

Databases Project – Spring 2017

Prof. Anastasia Ailamaki

Team No: 40

Names: Léonard Berney, Pierre Fouché, Chenyu Shi

Contents

Contents	1
Deliverable 1.....	2
Assumptions	2
Entity Relationship Schema	3
Schema	3
Description.....	3
Relational Schema	4
ER schema to Relational schema	4
DDL	4
General Comments.....	11
Deliverable 2.....	12
Assumptions	12
Data Loading.....	12
Query Implementation	12
Query a:	12
Description of logic:.....	12
SQL statement	13
Interface	13
Design logic Description	13
Screenshots	13

General Comments.....	13
Deliverable 3.....	14
Assumptions	14
Query Implementation	14
Query a:	14
Description of logic:.....	14
SQL statement	14
Query Analysis	14
Selected Queries (and why).....	14
Query 1	14
Query 2	14
Query 3	14
Interface	15
General Comments.....	15

Deliverable 1

Assumptions

We tried to make the smallest amount of assumptions possible.

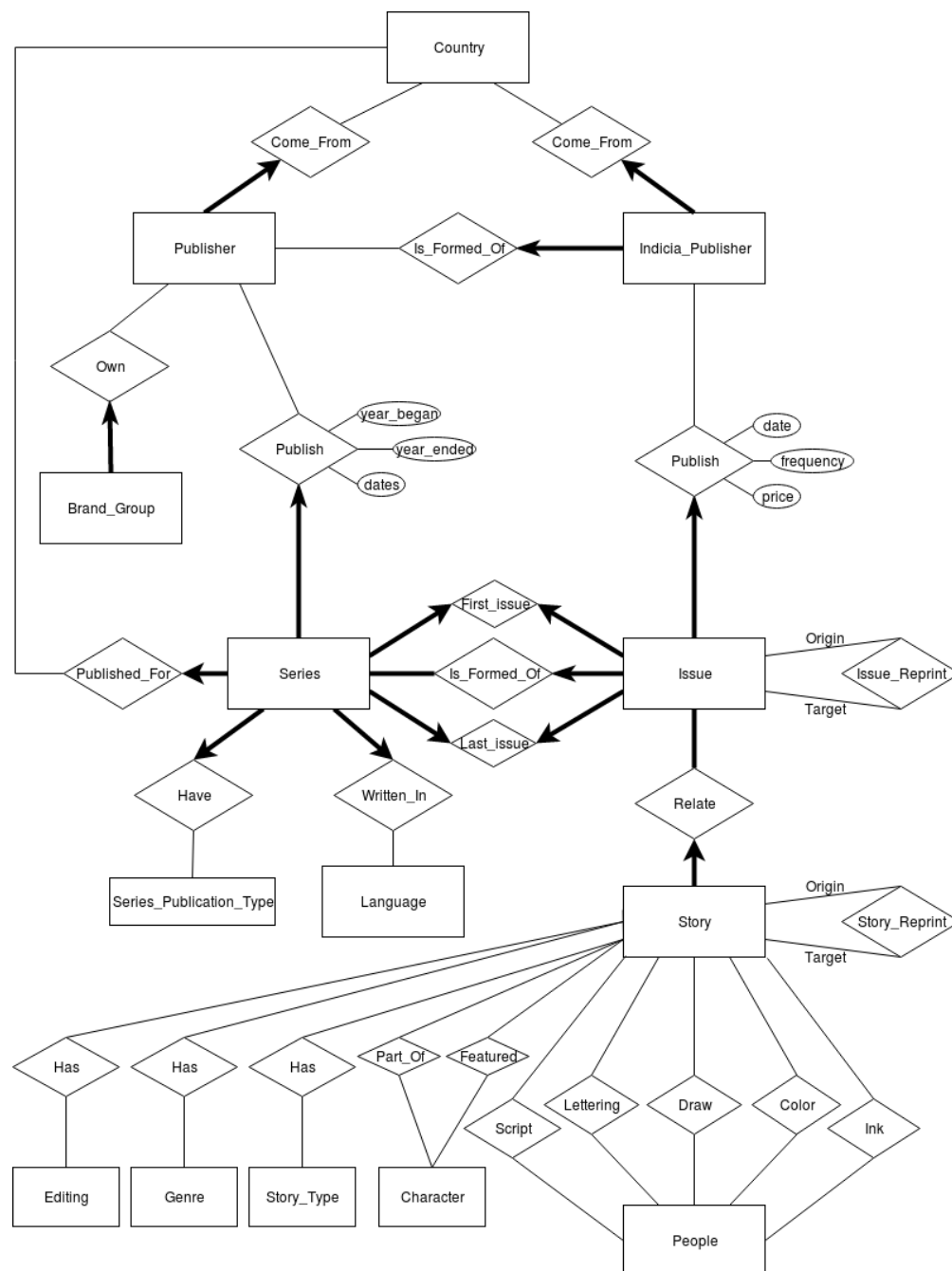
We made no assumption about any data being unique or non-null, except for the fields referring to IDs which we assumed to be either usable as primary keys or foreign keys. Countries and Languages were also an exception; because the dataset was small we figured out by looking at it the constraints that could be extracted.

