DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Project Description – Part 2 Description

CS-322 Introduction to Database Systems
Spring 2023

Table of Contents

Summary

This document describes the starting point for *part 2* of the project and supplements the initial description.

In **part 1**, based on the raw data, you have designed an ER model, translated it into a relational model, and discussed the data cleaning and data-driven decisions you had to take and potentially simplify your ER model accordingly.

In **part 2**, we provide the ER model, relational schema, DDL, and the corresponding data to load, as well as the queries to write. This document explains the elements we provide as the common starting point for every group. This document should also be considered educational and instructional, and we hope you find it useful.

READ THE DOCUMENT FULLY AND CAREFULLY.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Introduction

This document presents the ER model, some data-cleaning decisions, and reasoning for the model simplifications. The purpose of this document is to serve as a starting point for **project part 2** and to be instructive and educative regarding some common practices that we will explain.

In this document, we provide the following:

1. ER model,
2. SQL DDL corresponding to the relational schema – that you will use to make your database on our DBMS,
3. Link to download the data corresponding to the relational schema – that you will load into our DBMS,
4. Queries organized in two groups for you to write and run.

The link to download the elements above is here: https://drive.switch.ch/index.php/s/QktuztHrbiEMyaD (.zip file, 480.9MB)

NOTE: This is not the only, best, or ideal solution – it is one of the possible solutions. Modeling often requires simplifications and adapting to a particular use case of existing entities and relationships that can be captured in a best effort by the data related to some real-world process or informed by domain experts.

ER Model

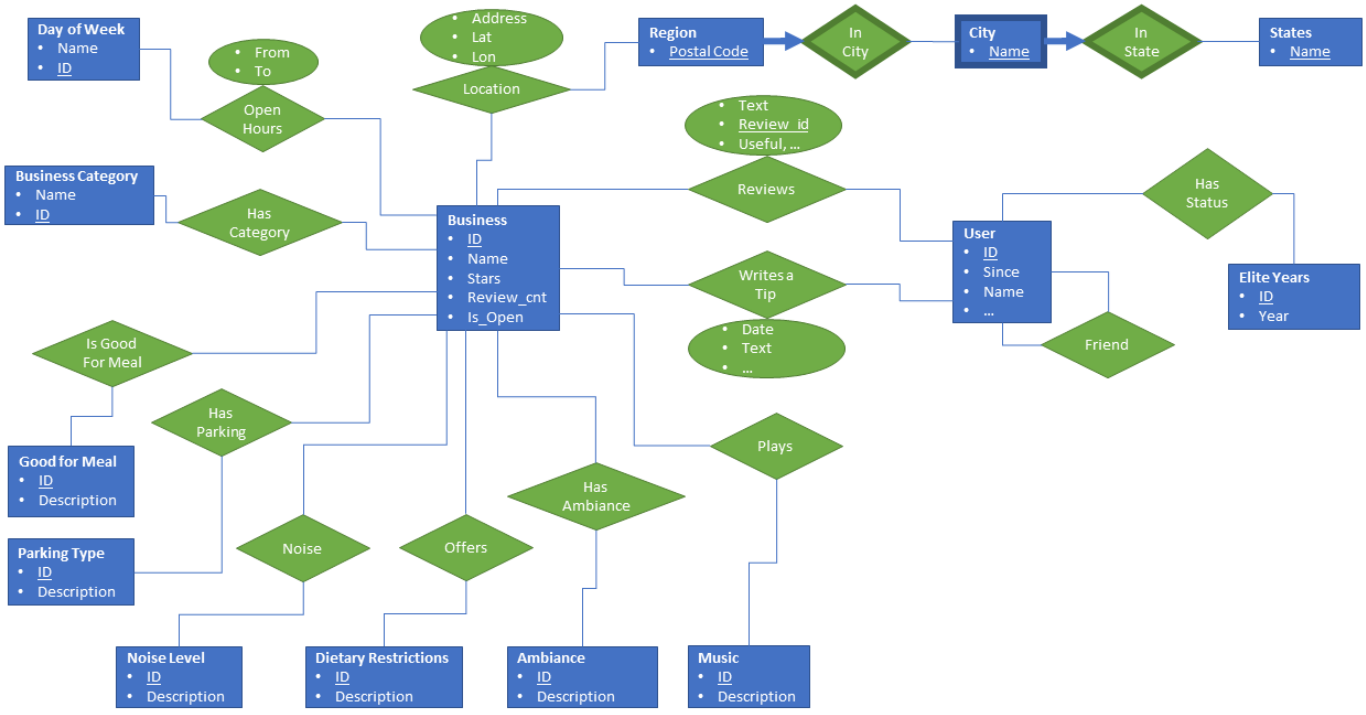

Figure 1. ER Model overview, simplified, without all the attributes and the keys

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

One common design for ER model schema is a snowflake schema – where fact tables (that contain and connect the information) and dimension tables (attributes or sets of attributes grouped as entities further describing the data) are normalized and organized so they resemble a snowflake. The effect of such design is normalization – where data is split up to avoid redundancy but introducing the need for joins. Formally, you can check out normalization and normal forms. 1st normal form is the basis of the relational model: each table column must have a single value, and sets of values or nested records are not allowed.

