Database System for Consumer Review Analysis on Yelp

CLASS PROJECT: DATABASE SYSTEM DESIGN & IMPLEMENTATION

IS 455 – DATABASE DESIGN & PROTOTYPING

Name : Le Kim Ngan
Email : lkhoang2@illinois.edu
NetID: lkhoang2

Table of Contents

I. Background and Objectives

As a student with a background in Marketing, I recognize that a high-quality database is essential for aligning teams around a shared understanding of customers and for tracking marketing performance over time. Reliable data enables consistent evaluation of campaigns and supports evidence-based decision-making.

From a competitive analysis perspective, access to up-to-date and relevant data allows businesses to assess their position within various markets—whether city-specific or nationwide—and to identify opportunities for differentiation. When it comes to understanding customers, uncovering their perceptions of a business requires a structured and well-designed database that can capture and organize insights drawn from authentic user feedback.

For this class project, I aim to leverage **Yelp—an established review platform that helps consumers discover, evaluate, and engage with local businesses across the United States and Canada (Yelp Press, 2025) .** By utilizing publicly available user review data, this project seeks to design and implement a database system capable of generating meaningful insights into customer behavior. The goal is to support data-driven decision-making for managers, marketers, and analytics teams, providing actionable intelligence about both customers and potential competitors.

II. Users' requirements

1. Data requirements

Required data includes **review information** (text, star rating), **users' information** (check-in frequency, number of reviews, etc.), **business information** (of their own company and other organizations publicly on Yelp). The information should be able to link back to the correct user/reviewer and business, and no duplicates should be included.

2. Functional requirements

Data should be systematically organized into well-defined relations that capture the logical associations among different entities within the system. Establishing clear relational structures not only enhances data integrity but also facilitates efficient querying, maintenance, and scalability of the database. Each relation should be thoroughly documented, including attribute descriptions, data types, and constraints, to ensure precise interpretation and reproducibility of analytical outcomes.

From a technical perspective, database functions must be implemented to operate reliably and efficiently, supporting the full cycle of data management activities. Depending on the level of user authorization, the system should allow data insertion, updating, importing, and exporting, ensuring flexibility while maintaining data security and governance. Moreover, the database should be capable of supporting both standardized and ad-hoc analyses, enabling users to generate reports that address specific managerial or research needs.

3. User access & Security requirements

3.1 **Potential users**

The database is designed for four main user groups with distinct levels of access. The first group includes the Marketing and Data Analytics teams, who directly utilize the data to monitor performance, generate reports, and conduct analytical tasks. The second group consists of upper-level managers, whose access focuses on reviewing summarized insights to support strategic decision-making and ensure transparency in reported outcomes.

Administrative officers from Yelp serve as system administrators responsible for maintaining database integrity, performing updates, and overseeing compliance with data governance standards. This structured access framework ensures efficient data use while maintaining accuracy, security, and accountability across all user roles.

3.2 Access level

Access levels must be clearly defined to ensure data security, privacy, and integrity throughout the database system. Managers are authorized to view datasets related to their own businesses and to compare performance metrics with other businesses using aggregated user review data. The Marketing and Data Analytics teams are permitted to extract datasets for analytical purposes, including in-depth exploration and model development. However, all database users from local businesses—including managers, Marketing, and Data Analytics teams—are restricted from accessing any confidential user information belonging to other businesses, such as credit card details. To maintain ethical data handling within this class project, all confidential attributes will be excluded from the dataset provided to users. Yelp administrators hold the highest level of authorization and are responsible for maintaining data integrity through activities such as adding, updating, and removing records, as well as monitoring system access and potential security incidents to ensure ongoing compliance and database reliability.

3.3 Authentication

All database users must be assigned unique credentials and access the system through individual accounts to ensure accountability and traceability. Given that the database may encompass data from multiple businesses, user email addresses should be masked, and IP address exposure must be strictly prevented to protect user privacy. Only publicly available Yelp data will be utilized in this project, and no private or personally identifiable information from Yelp users will be stored or processed within the database. This approach ensures compliance with data protection standards and ethical research practices.

