Databases Project – Spring 2017

Prof. Anastasia Ailamaki

Team No:

Names:

Contents

Contents 1

Deliverable 1 2

Assumptions 2

Entity Relationship Schema 2

Schema 2

Description 2

Relational Schema 2

ER schema to Relational schema 2

DDL 3

General Comments 3

Deliverable 2 4

Assumptions 4

Data Loading 4

Query Implementation 4

Query a: 4

Description of logic: 4

SQL statement 4

Interface 4

Design logic Description 4

Screenshots 4

General Comments 4

Deliverable 3 5

Assumptions 5

Query Implementation 5

Query a: 5

Description of logic: 5

SQL statement 5

Query Analysis 5

Selected Queries (and why) 5

Query 1 5

Query 2 5

Query 3 5

Interface 6

General Comments 6

# Deliverable 1

# Deliverable 1

## Assumptions

The assumptions we made were that we are supposed to delete and to create some tables from the data to optimize the efficiency and the size of our DB. For example, if one person has written thousands of books, it would be a loss of space to copy their name in every issue whey worked in. Instead, we would create a table “artist” with this person inside, and just this person’s id to all the books they have written.

## Entity Relationship Schema

### Schema

**TODO TIM METTRE L’IMAGE DRAW IO et refaire le bullshit**

### Description

We translate the given dataset description into entities. We add some entities to better sort data:

* Artist: id, name which describe the different persons for a story
* Price: id, amount, currency which describe the different price for an issue
* Genre: id, name
* Website: id, url
* Character: id, name

And instead of doing an entity for the issue\_reprint and story\_reprint, we add one relationship “Reprint” to issue and one relationship “Reprint” to story, which has an id attribute and repoints to the same entity. We hence added a normal arrow between story and is\_reprinted\_as

We then link the artist table to the story table through multiples relations “has\_...” corresponding to if that artist has worked for a certain story with the role cited in the relation “has\_...”. Similarily, we link the price entity to the issue entity through the “has\_price” relationship, since a single issue can have many prices, expressed in many currencies etc.

We also created a table character and linked it to the story entity with the “has\_character” relation, since some famous characters are featured in many stories (superheroes typically).

Similarly, we created an entity genre, since a story may have many genre, common with other series.

We created an entity website as the brands/incidia/publisher share common websites.

