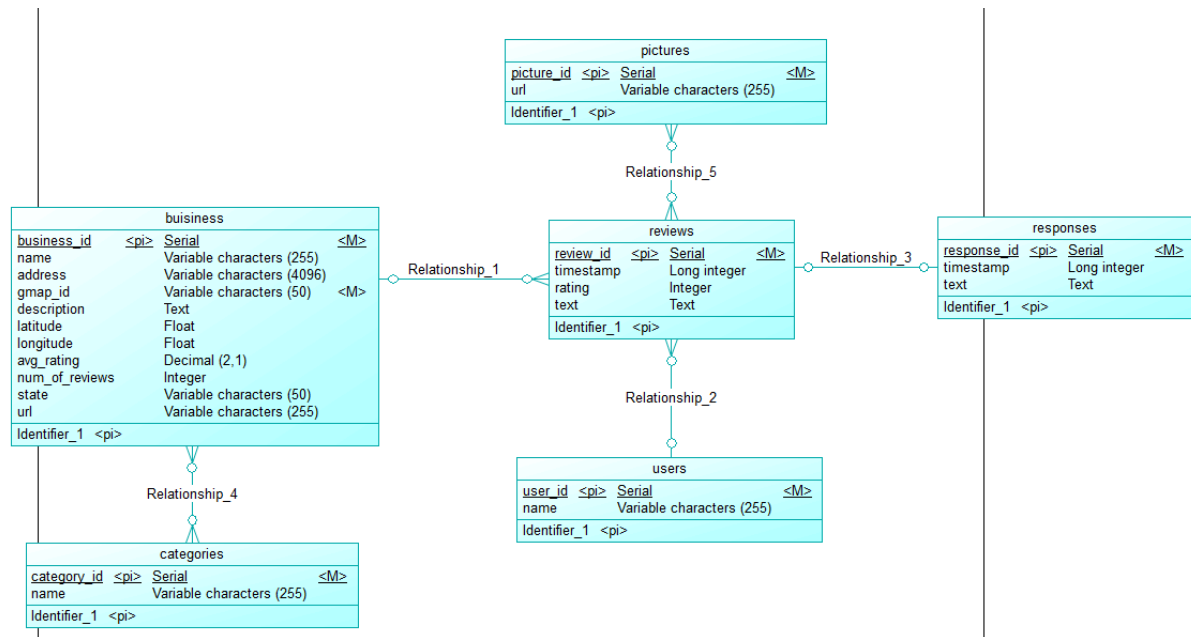
| | | |
|------------------|------------------|--|
| | | 'Accessibility': ['Wheelchair accessible entrance', 'Wheelchair accessible parking lot'], 'Planning': ['Quick visit'], 'Payments': ['Checks', 'Debit cards'] }, |
| state | String | 'Closes soon · 1:30PM · Reopens 2PM' |
| relative results | Array of Strings | ['0x881614cd49e4fa33:0x2d507c24ff4f1c74', '0x8816145bf5141c89:0x535c1d605109f94b', '0x881614cda24cc591:0xca426e3a9b826432', '0x88162894d98b91ef:0xd139b34de70d3e03', '0x881615400b5e57f9:0xc56d17dbe420a67f'], |
| url | String | 'https://www.google.com/maps/place//data=!4m2!3m1!1s0x881614ce7c13acb b:0x5c7b18bbf6ec4f7e?authuser=-1&hl=en&gl=us' |

| review_estados | | |
|----------------|------------------------------------|---|
| key | data type | example |
| user_id | String | '101463350189962023774' |
| name | String | 'Jordan Adams' |
| time | Integer, Timestamp | 1627750414677 |
| rating | Integer | 5 |
| text | String | "Cool place, great people, awesome dentist!" |
| pics | Array of Objects: Array of Strings | 'pics': [{ 'url': ['https://lh5.googleusercontent |

