# 데이터관리와 분석

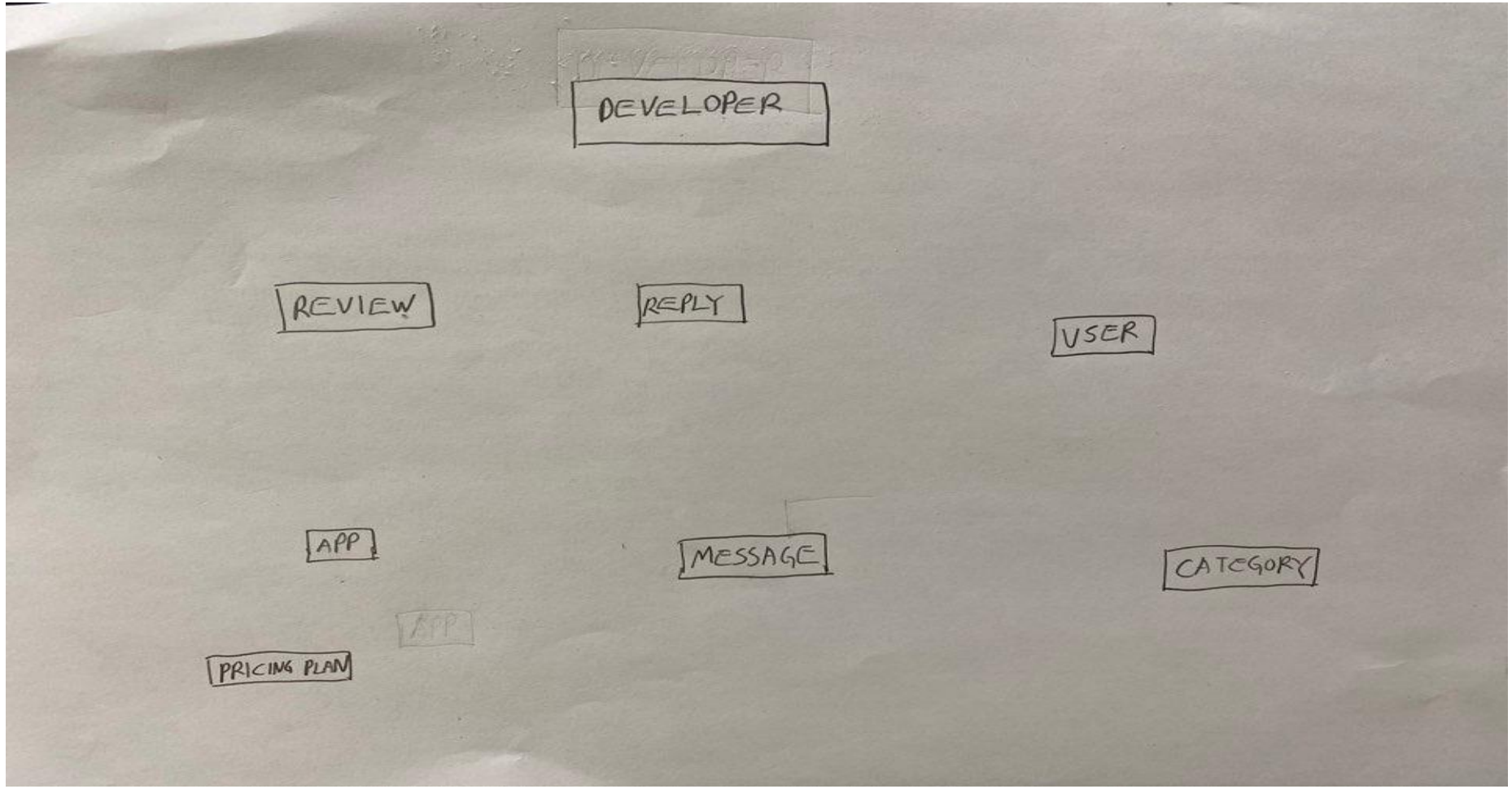
프로젝트#1: Conceptual DB design & DB implementation

