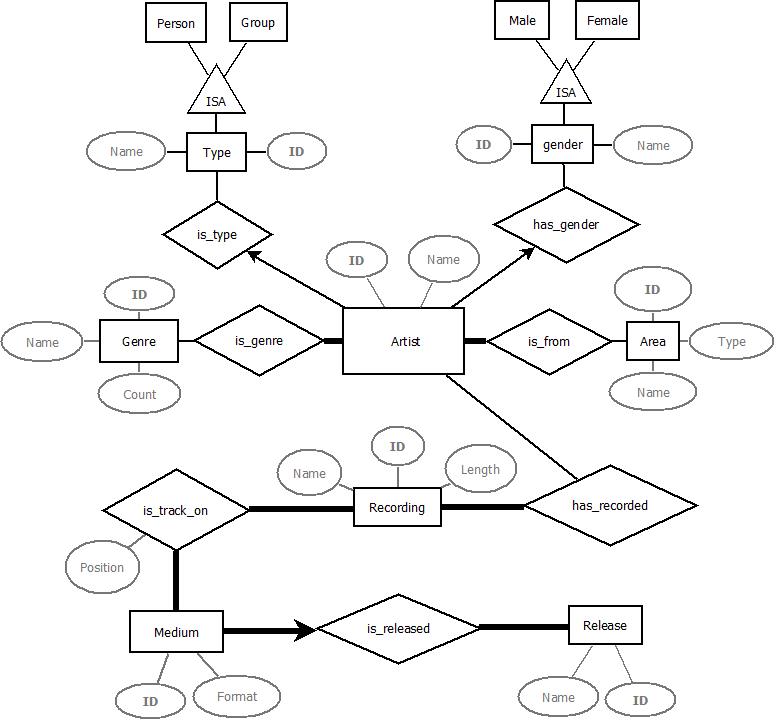
**1 Deliverable 1**

**1.1 ER Diagram**



**1.2 Design choices**

Because we cannot ask customers for their expectations about the database, we did several assumptions listed below. The goal was to maximize the number of caught constraints in our design while avoiding duplication. These are subject to change over time with our design evolution.

1. An artist can have several origins (AREAS) and several genres. However he can only have at most one GENDER and at most one TYPE when those are given.
2. Ideally every artist of type « Group » should have its gender attribute set to « Other ». This is a constraint that we cannot capture in our design.
3. We keep a name field in tables TYPE and GENDER to assure consistency in the beginning of this project, but it would be possible to get rid of them and use only the \_id field to establish an artist’s type or gender by the use of some arbitrary convention.
4. Every recordings in the database should be related to at least one artist, but an artist can have zero recording.
5. All the recordings must appear on at least one medium. A medium must have at least one recording.
6. With the add of a position attribute to the relation is\_track\_on, we capture the definition of a track.
7. A tuple of medium is related to exactly one release.
8. There is no cascade in deletion.

**1.3 SQL DDL code for table creation**

CREATE TABLE Type (

Type\_id INTEGER,

Name CHAR (10),

PRIMARY KEY (Type\_id)) ;

CREATE TABLE Gender (

Gender\_id INTEGER,

Name CHAR (10),

PRIMARY KEY (Gender\_id)) ;

CREATE TABLE Genre (

Genre\_id INTEGER,

Name CHAR (50),

Count INTEGER,

PRIMARY KEY (Genre\_id)) ;

CREATE TABLE Area (

Area\_id INTEGER,

Name CHAR (100),

Type\_of\_area CHAR (100),

PRIMARY KEY (Area\_id)) ;

CREATE TABLE Artist (

Artist\_id INTEGER,

Name CHAR (30),

Type\_id INTEGER,

Gender\_id INTEGER,

Genre\_id INTEGER NOT NULL,

Area\_id INTEGER NOT NULL,

PRIMARY KEY (Artist\_id),

FOREIGN KEY (Type\_id) REFERENCES Type,

FOREIGN KEY (Gender\_id) REFERENCES Gender,

FOREIGN KEY (Genre\_id) REFERENCES Genre,

FOREIGN KEY (Area\_id) REFERENCES Area) ;

CREATE TABLE Recording (

Recording\_id INTEGER,

Name CHAR (100),

Length INTEGER,

PRIMARY KEY (Recording\_id)) ;

CREATE TABLE has\_recorded (

Artist\_id INTEGER, // implicit NOT NULL (used in primary key)

Recording\_id INTEGER, // implicit NOT NULL (idem)

PRIMARY KEY (Artist\_id, Recording\_id),

FOREIGN KEY (Artist\_id) REFERENCES Artist,

FOREIGN KEY (Recording\_id) REFERENCES Recording) ;

CREATE TABLE Medium (

Medium\_id INTEGER,

Format CHAR (30),

Release\_id INTEGER NOT NULL,

PRIMARY KEY (Medium\_id),

FOREIGN KEY (Release\_id) REFERENCES Release) ;

CREATE TABLE Release (

Release\_id INTEGER,

Name CHAR (200),

PRIMARY KEY (Release\_id)) ;

