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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Databases Project – Spring 2017

Prof. Anastasia Ailamaki

Team No: 21

Names: Marc Bickel, Valentin Moullet, Quentin Laville

Contents

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/

Deliverable 1

Assumptions

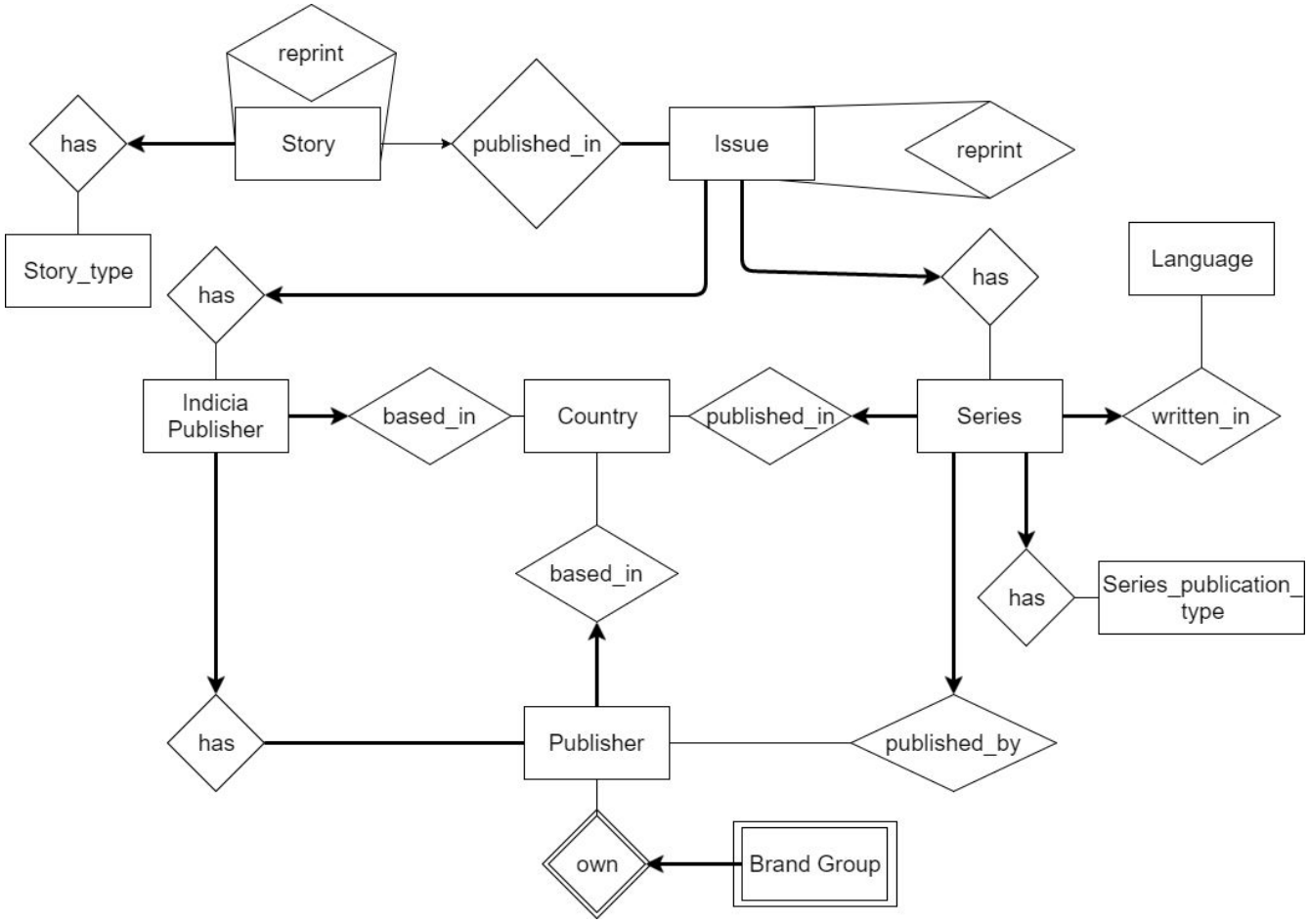
- For stories, we assume that we can have some story without any issue, so issue_id can be NULL.
- For series, we assume that we can have some series without any last issue (at least for the moment), so last_issue_id can be NULL.
- For all the other fields for which we could ask ourselves if it can be NULL or not, we assume they have to be NOT NULL.

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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Entity Relationship Schema

Schema



Description

Our ER schema follows is pretty straight-forward. The main point that needs to be discussed is the one about brand group. We think that this entity is the only one that we consider as a weak entity: it cannot exist without a publisher.

