Databases Project – Spring 2017

Prof. Anastasia Ailamaki

Team No: **14**

Names: Perrotta Lucie, Phan Hoang Kim Lan, Nguyen Tim

# 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

### C:\Users\Me\AppData\Local\Microsoft\Windows\INetCache\Content.Word\ER_model_w_relationships.pngSchema

### 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

**1 – SQL CODE FOR MILESTONE 1**

**2**

**3 CREATE SCHEMA IF NOT EXISTS `mydb` DEFAULT CHARACTER SET utf8 ;**

**4 USE `mydb` ;**

**5**

**6 -- `country**

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

**8 `id` INT NOT NULL,**

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

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

**11 PRIMARY KEY (`id`));**

**12**

**13 -- `website**

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

**15 `id` INT NOT NULL,**

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

**17 PRIMARY KEY (`id`));**

**18**

**19 -- `publisher**

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

**21 `id` INT NOT NULL,**

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

**23 `country\_id` INT NULL,**

**24 `year\_began` DATE NOT NULL,**

**25 `year\_ended` DATE NULL,**

**26 `notes` TEXT NULL,**

**27 `website\_id` INT NULL,**

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

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

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

**31 CONSTRAINT `country\_id`**

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

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

**34 ON DELETE SET NULL**

**35 ON UPDATE CASCADE,**

**36 CONSTRAINT `website\_id`**

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

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

**39 ON DELETE SET NULL**

**40 ON UPDATE CASCADE);**

**41**

**42 -- `language**

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

**44 `id` INT NOT NULL,**

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

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

**47 PRIMARY KEY (`id`));**

**48**

**49 -- `series**

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

**51 `id` INT NOT NULL,**

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

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

**54 `year\_began` DATE NOT NULL,**

**55 `year\_ended` DATE NULL,**

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

**57 `first\_issue\_id` INT NULL,**

**58 `last\_issue\_id` INT NULL,**

**59 `publisher\_id` INT NULL,**

**60 `country\_id` INT NULL,**

**61 `language\_id` INT NULL,**

**62 `notes` TEXT NULL,**

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

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

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

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

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

**68 `publication\_type\_id` VARCHAR(255) NULL,**

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

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

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

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

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

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

**75 CONSTRAINT `first\_issue\_id`**

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

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

**78 ON DELETE SET NULL,**

**79 CONSTRAINT `last\_issue\_id`**

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

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

**82 ON DELETE SET NULL,**

**83 CONSTRAINT `publisher\_id`**

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

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

**86 ON DELETE SET NULL,**

**87 CONSTRAINT `country\_id`**

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

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

**90 ON DELETE SET NULL,**

**91 CONSTRAINT `language\_id`**

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

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

**94 ON DELETE SET NULL);**

**95**

**96 -- `indicia\_publisher**

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

**98 `id` INT NOT NULL,**

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

**100 `publisher\_id` INT NOT NULL,**

**101 `country\_id` INT NULL,**

**102 `year\_began` DATE NOT NULL,**

**103 `year\_ended` DATE NULL,**

**104 `is\_surrogate` TINYINT NULL,**

**105 `notes` TEXT NULL,**

**106 `website\_id` INT NULL,**

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

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

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

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

**111 CONSTRAINT `publisher\_id`**

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

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

**114 ON DELETE CASCADE,**

**115 CONSTRAINT `country\_id`**

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

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

**118 ON DELETE CASCADE,**

**119 CONSTRAINT `website\_id`**

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

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

**122 ON DELETE SET NULL**

**123 ON UPDATE CASCADE);**

**124**

**125 -- `issue**

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

**127 `id` INT NOT NULL,**

**128 `number` INT NULL,**

**129 `series\_id` INT NOT NULL,**

**130 `indicia\_publisher\_id` INT NOT NULL,**

**131 `publication\_date` DATE NULL,**

**132 `page\_count` INT NULL,**

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