III. Business rules

General understanding : Each company has a separate review site on Yelp, so each **review** or **tip** value points to one business only.

USER – REVIEW (1:M)

- An user can write many reviews
- A review can be written by one user

USER – TIP (1:M)

- An user can write many tips
- A tip can be written by only one user

BUSINESS - REVIEW (1:M)

- A review can be written about one business
- A business can gather many reviews

BUSINESS - TIP (1:M)

- A tip can be given to a business only
- A business can receive many tips (from one or many users)

BUSINESS – CHECK-IN (1:M)

- A business can be checked in many times
- Each checked-in time is only for a business

IV. Major entities of the system & description

1. Major entities

USER : *This entity holds information about Yelp users*. Attributes include user ID, name, number of reviews written, the date the user joined Yelp, their list of friends, and the number of votes (useful, funny, cool) they have given. Other attributes considered include fan count, elite years, average rating across reviews, and counts of compliments received in different categories (hot, cute, photos, etc.).

REVIEW: *Review entity represents feedback written for a business*. Attributes are review ID, user ID, business ID, and details of the star rating given, the date, the full review text, and counts of votes (useful, funny, cool).

TIP: *Tips on Yelp are short pieces of advice from users*. Attributes are text, date, number of compliments, user ID, and business ID.

BUSINESS: *This entity holds content about businesses listed on Yelp*. The attributes that might be covered include business ID, name, address, city, state, postal code, latitude, longitude, average star rating, counts of reviews, and whether the business is open or not. It also stores information related to categories, opening hours, and business service attributes (like parking, takeout options).

CHECK-IN: *Visits to a business is recorded in this entity.* Business ID and list of timestamps when check-ins occurred are recorded here.

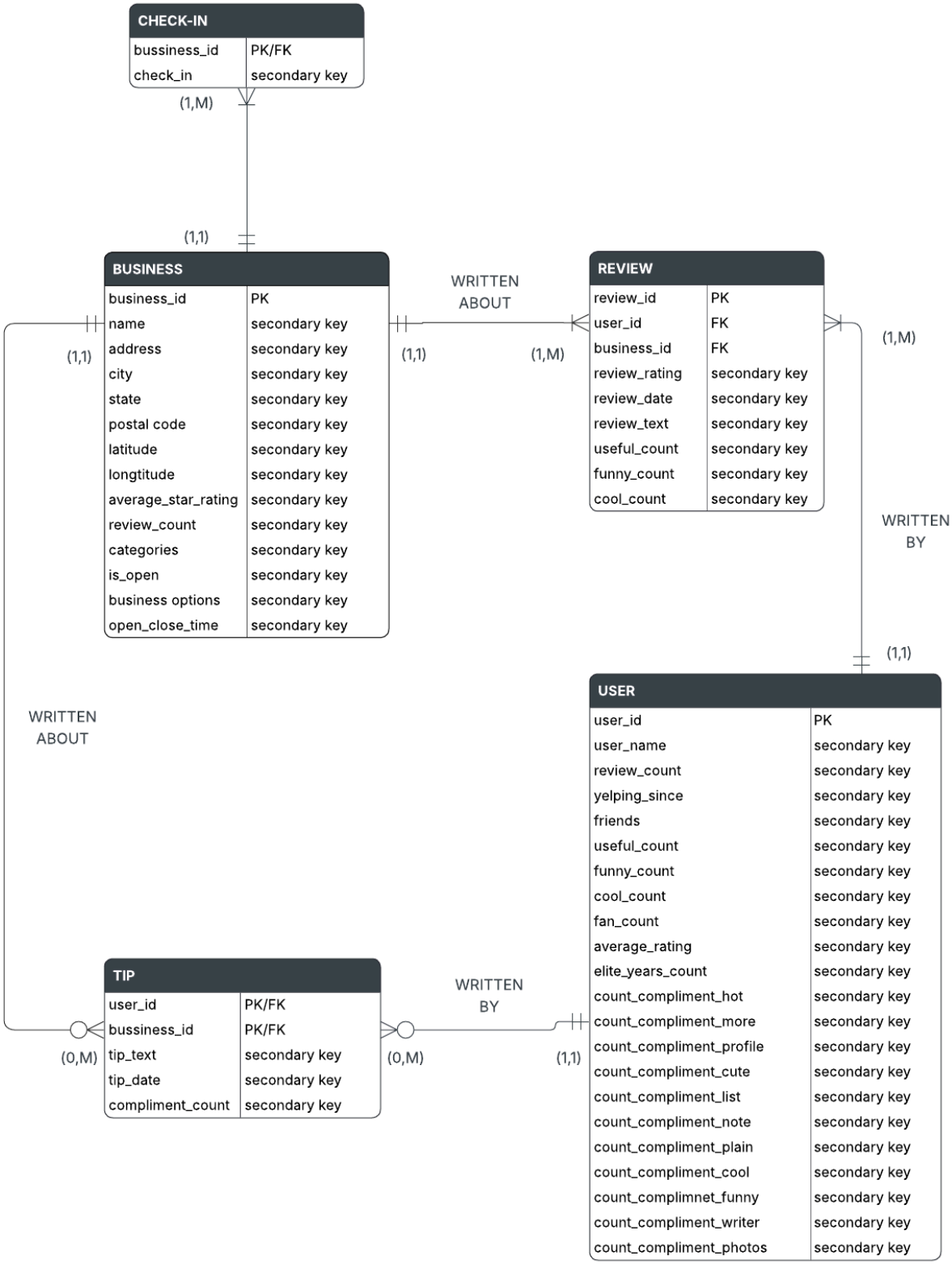
2. Detailed dictionary (Attributes' description)