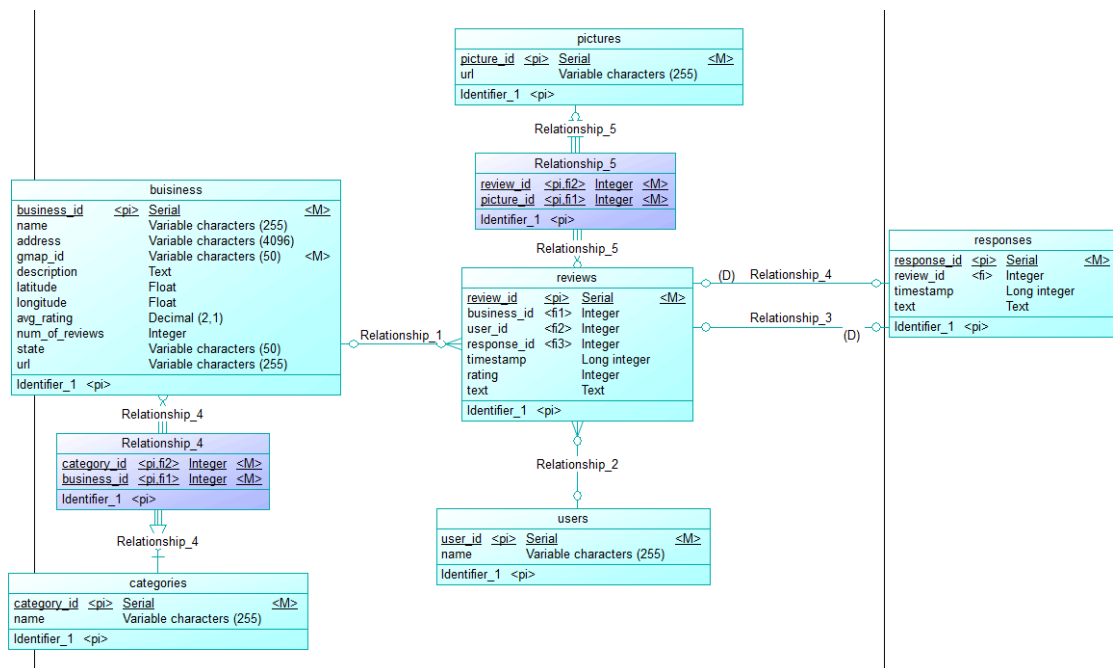
| | | |
|---------|--|--|
| | | nt.com/p/AF1QipNq2nZC5TH4_M7h5xRA 61hoTgvY1o9lozABgul=w150-h150-k-no-p] }], |
| resp | Object: time (Integer, Timestamp), text (String) | 'resp': { 'time': 1628455067818, 'text': 'Thank you for your five-star review! -Dr. Blake' }, |
| gmap_id | String | '0x87ec2394c2cd9d2d:0xd119cfbee0da6f3' |