**134 `editing` VARCHAR(255) NULL,**

**135 `notes` TEXT NULL,**

**136 `isbn` INT NULL,**

**137 `valid\_isbn` INT NULL,**

**138 `barcode` INT NULL,**

**139 `title` VARCHAR(255) NOT NULL,**

**140 `on\_sale\_date` DATE NULL,**

**141 `rating` INT NULL,**

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

**143 UNIQUE INDEX `id\_UNIQUE` (`id` ASC),**

**144 UNIQUE INDEX `series\_id\_UNIQUE` (`series\_id` ASC),**

**145 UNIQUE INDEX `indicia\_publisher\_id\_UNIQUE` (`indicia\_publisher\_id` ASC),**

**146 CONSTRAINT `series\_id`**

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

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

**149 ON DELETE CASCADE,**

**150 CONSTRAINT `indicia\_publisher\_id`**

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

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

**153 ON DELETE CASCADE);**

**154**

**155 -- `story\_type**

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

**157 `id` INT NOT NULL,**

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

**159 PRIMARY KEY (`id`));**

**160**

**161 -- `story**

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

**163 `id` INT NOT NULL,**

**164 `title` VARCHAR(255) NOT NULL,**

**165 `features` TEXT NULL,**

**166 `issue\_id` INT NULL,**

**167 `letters` VARCHAR(255) NULL,**

**168 `editing` VARCHAR(255) NULL,**

**169 `synopsis` VARCHAR(255) NULL,**

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

**171 `notes` TEXT NULL,**

**172 `type\_id` INT NULL,**

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

**174 UNIQUE INDEX `id\_UNIQUE` (`id` ASC),**

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

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

**177 CONSTRAINT `issue\_id`**

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

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

**180 ON DELETE CASCADE**

**181 ON UPDATE CASCADE,**

**182 CONSTRAINT `type\_id`**

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

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

**185 ON DELETE SET NULL**

**186 ON UPDATE CASCADE);**

**187**

**188 -- `brand\_group**

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

**190 `id` INT NOT NULL,**

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

**192 `year\_began` DATE NOT NULL,**

**193 `year\_ended` DATE NULL,**

**194 `notes` TEXT NULL,**

**195 `website\_Id` INT NULL,**

**196 `publisher\_id` INT NULL,**

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

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

**199 INDEX `website\_id\_idx` (`website\_Id` ASC),**

**200 CONSTRAINT `publisher\_id`**

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

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

**203 ON DELETE SET NULL,**

**204 CONSTRAINT `website\_id`**

**205 FOREIGN KEY (`website\_Id`)**

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

**207 ON DELETE SET NULL**

**208 ON UPDATE CASCADE);**

**209**

**210 -- `story\_reprint**

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

**212 `id` INT NOT NULL,**

**213 `origin\_id` INT NOT NULL,**

**214 `target\_id` INT NOT NULL,**

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

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

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

**218 CONSTRAINT `origin\_id`**

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

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

**221 ON DELETE CASCADE,**

**222 CONSTRAINT `target\_id`**

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

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

**225 ON DELETE CASCADE);**

**226**

**227 -- `issue\_reprint**

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

**229 `id` INT NOT NULL,**

**230 `origin\_issue\_id` INT NOT NULL,**

**231 `target\_issue\_id` INT NOT NULL,**

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

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

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

**235 CONSTRAINT `origin\_issue\_id`**

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

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

**238 ON DELETE CASCADE,**

**239 CONSTRAINT `target\_issue\_id`**

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

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

**242 ON DELETE CASCADE);**

**243**

**244 -- `series\_publication\_type**

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

**246 `id` INT NOT NULL,**

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

**248 PRIMARY KEY (`id`));**

**249**

**250 -- `artist**

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

**252 `id` INT NOT NULL,**

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

**254 PRIMARY KEY (`id`));**

**255**

**256 -- `character**

**257 CREATE TABLE IF NOT EXISTS `mydb`.`character` (**

**258 `id` INT NOT NULL,**

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

**260 PRIMARY KEY (`id`));**

**261**

**262 -- `genre**

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

**264 `id` INT NOT NULL,**

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

**266 PRIMARY KEY (`id`));**

**267**

**268 -- `has\_pencils**

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

**270 `id` INT NOT NULL,**

**271 `story\_id` INT NOT NULL,**

**272 `artist\_id` INT NOT NULL,**

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

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

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

**276 CONSTRAINT `story\_id`**

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