The dictionaries for each entity are based on the dictionaries provided by Yelp for their public dataset. Attributes' names were modified to match the naming requirements. Data dictionary will include **Table name, Attribute name, Contents (Attribute description), Data type, Format, Key, Referenced table.** Further information of the dictionary will be adjusted as class project progress.

Detailed dictionaries are provided in Appendix 1, Appendix 2, Appendix 3, Appendix 4, and Appendix 5.

V. Entity Relationship Diagram (ERD) – Crow-foot Model

The ER Diagram is drawn using Lucid – chart service, a link to the original work is provided in Appendix 6


CHECK-IN
bussiness_id
PK/FK
check_in
secondary key
(1,M)
(1,1)
BUSINESS
business_id
PK
name
secondary key
address
secondary key
city
secondary key
state
secondary key
postal code
secondary key
latitude
secondary key
longtitude
secondary key
average_star_rating
secondary key
review_count
secondary key
categories
secondary key
is_open
secondary key
business options
secondary key
open_close_time
secondary key
(1,1)
WRITTEN ABOUT
(1,1)
(1,M)
REVIEW
review_id
PK
user_id
FK
business_id
FK
review_rating
secondary key
review_date
secondary key
review_text
secondary key
useful_count
secondary key
funny_count
secondary key
cool_count
secondary key
(1,M)
WRITTEN BY
(1,1)
USER
user_id
PK
user_name
secondary key
review_count
secondary key
yelping_since
secondary key
friends
secondary key
useful_count
secondary key
funny_count
secondary key
cool_count
secondary key
fan_count
secondary key
average_rating
secondary key
elite_years_count
secondary key
count_compliment_hot
secondary key
count_compliment_more
secondary key
count_compliment_profile
secondary key
count_compliment_cute
secondary key
count_compliment_list
secondary key
count_compliment_note
secondary key
count_compliment_plain
secondary key
count_compliment_cool
secondary key
count_complimnet_funny
secondary key
count_compliment_writer
secondary key
count_compliment_photos
secondary key
WRITTEN ABOUT
(0,M)
TIP
user_id
PK/FK
bussiness_id
PK/FK
tip_text
secondary key
tip_date
secondary key
compliment_count
secondary key
(0,M)
WRITTEN BY
(1,1)

VI. Database Implementation

Database and table definition statements

Based on the Data dictionary, Business Rules and Constraints, database named yelp_analysis was created using MySQL Workbench. The table definition statements are written in an sql file, attached separately from the final report.

