

2022-2 산업정보관리론 기말 프로젝트

최종 발표 자료

24조

2020147050 정준우

2020147046 이학민

2017232044 김진우

1

제안하는 서비스의 필요성

추진 배경

아직도 수많은 교육 분야에서 심각한 **지역 간 교육 격차가 발생**하고 있다.

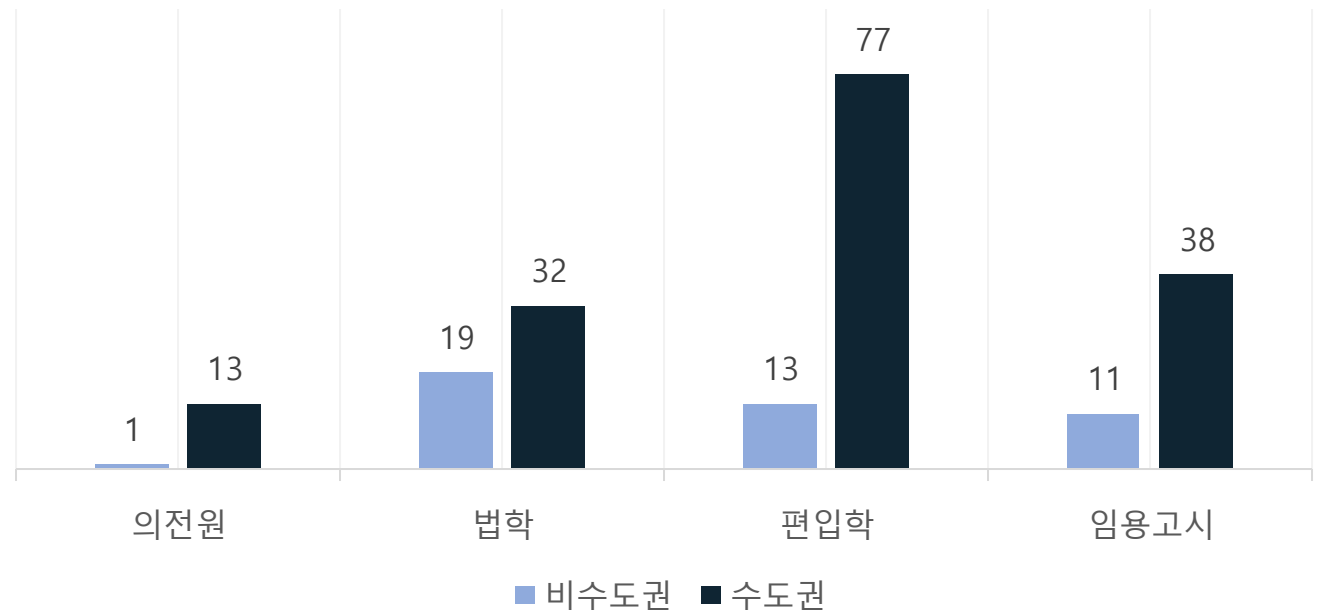
I 교육 인프라의 수도권 과밀화

특정 분야 학원의 수도권
집중으로 인해 비수도권 지역
학생의 교육 기회 부재

I 교육 정보의 지역 간 불균형

모의고사 석차, 성적 분석 등
고급 정보에 대한 진입장벽 존재

수도권-비수도권 학원 수 비교 (2022년)

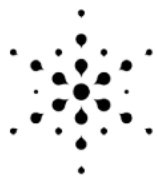


출처: 카카오맵

1

제안하는 서비스의 필요성

해결하고자 하는 문제점



SANTA TOEIC

- 설립: 2014년
- 문제 분야: TOEIC
- 서비스 1) 문제 추천
2) 점수 예측



- 설립: 2018년
- 문제 분야: 고등학교 내신
- 서비스 1) 유사 문제
2) 점수 예측

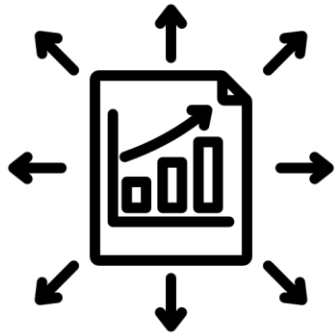
- ✓ 최근 산타(토익), 큐비(고등학교 내신) 등 인공지능 기술을 활용하여 저렴한 비용으로 학습의 접근성을 높이는 스타트업들이 등장하고 있다.
- ✓ 하지만 공무원 시험, 기사 자격증 시험, 편입 시험 등 국내 사교육 시장(23.4조)에 존재하는 수많은 교육 분야에서 4차 산업 기술의 혜택을 받지 못하고 있다.

1

제안하는 서비스의 필요성

제안하는 서비스와 기대효과

이에 따라 우리는 범 '시험'적인 **취약 유형 문제 추천 서비스**의 API를 만들고자 한다.



교육 시장 디지털 파급효과

교육 분야 전문가들은 개발 팀을 꾸리지 않더라도 **해당 API를 활용하여 IT 서비스를 만들 수 있고**, 사람들에게 학습 편의성 높이는 서비스를 제공할 수 있다.