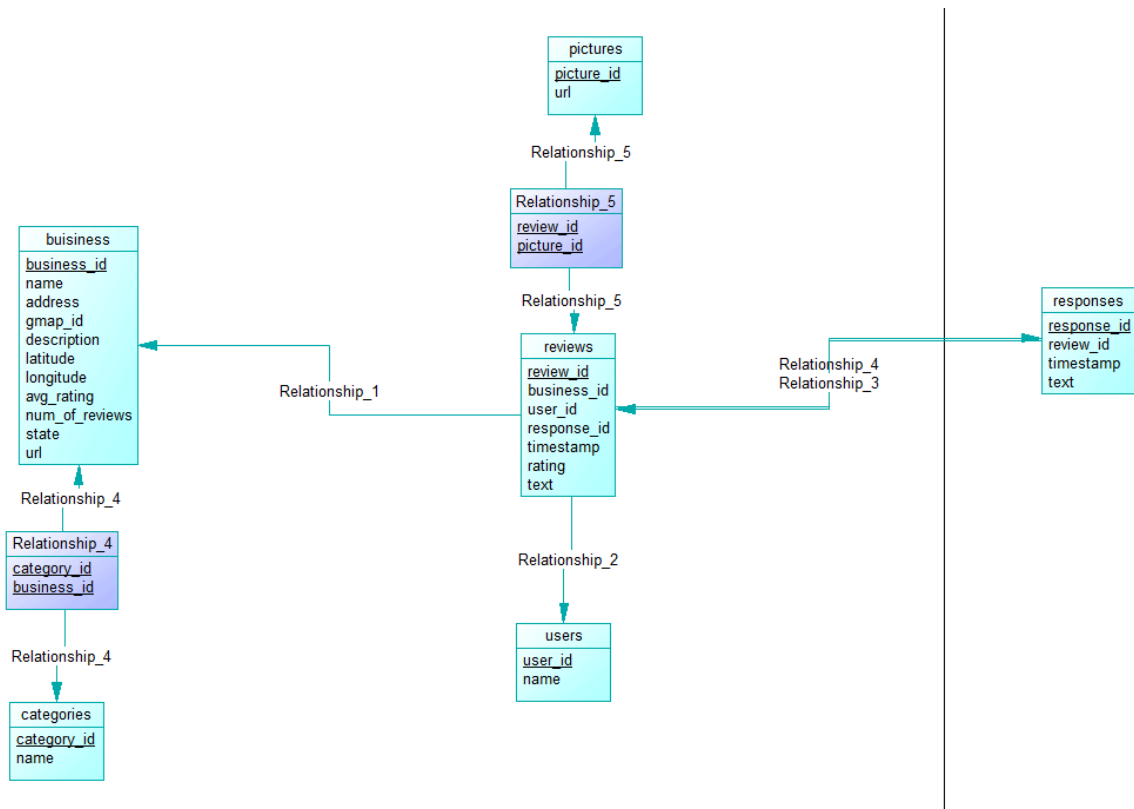
Taking that conceptual model into account we normalize tables and transform them to logical models.



Logical Model breaks many to many relationships to create intermediate tables.



Physical Data Model



SQL DB Generation Code for MySQL.

```

drop table if exists BUSINESS;

drop table if exists CATEGORIES;

drop table if exists PICTURES;

drop table if exists RELATIONSHIP_4;

drop table if exists RELATIONSHIP_5;

drop table if exists RESPONSES;

drop table if exists REVIEWS;

drop table if exists USERS;

/*=====*/
/* Table: BUSINESS */
/*=====*/
create table BUSINESS
(
    BUSINESS_ID          int not null auto_increment,

```

```

    NAME                varchar(255),
    ADDRESS              varchar(4096),
    GMAP_ID              varchar(50) not null,
    DESCRIPTION           text,
    LATITUDE              float,
    LONGITUDE             float,
    AVG_RATING            decimal(2,1),
    NUM_OF_REVIEWS        int,
    STATE                 varchar(50),
    URL                   varchar(255),
    primary key (BUSINESS_ID)
);

/*=====*/
/* Table: CATEGORIES */
/*=====*/
create table CATEGORIES
(
    CATEGORY_ID          int not null auto_increment,
    NAME                  varchar(255),
    primary key (CATEGORY_ID)
);

/*=====*/
/* Table: PICTURES */
/*=====*/
create table PICTURES
(
    PICTURE_ID           int not null auto_increment,
    URL                   varchar(255),
    primary key (PICTURE_ID)
);

/*=====*/
/* Table: RELATIONSHIP_4 */
/*=====*/
create table RELATIONSHIP_4
(
    CATEGORY_ID          int not null,
    BUSINESS_ID          int not null,
    primary key (CATEGORY_ID, BUSINESS_ID)
);

/*=====*/
/* Table: RELATIONSHIP_5 */
/*=====*/