Data insertion statements

To populate the MySQL database with the Yelp dataset, Python scripts were developed for each entity : USER, BUSINESS, REVIEW, TIP, CHECKIN.

The scripts were designed to read Yelp line-delimited JSON files and transform each record into its relational representation. Attribute names were mapped to the schema conventions defined in database design. Several non-relational attributes' structures in the entities, including friends, elite years (for USER), and business categories (BUSINESS), were encoded as JSON text where appropriate. Date and time fields, which are initially strings in the raw dataset, were converted into DATETIME format in MySQL.

As the source files are extremely large: USER (3.36 GB) and REVIEW (5.34 GB), conventional manual insertion using INSERT STATEMENTS is time-consuming and inefficient. Therefore, to ensure the reliability under this volume of data, the Python code was refined to execute the insertion task.

The code was refined to skip records with missing parent references, avoid duplicate inserts, and commit data in batches for efficiency. The scripts were also run in recommended referential order – USER, BUSINESS, REVIEW, TIP, and CHECKIN to maintain foreign-key constraints. This Python-based ETL process allowed the dataset to be inserted into MySQL while preserving the relational design of the database schema, as well as its integrity and performance.

The Python scripts includes large files, therefore the files are compressed and sent as Google drive link since the submission portal is unable to load the file:

https://drive.google.com/file/d/1x_jU0s0kBbsAEtgKpc-RUUIL_KN7X602/view?usp=sharing

MySQL sample queries (Plain English & SELECT statements)

The sample queries are attached separately in the SQL file.

Sample query 1

```sql
SELECT
    r.user_id,
    u.user_name,
    b.name AS business_name,
    COUNT(r.review_id) AS review_count_business_A
FROM review r
JOIN user u ON r.user_id = u.user_id
JOIN business b ON r.business_id = b.business_id
WHERE r.business_id = '_GGgSYM6yN3-2ZVxvp65HA'
GROUP BY r.user_id, u.user_name, b.name
LIMIT 10;
```

Plain English explanation: The query helps find the users who have written reviews for a specific business, and counts how many reviews each user has written for that business. This sample query involves joining three entities: REVIEW, USER, and BUSINESS to display the name of the review writer and the business. This query essentially helps managers see which users review their business and how active they are.

Result:

user_id	user_name	business_name	review_count_business...
kDJQgujsYnsSqOpwpMbiOg	Tarnisha	La Chancla	1
YigilpFMe1NTar61wFX8wg	Elaine	La Chancla	1
DBgwZXLdd5YK67ktxJ1R1A	Rich	La Chancla	1
IC8Uvs_vgsBkigqnwQeqag	Tom	La Chancla	1
QScz2C4EPz6VsFAZT0EDXQ	Antonina	La Chancla	1
3nmUGnCIR31g2SHweQkewg	Tran	La Chancla	1
WJu9RMMwJXInckCxmVim6w	Ryan	La Chancla	1
P2oU8BQvcMvJCiwk6Nz84w	John	La Chancla	1
D_0f1NhAdIW4kjbJPHKzWQ	Sheridan	La Chancla	2
IXIJsJxVnKAy27nJB7BoKw	Lawrence	La Chancla	1

Sample query 2

```
SELECT
    b.business_id,
    b.name,
    b.categories,
    b.city,
    COUNT(c.check_in) AS checkin_count
FROM BUSINESS b
LEFT JOIN CHECKIN c
    ON b.business_id = c.business_id
WHERE
    b.categories = (SELECT categories FROM business WHERE business_id = '_GGgSYM6yN3-2ZVxvp65HA') AND
    b.city = (SELECT city FROM business WHERE business_id = '_GGgSYM6yN3-2ZVxvp65HA')
GROUP BY b.business_id, b.name, b.categories, b.city
ORDER BY checkin_count DESC
LIMIT 10;
```

Plain English explanation: The query finds other businesses in the same city and in the same category as the chosen business (using business_id) and counts how many check-ins each one has. By filtering on category and city, the query helps managers to use their business_id to compare them with their local competitors. Ordering by the number of check-ins also helps identify which businesses are most visited among them.