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

**279 CONSTRAINT `artist\_id`**

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

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

**282**

**283 -- `has\_genre**

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

**285 `id` INT NOT NULL,**

**286 `genre\_id` INT NOT NULL,**

**287 `story\_id` INT NOT NULL,**

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

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

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

**291 CONSTRAINT `genre\_id`**

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

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

**294 CONSTRAINT `story\_id`**

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

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

**297**

**298 -- `has\_inks**

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

**300 `id` INT NOT NULL,**

**301 `story\_id` INT NOT NULL,**

**302 `artist\_id` INT NOT NULL,**

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

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

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

**306 CONSTRAINT `story\_Id`**

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

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

**309 CONSTRAINT `artist\_id`**

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

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

**312**

**313 -- `has\_colors**

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

**315 `id` INT NOT NULL,**

**316 `story\_id` INT NOT NULL,**

**317 `artist\_id` INT NOT NULL,**

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

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

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

**321 CONSTRAINT `story\_id`**

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

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

**324 CONSTRAINT `artist\_id`**

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

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

**327**

**328 -- `has\_script**

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

**330 `id` INT NOT NULL,**

**331 `story\_id` INT NOT NULL,**

**332 `artist\_id` INT NOT NULL,**

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

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

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

**336 CONSTRAINT `story\_id`**

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

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

**339 CONSTRAINT `artist\_id`**

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

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

**342**

**343 -- `price**

**344 CREATE TABLE IF NOT EXISTS `mydb`.`price` (**

**345 `id` INT NOT NULL,**

**346 `amount` DECIMAL NOT NULL,**

**347 `currency` VARCHAR(5) NOT NULL,**

**348 PRIMARY KEY (`id`));**

**349**

**350 -- `has\_price**

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

**352 `id` INT NOT NULL,**

**353 `issue\_id` INT NOT NULL,**

**354 `price\_id` INT NOT NULL,**

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

**356 INDEX `price\_id\_idx` (`price\_id` ASC),**

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

**358 CONSTRAINT `price\_id`**

**359 FOREIGN KEY (`price\_id`)**

**360 REFERENCES `mydb`.`price` (`id`),**

**361 CONSTRAINT `issue\_id`**

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

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

**364**

**365 -- `has\_characters**

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

**367 `id` INT NOT NULL,**

**368 `character\_id` INT NOT NULL,**

**369 `story\_id` INT NOT NULL,**

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

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

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

**373 CONSTRAINT `character\_id`**

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

**375 REFERENCES `mydb`.`character` (`id`),**

**376 CONSTRAINT `story\_id`**

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

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

**379 SET SQL\_MODE=@OLD\_SQL\_MODE;**

**380 SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;**

**381 SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;**

**382 )**

## General Comments

I think (Lucie) we all worked equally, both Kim Lan and I had already done SQL in the past so it way maybe more easy to get into it. It was totally new for Tim but the really made efforts and participated as much.

# Deliverable 2

## 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).
* 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”

## Parsing

We choose to use a local database, using wamp server (windows local server) and phpmyadmin.

We had to parse multiple times the csv 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

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 file and change their url values into website id (which become a foreign key referencing the website table).

[CODE]

Artist : Artists come from story csv file. We get names, parse null values, and add it in a new artist table. To not add an artist twice, we construct a comparative string, which is the name of author without spaces, dot and hyphen. Unfortunately, getting rid of construction like [as name] give us information about full name or nickname of artists. These can permit us to avoid duplicata or obtain full names of artists (our actual version uniquely keep first found entry).

The idea is then to add id of artist in has\_ table (same idea as used for websites).

[CODE]