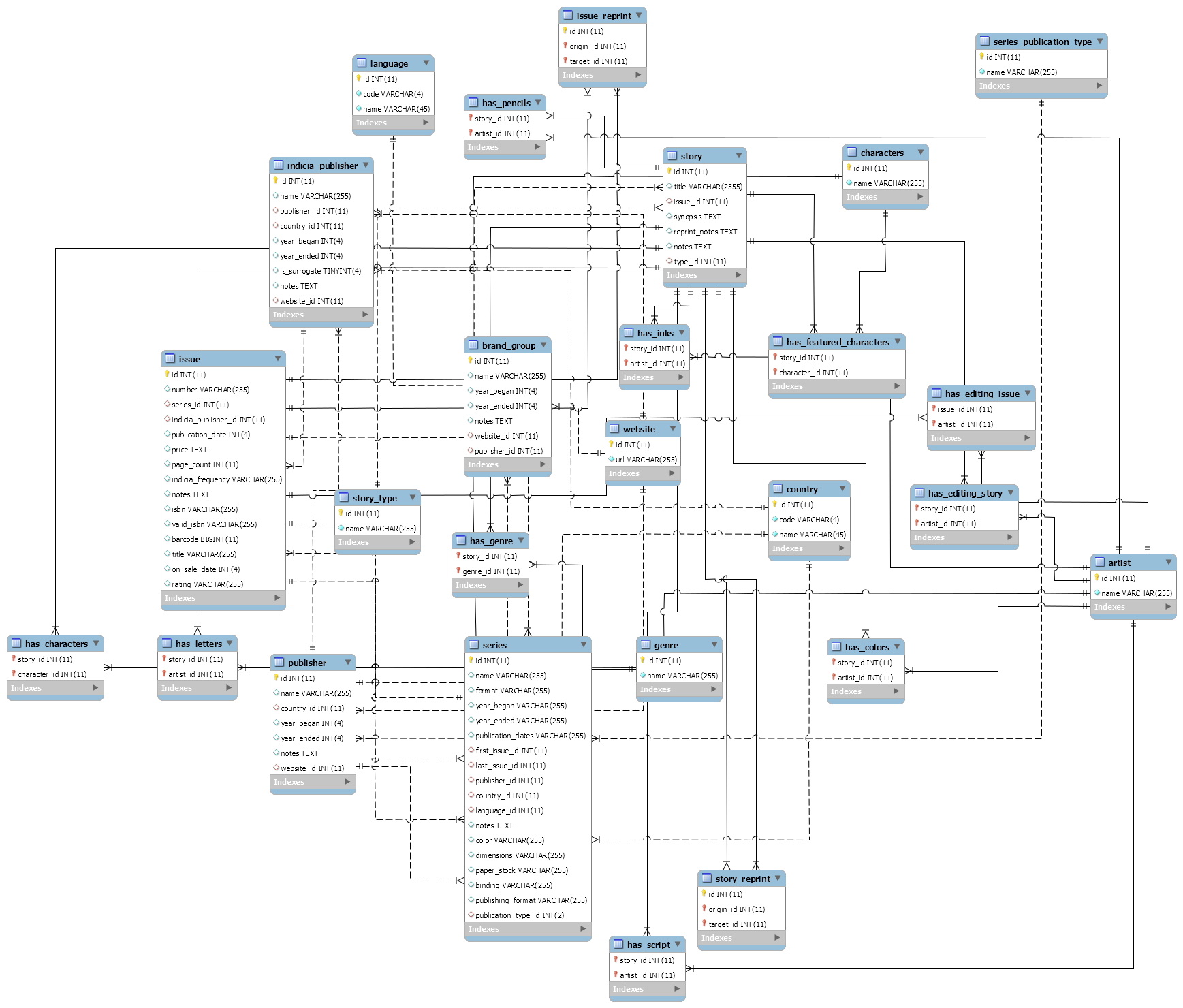
## Relational Schema

### ER schema to Relational schema

The translation straightly follows a certain logic. For each entity in the ER model we create a table. The arrow types have been translated in relations tables and foreign keys.

* For each relation 1 to 1 between entity A and entity B, we directly replaced the B’s name in A’s attributes by an id B\_id, and set it as a foreign key pointing on B’s id itself.
* For each relation from 1 to n, we created an intermediate table “has\_...” containing relations. A relation is composed of both ids of the two entities connected through this relation, and the id of the relation. This relation table contains then 2 foreign keys, pointing on both entities related. We hence have 7 “has tables”, for connecting for instance artists with the story they’ve been working on, or also for connecting the many prices an issue could have.

We then deleted all the “artist fields, cost field, etc…”, i.e. “inks, pencils etc.” since entities are directly connected through the relation.



### DDL

***- -----------------------------------------------------***

***-- Table `mydb`.`artist`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`artist` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`country`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`country` (**

**`id` INT(11) NOT NULL,**

**`code` VARCHAR(4) NOT NULL,**

**`name` VARCHAR(45) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`website`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`website` (**

**`id` INT(11) NOT NULL,**

**`url` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`publisher`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`publisher` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NULL DEFAULT NULL,**

**`country\_id` INT(11) NULL DEFAULT NULL,**

**`year\_began` INT(4) NULL DEFAULT NULL,**

**`year\_ended` INT(4) NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`website\_id` INT(11) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `country\_id\_idx` (`country\_id` ASC),**

**INDEX `website\_id` (`website\_id` ASC),**

**CONSTRAINT `country\_id\_publisher`**

**FOREIGN KEY (`country\_id`)**

**REFERENCES `mydb`.`country` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `website\_id\_publisher`**

**FOREIGN KEY (`website\_id`)**

**REFERENCES `mydb`.`website` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`brand\_group`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`brand\_group` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NULL DEFAULT NULL,**

**`year\_began` INT(4) NULL DEFAULT NULL,**

**`year\_ended` INT(4) NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`website\_id` INT(11) NULL DEFAULT NULL,**

**`publisher\_id` INT(11) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `publisher\_id\_idx` (`publisher\_id` ASC),**

**INDEX `website\_id` (`website\_id` ASC),**

**CONSTRAINT `publisher\_id\_brand`**

**FOREIGN KEY (`publisher\_id`)**

**REFERENCES `mydb`.`publisher` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `website\_id\_brand`**

**FOREIGN KEY (`website\_id`)**

**REFERENCES `mydb`.`website` (`id`)**

**ON DELETE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`characters`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`characters` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`genre`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`genre` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`indicia\_publisher`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`indicia\_publisher` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NULL DEFAULT NULL,**

**`publisher\_id` INT(11) NULL DEFAULT NULL,**

**`country\_id` INT(11) NULL DEFAULT NULL,**

**`year\_began` INT(4) NULL DEFAULT NULL,**

**`year\_ended` INT(4) NULL DEFAULT NULL,**

**`is\_surrogate` TINYINT(4) NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`website\_id` INT(11) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `publisher\_id\_idx` (`publisher\_id` ASC),**