One major assumption that we made (see next section) is that if two characters or people have the same name, we will consider them to be the same person or character.

Another assumption we did sometimes was to consider date columns (usually “year_started” and “year_ended”) as integer values, after seeing the data, we might have to reconsider this however.

Entity Relationship Schema

Schema



Description

We tried to stay as close to the provided data as possible because it will later be easier to import it in the database if we don't need to modify it too much. The provided data appeared to come from a database that was not normalized, so we modified it so that it was at least in the first normal form. The "Story" table contained multi-valued columns, the characters and the people who participated in making the story. We created two new entities, "characters" and "people". A character can be part of one or multiple stories, whereas a person can contribute in multiple ways to a story, for example by writing the script, drawing, coloring, etc. Other than that, we made "Issue_reprint" and "Story_reprint" weak entities because they cannot exist on their own. The other entities were extracted from the provided csv files in a pretty straightforward manner.

Relational Schema

ER schema to Relational schema

We created a table for every entity in the ER diagram. We had to create six additional tables to represent the many to many relationships between people, characters and stories.

You may notice that some relationships should have non-null foreign keys and other constraints. We intentionally did not include them in the DDL because we wanted to make it easier to import the data and avoid conflicts as much as possible. Later on we will add these missing constraints and filter out the invalid data.