12조
2017-15751 장준혁
2017-18868 조형찬
2017-10854 최의현
2017-13918 황재훈
2017-10089 윤성진

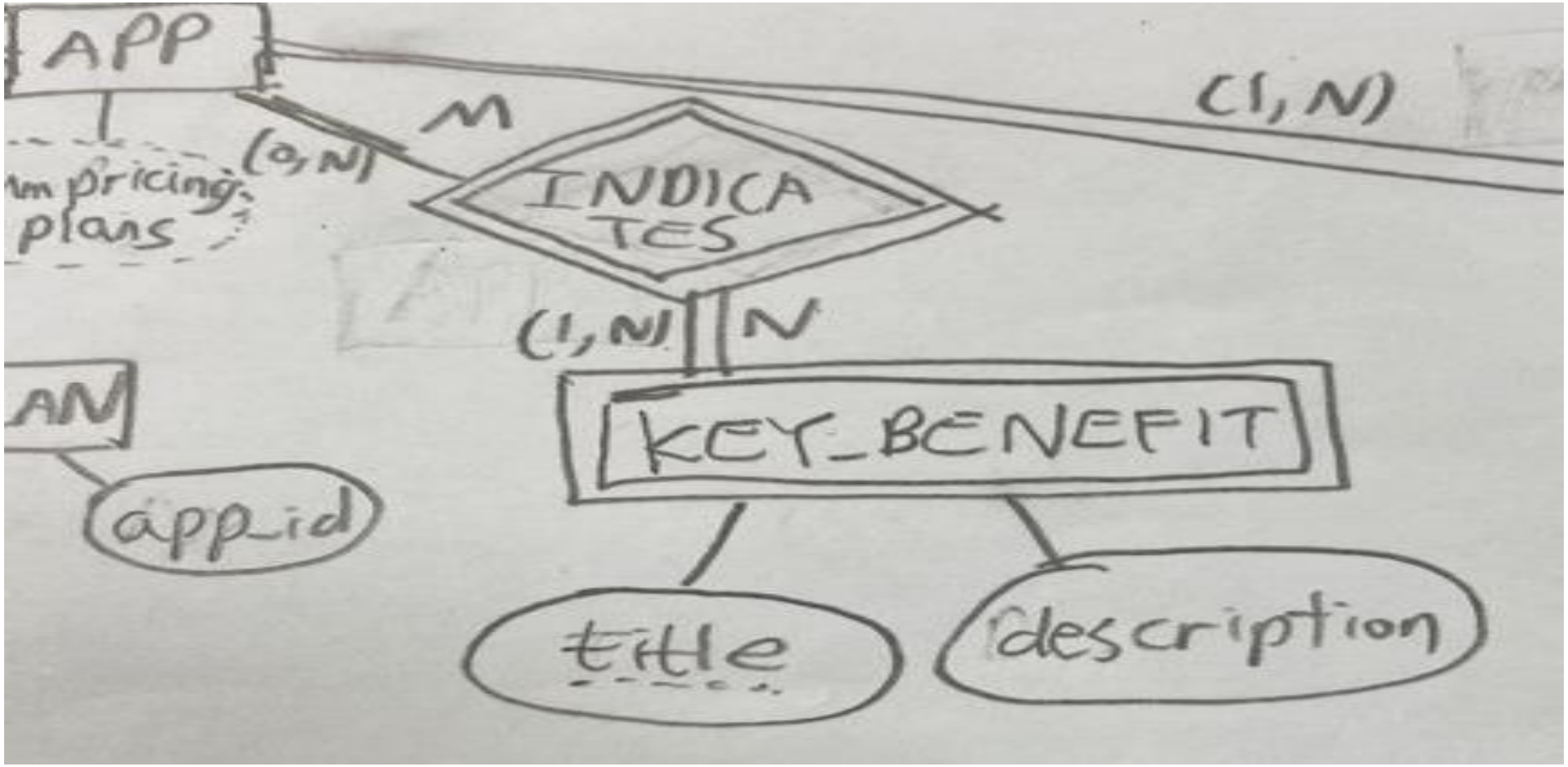
# 목차

# I. ER diagram 도식화

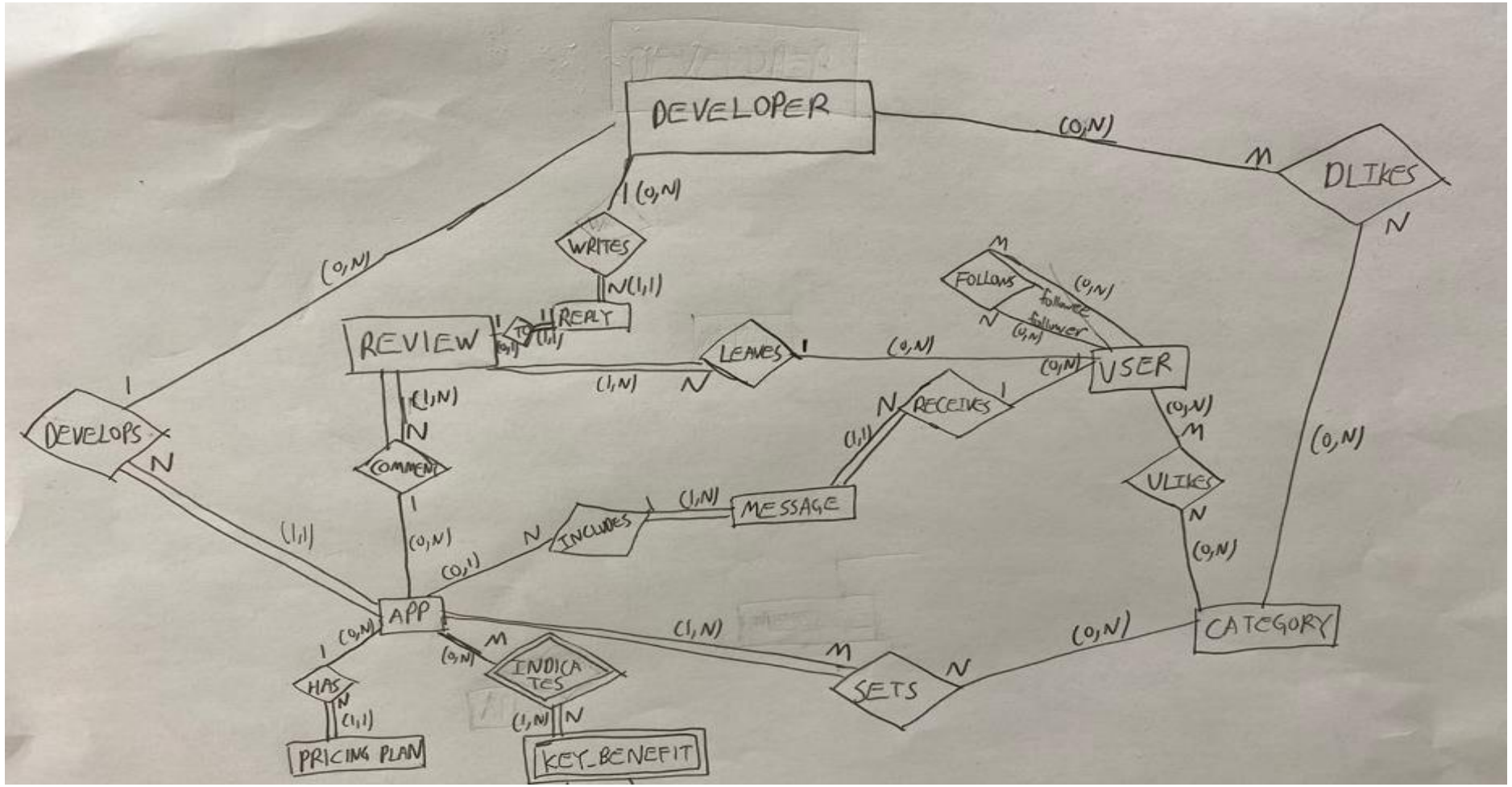
# 1) Entity Type 그리기



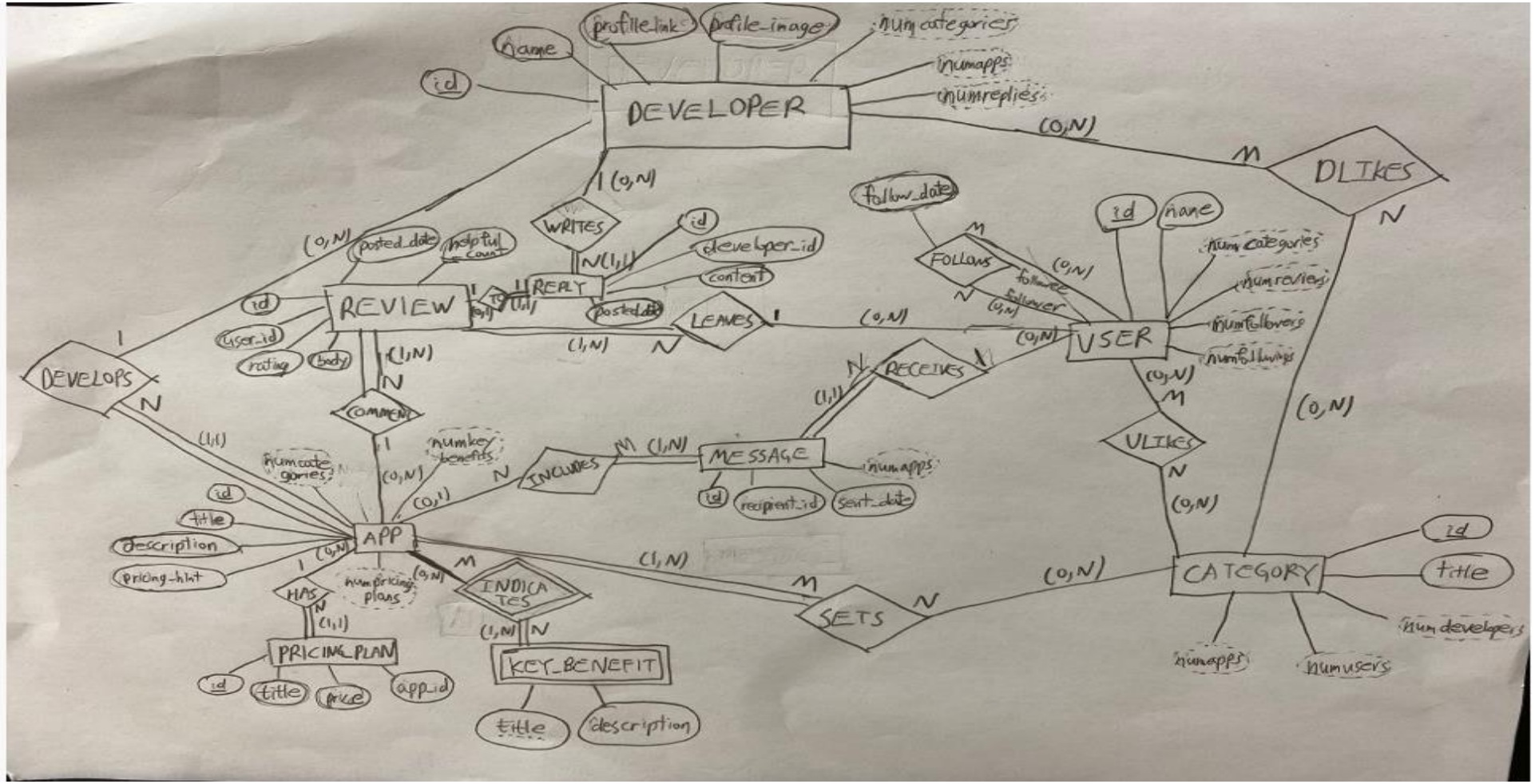
# 2) Weak Entity Type 그리기



# 3) Relationship Type 그리기



# 4) 최종 ER diagram



# Ⅱ. Relational DB model 구현

# 1) Regular Entity Type mapping

**DEVELOPER**

| id | name | profile_image | profile_link |
|---|---|---|---|

**USER**

| id | name |
|---|---|

**APP**

| id | title | developer_id | description | pricing_hint |
|---|---|---|---|---|

**CATEGORY**

| id | title |
|---|---|

**REPLY**

| id | review_id | developer_id | content | posted_date |
|---|---|---|---|---|

**REVIEW**

| id | user_id | rating | body | helpful_count | posted_date | app_id |
|---|---|---|---|---|---|---|