Relational Schema

ER schema to Relational schema

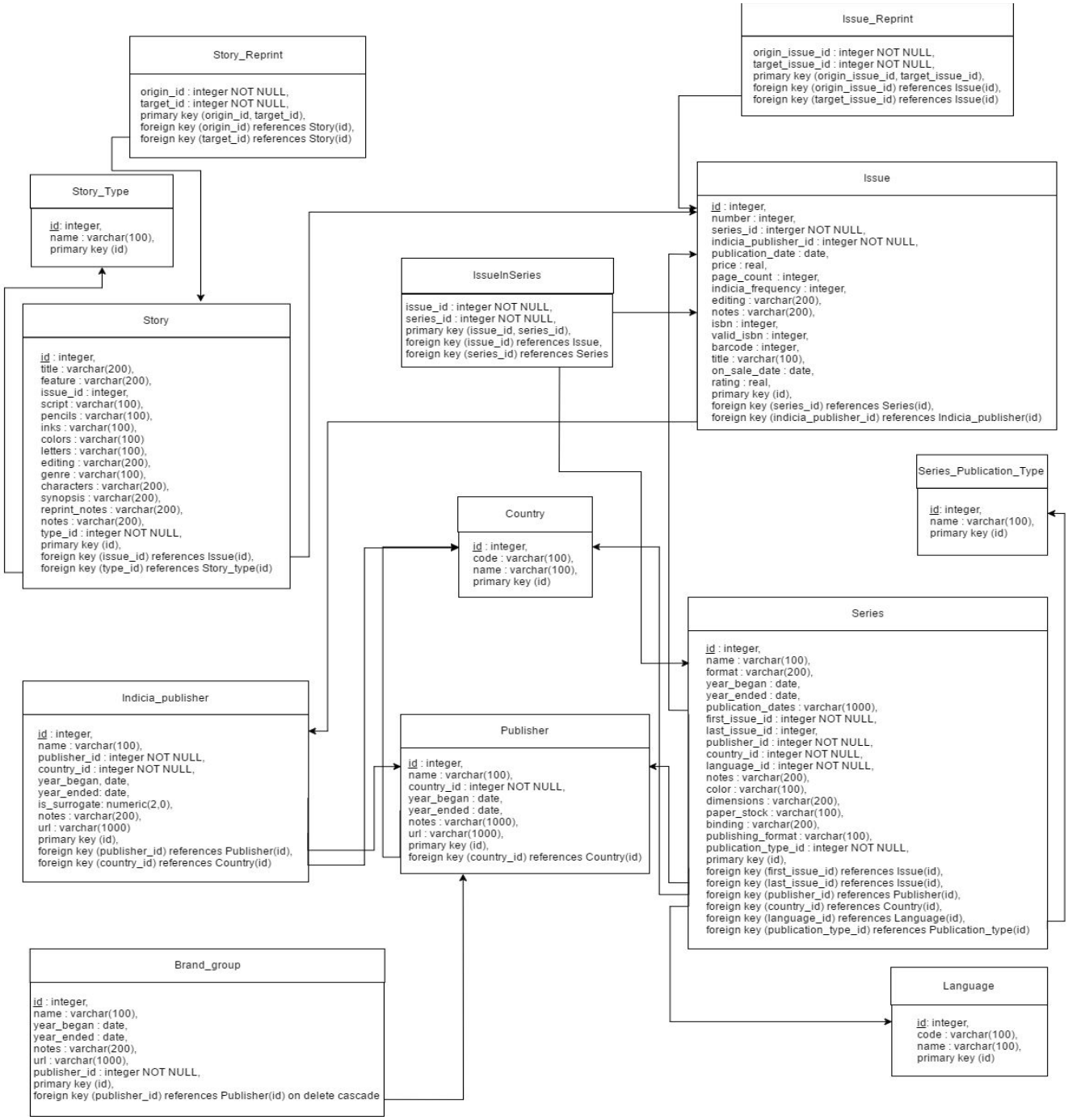
The transition is also pretty straight-forward. Every entity will become a table with some foreign key(s) corresponding to the one(s) they are linked to with a relation. The only table that needed to be created so that

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/

the Relational schema coincides with the ER schema while being correct is the table IssueInSeries. We need this table because otherwise we would have a cycle in our dependencies: Issue references to Series through series_id and Series to Issue through first_issue_id and last_issue_id. With this new table that contains a tuple for (issue_id, series_id), we still have the information about which issue is within which series, but we removed the cycle because we can now remove the field series_id in Issue.

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/

DDL



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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

General Comments

From the Relational schema, it is straight-forward to get the SQL needed to create the corresponding tables.