**INDEX `country\_id\_idx` (`country\_id` ASC),**

**INDEX `website\_id` (`website\_id` ASC),**

**CONSTRAINT `country\_id\_indicia`**

**FOREIGN KEY (`country\_id`)**

**REFERENCES `mydb`.`country` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `publisher\_id\_indicia`**

**FOREIGN KEY (`publisher\_id`)**

**REFERENCES `mydb`.`publisher` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `website\_id\_indicia`**

**FOREIGN KEY (`website\_id`)**

**REFERENCES `mydb`.`website` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`language`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`language` (**

**`id` INT(11) NOT NULL,**

**`code` VARCHAR(4) NOT NULL,**

**`name` VARCHAR(45) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`series\_publication\_type`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`series\_publication\_type` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`series`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`series` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NULL DEFAULT NULL,**

**`format` VARCHAR(255) NULL DEFAULT NULL,**

**`year\_began` VARCHAR(255) NULL DEFAULT NULL,**

**`year\_ended` VARCHAR(255) NULL DEFAULT NULL,**

**`publication\_dates` VARCHAR(255) NULL DEFAULT NULL,**

**`first\_issue\_id` INT(11) NULL DEFAULT NULL,**

**`last\_issue\_id` INT(11) NULL DEFAULT NULL,**

**`publisher\_id` INT(11) NULL DEFAULT NULL,**

**`country\_id` INT(11) NULL DEFAULT NULL,**

**`language\_id` INT(11) NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`color` VARCHAR(255) NULL DEFAULT NULL,**

**`dimensions` VARCHAR(255) NULL DEFAULT NULL,**

**`paper\_stock` VARCHAR(255) NULL DEFAULT NULL,**

**`binding` VARCHAR(255) NULL DEFAULT NULL,**

**`publishing\_format` VARCHAR(255) NULL DEFAULT NULL,**

**`publication\_type\_id` INT(2) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `first\_issue\_id\_idx` (`first\_issue\_id` ASC),**

**INDEX `last\_issue\_id\_idx` (`last\_issue\_id` ASC),**

**INDEX `publisher\_id\_idx` (`publisher\_id` ASC),**

**INDEX `country\_id\_idx` (`country\_id` ASC),**

**INDEX `language\_id\_idx` (`language\_id` ASC),**

**INDEX `publication\_type\_id` (`publication\_type\_id` ASC),**

**CONSTRAINT `country\_id\_series`**

**FOREIGN KEY (`country\_id`)**

**REFERENCES `mydb`.`country` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `first\_issue\_id\_series`**

**FOREIGN KEY (`first\_issue\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `language\_id\_series`**

**FOREIGN KEY (`language\_id`)**

**REFERENCES `mydb`.`language` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `last\_issue\_id\_series`**

**FOREIGN KEY (`last\_issue\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION,**

**CONSTRAINT `publication\_type\_id\_series`**

**FOREIGN KEY (`publication\_type\_id`)**

**REFERENCES `mydb`.`series\_publication\_type` (`id`)**

**ON DELETE NO ACTION**

**ON UPDATE NO ACTION,**

**CONSTRAINT `publisher\_id\_series`**

**FOREIGN KEY (`publisher\_id`)**

**REFERENCES `mydb`.`publisher` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`issue`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`issue` (**

**`id` INT(11) NOT NULL,**

**`number` VARCHAR(255) NULL DEFAULT NULL,**

**`series\_id` INT(11) NULL DEFAULT NULL,**

**`indicia\_publisher\_id` INT(11) NULL DEFAULT NULL,**

**`publication\_date` INT(4) NULL DEFAULT NULL,**

**`price` TEXT NULL DEFAULT NULL,**

**`page\_count` INT(11) NULL DEFAULT NULL,**

**`indicia\_frequency` VARCHAR(255) NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`isbn` VARCHAR(255) NULL DEFAULT NULL,**

**`valid\_isbn` VARCHAR(255) NULL DEFAULT NULL,**

**`barcode` BIGINT(11) NULL DEFAULT NULL,**

**`title` VARCHAR(255) NULL DEFAULT NULL,**

**`on\_sale\_date` INT(4) NULL DEFAULT NULL,**

**`rating` VARCHAR(255) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `indicia\_publisher\_id` (`indicia\_publisher\_id` ASC),**

**INDEX `series\_id` (`series\_id` ASC),**

**CONSTRAINT `indicia\_publishier\_id\_issue`**

**FOREIGN KEY (`indicia\_publisher\_id`)**

**REFERENCES `mydb`.`indicia\_publisher` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE,**

**CONSTRAINT `series\_id\_issue`**

**FOREIGN KEY (`series\_id`)**

**REFERENCES `mydb`.`series` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`story\_type`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`story\_type` (**