DDL

```
DROP TABLE IF EXISTS `brand_groups`;
```

```
CREATE TABLE `brand_groups` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) DEFAULT NULL,  
  `year_began` int(11) DEFAULT NULL,  
  `year_ended` int(11) DEFAULT NULL,  
  `notes` text,  
  `url` text,  
  `publisher_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `_idx` (`publisher_id`),  
  CONSTRAINT `brand_groups_publishers_fk` FOREIGN KEY (`publisher_id`) REFERENCES  
  `publishers` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `characters`;
```

```
CREATE TABLE `characters` (  
  `name` varchar(255) NOT NULL,  
  `id` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  UNIQUE KEY `name_UNIQUE` (`name`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `countries`;
```

```
CREATE TABLE `countries` (  
  `id` int(11) NOT NULL,  
  `code` varchar(255) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `name_UNIQUE` (`name`),  
  UNIQUE KEY `code_UNIQUE` (`code`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `indicia_publishers`;
```

```
CREATE TABLE `indicia_publishers` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) DEFAULT NULL,  
  `publisher_id` int(11) DEFAULT NULL,  
  `country_id` int(11) DEFAULT NULL,  
  `year_began` int(11) DEFAULT NULL,  
  `year_ended` int(11) DEFAULT NULL,  
  `is_surrogate` tinyint(4) DEFAULT NULL,  
  `notes` text,  
  `url` text,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `indicia_publishers_countries_fk_idx` (`country_id`),  
  KEY `indicia_publishers_publishers_fk_idx` (`publisher_id`),  
  CONSTRAINT `indicia_publishers_countries_fk` FOREIGN KEY (`country_id`)  
REFERENCES `countries` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `indicia_publishers_publishers_fk` FOREIGN KEY (`publisher_id`)  
REFERENCES `publishers` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `issue_reprints`;
```

```
CREATE TABLE `issue_reprints` (  
  `id` int(11) NOT NULL,  
  `origin_issue_id` int(11) DEFAULT NULL,  
  `target_issue_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  KEY `issue_reprints_target_issues_fk_idx` (`target_issue_id`),  
  KEY `issue_reprints_origin_issues_fk_idx` (`origin_issue_id`),  
  CONSTRAINT `issue_reprints_origin_issues_fk` FOREIGN KEY (`origin_issue_id`)  
REFERENCES `issues` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `issue_reprints_target_issues_fk` FOREIGN KEY (`target_issue_id`)  
REFERENCES `issues` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `issues`;
```

```
CREATE TABLE `issues` (  
  `id` int(11) NOT NULL,  
  `number` varchar(255) DEFAULT NULL,  
  `series_id` int(11) DEFAULT NULL,  
  `indicia_publisher_id` int(11) DEFAULT NULL,  
  `publication_date` varchar(255) DEFAULT NULL,  
  `price` varchar(255) DEFAULT NULL,  
  `page_count` int(11) DEFAULT NULL,  
  `indicia_frequency` varchar(255) DEFAULT NULL,  
  `editing` text,  
  `notes` text,  
  `isbn` varchar(255) DEFAULT NULL,  
  `valid_isbn` varchar(255) DEFAULT NULL,  
  `barcode` varchar(255) DEFAULT NULL,  
  `title` varchar(255) DEFAULT NULL,  
  `on_sale_date` varchar(255) DEFAULT NULL,  
  `rating` varchar(255) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `issues_indicia_publishers_fk` (`indicia_publisher_id`),  
  KEY `issues_series_fk_idx` (`series_id`),  
  CONSTRAINT `issues_indicia_publishers_fk` FOREIGN KEY (`indicia_publisher_id`)  
REFERENCES `indicia_publishers` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `issues_series_fk` FOREIGN KEY (`series_id`) REFERENCES `series`  
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `languages`;
```

```
CREATE TABLE `languages` (  
  `id` int(11) NOT NULL,  
  `code` varchar(255) NOT NULL,  
  `name` varchar(255) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `name_UNIQUE` (`name`),  
  UNIQUE KEY `code_UNIQUE` (`code`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `people`;
```

```
CREATE TABLE `people` (  
  `first_name` varchar(255) DEFAULT NULL,  
  `last_name` varchar(255) DEFAULT NULL,  
  `id` int(11) NOT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `publishers`;
```

```
CREATE TABLE `publishers` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) DEFAULT NULL,  
  `country_id` int(11) DEFAULT NULL,  
  `year_began` int(11) DEFAULT NULL,  
  `year_ended` int(11) DEFAULT NULL,  
  `notes` text,  
  `url` text,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `publishers_countries_fk_idx` (`country_id`),  
  CONSTRAINT `publishers_countries_fk` FOREIGN KEY (`country_id`) REFERENCES  
  `countries` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `series`;
```

```
CREATE TABLE `series` (  
  `id` int(11) NOT NULL,  
  `name` varchar(255) DEFAULT NULL,  
  `format` varchar(255) DEFAULT NULL,  
  `year_began` int(11) DEFAULT NULL,  
  `year_ended` int(11) DEFAULT NULL,  
  `publication_dates` varchar(255) DEFAULT NULL,  
  `first_issue_id` int(11) DEFAULT NULL,  
  `last_issue_id` int(11) DEFAULT NULL,  
  `publisher_id` int(11) DEFAULT NULL,  
  `country_id` int(11) DEFAULT NULL,  
  `language_id` int(11) DEFAULT NULL,  
  `notes` text,  
  `color` varchar(255) DEFAULT NULL,  
  `dimensions` varchar(255) DEFAULT NULL,  
  `paper_stock` varchar(255) DEFAULT NULL,  
  `binding` varchar(255) DEFAULT NULL,  
  `publishing_format` varchar(255) DEFAULT NULL,  
  `publication_type_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `series_first_issues_fk_idx` (`first_issue_id`),  
  KEY `series_last_issues_fk_idx` (`last_issue_id`),  
  KEY `series_publishers_fk_idx` (`publisher_id`),  
  KEY `series_countries_fk_idx` (`country_id`),  
  KEY `series_languages_fk_idx` (`language_id`),  
  CONSTRAINT `series_countries_fk` FOREIGN KEY (`country_id`) REFERENCES  