Result:

business_id	name	categories	city	checkin_cou...
_GGgSYM6yN3-2ZVxvp65HA	La Chancla	"Restaurants, Mexican"	Norristown	1
-Rn6U-ySYo_Lx9vCgDSooA	Taqueria Mexico Lindo	"Restaurants, Mexican"	Norristown	1
ckc6K5-28tDo9-Uzw4RKaQ	El Puerto Jarocho	"Restaurants, Mexican"	Norristown	1
E01_x43vPxeOQ9MFbhSKDQ	Paraiso	"Restaurants, Mexican"	Norristown	1
EoQiJ5D-pyWczjEIN24oZg	El Primo Taqueria	"Restaurants, Mexican"	Norristown	1
F43loeSCbFosPOBUQwr4oA	El Rincon de Mexico	"Restaurants, Mexican"	Norristown	1
q-hpAt4iYH5pkGE32O8J0w	La Poblanita Mexican Bar	"Restaurants, Mexican"	Norristown	1
vlHYjFR8_hmmGzATiVSr-Q	TreJo American & Mexican Restaurant	"Restaurants, Mexican"	Norristown	1
x1CtK2qnlCr_1DJFNo1_vw	El Limon - Norristown	"Restaurants, Mexican"	Norristown	1
zo2p1GbzhtFZ9vKynK1Hiw	El Paraiso Mexicano	"Restaurants, Mexican"	Norristown	1

Sample query 3

```
SELECT
    user_id,
    business_id,
    tip_text,
    YEAR(tip_date) AS tip_year_entered,
    compliment_count
FROM tip
WHERE
    business_id = (SELECT business_id FROM business WHERE name = 'D\'Angelo\'s Bakery')
    AND YEAR(tip_date) = 2011
ORDER BY compliment_count DESC
LIMIT 10;
```

Plain English explanation:

This query shows the tips written for business (D'Angelo's Bakery) during the year 2011. It orders the tips by how many compliments they received, so the tips that were the most appreciated appear first. This helps businesses understand which tips received agreements among customers and make suitable adjustments to improve business performance accordingly.

Result:

	user_id	business_id	tip_text	tip_year_enter...	compliment_count
	__3gdV_ALx9QQzdXWHHUew	YrNtBUOUOYwmRZ_UVwH8iQ	Tri tip omelet is fantastic!	2011	0
	__3gdV_ALx9QQzdXWHHUew	YrNtBUOUOYwmRZ_UVwH8iQ	Eggs rose. Best breakfast in SB.	2011	0
	2hMwl_rbESY9-79RWJMx7w	YrNtBUOUOYwmRZ_UVwH8iQ	Eggs "Rose"	2011	0
	3GJQr9b-FN0uCmwy-pqVNQ	YrNtBUOUOYwmRZ_UVwH8iQ	good food, mediocre service.	2011	0
	8ThZxs4A8RZSkQTnHlvBzw	YrNtBUOUOYwmRZ_UVwH8iQ	Here nice and early (830) on Sunday and no wa...	2011	0
	edO_pZxGW0MrCQH3am7SZQ	YrNtBUOUOYwmRZ_UVwH8iQ	love their chocolate croissants!	2011	0
	gHXnebmJvPY-eYbrEVXiKw	YrNtBUOUOYwmRZ_UVwH8iQ	Waiting for table. Painful.	2011	0
	lfgQ_Wcn8SNifDi-hv5zEg	YrNtBUOUOYwmRZ_UVwH8iQ	Danish Breakfast :)	2011	0
	LtmqydFC5-6ll2Gn_i486w	YrNtBUOUOYwmRZ_UVwH8iQ	There's no meat here unless u want to spend a l...	2011	0
	m259Y2a9ohsgFn17FwC_LA	YrNtBUOUOYwmRZ_UVwH8iQ	Crowded seat yourself Saturday's	2011	0

VII. Database Evaluation Plan

Objective: The goal is to cover evaluation methods that help ensure the integrity, security, performance, scalability, and maintainability of the database.