**`id` INT(11) NOT NULL,**

**`name` VARCHAR(255) NOT NULL,**

**PRIMARY KEY (`id`))**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`story`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`story` (**

**`id` INT(11) NOT NULL,**

**`title` VARCHAR(2555) NULL DEFAULT NULL,**

**`issue\_id` INT(11) NULL DEFAULT NULL,**

**`synopsis` TEXT NULL DEFAULT NULL,**

**`reprint\_notes` TEXT NULL DEFAULT NULL,**

**`notes` TEXT NULL DEFAULT NULL,**

**`type\_id` INT(11) NULL DEFAULT NULL,**

**PRIMARY KEY (`id`),**

**INDEX `issue\_id\_idx` (`issue\_id` ASC),**

**INDEX `type\_id\_idx` (`type\_id` ASC),**

**CONSTRAINT `issue\_id\_story`**

**FOREIGN KEY (`issue\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE CASCADE,**

**CONSTRAINT `type\_id\_story`**

**FOREIGN KEY (`type\_id`)**

**REFERENCES `mydb`.`story\_type` (`id`)**

**ON DELETE SET NULL**

**ON UPDATE CASCADE)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_characters`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_characters` (**

**`story\_id` INT(11) NOT NULL,**

**`character\_id` INT(11) NOT NULL,**

**PRIMARY KEY USING BTREE (`story\_id`, `character\_id`),**

**INDEX `character\_id\_idx` (`character\_id` ASC),**

**INDEX `story\_id\_idx` (`story\_id` ASC),**

**CONSTRAINT `character\_id\_characters`**

**FOREIGN KEY (`character\_id`)**

**REFERENCES `mydb`.`characters` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_characters`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_colors`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_colors` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `story\_id\_idx` (`story\_id` ASC),**

**INDEX `artist\_id\_idx` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_colors`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_colors`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_editing\_issue`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_editing\_issue` (**

**`issue\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`issue\_id`, `artist\_id`),**

**INDEX `artist\_id\_editing\_issue` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_editing\_issue`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `issue\_id\_editing\_issue`**

**FOREIGN KEY (`issue\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = latin1;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_editing\_story`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_editing\_story` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `artist\_id\_editing` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_editing`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE,**

**CONSTRAINT `story\_id\_editing`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = latin1;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_featured\_characters`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_featured\_characters` (**

**`story\_id` INT(11) NOT NULL,**

**`character\_id` INT(11) NOT NULL,**

**PRIMARY KEY USING BTREE (`story\_id`, `character\_id`),**

**INDEX `character\_id\_feat` (`character\_id` ASC),**

**CONSTRAINT `character\_id\_feat`**

**FOREIGN KEY (`character\_id`)**

**REFERENCES `mydb`.`characters` (`id`)**

**ON DELETE CASCADE,**

**CONSTRAINT `story\_id\_feat`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = latin1;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_genre`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_genre` (**

**`story\_id` INT(11) NOT NULL,**

**`genre\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`genre\_id`, `story\_id`),**

**INDEX `genre\_id\_idx` (`genre\_id` ASC),**

**INDEX `story\_id\_idx` (`story\_id` ASC),**

**CONSTRAINT `genre\_id\_genre`**

**FOREIGN KEY (`genre\_id`)**

**REFERENCES `mydb`.`genre` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_genre`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_inks`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_inks` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `stpry\_Id\_idx` (`story\_id` ASC),**

**INDEX `artist\_id\_idx` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_inks`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_inks`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_letters`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_letters` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `artist\_id\_letters` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_letters`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE,**

**CONSTRAINT `story\_id\_letters`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8**

**COLLATE = utf8\_bin;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_pencils`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_pencils` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `id\_artist\_idx` (`artist\_id` ASC),**

**INDEX `story\_id\_idx` (`story\_id` ASC),**

**CONSTRAINT `artist\_id\_pencils`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_pencils`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`has\_script`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`has\_script` (**

**`story\_id` INT(11) NOT NULL,**

**`artist\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`story\_id`, `artist\_id`),**

**INDEX `story\_id\_idx` (`story\_id` ASC),**

**INDEX `artist\_id\_idx` (`artist\_id` ASC),**

**CONSTRAINT `artist\_id\_script`**

**FOREIGN KEY (`artist\_id`)**

**REFERENCES `mydb`.`artist` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `story\_id\_script`**

**FOREIGN KEY (`story\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`issue\_reprint`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`issue\_reprint` (**