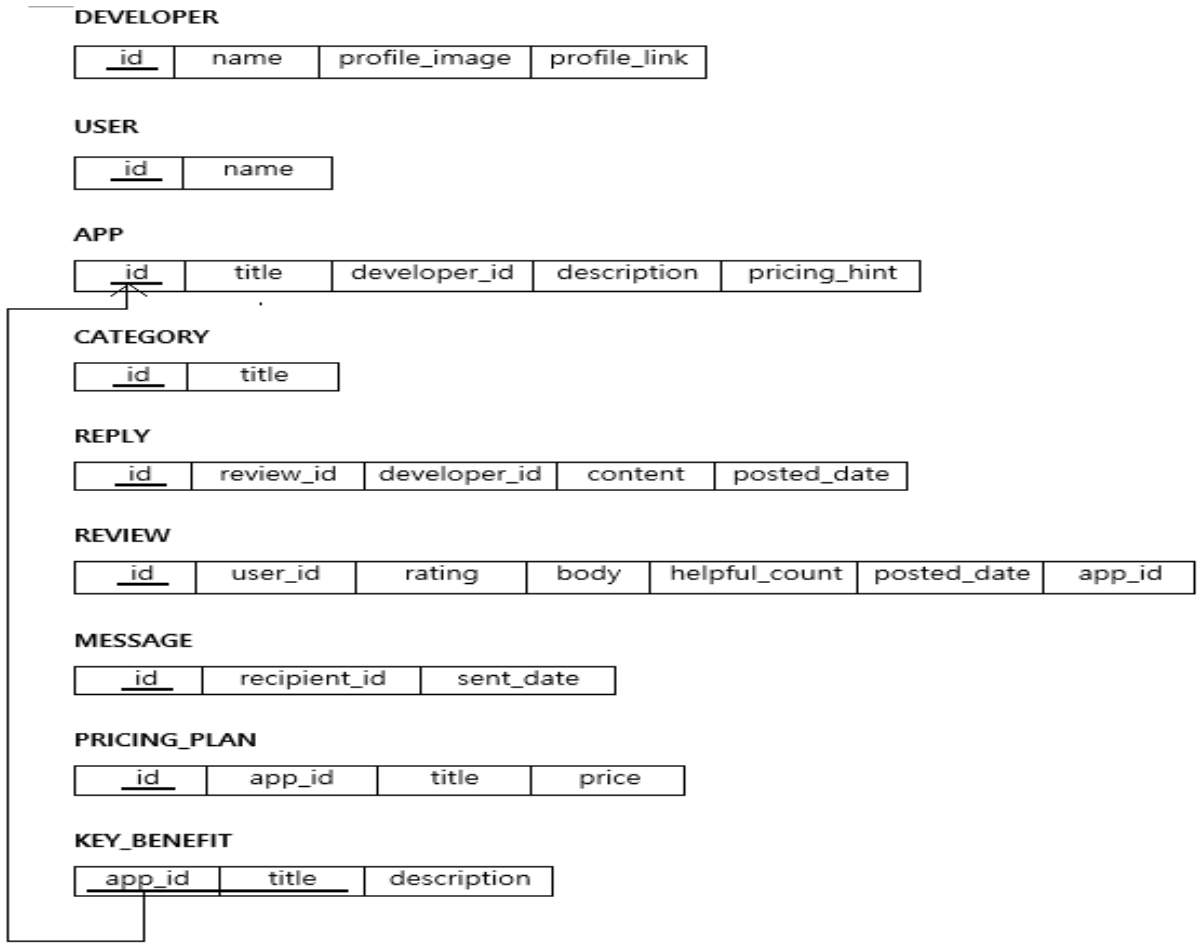
**MESSAGE**

| id | recipient_id | sent_date |
|---|---|---|

**PRICING_PLAN**

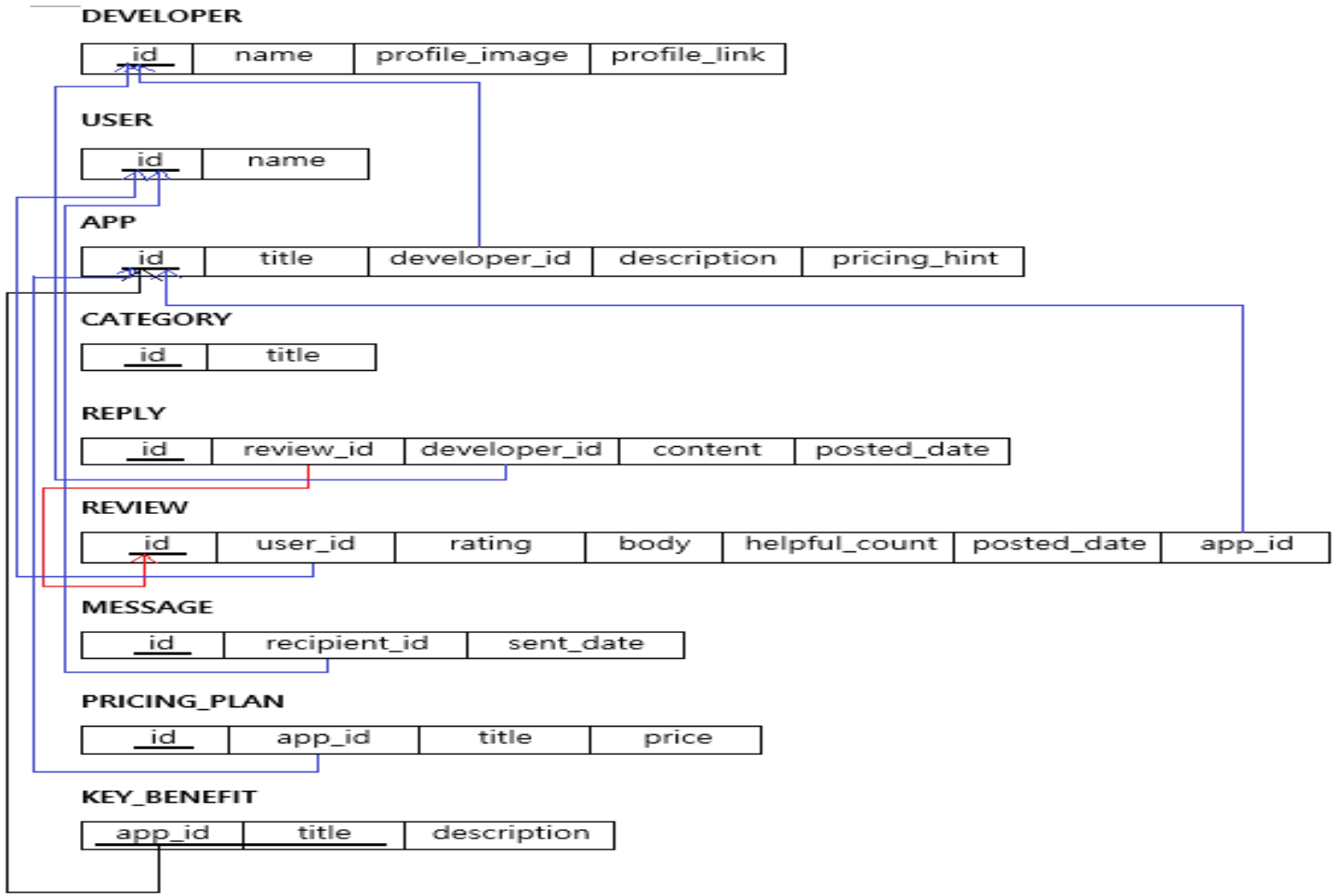
| id | app_id | title | price |
|---|---|---|---|