학습 보조

사용자는 수도권 학원을 가거나 비싼 인터넷 강의를 듣지 않아도 **문제 취약 유형 분석, 전국 석차 제공 등의 학습 보조**를 받을 수 있다.



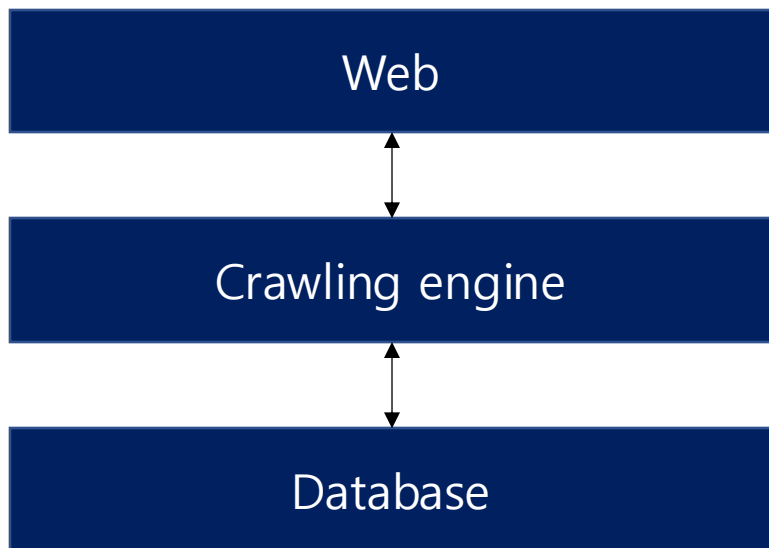
학습자 학습 효율성

무엇보다 이전에 100문제를 풀어야 알 수 있었던 본인의 공부 약점을, 학습자는 **본 서비스를 통해 10문제만 풀어도 알 수 있게 된다.**

2 데이터

데이터 수집

- 문제 데이터 수집: 웹 크롤링 알고리즘



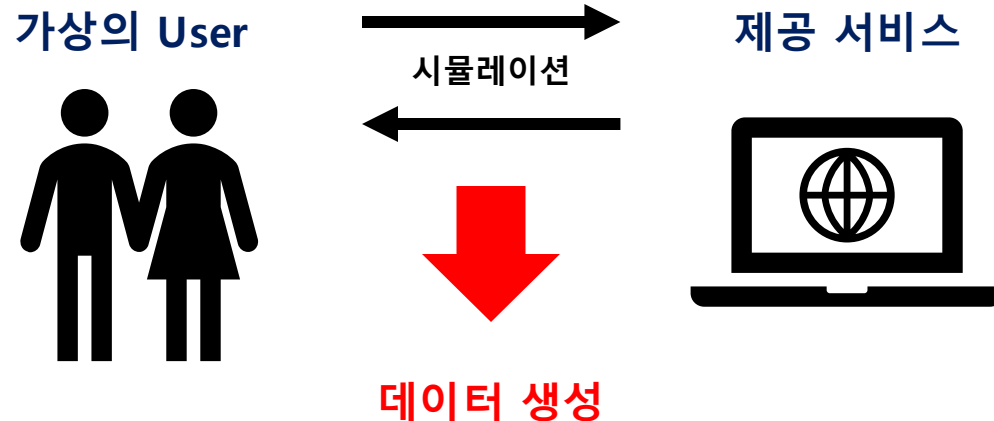
- 공개되어있는 국가자격시험 문제 데이터 수집
Ex. 정보처리기사 필기 자격증 시험 등
- 크롤링 사이트: <http://q.fran.kr>
- 크롤링 엔진: 리스틀리 무료 크롬확장프로그램 사용
- 수집 데이터: "PROBLEM" 테이블의 ProblemType, MainText, ChoiceText

2

데이터

데이터 생성

- 사용자 데이터 생성: **시뮬레이션**



- **구현 방법:** 실제 사용자가 문제를 푸는 여정에 가깝게 Python 모델을 구현
- **구현 내용:** 수집된 데이터에 대해 기본적인 전처리를 진행하고, 가상의 User가 시뮬레이션 모델에서 서비스를 이용하며 데이터 생성
- **생성 데이터:** 수집 데이터를 제외한 모든 데이터의 초기 데이터*

