Deliverable 1

Assumptions

Using the given data, we identify several attributes that are not filled most of the time, those attributes are marked in red in the ER schema. If future queries don't need those attributes we might want to delete them since they don't bring much information to the user.
By looking at the data, we deduced the relationships between the entities. Furthermore, it allowed us to lessen our constraints that we initially thought.
Entity Relationship Schema

Schema

See annex 1.

Description

We found several constraints on the relationship.

- **Story:** Since the Story entity contains an issue_id attribute, we defined a relationship between Issue and Story. By looking at the data, we saw that issue_id is not always filled, hence we conclude that Story is contained in at most one issue. Furthermore, Story always contains a single type.
- **Story_Reprint:** We decided to make a weak entity, because if Story is deleted, Story_Reprint doesn't have any sense anymore, so it needs to be deleted too.
- **Issue_Reprint:** Same reasoning was applied to issue_reprint. We decided to make a weak entity, because if Issue is deleted, this entity should be deleted too.
- **Issue:** Issue contains several relationships. For instance it is contained in at most one series because by looking at the data, we saw that series_id is not always given. We apply the same reasoning with indicia_publisher_id.
- **Series:** Series has a lot of relationship. First at all, it contains at least one issue. Indeed, by looking at the data, we saw that the first_issue_id and last_issue_id are always set, so we deduces that there was always at least one Issue in a Serie, and exactly one first_issue and last_issue. Moreover, the country_id is always given and points to only one country. Language, Series_Publication_Type and Publisher are not always given, and when they are, there is only one. So we used an at most one constraint.
- **Publisher:** The country_id of Publisher is always given.
- **Indicia Publisher:** An Indicia_publisher is always contained in exactly one publisher and is located in exactly one country.
- **Brand Group:** A Brand Group is always contained in exactly one publisher.

Relational Schema

ER schema to Relational schema

Since we always have an exactly one or at most one constraint in every relationship, we don't need any intermediate table to represent the relationship. Moreover, defining the primary keys and the foreign keys was straightforward.
The primary keys are always the id of the entity, and the foreign keys the attributes that point to another entity.

DDL

See annex 2.

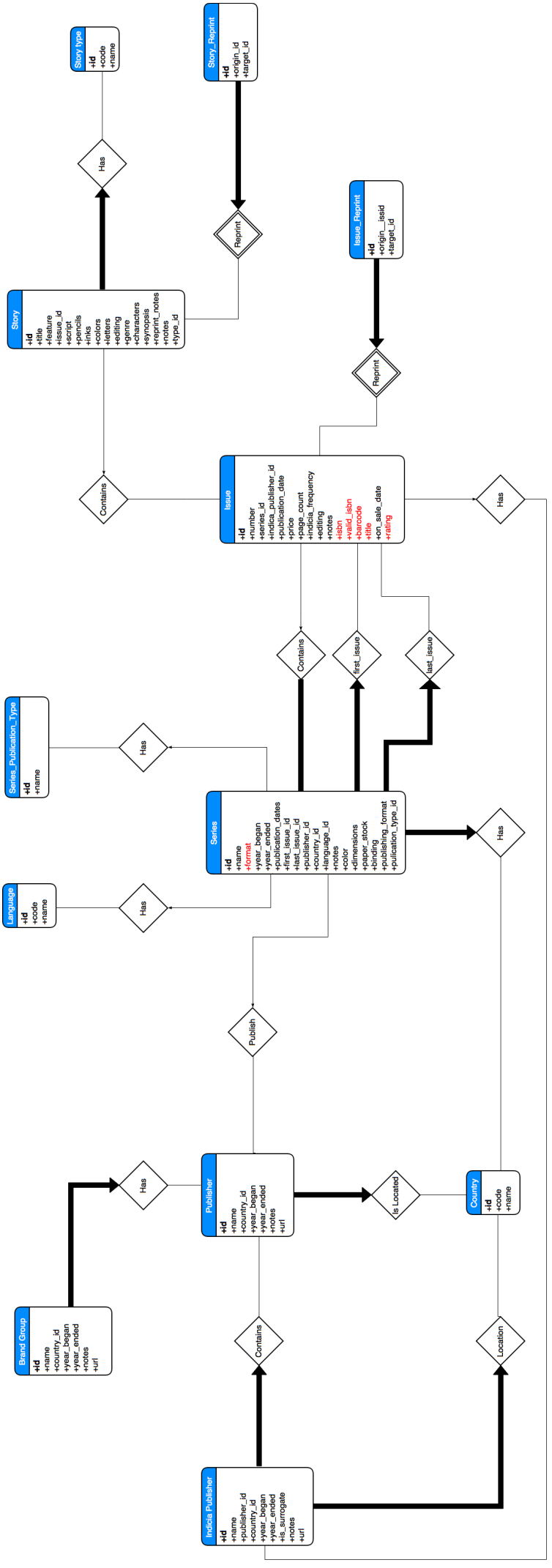
General Comments

At the beginning, we started by trying to define the relationships logically, using common sense and what we could expect to have in the real world. But then we noticed that our constraints were too restrictive. So we decided to look at the given data and conclude the relationships from there, this allowed us to lessen our constraints and provide a more flexible design.
The work has been equally split up between the 3 members of the group, each one was given a different task and we met up every Wednesday to evaluate the progress of everyone and to define the tasks for the upcoming weeks.