CREATE TABLE is\_track\_on (

Recording\_id INTEGER,

Medium\_id INTEGER,

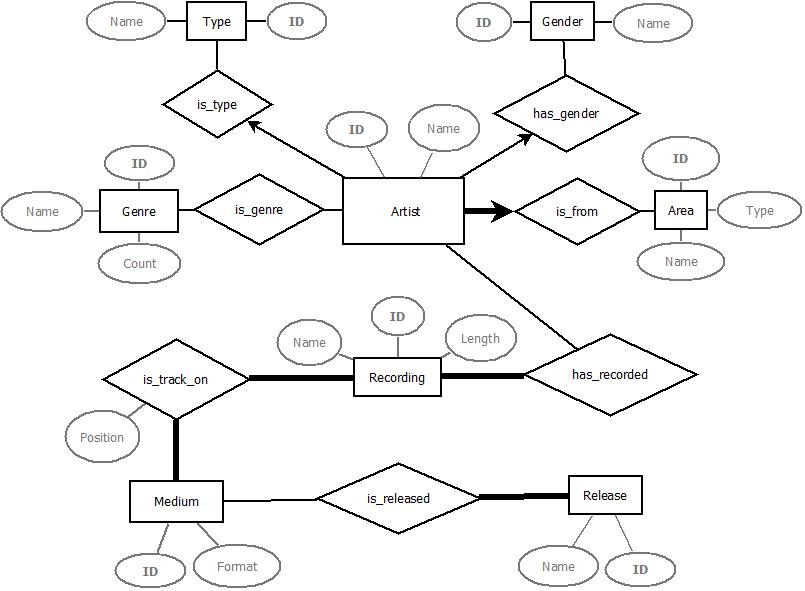
Position INTEGER,

PRIMARY KEY (Recording\_id, Medium\_id, Position),

FOREIGN KEY (Recording\_id) REFERENCES Recording,

FOREIGN KEY (Medium\_id) REFERENCES Medium) ;

**2 Deliverable 2**

**2.1 ER Diagram**

**2.2 Design Choices**

From the feedbacks of the deliverable 1, we made some improvements to our diagram.

First we got rid of the ISAs relationships because they were carrying more complexity than help. Each Type and Gender tables will contain few entries, respectively {‘Person’, ‘Group’} and {‘Male’, ‘Female’, ‘Other’}. This could allow us to add an eventual new Type if needed (not expected in this project though).

We limited the ability for an artist to be from more than one Area as proposed in the feedback. Now an artist must be from exactly one Area.

We allowed an artist to have no genre. Even though the number of genre should cover most needs we thought it was a useless limitation.

The new relation between Medium and Release allow the first to be part of several or none Release (many-to-many relation). We kept the fact that a Release must have at least one Medium.

**2.3 SQL DDL Code for table creation**

CREATE TABLE Type (

Type\_id INTEGER,

Name CHAR (10),

PRIMARY KEY (Type\_id)) ;

CREATE TABLE Gender (

Gender\_id INTEGER,

Name CHAR (10),

PRIMARY KEY (Gender\_id)) ;

CREATE TABLE Genre (

Genre\_id INTEGER,

Name CHAR (50),

Count INTEGER,

PRIMARY KEY (Genre\_id)) ;

CREATE TABLE Area (

Area\_id INTEGER,

Name CHAR (100),

Type\_of\_area CHAR (100),

PRIMARY KEY (Area\_id)) ;

CREATE TABLE Artist (

Artist\_id INTEGER,

Name CHAR (30),

Type\_id INTEGER,

Gender\_id INTEGER,

Area\_id INTEGER NOT NULL,

PRIMARY KEY (Artist\_id),

FOREIGN KEY (Type\_id) REFERENCES Type,

FOREIGN KEY (Gender\_id) REFERENCES Gender,

FOREIGN KEY (Area\_id) REFERENCES Area) ;

CREATE TABLE is\_genre (

Artist\_id INTEGER,

Genre\_id INTEGER,

PRIMARY KEY (Artist\_id, Genre\_id),

FOREGIN KEY (Artist\_id) REFERENCES Artist,

FOREIGN KEY (Genre\_id) REFERENCES Genre) ;

CREATE TABLE Recording (

Recording\_id INTEGER,

Name CHAR (100),

Length INTEGER,

PRIMARY KEY (Recording\_id)) ;

CREATE TABLE has\_recorded (

Artist\_id INTEGER, // implicit NOT NULL (used in primary key)

Recording\_id INTEGER, // implicit NOT NULL (idem)

PRIMARY KEY (Artist\_id, Recording\_id),

FOREIGN KEY (Artist\_id) REFERENCES Artist,

FOREIGN KEY (Recording\_id) REFERENCES Recording) ;

CREATE TABLE Medium (

Medium\_id INTEGER,

Format CHAR (30),

PRIMARY KEY (Medium\_id)) ;

CREATE TABLE Release (

Release\_id INTEGER,

Name CHAR (200),

PRIMARY KEY (Release\_id)) ;

CREATE TABLE is\_track\_on (

Recording\_id INTEGER,

Medium\_id INTEGER,

Position INTEGER,

PRIMARY KEY (Recording\_id, Medium\_id, Position),

FOREIGN KEY (Recording\_id) REFERENCES Recording,

FOREIGN KEY (Medium\_id) REFERENCES Medium) ;

CREATE TABLE is\_released (

Medium\_id INTEGER,

Release\_id INTEGER,

PRIMARY KEY (Medium\_id, Release\_id),

FOREIGN KEY (Medium\_id) REFERENCES Medium,

FOREIGN KEY (Release\_id) REFERENCES Release) ;