* TEST_PROBLEM 테이블의 Ranking 열은 SQL에서 구현

데이터 생성

• 사용자 데이터 생성: Python source code

```
import pandas as pd
import random
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
problem_set = pd.read_excel("problem_set.xlsx", engine = "openpyxl")
```

```
#파라미터 입력
User_num = 20
```

```
# 데이터 생성
PROBLEM = pd.DataFrame({'ProblemID': [], 'TestID': [], 'ProblemTypeID': [], 'Accuracy': [], 'MainText': [],
                        'ChoiceText1': [], 'ChoiceText2': [], 'ChoiceText3': [], 'ChoiceText4': [],
                        'SolvingTimeAvg': [], 'SolvingTimeStd': []})
TEST_DOMAIN = pd.DataFrame({'TestDomainID': [], 'PublisherID': [], 'TestTimeLimit': []})
PROBLEM_TYPE = pd.DataFrame({'ProblemTypeID': [], 'TestDomainID': []})
TEST_FEEDBACK = pd.DataFrame({'TestID': [], 'UserID': [], 'Score': []})
USER = pd.DataFrame({'UserID': []})
PROBLEM_FEEDBACK = pd.DataFrame({'ProblemID': [], 'UserID': [], 'RightWrong': [], 'SolvingTime': []})
WEAK_OR_NOT = pd.DataFrame({'ProblemTypeID': [], 'UserID': [], 'WeakOrNot': []})
```

```
for j in range(len(TEST['TestDomainID'].unique())):
    TestDomainID = TEST[TEST['TestDomainID'].unique()[j]]
    TEST_DOMAIN = TEST_DOMAIN.append({'TestDomainID': TestDomainID, 'ignore_index=True'})
```

```
for j in range(len(PROBLEM['ProblemTypeID'].unique())):
    ProblemTypeID = PROBLEM[PROBLEM['ProblemTypeID'].unique()[j]]
    TestDomainID = problem_set[problem_set['ProblemTypeID'] == ProblemTypeID]['Testname'].unique()[0]
    PROBLEM_TYPE = PROBLEM_TYPE.append({'ProblemTypeID': ProblemTypeID, 'TestDomainID': TestDomainID, 'ignore_index=True'})
```

```
for j in range(1, User_num+1):
    User_num_str = str(j).zfill(3)
    UserID = 'U00000' + User_num_str
    USER = USER.append({'UserID': UserID, 'ignore_index=True'})
```

```
for i in range(len(PROBLEM['ProblemID'].unique())):
    ProblemID = PROBLEM[PROBLEM['ProblemID'].unique()[i]]
    for j in range(len(USER['UserID'].unique())):
        UserID = USER[USER['UserID'].unique()[j]]

        RightWrong_percent = random.random()
        if RightWrong_percent > 0.3:
            RightWrong = 1
        else:
            RightWrong = 0

        SolvingTime_line = PROBLEM[PROBLEM['ProblemID'] == ProblemID]
        SolvingTimeAvg = PROBLEM[SolvingTime_line['SolvingTimeAvg'].unique()[0]]
        SolvingTimeStd = PROBLEM[SolvingTime_line['SolvingTimeStd'].unique()[0]]
        SolvingTime = np.random.normal(SolvingTimeAvg, SolvingTimeStd, size=1)[0]

        PROBLEM_FEEDBACK = PROBLEM_FEEDBACK.append({'ProblemID': ProblemID, 'UserID': UserID, 'RightWrong': RightWrong,
                                                    'SolvingTime': SolvingTime, 'ignore_index=True'})
```

```
for i in range(len(problem_set)):
    problem_set_line = problem_set.iloc[i]
    ProblemID = problem_set_line['ProblemID']
    TestID = problem_set_line['TestID']
    ProblemTypeID = problem_set_line['ProblemTypeID']
    MainText = problem_set_line['MainText']
    ChoiceText1 = problem_set_line['ChoiceText1']
    ChoiceText2 = problem_set_line['ChoiceText2']
    ChoiceText3 = problem_set_line['ChoiceText3']
    ChoiceText4 = problem_set_line['ChoiceText4']
    Accuracy = random.random()
    SolvingTimeAvg = random.uniform(30, 140)
    SolvingTimeStd = random.uniform(SolvingTimeAvg*0.025, SolvingTimeAvg*0.34)
    PROBLEM = PROBLEM.append({'ProblemID': ProblemID,
                              'TestID': TestID, 'ProblemTypeID': ProblemTypeID,