Group 2

27.03.2017

Annex 1



Annex 2

```
CREATE TABLE Story (
        id INTEGER,
        title TEXT,
        feature CHAR(64),
        issue_id INTEGER,
        script TEXT,
        pencils CHAR(64),
        inks CHAR(64),
        colors CHAR(64),
        letters CHAR(64),
        editing TEXT,
        genre CHAR(64),
        characters TEXT,
        synopsis TEXT,
        reprint_notes TEXT,
        notes TEXT,
        type_id INTEGER NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (issue_id) REFERENCES Issue(id),
        FOREIGN KEY (type_id) REFERENCES Story_Type(id)
)

CREATE TABLE Issue (
        id INTEGER,
        issue_number CHAR(32),
        series_id INTEGER,
        indicia_publisher_id INTEGER,
        publication_date CHAR(64),
        price TEXT,
        page_count INTEGER,
        indicia_frequency CHAR(64),
        editing TEXT,
        notes TEXT,
        isbn TEXT,
        valid_isbn TEXT,
        barcode TEXT,
        title TEXT,
        on_sale_date DATE,
        rating INTEGER,
        PRIMARY KEY (id),
        FOREIGN KEY (series_id) REFERENCES Series(id),
        FOREIGN KEY (indicia_publisher_id) REFERENCES Indicia_Publisher(id)
)
```

```
CREATE TABLE Series (
        id INTEGER,
        name TEXT,
        format TEXT,
        year_began INTEGER,
        year_ended INTEGER,
        publication_dates CHAR(64),
        first_issue_id INTEGER NOT NULL,
        last_issue_id INTEGER NOT NULL,
        publisher_id INTEGER,
        country_id INTEGER NOT NULL,
        language_id INTEGER,
        notes TEXT,
        color TEXT,
        dimensions TEXT,
        paper_stock TEXT,
        binding TEXT,
        publishing_format TEXT,
        publication_type_id INTEGER,
        PRIMARY KEY (id),
        FOREIGN KEY (first_issue_id) REFERENCES Issue(id),
        FOREIGN KEY (last_issue_id) REFERENCES Issue(id),
        FOREIGN KEY (publisher_id) REFERENCES Publisher(id),
        FOREIGN KEY (language_id) REFERENCES Language(id),
        FOREIGN KEY (country_id) REFERENCES Country(id)
)

CREATE TABLE Indicia_Publisher (
        id INTEGER,
        name TEXT,
        publisher_id INTEGER NOT NULL,
        country_id INTEGER NOT NULL,
        year_began INTEGER,
        year_ended INTEGER,
        is_surrogate INTEGER,
        notes TEXT,
        url CHAR(256),
        PRIMARY KEY (id),
        FOREIGN KEY (publisher_id) REFERENCES Publisher(id),
        FOREIGN KEY (country_id) REFERENCES Country(id)
)
```

```
CREATE TABLE Publisher (
        id INTEGER,
        name TEXT,
        country_id INTEGER NOT NULL,
        year_began INTEGER,
        year_ended INTEGER,
        notes TEXT,
        url CHAR(256),
        PRIMARY KEY (id),
        FOREIGN KEY (country_id) REFERENCES Country(id)
)

CREATE TABLE Brand_Group (
        id INTEGER,
        name TEXT,
        year_began INTEGER,
        year_ended INTEGER,
        notes TEXT,
        url CHAR(256),
        publisher_id INTEGER NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (publisher_id) REFERENCES Publisher(id)
)

CREATE TABLE Country (
        id INTEGER,
        code CHAR(8),
        name TEXT,
        PRIMARY KEY (id)
)

CREATE TABLE Language (
        id INTEGER,
        code CHAR(8),
        name TEXT,
        PRIMARY KEY (id)
)

CREATE TABLE Series_Publication_Type (
        id INTEGER,
        name CHAR(16),
        PRIMARY KEY (id)
)
```

```
CREATE TABLE Story_Type (
        id INTEGER,
        name TEXT,
        PRIMARY KEY (id)
)

CREATE TABLE Story_Reprint (
        id INTEGER,
        origin_id INTEGER NOT NULL,
        target_id INTEGER NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (origin_id) REFERENCES Story(id),
        FOREIGN KEY (target_id) REFERENCES Story(id)
)

CREATE TABLE Issue_Reprint (
        id INTEGER,
        origin_issue_id INTEGER NOT NULL,
        target_target_id INTEGER NOT NULL,
        PRIMARY KEY (id),
        FOREIGN KEY (origin_issue_id) REFERENCES Issue(id),
        FOREIGN KEY (target_target_id) REFERENCES Issue(id)
)
```