Evaluation criteria

1. **Data integrity and consistency**

The main purpose of this is to guarantee that the database follows the defined schema and constraints, including primary keys, foreign keys, and unique constraints.

Assessment method:

- Verify that foreign key relationships are enforced. For instance, ensure that all REVIEW.user_id exist in the USER table.
- Check for duplicates in entities that enforce unique constraints: check for user_id, review_id, business_id duplicates if they exist

2. **Data security**

Data stored in the database must be protected from access by unauthorized users.

Assessment method:

Recommended by Carlos Coronel, Steven Morris in "Database Systems Design, Implementation, Management", testing can be done on different aspects of security. Applied in this specific domain, testing can be performed on :

- Password security: Enforced at login time at the operating system level
- Access rights: Test the restrictions on operating statements and predetermined objects, such as databases or views
- Audit trails: Provided by the DBMS to check for access violations
- Diskless workstations: Allow users to access without being able to download the information

3. **Query Accuracy**

This criteria confirm SQL queries return correct and expected results.

Assessment method:

- Create test cases for queries related to users' interests, such as: top reviewers for a business, businesses with the highest number of check-ins
- Take a small subset of the dataset and compare query results with the expected outcomes

4. **Performance and Efficiency**

Test the efficiency of data retrieval and insertion operations

Assessment method:

- Test key operations and measure time running, for example with join function
- Test another batch insertion performance for large dataset, can be around 3 - 5 GB like the dataset for REVIEW and USER

5. **Scalability and Maintainability**

Ensure that the database design supports growth in data volume and is easy to maintain.

Assessment method:

- Review the schema design to check naming conventions, documentation for clear future maintenance
- Evaluate database design and normalization process to check for redundancy and support storage

VIII. Database Administration

To support database administration, 3 stored programs were created in MySQL Workbench to automate monitoring and maintenance tasks.

The first procedure was written to report database statistics, including table sizes and total row counts. This procedure would help the administrator inspect the storage usage and identify the fast-growing tables

As REVIEW and USER are two entities that grow rapidly, the second routine was implemented to automatically back up the database by running a system-level mysqldump command and exporting data to timestamped backup files. The main purpose of this procedure protect Yelp from accidental data loss.

The third procedure was written to be an event scheduler, which automates the process of running these procedures at defined intervals.

The stored procedures are written and attached separately in a SQL file.

IX. Reflection & Summary

This project explores fundamental database design and prototyping techniques using a real dataset from the retail domain. The work moves from conceptual design, which includes developing Crow's Foot data models, defining entities and business rules, and specifying relationships and cardinalities, toward logical design and full implementation in MySQL Workbench. The project includes the creation of database structures (tables, constraints, and schema definitions) and operational SQL, such as CREATE statements and practical query examples written in both natural language and SQL. To support real-world data loading, Python scripts were developed to automate large-scale insertion and ensure proper handling of data quality and dependencies.

Across the course timeline, the weekly lectures guided the gradual progression from theory to implementation, showing how design choices translate into usable database systems. Working with real Yelp retail data highlighted the importance of well-structured databases for data management in modern business settings and provided hands-on experience with the technical and conceptual skills required for professional database construction.