```

```

create table RELATIONSHIP_5
(
    REVIEW_ID          int not null,
    PICTURE_ID         int not null,
    primary key (REVIEW_ID, PICTURE_ID)
);

/*=====*/
/* Table: RESPONSES                                     */
/*=====*/
create table RESPONSES
(
    RESPONSE_ID        int not null auto_increment,
    REVIEW_ID          int,
    TIMESTAMP           bigint,
    TEXT               text,
    primary key (RESPONSE_ID)
);

/*=====*/
/* Table: REVIEWS                                       */
/*=====*/
create table REVIEWS
(
    REVIEW_ID          int not null auto_increment,
    BUSINESS_ID        int,
    USER_ID            int,
    RESPONSE_ID        int,
    TIMESTAMP           bigint,
    RATING             int,
    TEXT              text,
    primary key (REVIEW_ID)
);

/*=====*/
/* Table: USERS                                         */
/*=====*/
create table USERS
(
    USER_ID            int not null auto_increment,
    NAME              varchar(255),
    primary key (USER_ID)
);

alter table RELATIONSHIP_4 add constraint FK_RELATIONSHIP_4 foreign key
(CATEGORY_ID)

```

```

        references CATEGORIES (CATEGORY_ID) on delete restrict on update
restrict;

alter table RELATIONSHIP_4 add constraint FK_RELATIONSHIP_7 foreign key
(BUSINESS_ID)
        references BUISINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table RELATIONSHIP_5 add constraint FK_RELATIONSHIP_5 foreign key
(REVIEW_ID)
        references REVIEWS (REVIEW_ID) on delete restrict on update
restrict;

alter table RELATIONSHIP_5 add constraint FK_RELATIONSHIP_8 foreign key
(PICTURE_ID)
        references PICTURES (PICTURE_ID) on delete restrict on update
restrict;

alter table RESPONSES add constraint FK_RELATIONSHIP_6 foreign key
(REVIEW_ID)
        references REVIEWS (REVIEW_ID) on delete restrict on update
restrict;

alter table REVIEWS add constraint FK_RELATIONSHIP_1 foreign key
(BUSINESS_ID)
        references BUISINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table REVIEWS add constraint FK_RELATIONSHIP_2 foreign key
(USER_ID)
        references USERS (USER_ID) on delete restrict on update restrict;

alter table REVIEWS add constraint FK_RELATIONSHIP_3 foreign key
(RESPONSE_ID)
        references RESPONSES (RESPONSE_ID) on delete restrict on update
restrict;

```

Yelp Dataset

| business | | |
|----------|-----------|---------|
| key | data type | example |

| | | |
|-----------------|-------------------|--|
| business_id | String | tnhfDv5II8EaGSXZGiuQGg |
| name | String | Garaje |
| address | String | 475 3rd St |
| city | String | San Francisco |
| state | String | CA |
| postal code | String | 94107 |
| latitude | Latitude | 37.7817529521 |
| longitude | Longitude | -122.39612197 |
| stars | Float | 4.5 |
| review_count | Integer | 1198 |
| is_open | Integer | 1 |
| attributes | Object of Objects | "attributes": { "RestaurantsTakeOut": true, |
| BusinessParking | Objects | "BusinessParking": { "garage": false, "street": true, "validated": false, "lot": false, "valet": false }, }, |
| categories | List | "categories": ["Mexican", "Burgers", "Gastropubs"], |
| hours | Objects | "hours": { "Monday": "10:00-21:00", "Tuesday": "10:00-21:00", "Friday": "10:00-21:00", "Wednesday": "10:00-21:00", "Thursday": "10:00-21:00", "Sunday": "11:00-18:00", |