# 2) Weak Entity Type mapping

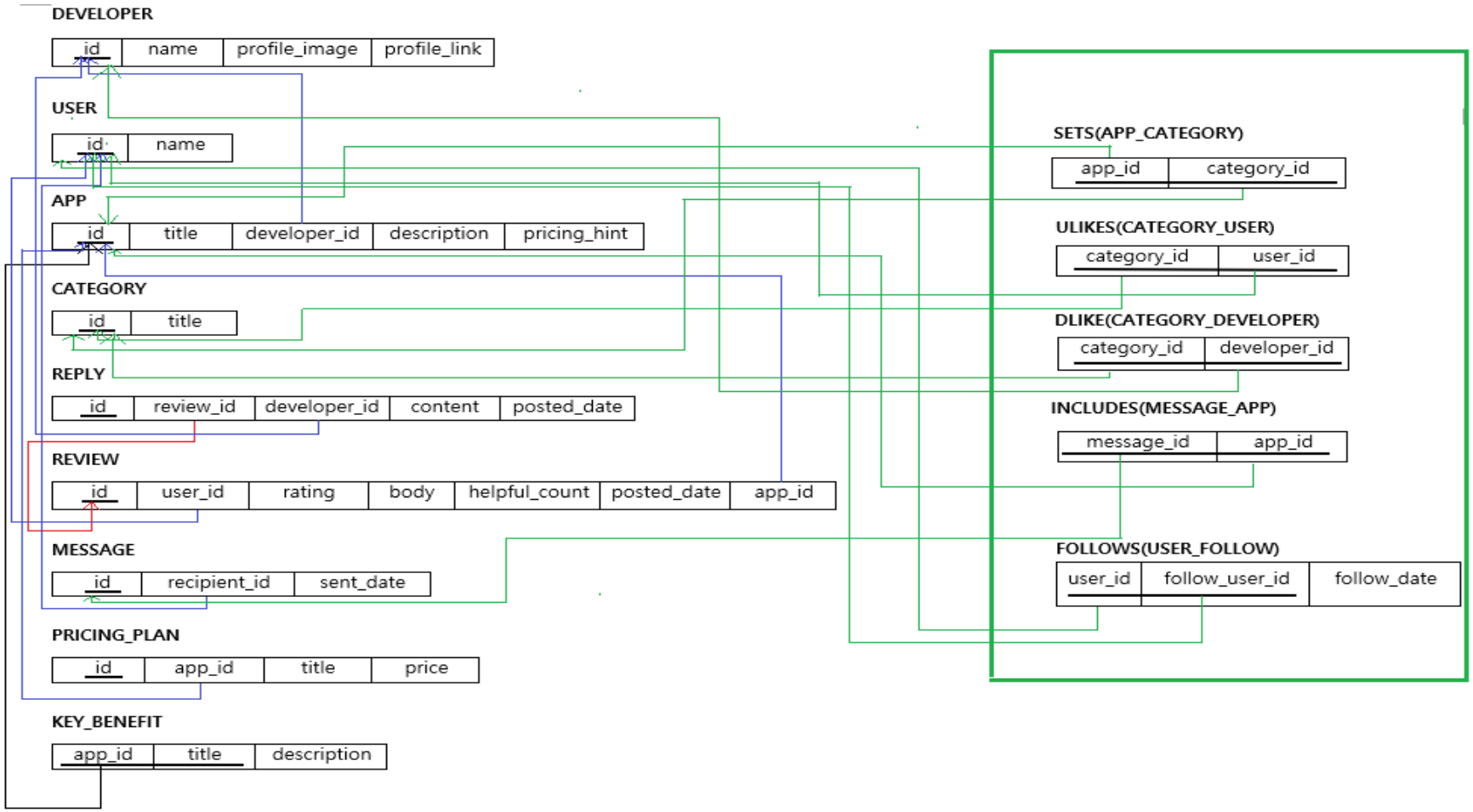


# 3) Binary 1:1 Relationship Type mapping

# 4) Binary 1:N Relationship Type mapping



# 5) Binary M:N Relationship Type mapping



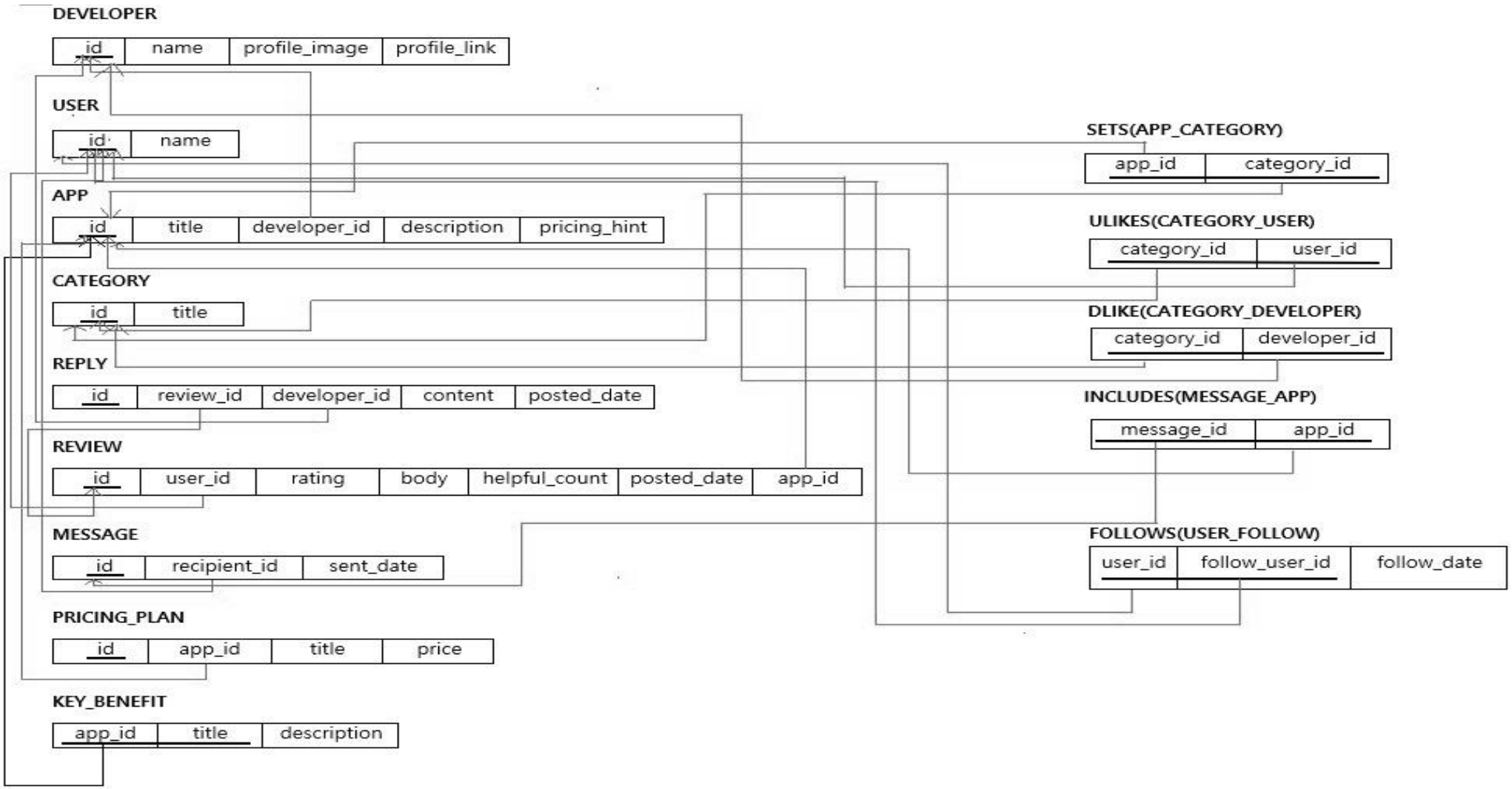
# 6) Multivalued Attributes mapping

Multivalued Attribute 존재하지 않음

# 7) N-ary Relationship Type mapping

N-ary Relationship 존재하지 않음

# 8) 최종 Relation DB model



# III. 데이터 입력

# 1) 프로그램 준비 작업

```
import mysql.connector

# TODO: REPLACE THE VALUE OF VARIABLE team (EX. TEAM 1 --> team = 1)
team = 12
```

# 2) R2-1: Create Schema

```
# Requirement1: create schema ( name: DMA_team## )
def requirement1(host, user, password):
    cnx = mysql.connector.connect(host=host, user=user, password=password)
    cursor = cnx.cursor()
    cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')

    # TODO: WRITE CODE HERE
    cursor.execute('CREATE DATABASE IF NOT EXISTS DMA_team12;')
    cursor.execute('USE DMA_team12;')

    # TODO: WRITE CODE HERE
    cursor.close()
```