                              'MainText': MainText, 'ChoiceText1': ChoiceText1,
                              'ChoiceText2': ChoiceText2, 'ChoiceText3': ChoiceText3,
                              'ChoiceText4': ChoiceText4, 'Accuracy': Accuracy,
                              'SolvingTimeAvg': SolvingTimeAvg, 'SolvingTimeStd': SolvingTimeStd, 'ignore_index=True'})
```

```
for j in range(len(PROBLEM['TestID'].unique())):
    TestID = PROBLEM[PROBLEM['TestID'].unique()[j]]
    TestDomainID = problem_set[problem_set['TestID'] == TestID]['Testname'].unique()[0]
    PublisherID = 'P000' + TestID[-4:]
    TestTimeLimit = 5000
    TEST = TEST.append({'TestID': TestID, 'TestDomainID': TestDomainID, 'PublisherID': PublisherID,
                       'TestTimeLimit': TestTimeLimit, 'ignore_index=True'})
```

```
for i in range(len(PROBLEM['TestID'].unique())):
    TestID = PROBLEM[PROBLEM['TestID'].unique()[i]]
```

```
for j in range(len(USER['UserID'].unique())):
    score_num = 0
    UserID = USER[USER['UserID'].unique()[j]]
    TestID_line = PROBLEM[PROBLEM['TestID'] == TestID]
```

```
for k in range(len(PROBLEM[TestID_line]['ProblemID'].unique())):
    ProblemID = PROBLEM[TestID_line][PROBLEM['ProblemID'].unique()[k]]
    ProblemID_line = PROBLEM_FEEDBACK[PROBLEM['ProblemID'] == ProblemID]
    UserID_line = PROBLEM_FEEDBACK[PROBLEM['UserID'] == UserID]
    score_line = PROBLEM_FEEDBACK[ProblemID_line & UserID_line]
    score_num += score_line['RightWrong'].unique()[0]
```

```
TEST_FEEDBACK = TEST_FEEDBACK.append({'TestID': TestID, 'UserID': UserID, 'Score': score_num, 'ignore_index=True'})
```

```
WEAK_OR_NOT = pd.DataFrame({'ProblemTypeID': [], 'UserID': [], 'WeakOrNot': []})
```

```
for i in range(len(PROBLEM[PROBLEM['ProblemTypeID'].unique())):
    ProblemTypeID = PROBLEM[PROBLEM['ProblemTypeID'].unique()[i]]
```

```
for j in range(len(USER[USER['UserID'].unique())):
    score_num = 0
    UserID = USER[USER['UserID'].unique()[j]]
    ProblemTypeID_line = PROBLEM[PROBLEM['ProblemTypeID'] == ProblemTypeID]
```

```
for k in range(len(PROBLEM[ProblemTypeID_line][PROBLEM['ProblemID'].unique())):
    ProblemID = PROBLEM[ProblemTypeID_line][PROBLEM['ProblemID'].unique()[k]]
    ProblemID_line = PROBLEM_FEEDBACK[PROBLEM['ProblemID'] == ProblemID]
    UserID_line = PROBLEM_FEEDBACK[PROBLEM['UserID'] == UserID]
    score_line = PROBLEM_FEEDBACK[ProblemID_line & UserID_line]
    score_num += score_line['RightWrong'].unique()[0]
    if score_num > len(PROBLEM[ProblemTypeID_line][PROBLEM['ProblemID'].unique()))*0.65:
        WeakOrNot = 1
    else:
        WeakOrNot = 0
```

```
WEAK_OR_NOT = WEAK_OR_NOT.append({'ProblemTypeID': ProblemTypeID, 'UserID': UserID, 'WeakOrNot': WeakOrNot, 'ignore_index=True'})
```

```
writer=pd.ExcelWriter('Dataset.xlsx', engine='openpyxl')
PROBLEM.to_excel(writer, sheet_name='PROBLEM', index=False)
TEST.to_excel(writer, sheet_name='TEST', index=False)
TEST_DOMAIN.to_excel(writer, sheet_name='TEST_DOMAIN', index=False)
PROBLEM_TYPE.to_excel(writer, sheet_name='PROBLEM_TYPE', index=False)
TEST_FEEDBACK.to_excel(writer, sheet_name='TEST_FEEDBACK', index=False)
USER.to_excel(writer, sheet_name='USER', index=False)
PROBLEM_FEEDBACK.to_excel(writer, sheet_name='PROBLEM_FEEDBACK', index=False)
WEAK_OR_NOT.to_excel(writer, sheet_name='WEAK_OR_NOT', index=False)
writer.save()
pd.ExcelWriter.close(writer)
```

