1. Determine the concept that needs to be stored
    - Magazine
    - MagazineVolume
    - Articles
    - Author
    - ArticleAuthor
    - MagazineOrder
2. Determine attributes of each concept
    (a) Magazine
        - issn
        - title
        - number
        - year

    (b) Articles
        - art_no
        - title
        - begin_page
        - end_page

    (c) Author
        - auth_no
        - name
        - email
        - ascription

    (d) MagazineOrder
        - ord_no
        - number
3. Determine links (relationships) between them
    - Magazine (has) Articles
    - Articles (has) Authors
    - Magazine (has) MagazineOrder
4. Determine types of attributes
    (a) Magazine
        - issn **varchar(20)**
        - title **varchar(50)**

- number **int**
- year **varchar(10)**

(b) Articles

- art_no **int**
- title **varchar(50)**
- begin_page **int**
- end_page **int**

(c) Author

- auth_no **int**
- name **varchar(50)**
- email **varchar(50)**
- ascription **varchar(50)**

(d) MagazineOrder

- ord_no **int**
- number **int**

5. Solve foreign key links:

(a) add primary key

i. Magazine

- **issn varchar(20)**
- title varchar(50)
- number int
- year varchar(10)

ii. Articles

- **art_no int**
- title varchar(50)
- begin_page int
- end_page int

iii. Author

- **auth_no int**
- name varchar(50)
- email varchar(50)
- ascription varchar(50)

iv. MagazineOrder

- **ord_no int**
- number int

(b) add foreign key for n-n

i. Magazine

- issn varchar(20)
- title varchar(50)
- ~~number int~~
- year varchar(10)

ii. MagazineVolume
- mag_no int
- issn varchar(20)
- number int

iii. Articles
- art_no int
- **mag_no int**
- title varchar(50)
- begin_page int
- end_page int

iv. Author
- auth_no int
- name varchar(50)
- email varchar(50)
- ascription varchar(50)

v. ArticleAuthor
- id int
- art_no int
- auth_no int

vi. MagazineOrder
- ord_no int
- **mag_no int**
- number int

(c) add foreign key for 1-n

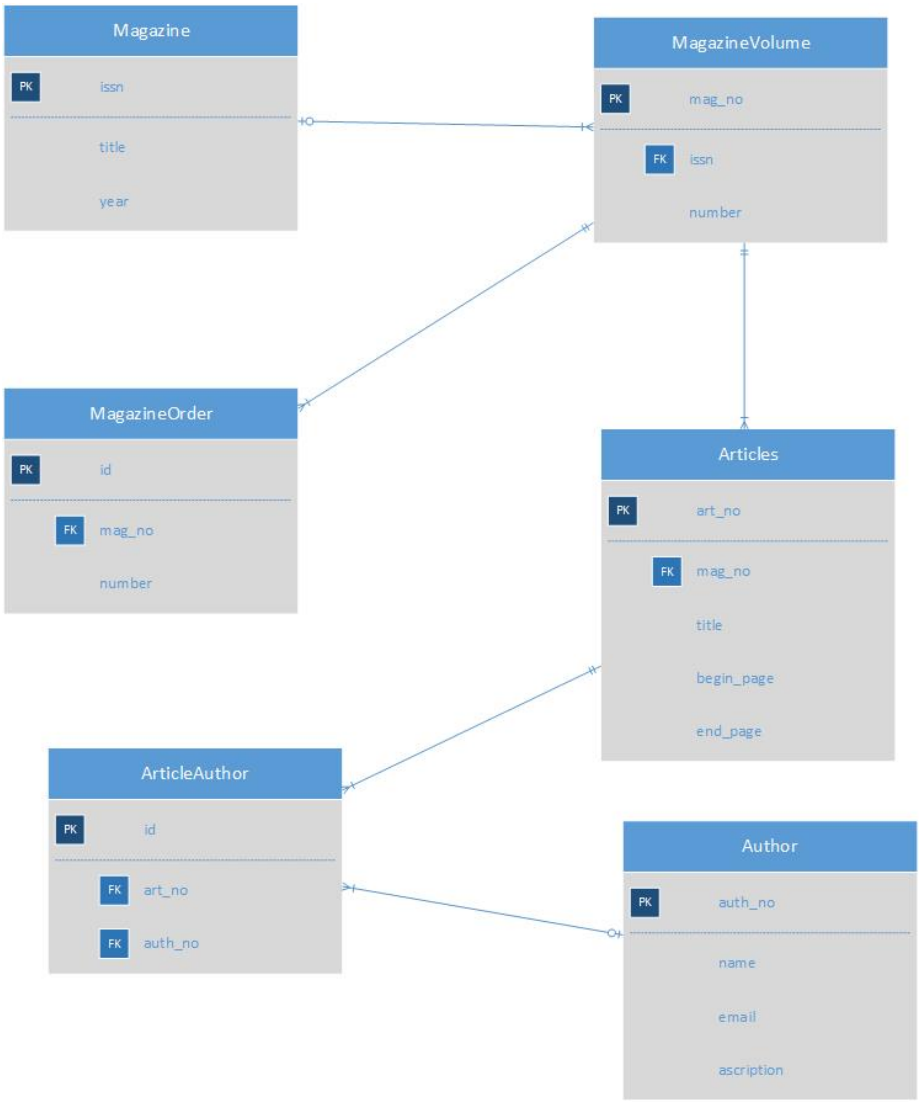
i. Magazine
- issn varchar(20)
- title varchar(50)
- number int
- year varchar(10)

ii. Articles
- art_no int
- **mag_no int**
- title varchar(50)
- begin_page int
- end_page int

iii. Author
   - auth_no int
   - name varchar(50)
   - email varchar(50)
   - ascription varchar(50)

iv. MagazineOrder
   - ord_no int
   - **mag_no int**
   - number int

6. EERD Diagram



7. Implementation

   (a) Database

   ```
   CREATE database MagazineDB;
   ```

   (b) Tables

   ```
   CREATE TABLE Magazine(
   issn varchar(20) primary key,
   title varchar(50),
   ```

```
year varchar(10)
);

CREATE TABLE MagazineVolume(
mag_no int primary key auto_increment,
issn varchar(20),
number int,
foreign key(issn) references Magazine(issn)
);

CREATE TABLE Articles(
art_no int primary key auto_increment,
mag_no int,
title varchar(50) UNIQUE,
begin_page int,
end_page int,
foreign key(mag_no) references MagazineVolume(mag_no)
);

CREATE TABLE Author(
auth_no int primary key auto_increment,
name varchar(50),
email varchar(50),
ascription varchar(50)
);

CREATE TABLE ArticleAuthor(
id int primary key auto_increment,
art_no int,
auth_no int,
foreign key(art_no) references Articles(art_no),
foreign key(auth_no) references Author(auth_no)
);

CREATE TABLE MagazineOrder(
ord_id int primary key auto_increment,
mag_no int,
number int,
foreign key(mag_no) references MagazineVolume(mag_no)
);
```

(c) Show Tables

```
SHOW tables;

+----------------------+
| Tables_in_magazinedb |
```

```
+----------------------+
| articleauthor        |
| articles             |
| author               |
| magazine             |
| magazineorder        |
| magazinevolume       |
+----------------------+
6 rows in set (0.00 sec)
```