We would be glad to have comments on our solution!

Deliverable 2

Our GitHub: https://github.com/Gangstere44/cs322proj

Assumptions

No direct assumption were made. We wanted to assume the data was clean, but obviously it wasn't AT ALL. We had quite a hard time to clean it, and are still not done now. For example, some fields that are supposed to be numbers (price for example), contain thing that are not numbers (e.g. "0.00 FREE", or "2.10 USD"). In those cases, we want to put them at null if is not in the format we are looking for. The goal would be, for every field that contains some "uncleaned" data, to find what is the norm and to drop everything that is not in the same format. Is it a good solution?
Also, we didn't have time to import everything in the DB (the one you set up for us), because when we tried to insert all the lines, it was taking too long. We are using INSERT ALL (with 1000 lines each time) to insert data in the table we created, but there is so much data to insert that it takes too much time. Is it normal? What would be a better solution?

Data Loading

Query Implementation

<For each query>

Query a:

Description of logic:

Looks up the name of brand_groups where certain conditions are met

SQL statement

SELECT BG.name FROM indicia_publisher IP, publisher P, brand_group BG, country C WHERE MAX(COUNT(C.name = "Belgium" AND C.id = IP.country_id AND BG.publisher_id = IP.publisher_id))

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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query b:

Description of logic:

Get the danish series

SQL statement

SELECT P.id, P.name FROM publisher P, country C, series s WHERE c.name = 'Danemark' AND c.id = S.country_id AND P.id = S.publisher_id

Query c:

Description of logic:

Nothing hard to see

SQL statement

SELECT S.name FROM series S, series_publication_type SPT, country C WHERE S.country_id = C.id AND C.name = 'Switzerland' AND SPT.id = S.publication_type_id AND SPT.name = 'magazine'

Query d:

Description of logic:

We group the issues by year having year >= 1990, and then we count them

SQL statement

SELECT COUNT(I.number) FROM issue I GROUP BY (EXTRACT year FROM I.publication_date) HAVING (EXTRACT year FROM I.on_sale_date) >= 1990

Query e:

Description of logic:

We look for everything that contains, at some point, the string DC Comics.

SQL statement

SELECT COUNT(series) FROM indicia_publishers IP GROUP BY IP.name HAVING IP.name LIKE '%DC Comics%'

Query f:

Description of logic:

Simply find the max of the count of ids appearing as origin in story_reprint. This shows the most reprinted, but we don't know how to do the 10 most.

SQL statement

SELECT S.title FROM story S, story_reprint SR WHERE MAX(COUNT(SR.origin_id)) AND S.id = SR.origin_id --only finds the most reprinted, no idea how to do the 10 most

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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Query g:

Description of logic:

We group all the tables by story_id, and we check that each role (that we focus on) is done by the same artist_id

SQL statement

SELECT DISTINCT A.name FROM Artist A, Story_to_Script STS, Story_to_Pencils STP, Story_to_Colors STC WHERE STS.artist_id = STP.artist_id AND STS.artist_id = STC.artist_id GROUP BY story_id

Query h:

Description of logic:

Not sure about this one. We find the stories such that Batman appears as in the table story_to_chracters, but not in the table story_to_feature.

SQL statement

SELECT S.* FROM story S, story_to_characters STC, story_reprint SR, character C, story_to_feature STF WHERE (S.id NOT IN SR.origin_id) AND C.name = 'Batman' AND C.id = STC.character_id AND (C.id NOT IN STF.feature_id) --not sure

Interface

Design logic Description

Our goal is to have a simple interface, really close to the one you describe in the Interface section. We will have a simple web page (HTML + CSS), and we will use JavaScript to allow interactions with it (tabs, button clicks, ...). To do SQL requests, we will send a request to a server running in Python with the description of the SQL request, the server will query the DB (we will be using the DB that you set up for us), and return the answer to the client in a JSON format. The client will then display the answer of the query to the user.
We are using the Django framework to help us doing that.