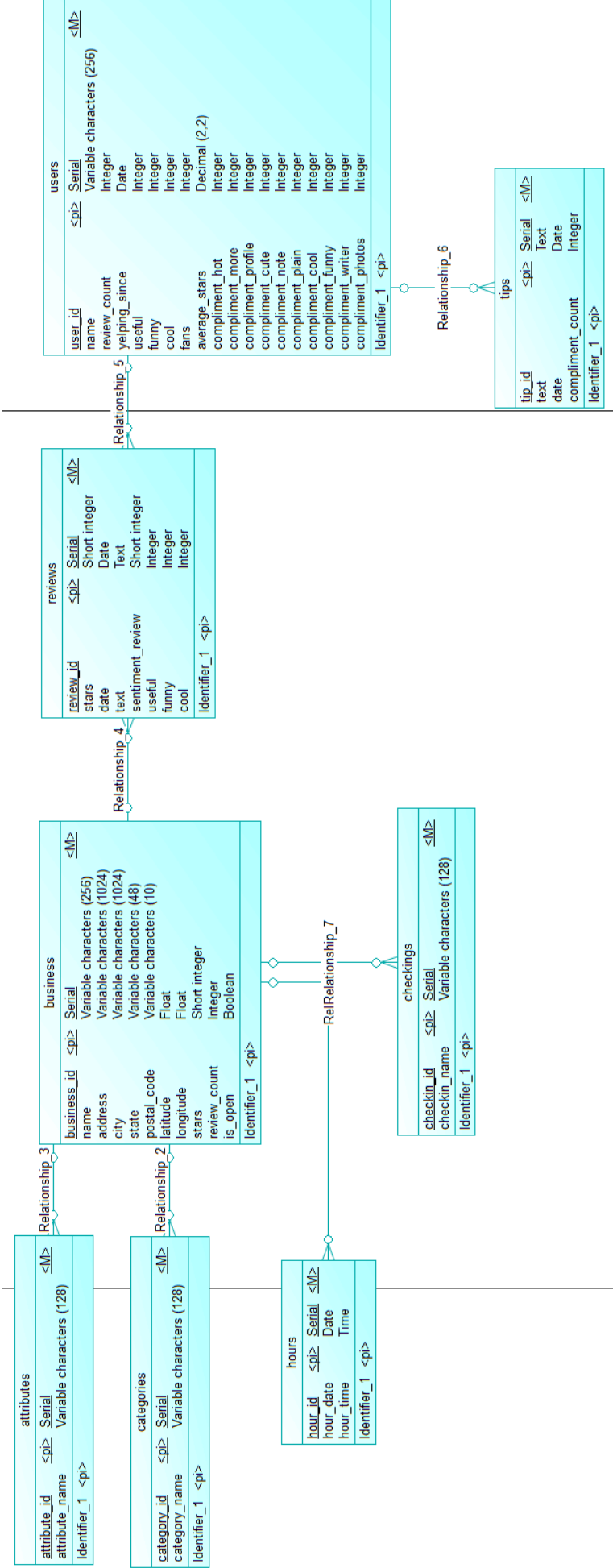
| | | |
|--|--|---------------------------|
| | | "Saturday": "10:00-21:00" |
|--|--|---------------------------|

| review | | |
|-------------|-----------|--|
| key | data type | example |
| review_id | String | zdSx_SD6obEhz9VrW9uA WA |
| user_id | String | Ha3iJu77CxlRfm-vQRs_8g |
| business_id | String | tnhfDv5ll8EaGSXZGiuQGg |
| stars | Float | 4 |
| date | String | 2016-03-09 |
| text | String | "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks." |
| useful | Integer | 0 |
| funny | Integer | 0 |
| cool | Integer | 0 |