**`id` INT(11) NOT NULL,**

**`origin\_id` INT(11) NOT NULL,**

**`target\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`id`, `origin\_id`, `target\_id`),**

**INDEX `origin\_issue\_id\_idx` (`origin\_id` ASC),**

**INDEX `target\_issue\_id\_idx` (`target\_id` ASC),**

**CONSTRAINT `origin\_issue\_id\_issue\_reprint`**

**FOREIGN KEY (`origin\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `target\_issue\_id\_issue\_reprint`**

**FOREIGN KEY (`target\_id`)**

**REFERENCES `mydb`.`issue` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

***-- -----------------------------------------------------***

***-- Table `mydb`.`story\_reprint`***

***-- -----------------------------------------------------***

**CREATE TABLE IF NOT EXISTS `mydb`.`story\_reprint` (**

**`id` INT(11) NOT NULL,**

**`origin\_id` INT(11) NOT NULL,**

**`target\_id` INT(11) NOT NULL,**

**PRIMARY KEY (`id`, `origin\_id`, `target\_id`),**

**INDEX `origin\_id\_idx` (`origin\_id` ASC),**

**INDEX `target\_id\_idx` (`target\_id` ASC),**

**CONSTRAINT `origin\_id\_reprint`**

**FOREIGN KEY (`origin\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION,**

**CONSTRAINT `target\_id\_reprint`**

**FOREIGN KEY (`target\_id`)**

**REFERENCES `mydb`.`story` (`id`)**

**ON DELETE CASCADE**

**ON UPDATE NO ACTION)**

**ENGINE = InnoDB**

**DEFAULT CHARACTER SET = utf8;**

# Deliverable 2

Design evolved a bit from milestone 1. Tables we added are has\_featured\_character(story\_id, character\_id) as well as has\_editing(story\_id, artist\_id).

## Assumptions

Parsing leads to lots of assumptions. Here the ones we had to assume :

* Names are full of information between parenthesis or brackets (such as (signed), (translator)…). We delete them to be able to rely them (instead of having twice the same author). Also to make them match even more, we construct a comparative string, which is the name of author without spaces, dot and hyphen. These can permit us to avoid duplicata or obtain full names of artists (our actual version uniquely keep first found entry but should be modify so that we keep the longer and most detailed ones).
* Dates are difficult to retrieve from given csv file, because of its non-uniform format. We assume:
  + “1870’s” become “1870”
  + “July 10 1870” become “1870”
  + “1870-07-10” become “1870”
* Stories with null titles are deleted.

## Parsing and cleaning of the data

We choose to use a local database, using **wamp server** (windows local server) and **phpmyadmin**. Thus we are parsing the csv file using php.

We had to parse multiple times the csv’s given to us.

For the beginning, we created scripts which create sql commands from csv data. During the process, we get rid of null values such as “Null”, “none”, “[nn]”, “?”, etc. We also had to change some column types such as:

* isbn , rating, number of issue becoming varchar
* synospsis of story becoming text
* dates becoming integer

**TODO KL changer les commentaires sur les fichers php et beaucoup parler du cleaning**

We used the following php commands to do the parsing. This is the “*functions.php*” file, used in every php parsing files.

[functions.php]

<?php

function parseNullValue($s) {

if(empty($s) && $s!='0') return true;

$nullValues = ['NULL', '[nn]', 'nn', 'none', '[none]','?', '(unknown)','None'];

foreach($nullValues as $n) {

if ($s === $n) {

return true;

}

}

return false;

}

function parseDoubleQuote($s) {

if(parseNullValue($s)) return "NULL";

$res = str\_replace('"', '\"', $s);

$res = str\_replace('\\\\"', '\"', $res);

return '"'.$res.'"';

}

function getDateFromYear($year) {

if(parseNullValue($year)) return "NULL";

$res = preg\_replace("/[^\d\s-]/" , "", $year);

$res1 = preg\_split("/[\s-]/" , $res);

for ($i = 0; $i < sizeof($res1); $i++) {

if(strlen($res1[$i])==4) {

return $res1[$i];

}

}

$month=1; $day=1;

$hour=0; $minute=0; $second=0;

// no date with 4 digits

return "0";

}

function getInt($i) {

if(parseNullValue($i)) return "NULL";

// suppress [ ] char

$i = preg\_replace("/\[\*\]\*/" ,"", $i);

return $i;

}

// return last index used in csv - useful for assigning id

function getLastIndex($file) {

$index;

while((!feof($file)) && ($val = fgetcsv($file))){

$index = $val[0];

}

return (empty($index)) ? 0 : $index+1;

}

// return words separated by delimiter

function parseNames($s, $delimiter=";"){

// get rid of first and last double quotes

$string = substr($s, 1, -1);