DDL

In the .zip file, you can find the SQL DDL commands corresponding to the relational schema. Remember the constraints we define – especially the foreign key constraints. You cannot define a constraint to the table or column you have not defined yet. For this reason, sometimes it is necessary to first define the tables without some constraints (not inline, immediately in the *create table* statement) and then alter the table subsequently to add those constraints.

NOTE: Do not modify the provided table and column names – this will result in a grading penalty.

The connection information, except the username and password we will provide, are the same as in the first weeks – you should have connected already before and have the environment set up and tested, as this was the topic of the first weeks and practical SQL exercises – do not forget to use EPFL VPN if not on EPFL network.

Every group has a separate username and password that you must use for your project work. In case you change your password, please get in touch with your assigned TA. We can monitor the DBMS activity, so please do not log into or modify *other groups'* databases, as this will result in disciplinary action.

hostname: cs322-db.epfl.ch, **port:** 1521, **SID:** ORCLCDB
your group's username: C##DB2023_G[groupid]
your group's password: ORCL23_G[groupid]

If the group number is lower than 10, prefix it with 0; e.g., group 1 is 01.
For example, for group 1, the credentials are: username: C##DB2023_G01, password: ORCL23_G01

Data loading

The specified constraints guarantee that modifying the tables by inserting, updating, and deleting values and rows will preserve the integrity of the data. Checking if all the constraints (for example, foreign key constraints) hold is not an operation that comes for free – and if one is *bulk loading* the data, disabling the constraints temporarily can improve the performance of inserting large amounts of data at once (for example, from files). However, finally, the data must match the required constraints, which need to be enabled again – which will fail if your data is inconsistent (for example, the field you defined as a primary key appears multiple times). This is why this mechanism should not be misused and should be used with care.

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

NOTE: When inserting or modifying data, make sure you *commit* the changes to the DBMS to see the changes.

However, sometimes we must temporarily disable or defer constraint checks. Imagine two tables referencing one another: inserting a row into one that does not exist in the other is impossible, and vice versa. We provide you with the PL/SQL code – a procedural Oracle SQL extension – for disabling constraints. You can select the command and run it in the terminal as any other SQL command. **You can run this command before you start loading the files to the defined tables – it auto-generates and runs *alter table* queries to disable constraints.**

```
BEGIN
  FOR c IN
  (SELECT c.owner, c.table_name, c.constraint_name
   FROM user_constraints c, user_tables t
   WHERE c.table_name = t.table_name
   AND c.status = 'ENABLED'
   AND NOT (t.iot_type IS NOT NULL AND c.constraint_type = 'P')
   ORDER BY c.constraint_type DESC)
  LOOP
    dbms_utility.exec_ddl_statement('alter table "' || c.owner || '"."' || c.table_name || '"
disable constraint ' || c.constraint_name);
  END LOOP;
END;
```

Conversely, the following code snippet enables all the disabled constraints. However, if the loaded data does not satisfy the constraints, you will get an error that DBMS cannot enable (certain) constraints. Then you need to, based on the error and the constraint, figure out and fix the data such that the constraints hold.

After you have finished loading the data, <u>you must enable all the constraints – else the data loading part is not considered complete.</u>

```
BEGIN
  FOR c IN
  (SELECT c.owner, c.table_name, c.constraint_name
   FROM user_constraints c, user_tables t
   WHERE c.table_name = t.table_name
   AND c.status = 'DISABLED'
   ORDER BY c.constraint_type)
  LOOP
    dbms_utility.exec_ddl_statement('alter table "' || c.owner || '"."' || c.table_name || '" enable
constraint ' || c.constraint_name);
  END LOOP;
END;
```

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Ultimately, the goal is to load the data files into the corresponding tables. We have tested the data loading and will provide the expected number of rows each table will have – so you are certain that you have loaded the data correctly. We provide you with both CSV and pipe-separated files. These are common file formats that most IDEs know how to process (if you use their GUI for loading the data to a table). You must handle the case if there is a header, if you need to handle quotations or pipes. **You should not modify the data at all** – you need to explore the available options and make sure you know how to use options that your development tools provide.

TABLE NAME	*COUNT*
AMBIANCE	9
BUSINESS	192609
BUSINESS_AMBIANCE	30739
BUSINESS_CATEGORIES	1300
BUSINESS_DIETARY_RESTRICTIONS	74
BUSINESS_GOOD_FOR_MEAL	41047
BUSINESS_HAS_CATEGORIES	788280
BUSINESS_HOURS	932626
BUSINESS_LOCATION	191950
BUSINESS_MUSIC	2759
BUSINESS_NOISE_LEVEL	43806
BUSINESS_PARKING_TYPE	69363
CITIES	1149
DAY_OF_WEEK	7
DIETARY_RESTRICTIONS	8
ELITE_YEARS	13
FRIENDS	14784576
GOOD_FOR_MEAL	6
MUSIC	8
NOISE_LEVEL	4
PARKING_TYPE	5
REGIONS	20186
REVIEWS	918679
STATES	36
TIPS	~~1029047~~ **1004279**
USER_ELITE	233177
USER_YELP	778651

The data must be loaded into the DBMS using your group's credentials. We will check this!
A brief reminder that data loading and implementation carries 5 points (from project description)

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Queries

After loading the data and verifying the row counts of the table, you can proceed with the queries, which we split into two categories: Initial insights (1.5 points per query) and Advanced insights (3.5 points per query).