Screenshots

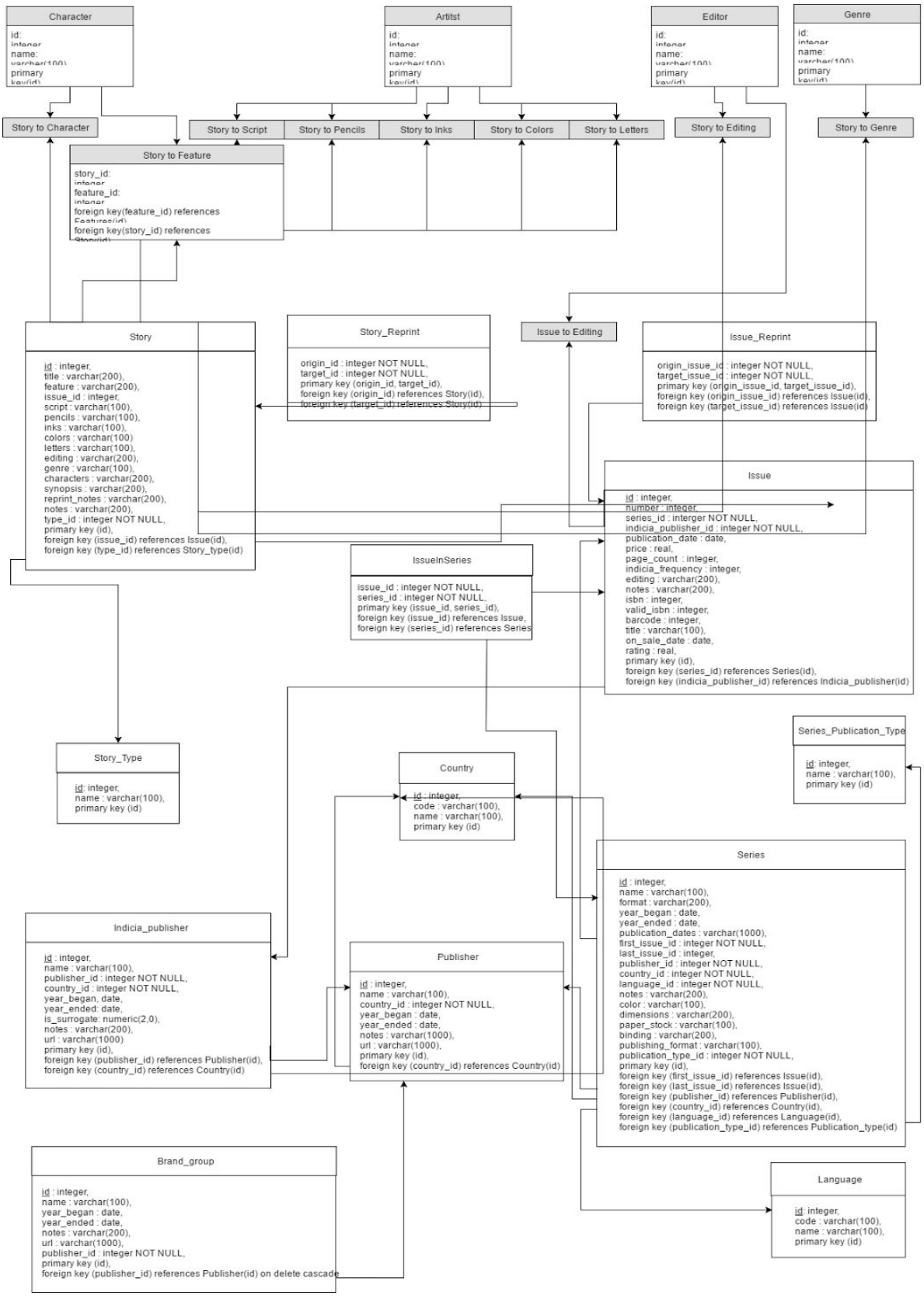
We do not have anything ready to show yet. Our server is created (we are using an EC2 instance from AWS so that we can have our webpage online and easily accessible). The design should be similar to the interface you described as an example.

General Comments

We modified the DDL in this milestone, based on the feedback we got plus what we understood, additionally to last time. The new relations/relationships have grey headers. Each Story_to_... relationship has the same structure: a story_id, and an character/artist/genre id, and it simply associates the two.

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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE



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/

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Deliverable 3

Assumptions

<In this section write down the assumptions you made about the data. Write a sentence for each assumption you made>

Query Implementation

<For each query>

Query a:

Description of logic:

<What does the query do and how do I decide to solve it>

SQL statement

<The SQL statement>

Query Analysis

Selected Queries (and why)

Query 1

<Initial Running time:
Optimized Running time:
Explain the improvement:
Initial plan
Improved plan>

Query 2

<Initial Running time:
Optimized Running time:
Explain the improvement:
Initial plan
Improved plan>

Query 3

<Initial Running time:
Optimized Running time:
Explain the improvement:
Initial plan
Improved plan>

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/

Interface

Design logic Description

<Describe the general logic of your design as well as the technology you decided to use>

Screenshots

<Provide some initial screen shots of your interface>

General Comments

<In this section write general comments about your deliverable (comments and work allocation between team members>