  `countries` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `series_first_issues_fk` FOREIGN KEY (`first_issue_id`) REFERENCES  
  `issues` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
```

```
CONSTRAINT `series_languages_fk` FOREIGN KEY (`language_id`) REFERENCES
`languages` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `series_last_issues_fk` FOREIGN KEY (`last_issue_id`) REFERENCES
`issues` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `series_publishers_fk` FOREIGN KEY (`publisher_id`) REFERENCES
`publishers` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `series_publication_types`;

CREATE TABLE `series_publication_types` (
  `id` int(11) NOT NULL,
  `name` varchar(255) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id_UNIQUE` (`id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `stories`;

CREATE TABLE `stories` (
  `id` int(11) NOT NULL,
  `title` varchar(255) DEFAULT NULL,
  `feature` varchar(255) DEFAULT NULL,
  `issue_id` int(11) DEFAULT NULL,
  `script_id` int(11) DEFAULT NULL,
  `pencils_id` int(11) DEFAULT NULL,
  `inks_id` int(11) DEFAULT NULL,
  `colors_id` int(11) DEFAULT NULL,
  `letters_id` int(11) DEFAULT NULL,
  `editing` text,
  `genre` varchar(255) DEFAULT NULL,
  `characters_id` int(11) DEFAULT NULL,
  `synopsis` text,
  `reprint_notes` text,
  `notes` text,
  `type_id` int(11) DEFAULT NULL,
  PRIMARY KEY (`id`),
  UNIQUE KEY `id_UNIQUE` (`id`),
  KEY `stories_issues_fk_idx` (`issue_id`),
  KEY `stories_script_fk_idx` (`script_id`),
  KEY `stories_pencils_fk_idx` (`pencils_id`),
  KEY `stories_inks_fk_idx` (`inks_id`),
  KEY `stories_colors_fk_idx` (`colors_id`),
  KEY `stories_letters_fk_idx` (`letters_id`),
  KEY `stories_characters_fk_idx` (`characters_id`),
  KEY `stories_story_types_fk_idx` (`type_id`),
  CONSTRAINT `stories_characters_fk` FOREIGN KEY (`characters_id`) REFERENCES
`stories_characters` (`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
  CONSTRAINT `stories_colors_fk` FOREIGN KEY (`colors_id`) REFERENCES
`stories_colors` (`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
```



```
CONSTRAINT `stories_inks_fk` FOREIGN KEY (`inks_id`) REFERENCES `stories_inks`
(`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `stories_issues_fk` FOREIGN KEY (`issue_id`) REFERENCES `issues`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `stories_letters_fk` FOREIGN KEY (`letters_id`) REFERENCES
`stories_letters` (`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `stories_pencils_fk` FOREIGN KEY (`pencils_id`) REFERENCES
`stories_pencils` (`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `stories_script_fk` FOREIGN KEY (`script_id`) REFERENCES
`stories_script` (`story_id`) ON DELETE NO ACTION ON UPDATE NO ACTION,
CONSTRAINT `stories_story_types_fk` FOREIGN KEY (`type_id`) REFERENCES
`story_types` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `stories_characters`;

CREATE TABLE `stories_characters` (
  `story_id` int(11) NOT NULL,
  `character_id` int(11) NOT NULL,
  PRIMARY KEY (`story_id`,`character_id`),
  KEY `stories_characters_characters_fk_idx` (`character_id`),
  CONSTRAINT `stories_characters_characters_fk` FOREIGN KEY (`character_id`)
REFERENCES `characters` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `stories_colors`;

CREATE TABLE `stories_colors` (
  `story_id` int(11) NOT NULL,
  `person_id` int(11) NOT NULL,
  PRIMARY KEY (`story_id`,`person_id`),
  KEY `stories_colors_people_fk_idx` (`person_id`),
  CONSTRAINT `stories_colors_people_fk` FOREIGN KEY (`person_id`) REFERENCES
`people` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `stories_inks`;

CREATE TABLE `stories_inks` (
  `story_id` int(11) NOT NULL,
  `person_id` int(11) NOT NULL,
  PRIMARY KEY (`story_id`,`person_id`),
  KEY `stories_inks_people_fk_idx` (`person_id`),
  CONSTRAINT `stories_inks_people_fk` FOREIGN KEY (`person_id`) REFERENCES `people`
(`id`) ON DELETE NO ACTION ON UPDATE NO ACTION
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

DROP TABLE IF EXISTS `stories_letters`;

CREATE TABLE `stories_letters` (
```

```
`story_id` int(11) NOT NULL,  
`person_id` int(11) NOT NULL,  
PRIMARY KEY (`story_id`,`person_id`),  
KEY `stories_letters_people_fk_idx` (`person_id`),  
CONSTRAINT `stories_letters_people_fk` FOREIGN KEY (`person_id`) REFERENCES  
`people` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `stories_pencils`;
```

```
CREATE TABLE `stories_pencils` (  
  `story_id` int(11) NOT NULL,  
  `person_id` int(11) NOT NULL,  
  PRIMARY KEY (`story_id`,`person_id`),  
  KEY `stories_pencils_people_fk_idx` (`person_id`),  
  CONSTRAINT `stories_pencils_people_fk` FOREIGN KEY (`person_id`) REFERENCES  
  `people` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `stories_script`;
```

```
CREATE TABLE `stories_script` (  
  `story_id` int(11) NOT NULL,  
  `person_id` int(11) NOT NULL,  
  PRIMARY KEY (`story_id`,`person_id`),  
  KEY `stories_script_people_fk_idx` (`person_id`),  
  CONSTRAINT `stories_script_people_fk` FOREIGN KEY (`person_id`) REFERENCES  
  `people` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `story_reprints`;
```

```
CREATE TABLE `story_reprints` (  
  `id` int(11) NOT NULL,  
  `origin_id` int(11) DEFAULT NULL,  
  `target_id` int(11) DEFAULT NULL,  
  PRIMARY KEY (`id`),  
  UNIQUE KEY `id_UNIQUE` (`id`),  
  KEY `story_reprints_origin_stories_fk_idx` (`origin_id`),  
  KEY `story_reprints_target_stories_fk_idx` (`target_id`),  
  CONSTRAINT `story_reprints_origin_stories_fk` FOREIGN KEY (`origin_id`)  
REFERENCES `stories` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION,  
  CONSTRAINT `story_reprints_target_stories_fk` FOREIGN KEY (`target_id`)  
REFERENCES `stories` (`id`) ON DELETE NO ACTION ON UPDATE NO ACTION  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