데이터 생성

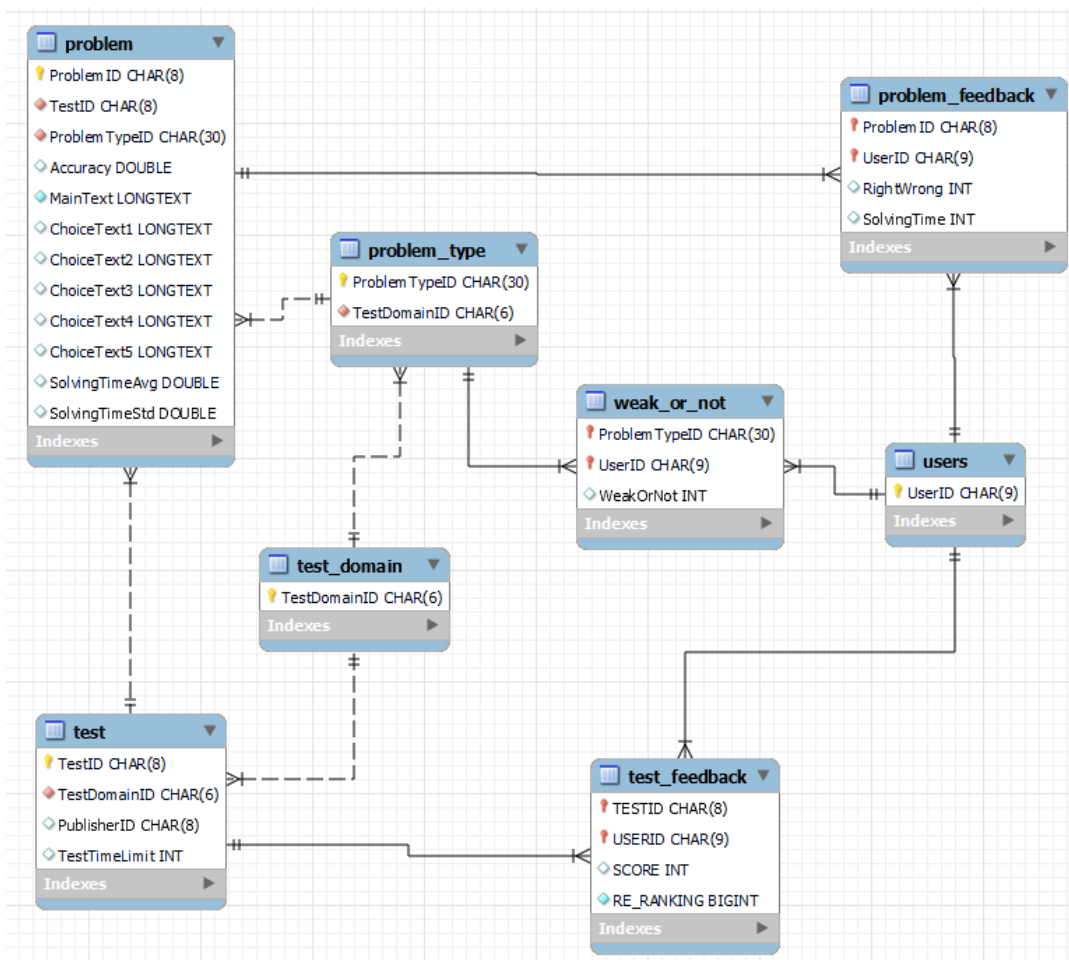
• 사용자 데이터 생성: SQL DDL을 사용한 Rank 데이터 재입력

```
1  -- 전체 킬링 등수에서 시험별 등수로 변경
2
3  • CREATE TABLE TEST_FEEDBACK AS
4    SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
5    FROM test_feedback_temp
6    WHERE TestID = 'T000001';
7
8  • INSERT INTO TEST_FEEDBACK
9    SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
10   FROM test_feedback_temp
11   WHERE TestID = 'T000002';
12
13 • INSERT INTO TEST_FEEDBACK
14   SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
15   FROM test_feedback_temp
16   WHERE TestID = 'T000003';
17
18 • INSERT INTO TEST_FEEDBACK
19   SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
20   FROM test_feedback_temp
21   WHERE TestID = 'T000004';
22
23 • INSERT INTO TEST_FEEDBACK
24   SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
25   FROM test_feedback_temp
26   WHERE TestID = 'T000005';
```

- Python 코드로 생성한 Ranking이 잘못되어있기 때문에, 이를 MySQL을 사용하여 올바르게 수정
 - DDL CREATE를 사용하여 우선 잘못된 Ranking 데이터를 포함하고 있는 TEST_FEEDBACK_TEMP Table을 생성
 - 기존 TEST_FEEDBACK_TEMP Table에 있던 TestID, UserID, Score는 그대로 가져오고, TestID 별로 User 등수를 새롭게 매겨 RE_Ranking 열을 생성하여 TEST_FEEDBACK 테이블 생성
- * 구체적인 과정은 추후 설명

2 데이터

EER Diagram



- 데이터 수집과 시뮬레이션을 통해 생성된 테이블들간의 관계를 고려하여 ERR Diagram을 설계
- BCNF (Boyce and Codd Normal Form) 정규화 완료

테이블 생성

• SQL DDL CREATE TABLE을 사용한 테이블 생성

```
1 CREATE TABLE TEST_DOMAIN(  
2     TestDomainID CHAR(6) NOT NULL,  
3     CONSTRAINT TESTDOMAINID_PK PRIMARY KEY(TestDomainID)  
4 );  
5  
6 CREATE TABLE TEST(  
7     TestID CHAR(8) NOT NULL,  
8     TestDomainID CHAR(6) NOT NULL,  
9     PublisherID CHAR(8),  
10    TestTimeLimit INT,  
11    CONSTRAINT TESTID_PK PRIMARY KEY(TestID),  
12    CONSTRAINT TESTDOMAINID_FK FOREIGN KEY(TestDomainID)  
13        REFERENCES TEST_DOMAIN(TestDomainID)  
14        ON UPDATE NO ACTION  
15        ON DELETE NO ACTION  
16 );  
17  
18 CREATE TABLE PROBLEM_TYPE(  
19     ProblemTypeID CHAR(30) NOT NULL,  
20     TestDomainID CHAR(6) NOT NULL,  
21     CONSTRAINT PROBLEMTYPEID_PK PRIMARY KEY(ProblemTypeID),  
22     CONSTRAINT TESTDOMAINID1_FK FOREIGN KEY(TestDomainID)  
23        REFERENCES TEST_DOMAIN(TestDomainID)  
24        ON UPDATE NO ACTION  
25        ON DELETE NO ACTION  
26 );
```

- 앞서 설계한 EER-Diagram을 바탕으로 CREATE TABLE을 작성
- PK (Primary Key)와 FK (Foreign Key)를 선언
- 데이터의 무결성을 지키기 위해 Foreign Key를 UPDATA하거나 DELETE하는 것을 허용하지 않음