Make sure to follow the query specification carefully with the attributes, order, and number of the results. Write each query in a separate .sql file. For initial insights, use the prefix E, for example, E_1.sql is the 1st query. For advanced insights, please use the prefix D, for example, D_5.sql is the 5th query. Then, save these files in a folder named G[groupId], e.g. G05. You must follow this file naming convention, or else you will get penalized.

Initial insights (15 points)

1. Find the businesses (name, review_count) in the city of 'las vegas' that have 'valet' parking, 5 stars and are open on Friday. Order by the business name in alphabetical order.

2. Find the top-10 (according to the number of stars) businesses in the state of California. For every business list the business name and the number of stars (business_name, stars). Sort the result in descending order according to the number of stars, and break tie by the business names in alphabetical order.
Note that states are stored by their abbreviation.

3. Find the ids (business_id) of the businesses that have been reviewed by more than 1030 unique users. List only the ids and sort by ascending order.

4. Find all businesses that have more than 3000 reviews and more than 2 dietary restriction categories. For every business list the business id, the business name and the review count (business_id, business_name, review_count). Order the results in descending order according to the number of reviews.

5. Find all users that started using Yelp at 2006 or before and their usernames consist of only one character. List their user ids, usernames and the date they started using Yelp (user_id, user_name, yelping_since). Sort the result in alphabetical order of usernames. Limit the number of returned rows to 50.

6. Find the maximum number of different businesses that have been reviewed by a single user. Return one column "count".

7. For each state, find the number of distinct businesses having the tag "vegetarian". List the state name and the number of businesses (state_name, business_count). Order by the number of businesses in descending order.

8. Find the minimum, maximum, mean, and median number of categories per business. Return 4 columns (min_categories, max_categories, mean_categories, median_categories).

9. What is the maximum number of categories assigned to a business? Return one column "count".

10. How many businesses labeled as "Dry Cleaners" are open on the weekend? Return one column "count".

DIAS: Data-Intensive Applications and Systems Laboratory
School of Computer and Communication Sciences
Ecole Polytechnique Fédérale de Lausanne
Building BC, Station 14
CH-1015 Lausanne
URL: http://dias.epfl.ch/

Advanced insights (35 points)

1. Find the cities where all businesses work less than 5 days a week. List only the names of the cities (city_name) and sort the results in alphabetical order.

2. Find the top-10 states with the highest number of registered businesses. List only the state and the number of businesses (state_name, num_businesses) and sort the results in descending order according to the number of businesses.

3. List the name, number of stars, and review count (business_name, stars, review_count) of the businesses that are in the category 'Irish Pub' and offer 'live' music. Sort in alphabetical order of the business names.

4. Find all cities that satisfy the following: each business in the city has at least two reviews. List only the names of the cities (city_name) in alphabetical order. Limit the number of returned rows to 50.

5. Find the average rating and the total number of reviews for all businesses which have a minimum of two categories and more than one parking type. Return two columns (stars, review_count).

6. Find the number of businesses for which every user that gave the business a positive tip (containing 'awesome') has also given some business a positive tip within the previous day. Return one column "count".

7. Compute the difference between the average stars of businesses considered 'good for dinner' with a (1) "divey" and (2) an "upscale" ambience. Name the column of the result 'DIFFERENCE_OF_AVERAGES'.

8. Find the names of the cities that satisfy the following: the combined number of reviews for the top-100 (wrt to reviews) businesses in the city is at least double the combined number of reviews for the rest of the businesses in the city. Return one column 'city'.

9. For each of the top-10 (in terms of number of reviews) businesses, find the top-3 reviewers: among those who reviewed the business, the ones that have the three highest numbers of total reviews across all businesses. Return 3 columns: the business id (as business_id), rank (1-3) of the top reviewer (as reviewer_rank) and the number of reviews made (as review_count). Order the results by business id in alphabetical order, and then rank (1 - 3).

10. Find the city whose businesses have the lowest combined number of reviews; only reviews from users with less than 3 friends should be taken into account. If there is a tie, choose the first city by alphabetical order. Return 2 columns, the number of reviews (as review_count) and the city name (as city_name).

Reminder: query optimization and query plan analysis carry 10 points and is a separate task that needs to be delivered as a short writeup (see project description!). We will also write a short recap as a Moodle announcement on how you should deliver Project Part 2 to avoid any confusion!

GOOD LUCK!