**Yelp Data Exploration**

Tables:

Documentation of the tables can be found here: <https://www.yelp.com/dataset/documentation/main>

|  |  |
| --- | --- |
| **Table** | **Description** |
| business | Information about a business. Note: please ignore the is\_open column as it is not accurate |
| category | The categories that a business falls into |
| user | Information about a user |
| review | Information about a review by a user |

# **1. business table**

An example ( business\_id: \_ab50qdWOk0DdB6XOrBitw )

<https://www.yelp.com/biz/_ab50qdWOk0DdB6XOrBitw>

Graphical user interface, website

Description automatically generated

-- Show the first 20 rows of table

SELECT \*

SELECT \*

FROM yelp.business

LIMIT 20;

Graphical user interface, text, table

Description automatically generated

-- Count the total number of rows (businesses) in the table

SELECT COUNT(\*)

FROM yelp.business;

-- Count the total number of businesses in the "business" table (use distinct in case there are duplicate rows with the same businss\_id)

SELECT COUNT(DISTINCT business\_id)

FROM yelp.business;

Graphical user interface, application

Description automatically generated

-- The postal\_code column seems to use the numeric data type, which is problematic (states on east coast will have postal\_codes that look like 01234, but this will become 1234 in the data). Let’s change the data type to be CHAR (5), which is a 5-digit string. We then left filled the postal\_code with 0 so that 1234 becomes 01234.

ALTER TABLE yelp.business

MODIFY postal\_code CHAR(5);

UPDATE yelp.business

SET postal\_code=LPAD(postal\_code,5,'0');

-- Show the number of businesses in each state

SELECT state, COUNT(\*)

FROM yelp.business

GROUP BY state;

Table

Description automatically generated

-- Summary statistics for stars and review\_count

SELECT 'Total',

COUNT(stars) as stars,

COUNT(review\_count) as review\_count

FROM yelp.business

UNION

SELECT 'Average',

Table

Description automatically generated

AVG(stars),

AVG(review\_count)

FROM yelp.business

UNION

SELECT 'Min',

MIN(stars),

MIN(review\_count)

FROM yelp.business

UNION

SELECT 'Max',

MAX(stars),

MAX(review\_count)

FROM yelp.business;

# **2. category table**

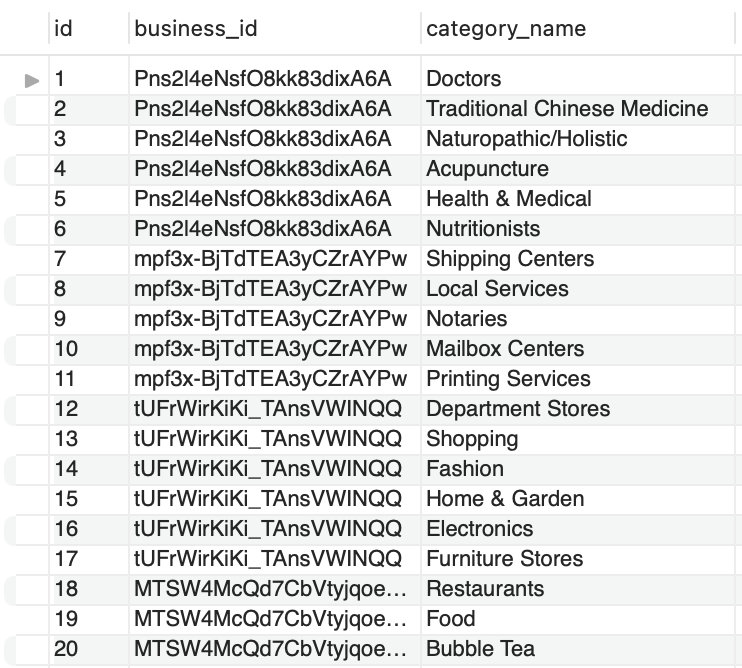
-- Show the first 20 rows of table

SELECT \*

SELECT \*

FROM yelp.category

LIMIT 20;



-- Show the number of unique categories

SELECT COUNT(DISTINCT category\_name)

FROM yelp.category;

Graphical user interface, application

Description automatically generated with medium confidence

-- Order by category\_name, show 20 unique categories

SELECT DISTINCT category\_name

FROM yelp.category

ORDER BY category\_name

LIMIT 20;

Table

Description automatically generated

# **3. user table**

An example ( user\_id: Hi10sGSZNxQH3NLyWSZ1oA)

<https://www.yelp.com/user_details?userid=Hi10sGSZNxQH3NLyWSZ1oA>

Graphical user interface, website

Description automatically generated

-- Show the first 20 rows of table

SELECT \*

SELECT \*

FROM yelp.user

LIMIT 20;

Table

Description automatically generated

-- Summary statistics for review\_count, average\_stars, and fans

SELECT 'Total',

COUNT(review\_count) as review\_count,

COUNT(average\_stars) as average\_stars,

Table

Description automatically generated

COUNT(fans) as fans

FROM yelp.user

UNION

SELECT 'Average',

AVG(review\_count) as review\_count,

AVG(average\_stars) as average\_stars,

AVG(fans) as fans

FROM yelp.user

UNION

SELECT 'Min',

MIN(review\_count) as review\_count,

MIN(average\_stars) as average\_stars,

MIN(fans) as fans

FROM yelp.user

UNION

SELECT 'Max',

MAX(review\_count) as review\_count,

MAX(average\_stars) as average\_stars,

MAX(fans) as fans

FROM yelp.user;

# **4. review table**

Graphical user interface, text

Description automatically generated

-- Show the first 20 rows of table

SELECT \*

SELECT \*

FROM yelp.review

LIMIT 20;



-- Show the dates for the earliest/latest reviews

SELECT MIN(date), MAX(date)

FROM yelp.review;

Table

Description automatically generated