$array = explode($delimiter, $string);

for($i = 0; $i< sizeof($array); $i++){

$array[$i] = ltrim($array[$i]);

}

return array\_filter($array, function($value) { return $value !== ''; });

}

// return true if $s in contained in $csv file (opened) (at position $pos), false otherwise

function isInCsv($file, $s, $pos){

rewind($file);

if(empty($s)) {

return false;

}

while(! feof($file)){

$val = fgetcsv($file);

if($val[$pos]==$s) {

return $val[0];

}

}

return false;

}

// return true if $s in contained in $csv file (opened) (at position $pos), false otherwise

function isInCsvName($file, $s, $pos){

rewind($file);

if(empty($s)) {

return false;

}

$string = parseToCompare($s);

while(! feof($file)){

$content = fgetcsv($file);

$val = parseToCompare($content[$pos]);

if($val==$string) {

return $content[0];

}

}

return false;

}

// modify string so that it can match even with the following differences : whitespaces, dot, dash (essentially for names)

// in csv -> will keep first occurence

function parseToCompare($s){

$res = preg\_replace("/\s/", "", $s);

$res = preg\_replace("/\-/", "", $res);

$res = preg\_replace("/\./", "", $res);

$res = strtolower($res);

return $res;

}

// delete from $s all content between () or [] or ?

function parseComments($s) {

$res = preg\_replace("/\[.\*\]/", "", $s);

$res = preg\_replace("/\(.\*\)/", "", $res);

$res = preg\_replace("/\?/", "", $res);

if(empty(preg\_replace("/\s/","",$res))){

return "NULL";

}

$res = trim($res);

return $res;

}

**Website** :

Websites come from publisher, indicia\_publisher and brand\_group csv file. We get the url values, parse null values, and add it in a new website table. We never add twice the same website. Once done, we process publisher, indicia\_publisher and brand\_group csv files and write into a new csv file their values plus their url values changed into website ids (which become a foreign key referencing the website table). These csv files are then process into sql commands to import them into our database.

[website\_tocsv.php]

<?php

include("db.php");

include("functions.php");

$files = array("comics/brand\_group.csv" => 5 , "comics/indicia\_publisher.csv" => 8, "comics/publisher.csv" => 6);

$csv = fopen("comics/website.csv", "a+"); // write into this sql to import

echo "<h1>Websites</h1>";

$index = getLastIndex($csv);

$min = 0;

$max = 10000000;

// get websites from all given files

foreach ($files as $f => $pos) {

$i = 0;

echo "begin with " . $f;

$file = fopen($f,"r");

$val = fgetcsv($file); // avoid url column name

while(! feof($file)){

$i++;

$val = fgetcsv($file);

if($i > $min){

$url = $val[$pos];

// if the websie

if(!parseNullValue($url)){

if(isInCsv($csv, $url,1)===false) {

$add = $index . ",".$url."\n";

fwrite($csv, $add);

$index++;

}

}

}

if($i==$max){

return;

}

}

fclose($file);

}

fclose($csv);

?>

[website\_to\_id.php]

<?php

include("db.php");

include("functions.php");

$files = array("comics/brand\_group.csv" => 5 , "comics/indicia\_publisher.csv" => 8, "comics/publisher.csv" => 6);

$filesnames = array("comics/brand\_group.csv" => "comics/brand\_group\_id.csv" , "comics/indicia\_publisher.csv" => "comics/indicia\_publisher\_id.csv", "comics/publisher.csv" => "comics/publisher\_id.csv");

$csv = fopen("comics/website.csv", "r"); // write into this sql to import

echo "<h1>Websites to id</h1>";

echo "<h2>Don't forget to run websites script first !!</h2>";

$index = getLastIndex($csv);

$min = 0;

$max = 1000000000;

// get websites from all given files

foreach ($files as $f => $pos) {

$i = 0;

$file = fopen($f,"r");

$out = implode(",",fgetcsv($file))."\n";

while(! feof($file) && ($i<$max)){

$i++;

$val = fgetcsv($file);

if($i > $min){

$url = $val[$pos];

if(!parseNullValue($url)){

$index = isInCsvName($csv, $url,1);

$val[$pos] = $index;

}

//var\_dump($val);

$out .= implode(",",$val)."\n";

if($i>$max) break;

}

}

fclose($file);

file\_put\_contents($filesnames[$f], $out);

}

fclose($csv);

?>

[website.php]

<?php

include("db.php");

include("functions.php");

$file = fopen("comics/website.csv","r");

$mysql = fopen("website.sql", "w"); // write into this sql to import

/\*

website id ->> mettre dans table website -> remettre foreign key

year -> date

\*/

$min = 0;

$max = 100000;

$i = 0;

/\*

0 => string 'id' (length=2)

1 => string 'code' (length=4)

2 => string 'name' (length=4)

\*/

while(! feof($file)){

$i++;

$val = fgetcsv($file);

if($i > $min){

$id = getInt($val[0]);

$url = parseDoubleQuote($val[1]);

$query = 'INSERT INTO website(id,url) VALUES(

'.$id.', '.$url.'