2 데이터

테이블 생성

- SQL DDL CREATE TABLE을 사용한 테이블 생성

```
CREATE TABLE PROBLEM(  
    ProblemID CHAR(8) NOT NULL,  
    TestID CHAR(8) NOT NULL,  
    ProblemTypeID CHAR(30) NOT NULL,  
    Accuracy double,  
    MainText Longtext NOT NULL,  
    ChoiceText1 Longtext,  
    ChoiceText2 Longtext,  
    ChoiceText3 Longtext,  
    ChoiceText4 Longtext,  
    ChoiceText5 Longtext,  
    SolvingTimeAvg Double,  
    SolvingTimeStd Double,  
    CONSTRAINT PROBLEMID_PK PRIMARY KEY(ProblemID),  
    CONSTRAINT TESTID_FK FOREIGN KEY(TestID)  
        REFERENCES TEST(TestID)  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION,  
    CONSTRAINT PROBLEMTYPEID_FK FOREIGN KEY(ProblemTypeID)  
        REFERENCES PROBLEM_TYPE(ProblemTypeID)  
        ON UPDATE NO ACTION  
        ON DELETE NO ACTION  
);
```

```
52 CREATE TABLE USERS(  
53     UserID CHAR(9) NOT NULL,  
54     CONSTRAINT USERID_PK PRIMARY KEY(UserID)  
55 );  
56  
57 CREATE TABLE WEAK_OR_NOT(  
58     ProblemTypeID CHAR(30) NOT NULL,  
59     UserID CHAR(9) NOT NULL,  
60     WeakOrNot Int,  
61     CONSTRAINT ALLNONE_PK PRIMARY KEY(ProblemTypeID, UserID),  
62     CONSTRAINT PROBLEMTYPEID1_FK FOREIGN KEY(ProblemTypeID)  
63         REFERENCES PROBLEM_TYPE(ProblemTypeID)  
64         ON UPDATE NO ACTION  
65         ON DELETE NO ACTION,  
66     CONSTRAINT USERID_FK FOREIGN KEY(UserID)  
67         REFERENCES USERS(UserID)  
68         ON UPDATE NO ACTION  
69         ON DELETE NO ACTION  
70 );
```

```
72 CREATE TABLE PROBLEM_FEEDBACK(  
73     ProblemID CHAR(8) NOT NULL,  
74     UserID CHAR(9) NOT NULL,  
75     RightWrong Int,  
76     SolvingTime INT,  
77     CONSTRAINT ALLNONE1_PK PRIMARY KEY(ProblemID, UserID),  
78     CONSTRAINT PROBLEMID_FK FOREIGN KEY(ProblemID)  
79         REFERENCES PROBLEM(ProblemID)  
80         ON UPDATE NO ACTION  
81         ON DELETE NO ACTION,  
82     CONSTRAINT USERID1_FK FOREIGN KEY(UserID)  
83         REFERENCES USERS(UserID)  
84         ON UPDATE NO ACTION  
85         ON DELETE NO ACTION  
86 );
```

2 데이터

테이블 생성

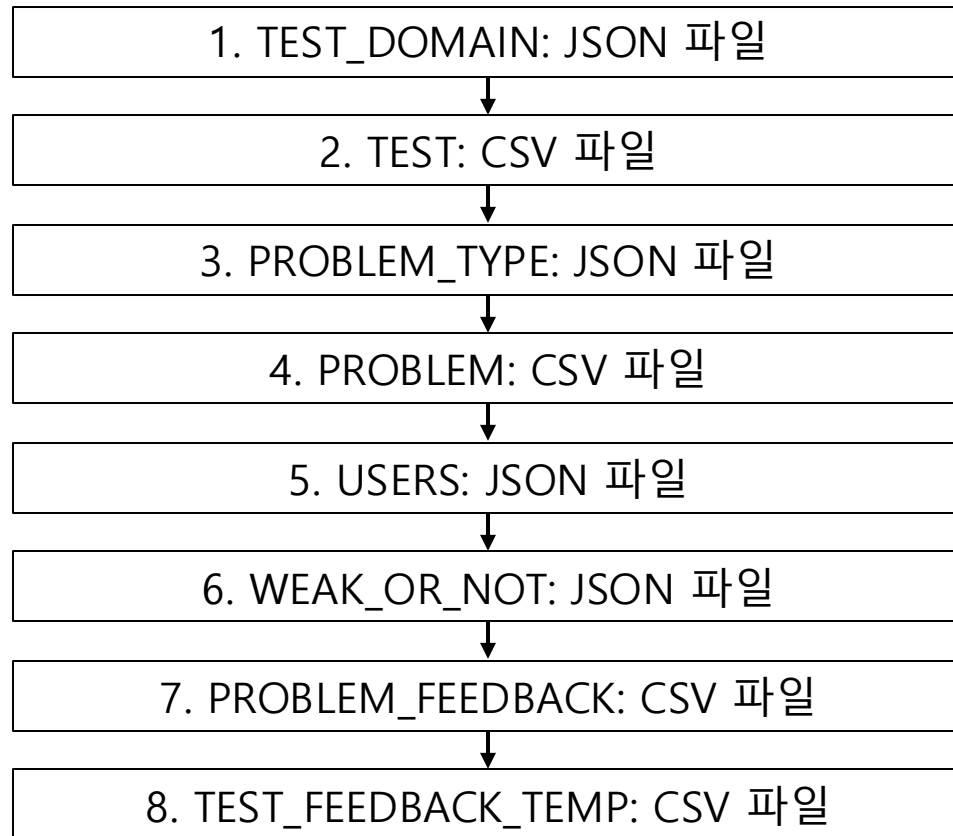
- SQL DDL CREATE TABLE을 사용한 테이블 생성

```
88 • CREATE TABLE TEST_FEEDBACK_TEMP(  
89     TestID      CHAR(8)      NOT NULL,  
90     UserID      CHAR(9)      NOT NULL,  
91     Ranking     INT,  
92     Score       INT,  
93     CONSTRAINT ALLINONE2_PK PRIMARY KEY(TestID, UserID),  
94     CONSTRAINT TESTID1_FK   FOREIGN KEY(TestID)  
95                             REFERENCES TEST(TestID)  
96                             ON UPDATE NO ACTION  
97                             ON DELETE NO ACTION,  
98     CONSTRAINT USERID2_FK   FOREIGN KEY(UserID)  
99                             REFERENCES USERS(UserID)  
100                             ON UPDATE NO ACTION  
101                             ON DELETE NO ACTION  
102 );
```

