

Database Project 2

Implementing a Simple Database Application

[도서관 관리 Application]

Overview

- Python과 MySQL을 이용한 응용 시스템 구현
 - 도서관 관리 시스템
- 주요 기능
 - 도서 정보, 회원 정보 삽입 / 삭제 / 출력
 - 회원이 도서를 대출, 반납과 평점 부여
 - 회원을 위한 도서 추천
- 과제 Due : 6/14(금) 23:59
 - 하루 Delay 마다 -10%, 3일 Delay 시 0점

Overview

- Raw data 제공 (data.csv)
- DB schema는 자유롭게 설계.
 - 주어진 raw data를 기반으로 적합한 DB schema를 설계.
 - csv 파일을 읽어와서 해당 내용으로 데이터베이스를 초기화하는 것으로 프로그램 시작.

- 예시 :

도서ID	도서명	도서 저자	회원ID	회원명	회원이 도서에 남긴 평점
14	The Giving Tree	Shel Silverstein	19	William	3
15	Gone with the Wind	Margaret Mitchell	1	Ava	4
16	The Perks of Being a Wallflower	Stephen Chbosky	20	Daniel	4
16	The Perks of Being a Wallflower	Stephen Chbosky	18	Sebastian	4
17	The Great Gatsby	F. Scott Fitzgerald	21	Abigail	5
17	The Great Gatsby	F. Scott Fitzgerald	22	Mia	3
17	The Great Gatsby	F. Scott Fitzgerald	23	Camila	4
18	The Little Prince	Antoine	9	Olivia	4
19	The Da Vinci Code	Dan Brown	21	Abigail	4

[data.csv 예시]

Overview

- 도서 관리
 - 도서 삽입 / 삭제 / 출력
- 회원 관리
 - 회원 삽입 / 삭제 / 출력
- 도서 대출
 - 회원이 지정한 도서를 대출
- 반납과 평점 부여
 - 회원이 대출한 도서를 반납하며 평점 부여 (1~5점)
- 도서 추천
 - Popularity-based Recommendation
 - User-based Collaborative Filtering

Popularity-based Recommendation (1/3)

- 예시 :
 - User 1~4, Item 1~4
 - 목표 : **User4** 에게 도서 추천하기
 - 평균 평점 기반 추천
 - 평점 개수 기반 도서 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5

Popularity-based Recommendation (2/3)

- 예시 : 평균 평점 기반 추천

- User 1~4, Item 1~4
- 목표 : **User4** 에게 도서 추천하기

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5
Average	2.67	3	2.5	4.5

→ 평균 평점 기준, User 4가 보지 않은 도서 중 평균 평점이 가장 높은 **Item 2** 추천

Popularity-based Recommendation (3/3)

- 예시 : 평점 수 기반 도서 추천
 - User 1~4, Item 1~4
 - 목표 : **User4** 에게 도서 추천하기

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		
User 3		2	1	4
User 4	2			5
Count	3	3	2	2

→ 평점 수 기준, User 4가 보지 않은 도서 중 평점 수가 가장 많은 **Item 2** 추천

User-based Collaborative Filtering (1/9)

- 회원이 평점을 남기지 않은 도서 중 회원이 가장 높은 평점을 부여할 것이라고 예상되는 도서를 추천
- 추천 알고리즘인 **User-based Collaborative Filtering** 사용
→ “나와 취향이 비슷한 사람들의 의견을 참고한다.”
- 예시: User 4에게 가장 적합한 Item을 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		1
User 3		2	1	4
User 4	2			5

User-item matrix

User-based Collaborative Filtering (2/9)

- 예시: User 4에게 가장 적합한 Item을 추천
 - User 4가 아직 평점을 부여하지 않은 Item 2, Item 3에 대한 평점을 예측
 - Item 2, Item 3 중, 가장 높은 예측평점을 가진 Item 을 추천

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	
User 2	1	3		1
User 3		2	1	4
User 4	2	?	?	5

User-item matrix

User-based Collaborative Filtering (3/9)

- 예시: User 4에게 가장 적합한 Item을 추천

회원이 아직 평점을 부여하지 않은 Item에 대해서는

해당 회원이 이미 부여한 평점들의 평균으로 임시 평점을 부여한다.

해당 회원이 평점을 부여한 적이 없다면, 0점으로 임시 평점을 부여한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	3.5	3.5	5

User-based Collaborative Filtering (4/9)

- 예시: User 4에게 가장 적합한 Item을 추천

각 사용자가 부여한 평점의 유사도를 측정한다.

→ 두 부여한 평점의 유사도가 높은 경우 두 회원의 취향이 유사하다고 판단한다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	3.5	3.5	5

User-based Collaborative Filtering (5/9)

- 예시: User 4에게 가장 적합한 Item을 추천

유사도 계산 방식: $\text{cosine-similarity} = \frac{A \cdot B}{\|A\| \cdot \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$

User 1 과 User 2 사이의 유사도 = $\frac{5 \times 1 + 4 \times 3 + 4 \times 1.67 + 4.33 \times 1}{\sqrt{5^2 + 4^2 + 4^2 + 4.33^2} \sqrt{1^2 + 3^2 + 1.67^2 + 1^2}} = \frac{28.1}{32.36} = 0.868 \approx \mathbf{0.87}$

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	3.5	3.5	5

User-based Collaborative Filtering (6/9)

- 예시: User 4에게 가장 적합한 Item을 추천

각 사용자가 부여한 평점의 유사도를 측정한다.