);';

//print\_r($query);

fwrite($mysql,$query);

$s1 = $con->query($query);

/\*var\_dump($s1);\*/

if($i==$max){

break;

}

}

}

fclose($file);

?>

**Artist, Characters, Genre** :

Artists come from story csv file. We get names, parse null values, and add it in a new artist table.

If a story has an artist (for script, letters, inks…) we add in an artist csv file the found artist (id, artist name) and we also add in a has\_ csv file the pairs (story\_id, artist\_id). These csv files are then process into sql commands to import them into our database.

This same idea is also applied to characters (features and characters from story) and genres.

[genre\_tocsv.php]

<?php

include("db.php");

include("functions.php");

$file = fopen("comics/story.csv","r");

$csv = fopen("comics/genre.csv", "a+");

$has\_csv = fopen("comics/has\_genre.csv", "w+");

echo "<h1>Genre</h1>";

$index = getLastIndex($csv);

$min = 0;

$max = 10000;

$i = 0;

var\_dump(fgetcsv($file));

while(! feof($file)){

$i++;

$val = fgetcsv($file);

if($i > $min){

$id = getInt($val[0]);

$genre = parseDoubleQuote($val[10]);

if($genre!="NULL"){

$genre\_array = parseNames($genre);

foreach ($genre\_array as $p){

$p = parseComments($p);

$exist = isInCsvName($csv, $p,1);

if(!is\_numeric($exist)) {

$add = $index . ",".$p."\n";

fwrite($csv, $add);

// has\_

$query = $id.",".$index ."\n";

fwrite($csv, $add);

$index++;

}

else {

$query = $id.",".$exist ."\n";

fwrite($has\_csv, $query);

}

}

}

if($i==$max){

break;

}

}

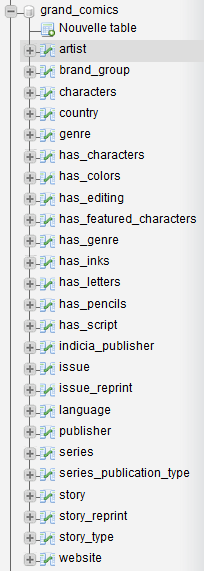
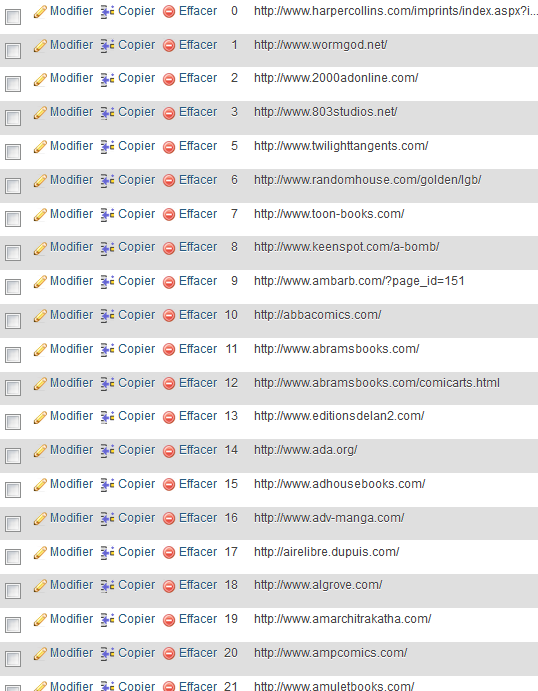
}

fclose($file);

fclose($csv);

?>

**Screenshots of the database, as seen from the phpMyAdmin interface**

**** 

All tablesWebsite table

Has\_genre table Language table

## Query Implementation

Here are the 8 queries we were asked to implement in MySQL.

For this first query we first select all the brand name possessing at least one indicia publisher from Belgium. We choose to get the name of the brand\_group and the number of Belgian indicia it possesses. Then we simply sort the resulting table by number of indicia and get the name only. Note that we don’t need to go through the Publisher table, which makes us gain some time.