- TEST_FEEDBACK 테이블의 경우 앞서 설명한 것처럼 MySQL Code를 이용하여 Ranking을 새롭게 새롭게 업데이트하기 때문에 기존 Python Code로 생성한 데이터를 TEST_FEEDBACK_TEMP로 생성하여 임시로 활용

데이터 입력

• CSV, JSON 파일을 사용한 데이터 입력



- CREATE TABLE을 사용하여 생성한 테이블들에 Table Data Import Wizard를 사용하여 데이터 입력
- FK 때문에 반드시 좌측 flow chart와 같은 순서대로 테이블 데이터를 import 해와야 함
- 특정 테이블들은 데이터에 한글이 들어 있어서 JSON 파일로 import 해와야 함
- 한글을 포함하지 않는 파일들도 전부 JSON으로 열리는 것은 아니라서 반드시 제시한 방식으로 import 해와야 한다.

데이터 입력

• TEST_FEEDBACK 테이블 생성

```
3 • CREATE TABLE TEST_FEEDBACK AS
4     SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
5     FROM test_feedback_temp
6     WHERE TestID = 'T000001';
7
8 • INSERT INTO TEST_FEEDBACK
9     SELECT TESTID, USERID, SCORE, RANK() OVER (ORDER BY SCORE DESC) AS RE_RANKING
10    FROM test_feedback_temp
11    WHERE TestID = 'T000002';
12
```

```
178 • ALTER TABLE TEST_FEEDBACK
179     ADD CONSTRAINT ALLINONE3_PK PRIMARY KEY(TestID, UserID);
180
181 • ALTER TABLE TEST_FEEDBACK
182     ADD CONSTRAINT TESTID2_FK FOREIGN KEY(TestID)
183     REFERENCES TEST(TestID)
184     ON UPDATE NO ACTION
185     ON DELETE NO ACTION;
186
187 • ALTER TABLE TEST_FEEDBACK
188     ADD CONSTRAINT USERID3_FK FOREIGN KEY(UserID)
189     REFERENCES USERS(UserID)
190     ON UPDATE NO ACTION
191     ON DELETE NO ACTION;
192
193 • DROP TABLE TEST_FEEDBACK_TEMP;
```

- TEST_FEEDBACK_TEMP 테이블을 활용하여 최종 TEST_FEEDBACK 테이블 생성
- 앞서 설명한 것 처럼 TestID, UserID, Score는 그대로 가져오고, TestID 별로 User 등수를 새롭게 매겨 RE_Ranking 열을 생성
- 이후 TEST_FEEDBACK 테이블의 PK와 FK를 설정하고 TEST_FEEDBACK_TEMP 테이블은 삭제

2

데이터

각 테이블 소개

• PROBLEM: 문제 정보

```
1 • SELECT * FROM teamproject.problem;
```
















Table Name:

Schema: **teamproject**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 ProblemID	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 TestID	CHAR(8)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 ProblemTypeID	CHAR(30)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 Accuracy	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 MainText	LONGTEXT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 ChoiceText1	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 ChoiceText2	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 ChoiceText3	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 ChoiceText4	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 ChoiceText5	LONGTEXT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 SolvingTimeAvg	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 SolvingTimeStd	DOUBLE	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