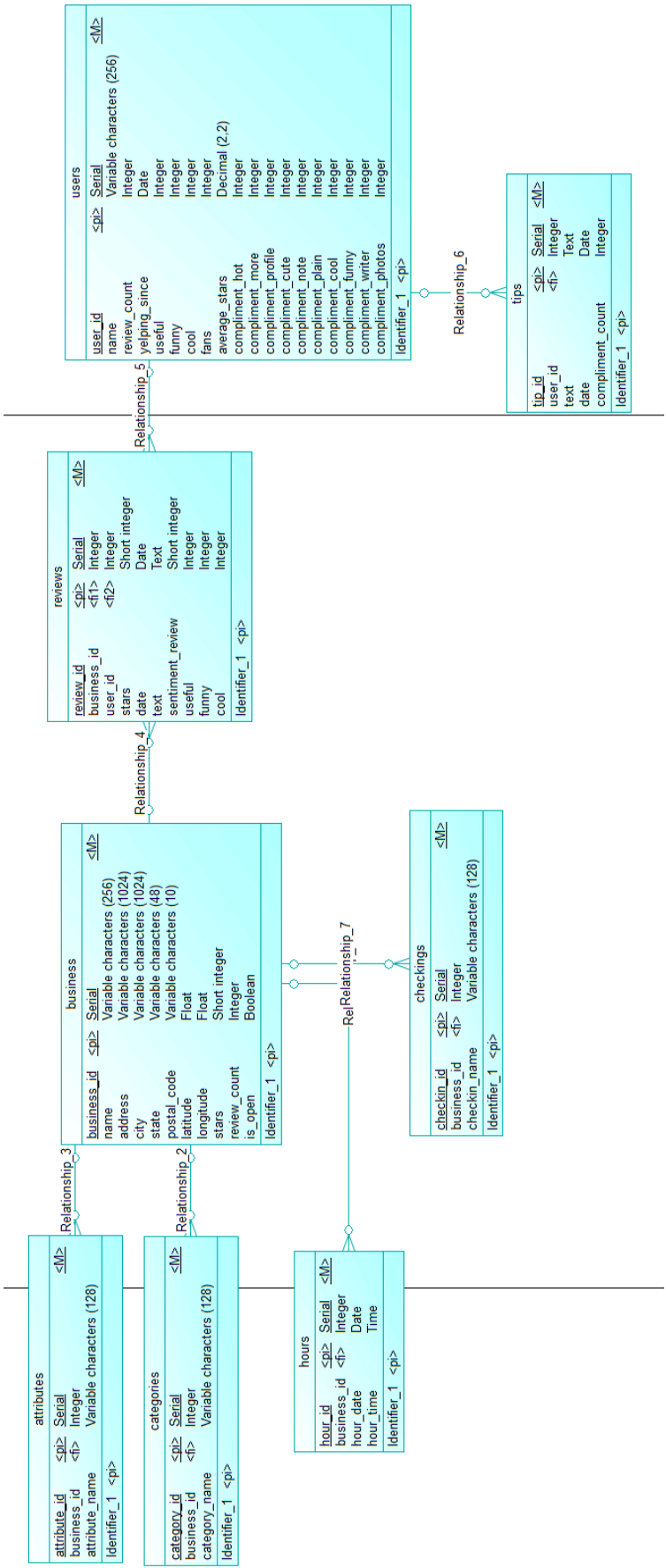
| checkin | | |
|-------------|---------------|---|
| key | data type | example |
| business_id | String | tnhfDv5ll8EaGSXZGiuQGg |
| date | String:Object | "date": "2016-04-26 19:49:16, 2016-08-30 18:36:57, 2016-10-15 02:45:18, 2016-11-18 01:54:50, 2017-04-20 |

| | | |
|--|--|-----------------------------------|
| | | 18:39:06, 2017-05-03 17:58:02" |
|--|--|-----------------------------------|

