Database Project

First Deliverable

1 Design Choices

The overall design choice we made was to stick as close as possible to the given data while keeping useful information and reducing redundancy.

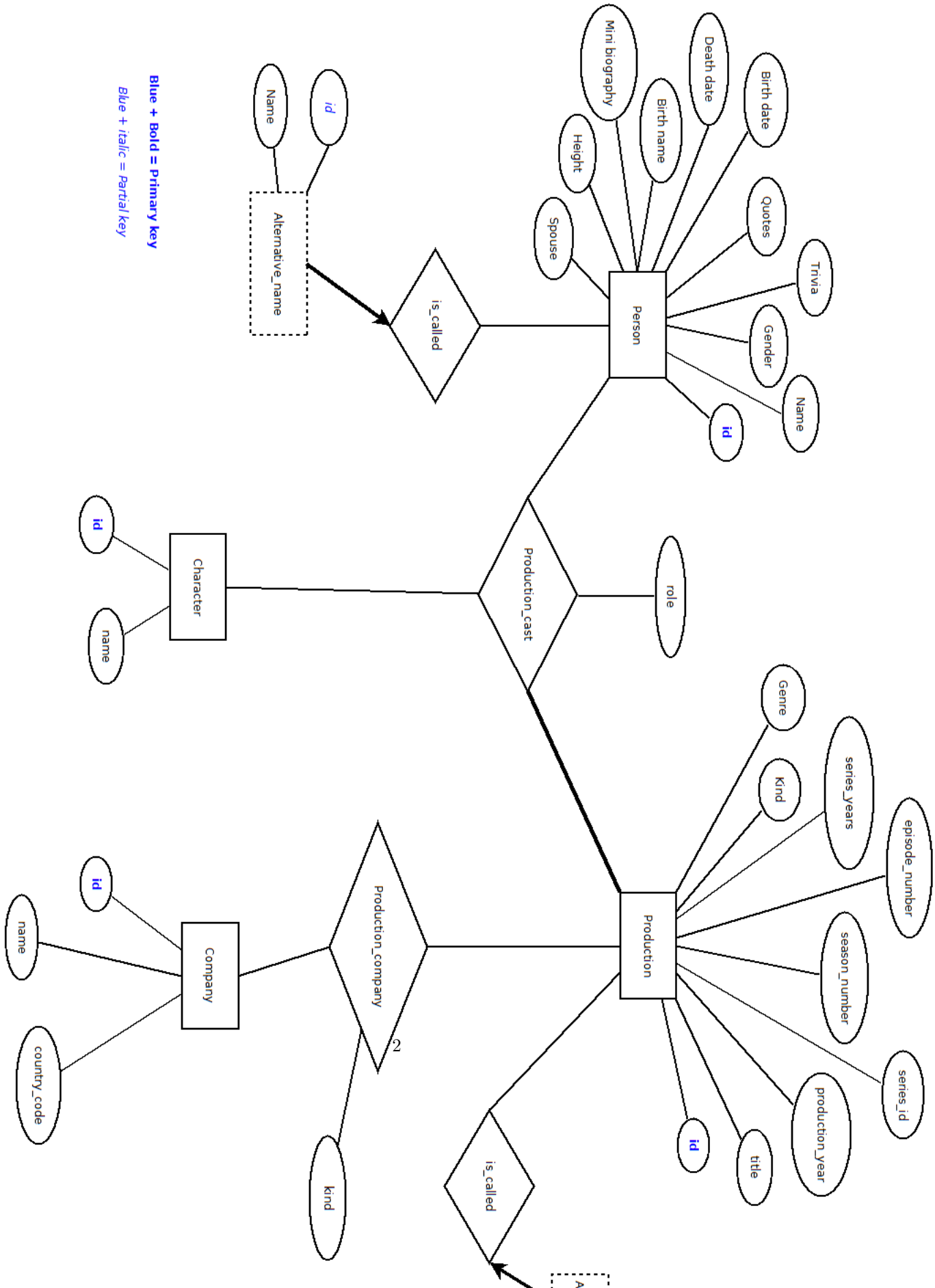
Let's start with real person in the universe, we obviously described them as an entity called Person. This entity is in a week entity relationship with the Alternative_name entity as it makes no sense to have an alternative name that is not linked to a person. We implemented the weak entity with "ON DELETE CASCADE" such that we never have existing alternative name linked to deleted Person in the database.