→ 두 부여한 평점의 유사도가 높은 경우 두 회원의 취향이 유사하다고 판단한다.

$$\text{User 1 과 User 2 사이의 유사도} = \frac{5 \times 1 + 4 \times 3 + 4 \times 1.67 + 4.33 \times 1}{\sqrt{5^2 + 4^2 + 4^2 + 4.33^2} \sqrt{1^2 + 3^2 + 1.67^2 + 1^2}} = \frac{28.1}{32.36} = 0.868 \approx \mathbf{0.87}$$

	User 1	User 2	User 3	User 4
User 1	1.0	0.87	0.92	0.94
User 2	0.87	1.0	0.73	0.86
User 3	0.92	0.73	1.0	0.93
User 4	0.94	0.86	0.93	1.0

User-based Collaborative Filtering (7/9)

- 예시: User 4에게 가장 적합한 Item을 추천

User 4 와 나머지 모든 User와의 유사도 값을 weight로 해서 weighted sum 값으로 각 도서에 대한 평점을 예측한다.

$$\textcircled{1} = \frac{0.94 \times 4 + 0.86 \times 3 + 0.93 \times 2}{0.94 + 0.86 + 0.93} = \frac{8.2}{2.73} = 3.00$$

$$\textcircled{2} = \frac{0.94 \times 4 + 0.86 \times 1.67 + 0.93 \times 1}{0.94 + 0.86 + 0.93} = \frac{6.13}{2.73} = 2.25$$

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	①	②	5

User-based Collaborative Filtering (8/9)

	User 1	User 2	User 3	User 4
User 1	1.0	0.87	0.92	0.94
User 2	0.87	1.0	0.73	0.86
User 3	0.92	0.73	1.0	0.93
User 4	0.94	0.86	0.93	1.0

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	①	②	5

$$\textcircled{1} = \frac{0.94 \times 4 + 0.86 \times 3 + 0.93 \times 2}{0.94 + 0.86 + 0.93} = \frac{8.2}{2.73} = 3.00$$

User-based Collaborative Filtering (9/9)

- 예시: User 4에게 가장 적합한 Item을 추천

추천 후보 (Item 2, Item 3) 중,

Item 2의 예상 평점이 다른 모든 추천 대상들의 예상 평점보다 높기 때문에,
User 4에게는 Item 2를 추천하게 된다.

	Item 1	Item 2	Item 3	Item 4
User 1	5	4	4	4.33
User 2	1	3	1.67	1
User 3	2.33	2	1	4
User 4	2	3.00	2.25	5

MySQL Connector Python Tutorial

MySQL Connector Python

- Database-Application 커넥터 (Connector)
 - 어플리케이션과 DB의 연동 기능을 구현할 수 있도록 커넥터(connector) 제공
- **mysql-connector-python**
 - Python에서 DB의 연동 기능 구현을 위한 MySQL 연동 패키지
 - 설치: **pip install mysql-connector-python**
 - Documentation:

<https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html>

Chapter 5 Connector/Python Coding Examples

Table of Contents

- 5.1 Connecting to MySQL Using Connector/Python
- 5.2 Creating Tables Using Connector/Python
- 5.3 Inserting Data Using Connector/Python
- 5.4 Querying Data Using Connector/Python

These coding examples illustrate how to develop Python applications and scripts which connect to MySQL Server using MySQL Connector/Python.

Python - MySQL 연동 예시 코드

Connection
object

```
from mysql.connector import connect
```

```
connection = connect(
```

```
    host='astronaut.snu.ac.kr',  
    port=7001,  
    user='TA1',  
    password='TA1',  
    db='TA1',  
    charset='utf8')
```

접속 정보

host, port (DB 서버 주소)

astronaut.snu.ac.kr (port: 7001)

user = password = db

DB[학번]

EX) DB2023_12345

MySQLCursor
object

```
with connection.cursor(dictionary=True) as cursor:
```

```
    cursor.execute("select * from stage")
```

```
    result = cursor.fetchall()
```

```
    print(result)
```

```
connection.close()
```

Python - MySQL 연동 예시 코드

`cursor.execute(sql_statement) → connection.commit()`

- Update/Insert/Modify 등 데이터베이스 레코드를 변경하는 statement의 경우,
- `execute` 만 실행하면 데이터베이스 레코드에 반영되지 않습니다.
 - “`commit()`” 수행해야 데이터베이스 레코드에 반영됩니다.

Notice

- **과제 Due : 6/14(금) 23:59**

- 하루 Delay 마다 -10%, 3일 Delay 시 0점

- 자세한 설명은 프로젝트 문서들을 참고
- 문서와 파일 포맷을 꼭 확인하고 제출
- 프로젝트 전반에 관한 질문은 ETL **Project 2** 게시판을 통해 질문
 - 비밀글 게시 및 이메일 문의는 지양
 - **질문 기한은 6/12 (수) 23:59 까지 입니다.**
- 개별 코드 구현에 대한 문의는 불가합니다