X. Appendix

APPENDIX 1

USER relation

Attribute name	Data type	Attribute description	Format	Key	Referenced table
user_id	String	22-character unique user id		PK	
user_name	String	User's first name	Xxxxxxxx		
review_count	Integer	The number of reviews user has written	999		
yelping_since	Datetime	When the user joined Yelp	YYYY-MM-DD		
friends	Array of str	Array of user_ids of user's friends			
userful_count	Integer	Number of useful voted **sent by** the user	999		
funny_count	Integer	Number of funny votes sent by the user	999		
cool_count	Integer	Number of cool votes sent by the user	999		
fan_count	Interger	Number of fans the user has	999		
average_rating	Float	The average rating of all reviews	9.9		
elite_years_count	Array of int	Array of years the user was elite	99		
count_compliment_hot	Integer	Number of hot compliments received by the user	999		
count_compliment_more	Integer	Number of more compliments received by the user	999		

count_compliment_profile	Integer	Number of profile compliment received by the user	999		
count_compliment_cute	Integer	Number of cute compliment received by the user	999		
count_compliment_list	Integer	Number of list compliment received by the user	999		
count_compliment_note	Integer	Number of note compliment received by the user	999		
count_compliment_plain	Integer	Number of plain compliment received by the user	999		
count_compliment_cool	Integer	Number of cool compliment received by the user	999		
count_compliment_funny	Integer	Number of funny compliment received by the user	999		
count_compliment_writer	Integer	Number of writer compliments received by users	999		
count_compliment_photos	Interger	Number of photo compliments received by users	999		

APPENDIX 2

BUSINESS relation

Attribute name	Data type	Attribute description	Format	Key	Referenced table
business_id	String	22 character unique string business id		PK	
name	String	The business's name	Xxxxxxxxx		
address	String	The full address of the business	Xxxxxxxxx		
city	String	The city	Xxxxxxxxx		
state	String	The state	Xxxxxxxxx		
postal_code	String	The postal code	99999		
latitude	Float	Latitude	99.9999999		
longtitude	Float	Longtitude	99.9999999		
average_star_rating	Float	Average star rating, rounded to half-stars	9.9		
review_count	Integer	Number of reviews	999		
categories	Array of str	Array of strings of business categories			
is_open	Integer	0 for closed, 1 for open			
business_options	Object	Business extra options : garage, lot, valet,..			
open_close_time	object	Object of key day to value hours, hours using 24 hour clock. Example : "Monday": "10:00 – 21:00"			

APPENDIX 3

REVIEW relation

Attribute name	Data type	Attribute description	Format	Key	Referenced table
review_id	String	22 character unique review id		PK	
user_id	String	22 character unique user id		FK	USER table
business_id	String	22 character unique business id		FK	BUSINESS table
review_rating	Integer	Star rating of the review	9.9		
review_date	String	Date of the review	YYYY-MM-DD		
review_text	String	The text of the review			
useful_count	Integer	Number of useful votes received	99		
funny_count	Integer	Number of funny votes received	99		
cool_count	Integer	Number of cool votes received	99		

APPENDIX 4

TIP relation

Attribute name	Data type	Attribute description	Format	Key	Referenced table
user_id	String	22 character unique user id		PK/FK	USER table
business_id	String	22 character unique business id		PK/FK	BUSINESS table
tip_text	String	Text of the tip			
tip_date	String	Date the tip was written	YYYY-MM-DD		
compliment_count	Integer	How many compliments it has	999		

APPENDIX 5

TIP relation

Attribute name	Data type	Attribute description	Format	Key	Referenced table
business_id	String	22 character unique business id		PK/FK	USER table
check_in	String	List of time stamps for each check in	YYYY-MM-DD HH:MM:SS		BUSINESS table

APPENDIX 6

Original work of ER Diagram :
https://lucid.app/lucidchart/373df8b0-2c91-44b6-826c-0d6f7c161637/edit?viewport_loc=-102%2C-364%2C2724%2C1170%2C0_0&invitationId=inv_eb2327d6-20bf-489b-b55c-584fcb887795

XI. Citation

Yelp Press. (2025, April). *Yelp News room-Fast facts*. Retrieved from Yelp Newsroom: https://www.yelp-press.com/company/fast-facts/default.aspx