We decided to have a 3 way relationship with the Person, Character and Production entities as we felt that linking these entities in a relationship was the best way to form tuples that represent something that produces movies. If a person produces a movies but isn't a character then the character_id field is set to null, this way let's us link directly person to character in the same table. We also link Person and Character to Production_cast with many to many relationship because a person can produce several movies in different production and play many different character also since a character could be played by no real person (in computer generated movies for example) the arrow between Character and Production_cast doesn't need any constraint, Finally anyone in the production cast needs at least one production therefore we connected the relationship with a bold line between Production and Production_Cast. The rest is self explanatory.

The production entity also posses a weak entity relationship with an alternative title the exact same way as Person does with Alternative_name.

And finally we have that the Company entity produces Production. This is seen as a many to many relationship with attributes because a Company can produces several production, but a production can have zero (or many) company that produces it (indie movie or collaborative movies).

2 ER Model


Blue + Bold = Primary key
Blue + italic = Partial key
Person
Mini biography
Death date
Birth date
Birth name
Height
Spouse
Quotes
Trivia
Gender
Name
id
is_called
Alternative_name
Name
id
Production_cast
role
Character
id
name
Production_company
Company
id
name
country_code
kind
Production
Genre
Kind
series_years
episode_number
season_number
series_id
production_year
title
id
is_called

3 QL DDL code for table creation

```
CREATE TABLE Person
(
        id VARCHAR(20) NOT NULL,
        Name VARCHAR(100),
        Gender VARCHAR(5),
        Trivia VARCHAR(255),
        Quotes VARCHAR(255),
        Birth_date DATE,
        Death_date DATE,
        Birth name VARCHAR(255),
        Mini biography TEXT, -- Using MySQL or PostgreSQL, this will allow to
                        -- have more than 4000 characters.
        Spouse VARCHAR(100),
        Height VARCHAR(5),

        PRIMARY KEY(id)
)

CREATE TABLE Characters
(
        id VARCHAR(20) NOT NULL,
        name VARCHAR(100),

        PRIMARY KEY(id)
)

CREATE TABLE Company
(
        id VARCHAR(20) NOT NULL,
        country_code VARCHAR(10),
        name VARCHAR(50),

        PRIMARY KEY(id)
)

CREATE TABLE Production
(
        id VARCHAR(20) NOT NULL,
        title VARCHAR(100),
        production_year CHAR(4),
        series_id CHAR(9), -- Here we fix the length to 9.
    -- No id will be > than 999 999 999. And it's faster than VARCHAR
        season_number CHAR(2),
        episode_number CHAR(4),
        series_years VARCHAR(10),
        kind CHAR(10),
```

```
        genre VARCHAR(20),

        PRIMARY KEY(id)
)


CREATE TABLE Alternative_title
(
        id VARCHAR(20) NOT NULL,
        title VARCHAR(50),
        prod_id VARCHAR(20) NOT NULL,

        PRIMARY KEY (id, prod_id),
        FOREIGN KEY (prod_id) REFERENCES Production(id),
                ON DELETE CASCADE
)

CREATE TABLE Alternative_name
(
        id VARCHAR(20) NOT NULL,
        Name VARCHAR(50),
        person_id VARCHAR(20) NOT NULL,

        PRIMARY KEY (id, person_id),
        FOREIGN KEY (person_id) REFERENCES Person(id),
                ON DELETE CASCADE
)

CREATE TABLE Production_cast
(
        production_id VARCHAR(20) NOT NULL,
        person_id VARCHAR(20),
        character_id VARCHAR(20),
        role VARCHAR(20),

        PRIMARY KEY (production_id, person_id, character_id),
        FOREIGN KEY (production_id) REFERENCES Production,
        FOREIGN KEY (person_id) REFERENCES Person,
        FOREIGN KEY (character_id) REFERENCES Characters
)

CREATE TABLE Production_company
(
        production_id VARCHAR(20),
        company_id VARCHAR(20),
        kind VARCHAR(50),

        PRIMARY KEY (production_id, company_id),
```

```
        FOREIGN KEY (production_id) REFERENCES Production,
        FOREIGN KEY (company_id) REFERENCES Company
)
```

Second Deliverable

4 Having trouble

26. April 2015

We had a lot of trouble parsing and cleaning the data and it took us way more time than expected so we were not able to finish all the steps for Deliverable 2.

We had to change our table design since we didn't take into account the fact that there were dates in the data so we added DATE fields in the SQL tables. We decided to use two date field for the date range in PRODUCTION. We also had to parse the height of person in the PERSON table. Regarding the feedback on Deliverable 1, we decided to add a surrogate key to the PRODUCTION_CAST table as a primary key.

Most of the data is imported into the given oracle DB right now (only PRODUCTION_CAST isn't) and we could establish a connection through php to the database so we are on good shape to catch up and build an interface.