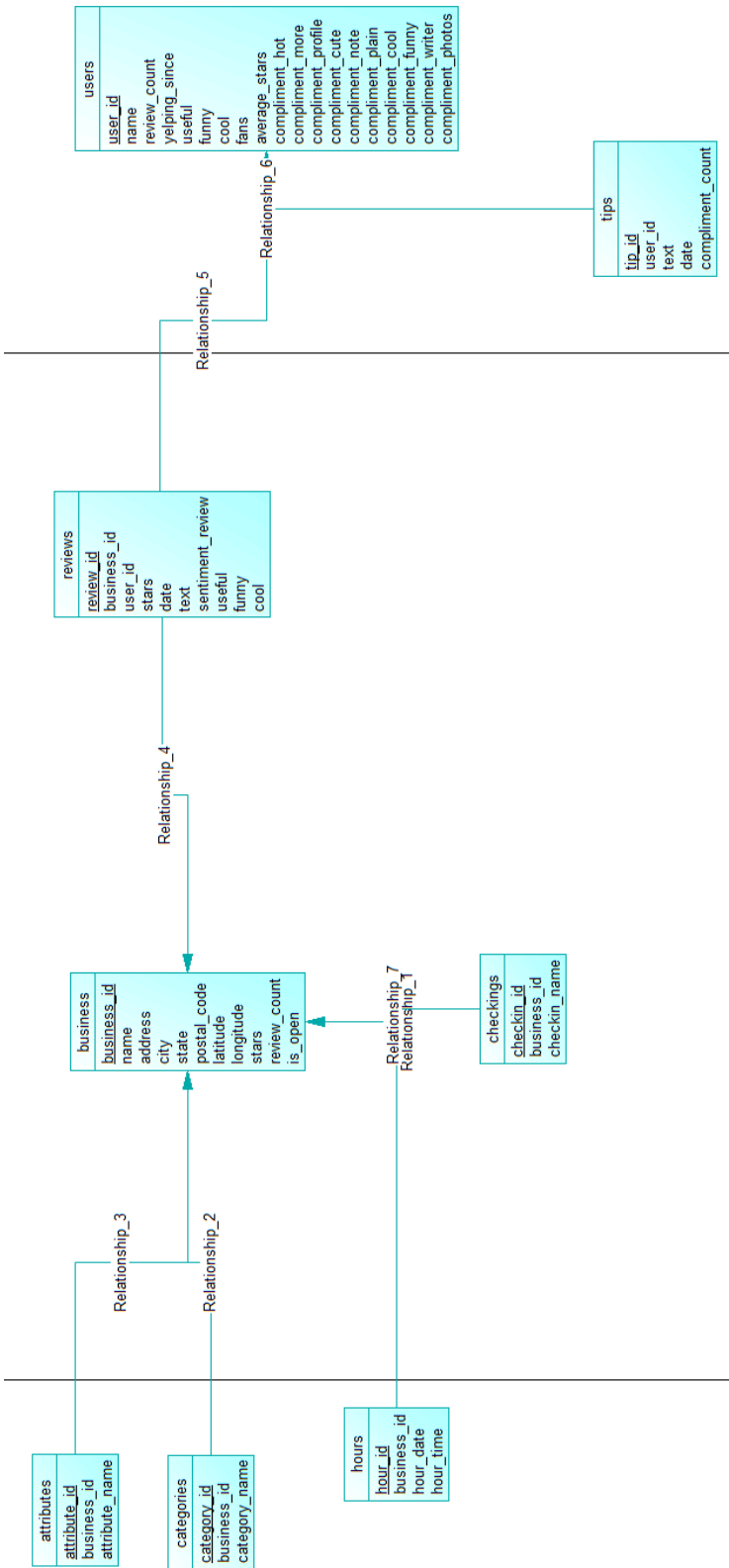
| tip | | |
|------------------|-----------|---|
| key | data type | example |
| text | String | "Secret menu - fried chicken sando is da bomb Their zapatos are good too." |
| date | String | 2013-09-20 |
| compliment_count | Integer | 172 |
| business_id | String | tnhfDv5ll8EaGSXZGiuQGg |
| user_id | String | 49JhAJh8vSQ-vM4Aourl0g |



Conceptual data model.



Physical Data Model.



```

drop table if exists ATTRIBUTES;

drop table if exists BUSINESS;

drop table if exists CATEGORIES;

drop table if exists CHECKINGS;

drop table if exists HOURS;

drop table if exists REVIEWS;

drop table if exists TIPS;

drop table if exists USERS;

/*=====*/
/* Table: ATTRIBUTES */
/*=====*/
create table ATTRIBUTES
(
    ATTRIBUTE_ID      int not null auto_increment,
    BUSINESS_ID       int,
    ATTRIBUTE_NAME     varchar(128),
    primary key (ATTRIBUTE_ID)
);

/*=====*/
/* Table: BUSINESS */
/*=====*/
create table BUSINESS
(
    BUSINESS_ID       int not null auto_increment,
    NAME              varchar(256),
    ADDRESS            varchar(1024),
    CITY               varchar(1024),
    STATE              varchar(48),
    POSTAL_CODE        varchar(10),
    LATITUDE           float,
    LONGITUDE          float,
    STARS              smallint,
    REVIEW_COUNT       int,
    IS_OPEN            bool,
    primary key (BUSINESS_ID)
);