# 3) R2-2: Create Table

```
# Requierement2: create table
def requirement2(host, user, password):
    cnx = mysql.connector.connect(host=host, user=user, password=password)
    cursor = cnx.cursor()
    cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')

    # TODO: WRITE CODE HERE
    cursor.execute('USE DMA_team12;')
    cursor.execute('''
            CREATE TABLE IF NOT EXISTS app(
            id VARCHAR(255) NOT NULL,
            title VARCHAR(255) NOT NULL,
            developer_id INT(11) NOT NULL,
            description INT(11) NOT NULL,
            pricing_hint VARCHAR(255),
            PRIMARY KEY(id)
            )Engine = InnoDB DEFAULT CHARSET = utf8mb4;
            ''')
```

# 4) R2-3: Insert Data

```
# Requirement3: insert data
def requirement3(host, user, password, directory):
    cnx = mysql.connector.connect(host=host, user=user, password=password)
    cursor = cnx.cursor()
    cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')

    # TODO: WRITE CODE HERE
    cursor.execute('USE DMA_team12;')

    f_app = open(directory + '₩₩app.csv', 'r', encoding='utf-8')
    data_app = f_app.readlines()

    for i in range(1, len(data_app)):

        line_app = data_app[i].replace('₩n', '')
        line_app = line_app.split(',')

        if line_app[4]=='':
            sql='INSERT INTO app VALUES (%s,%s,%s,%s,NULL)'
            cursor.execute(sql, (line_app[0], line_app[1], line_app[2], line_app[3]))
        else:
            sql = 'INSERT INTO app VALUES (%s,%s,%s,%s,%s)'
            cursor.execute(sql,(line_app[0], line_app[1], line_app[2], line_app[3], line_app[4]))
    cnx.commit()
    f_app.close()
```

# 5) R2-4: Add Constraint Key (Foreign Key)

```python
# Requirement4: add constraint (foreign key)
def requirement4(host, user, password):
    cnx = mysql.connector.connect(host=host, user=user, password=password)
    cursor = cnx.cursor()
    cursor.execute('SET GLOBAL innodb_buffer_pool_size=2*1024*1024*1024;')

    # TODO: WRITE CODE HERE
    cursor.execute('USE DMA_team12')
    cursor.execute('ALTER TABLE app ADD CONSTRAINT FOREIGN KEY (developer_id) REFERENCES developer(id);')
    cursor.execute('ALTER TABLE reply ADD CONSTRAINT FOREIGN KEY (review_id) REFERENCES review(id);')
    cursor.execute('ALTER TABLE reply ADD CONSTRAINT FOREIGN KEY (developer_id) REFERENCES developer(id);')
    cursor.execute('ALTER TABLE review ADD CONSTRAINT FOREIGN KEY (app_id) REFERENCES app(id);')
    cursor.execute('ALTER TABLE message ADD CONSTRAINT FOREIGN KEY (recipient_id) REFERENCES user(id);')
    cursor.execute('ALTER TABLE pricing_plan ADD CONSTRAINT FOREIGN KEY (app_id) REFERENCES app(id);')
    cursor.execute('ALTER TABLE key_benefit ADD CONSTRAINT FOREIGN KEY (app_id) REFERENCES app(id);')
    cursor.execute('ALTER TABLE app_category ADD CONSTRAINT FOREIGN KEY (app_id) REFERENCES app(id);')
    cursor.execute('ALTER TABLE app_category ADD CONSTRAINT FOREIGN KEY (category_id) REFERENCES category(id);')
    cursor.execute('ALTER TABLE category_user ADD CONSTRAINT FOREIGN KEY (category_id) REFERENCES category(id);')
    cursor.execute('ALTER TABLE category_user ADD CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id);')
    cursor.execute('ALTER TABLE category_developer ADD CONSTRAINT FOREIGN KEY (category_id) REFERENCES category(id);')
    cursor.execute('ALTER TABLE category_developer ADD CONSTRAINT FOREIGN KEY (developer_id) REFERENCES developer(id);')
    cursor.execute('ALTER TABLE message_app ADD CONSTRAINT FOREIGN KEY (message_id) REFERENCES message(id);')
    cursor.execute('ALTER TABLE message_app ADD CONSTRAINT FOREIGN KEY (app_id) REFERENCES app(id);')
    cursor.execute('ALTER TABLE user_follow ADD CONSTRAINT FOREIGN KEY (user_id) REFERENCES user(id);')
    cursor.execute('ALTER TABLE user_follow ADD CONSTRAINT FOREIGN KEY (follow_user_id) REFERENCES user(id);')
    # TODO: WRITE CODE HERE
    cnx.commit()
    cursor.close()
```