*-- a)*

**SELECT** **T**.**name**

**FROM** (

**SELECT** **B**.**name**,

**COUNT**(**\***) **AS** bid

**FROM** brand\_group B,

indicia\_publisher I,

country C

**WHERE** **C**.**name** **=** 'Belgium' **AND**

**C**.**id** **=** **I**.**country\_id** **AND**

**I**.**publisher\_id** **=** **B**.**publisher\_id**

**GROUP BY** **B**.**name**

) **AS** T

**ORDER BY** **T**.**bid**

For the query b), we simply use the chain AND rule to get all publishers from Denmark, in a straightforward fashion.

*-- b)*

**SELECT** **P**.**id**, **P**.**name**

**FROM** publisher P,

country C,

series S

**WHERE** **C**.**name** **=** 'Denmark' **AND**

**S**.**country\_id** **=** **C**.**id** **AND**

**S**.**publisher\_id** **=** **P**.**id** **AND**

**S**.**publication\_type\_id** **=** 1

For querry c), the fashion is similar to b), we simply apply the chain rule, to get series from Switzerland and published in a magazine.

*-- c)*

**SELECT** **S**.**name**

**FROM** series S,

country C,

series\_publication\_type T

**WHERE** **T**.**name** **=** 'magazine' **AND**

**T**.**id** **=** **S**.**publication\_type\_id** **AND**

**S**.**country\_id** **=** **C**.**id** **AND**

**C**.**name** **=** "Switzerland"

Here in d) we simply want to get all issues from 1990, sorted by year. Note that for simplicity purposes, all the dates have been converted to years stored as integers, since most of the dates simple consists as a year, often followed by text such as “circa”. It is hence easier to work with int(4) formatted years.

*-- d)*

**SELECT** **COUNT**(**\***)

**FROM** issue I

**WHERE** **I**.**publication\_date** **>=** 1990

**GROUP BY** **I**.**publication\_date**

For the query e), we simply do a left join between the indicias and the series (to avoid going through the publisher table). Then we just ask for names resembling Dc comics.

*-- e)*

**SELECT** **I**.**name** **AS** name,

**COUNT**(**I**.**id**) **AS** nb

**FROM** indicia\_publisher I

**LEFT JOIN** series S

**ON** **S**.**publisher\_id** **=** **I**.**publisher\_id**

**WHERE** **I**.**name** **LIKE** '%DC\_comics%'

**GROUP BY** **I**.**name**

For query f), we first select all stories that have been reprinted at least once, and then regroup them by original story. Finally, we count how many times each original story has been reprinted and sort them according to that. We only print the names of the stories for aesthetical purposes.

*-- f)*

**SELECT** **S**.**title**

**FROM** story S,

story\_reprint R

**WHERE** **S**.**id** **=** **R**.**origin\_id**

**GROUP BY** **R**.**origin\_id**

**ORDER BY** **COUNT**(**R**.**origin\_id**)

This query (g) was interesting since it uses the chain rule in a particular fashion. What we are seeking are artists who contributed to every part of the making of some story. That is, we want all artist who did color, write, draw and ink a story. We simply want an artist who did all 4 on a same story and a story who had all 4 done by a single artist.

*-- g)*

**SELECT distinct** **A**.**name**

**FROM** artist A,

has\_script SC,

has\_pencils P,

has\_colors C,

has\_inks I,

story S

**WHERE** **A**.**id** **=** **SC**.**artist\_id** **AND**

**A**.**id** **=** **P**.**artist\_id** **AND**

**A**.**id** **=** **C**.**artist\_id** **AND**

**A**.**id** **=** **I**.**artist\_id** **AND**

**S**.**id** **=** **SC**.**story\_id** **AND**

**S**.**id** **=** **P**.**story\_id** **AND**

**S**.**id** **=** **C**.**story\_id** **AND**

**S**.**id** **=** **I**.**story\_id**

For the last query h), we wanted Batman to be a non-featured character of a non-reprinted story. For that purpose, we seek all stories which were not reprinted, that is, which are not featured in the reprint table. We hence look for the story in the reprint table and expect the number of its occurrences to be 0.

Then, we simply say we want Batman to be in the non-featured characters. Note the utilization of the command LIKE in order to seek for all strings containing “Batman”.

*-- h)*

**SELECT** **S**.**title**

**FROM** story S,

characters C,

has\_characters HS

**WHERE** 0 **=** ( **SELECT** **COUNT**(distinct **R**.**origin\_id**)

**FROM** story\_reprint R

**WHERE** **S**.**id** **=** **R**.**origin\_id**

) **AND**

**HS**.**character\_id** **=** **C**.**id** **AND**

**HS**.**story\_id** **=** **S**.**id** **AND**

**C**.**name** **LIKE** '%Batman%' **AND**

**S**.**features** NOT **LIKE** '%Batman%'

# 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

**TODO LUCIE**

## 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>

**TODO KL**

### 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>