```

```

/*=====*/
/* Table: CATEGORIES */
/*=====*/
create table CATEGORIES
(
    CATEGORY_ID      int not null auto_increment,
    BUSINESS_ID      int,
    CATEGORY_NAME     varchar(128),
    primary key (CATEGORY_ID)
);

/*=====*/
/* Table: CHECKINGS */
/*=====*/
create table CHECKINGS
(
    CHECKIN_ID       int not null auto_increment,
    BUSINESS_ID      int,
    CHECKIN_NAME      varchar(128),
    primary key (CHECKIN_ID)
);

/*=====*/
/* Table: HOURS */
/*=====*/
create table HOURS
(
    HOUR_ID          int not null auto_increment,
    BUSINESS_ID      int,
    HOUR_DATE         date,
    HOUR_TIME         time,
    primary key (HOUR_ID)
);

/*=====*/
/* Table: REVIEWS */
/*=====*/
create table REVIEWS
(
    REVIEW_ID        int not null auto_increment,
    BUSINESS_ID      int,
    USER_ID          int,
    STARS            smallint,
    DATE             date,
    TEXT             text,
    SENTIMENT_REVIEW smallint,

```



```

        USEFUL            int,
        FUNNY             int,
        COOL              int,
        primary key (REVIEW_ID)
    );

/*=====*/
/* Table: TIPS */
/*=====*/
create table TIPS
(
    TIP_ID                int not null auto_increment,
    USER_ID               int,
    TEXT                  text,
    DATE                  date,
    COMPLIMENT_COUNT      int,
    primary key (TIP_ID)
);

/*=====*/
/* Table: USERS */
/*=====*/
create table USERS
(
    USER_ID               int not null auto_increment,
    NAME                  varchar(256),
    REVIEW_COUNT          int,
    YELPING_SINCE         date,
    USEFUL                int,
    FUNNY                 int,
    COOL                  int,
    FANS                  int,
    AVERAGE_STARS         decimal(2,2),
    COMPLIMENT_HOT         int,
    COMPLIMENT_MORE       int,
    COMPLIMENT_PROFILE    int,
    COMPLIMENT_CUTE       int,
    COMPLIMENT_NOTE       int,
    COMPLIMENT_PLAIN      int,
    COMPLIMENT_COOL       int,
    COMPLIMENT_FUNNY      int,
    COMPLIMENT_WRITER     int,
    COMPLIMENT_PHOTOS     int,
    primary key (USER_ID)
);

```

```
alter table ATTRIBUTES add constraint FK_RELATIONSHIP_3 foreign key
(BUSINESS_ID)
    references BUSINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table CATEGORIES add constraint FK_RELATIONSHIP_2 foreign key
(BUSINESS_ID)
    references BUSINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table CHECKINGS add constraint FK_RELATIONSHIP_7 foreign key
(BUSINESS_ID)
    references BUSINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table HOURS add constraint FK_RELATIONSHIP_1 foreign key
(BUSINESS_ID)
    references BUSINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table REVIEWS add constraint FK_RELATIONSHIP_4 foreign key
(BUSINESS_ID)
    references BUSINESS (BUSINESS_ID) on delete restrict on update
restrict;

alter table REVIEWS add constraint FK_RELATIONSHIP_5 foreign key
(USER_ID)
    references USERS (USER_ID) on delete restrict on update restrict;

alter table TIPS add constraint FK_RELATIONSHIP_6 foreign key (USER_ID)
    references USERS (USER_ID) on delete restrict on update restrict;
```