```
DROP TABLE IF EXISTS `story_types`;
```

```
CREATE TABLE `story_types` (  
  `id` int(11) NOT NULL,
```

```
`name` varchar(255) NOT NULL,  
PRIMARY KEY (`id`),  
UNIQUE KEY `name_UNIQUE` (`name`),  
UNIQUE KEY `id_UNIQUE` (`id`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8;
```

General Comments

We decided to use MySQL as a DBMS because some of us already had experience with it.

We used three different column types in our tables. The INT type was used for the indexes and the integer values such as years (full dates were represented using VARCHAR). The VARCHAR type was used for almost any text type field. The size of 255 was arbitrarily chosen and when it was too small, we instead used the TEXT type. Since we have not yet started importing the data, we might have to modify some tables to accommodate for larger text fields for example.

We split the work in two parts. Léonard did the DDL, Pierre and Chenyu did the ER diagram.

Deliverable 2

Assumptions

The assumptions we did in the first deliverable were pretty good and we did not have to make new ones, or revise existing ones.

Data Loading

Data loading was done in two parts. The first part consisted of parsing and cleaning the data, while the second part took care of importing the data in the database.

We cleaned the data using Python and Powershell scripts.

We first applied the same treatment to all CSV files; we inserted “\N” in the empty fields so that MySQL would recognize them as NULL values. We also replaced fields containing only a “?” or “NULL” by “\N”.

For the date fields that were some kind of string, we tried to extract a year from them in order to be able to work with integers. This was done by detecting a pattern of four numbers and assuming it to represent a year.

Extracting the list of characters from the story csv required quite a bit of parsing. According to the GCD official documentation for this field, a character or group of characters can be followed by “[]” in order to indicate an alternate identity to the character or to list the members of a group. We decided that it made more sense to extract every character as well as their alternate identities in separate characters. We had multiple cases where a single character for example, “The Flash” had multiple other identities, and by separating the data the way we did, we were able to keep them. We also removed comments written in parenthesis and other types of comments, usually written in capital letters.

We were able to apply a similar process to extract the data for the different tables from the story csv.

The process of loading the data was pretty straightforward because we previously created a csv for each table so we could just use the “load data infile” command from MySQL.

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>

Interface

Design logic Description

The application will be web based with a backend in PHP. We will use JTables to display the data in a searchable and sortable table.

Screenshots

We haven't yet had the opportunity to work on the front end, so no screenshot is available.

General Comments

Cleaning the data to make it somewhat useable was a big amount of work and we might need to go back to it later to make some minor adjustments. Because we have a large number of tables, it also took quite some time to write all the SQL scripts to load the data from the CSV and to add the missing foreign key constraints.

The queries were written by Chenyu. Pierre and Léonard did the data cleanup and importation and made the required modifications to the last deliverable.

Here are some comments concerning the modification of the ER diagram:

- We deleted the ternary relationship between Publisher, Indicia_Publisher and Country (if they come from different countries).
- The reprint relationships are now Many-to-Many relationships according to the data
- A series is published for exactly one country, has only one publication type and is written in exactly one language as it is defined in the data
- We separated 'genre' and 'editing' as two entity sets linked with the story entity through a relationship
- We realize that a story can only be published in one issue and can only have one type that is why we decide to keep 'issue_id' and 'type_id' as attributes of the story entity and not create separate ones.
- We represent 'first_issue' and 'last_issue' as separate 1-to-1 relationships.
- We represent 'featured character' and 'all other characters' using different relationships. Thus, different tables

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>

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>