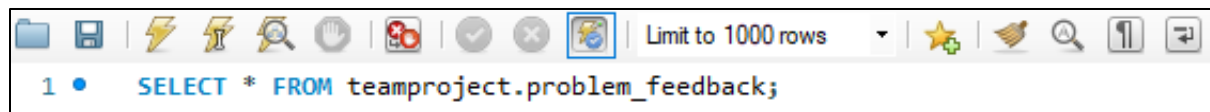
ProblemID	TestID	ProblemTypeID	Accuracy	MainText	ChoiceText1	ChoiceText2	ChoiceText3	ChoiceText4	ChoiceText5	SolvingTimeAvg	SolvingTimeStd
P0000001	T0000001	건축계획	0.25462...	1. 기...	②?소름	②?런드리	②?프로시...	②?인포메...	NULL	108.1373095	18.38466369
P0000002	T0000001	건축계획	0.82579...	2. 사...	②?공사비...	②?독립성...	②?방일이...	②?방일이...	NULL	109.7510116	3.692331702
P0000003	T0000001	건축계획	0.57683...	3. 주...	②?자형	②?루프(loop)	②?쿨데삭(...	②?오버브...	NULL	35.71864031	4.046591213
P0000004	T0000001	건축계획	0.57700...	4. 도...	②?폐가식	②?반개가식	②?자유개...	②?안전개...	NULL	132.1956738	41.1892408
P0000005	T0000001	건축계획	0.57525...	5. 단...	②?상출: ...	②?상출: ...	②?상출: ...	②?상출: 육...	NULL	53.13512651	8.521691095
P0000006	T0000001	건축계획	0.39391...	6. 다...	②?엘리베...	②?진열장...	②?지하주...	②?충돌 매...	NULL	53.46343978	5.357934784
P0000007	T0000001	건축계획	0.46017...	7. 학...	②?중합교...	②?교과교...	②?달트형...	②?플라톤...	NULL	69.03811366	22.30341105
P0000008	T0000001	건축계획	0.20793...	8. 종...	②?내과의 ...	②?환자의 ...	②?중앙주...	②?전체병...	NULL	40.57349319	1.978515518
P0000009	T0000001	건축계획	0.82825...	9. 공...	②?제품중...	②?레이아...	②?공정중...	②?고정식 ...	NULL	48.20645869	6.221804408
P0000010	T0000001	건축계획	0.36548...	10. 극...	②?엔티 룸...	②?그린 룸...	②?배경제...	②?의상실...	NULL	91.50418418	22.89678229

- ProblemID : PK, 문제, P0000001~
- TestID : FK, 시험 종류, T000001~
- ProblemTypeID : FK, 문제 종류, PT000001~
- Accuracy : 정답률, 0과 1사이의 실수
- MainText : 문제 본문 텍스트
- ChoiceText : 선지 텍스트
- SolvingTimeAvg : 문제 푸는 시간 평균, 시간(초)
- SolvingTimeStd : 문제 푸는 시간 표준편차, 시간(초)

2 데이터

각 테이블 소개

- **PROBLEM_FEEDBACK**: 사용자의 문제 풀이 정보






Table Name:

Schema: **teamproject**

Charset/Collation:





utf8mb4

utf8mb4_0900_ai_ci

Engine:

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 ProblemID	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 UserID	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 RightWrong	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 SolvingTime	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

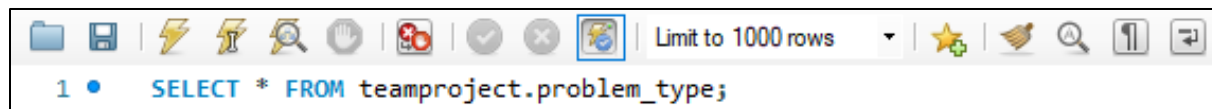
	ProblemID	UserID	RightWrong	SolvingTime
▶	P0000001	U00000001	1	112
	P0000001	U00000002	0	128
	P0000001	U00000003	1	113
	P0000001	U00000004	1	103
	P0000001	U00000005	1	91
	P0000001	U00000006	0	126
	P0000001	U00000007	1	98
	P0000001	U00000008	1	121
	P0000001	U00000009	0	112
	P0000001	U00000010	1	114

- ProblemID : PK, 문제, P0000001~
- UserID : PK, 사용자, U00000001~
- RightWrong : 문제 정답 여부, 1이면 right, 0이면 wrong
- SolvingTime : 문제 푸는 데 걸린 시간(초)

2 데이터

각 테이블 소개

- PROBLEM_TYPE: 문제의 유형






Table Name:

Schema: **teamproject**

Charset/Collation:



utf8mb4

utf8mb4_0900_ai_ci

Engine:

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 ProblemTypeID	CHAR(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 TestDomainID	CHAR(6)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

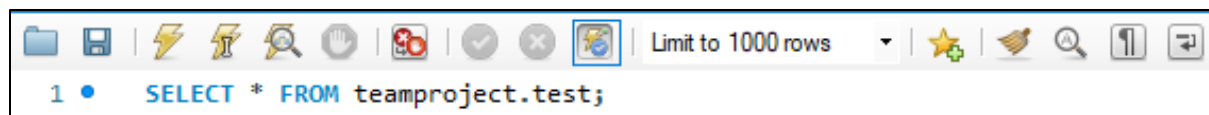
ProblemTypeID	TestDomainID
건축계획	TD0001
건축구조	TD0001
건축법규	TD0001
건축설비	TD0001
건축시공	TD0001
대기오염 개론	TD0002
대기오염 공정시험기준(방법)	TD0002
대기오염 방지기술	TD0002
대기환경관계법규	TD0002
연소공학	TD0002

- ProblemTypeID : PK, 문제 종류, PT000001 ~
- TestDomainID : FK, 시험 분야, TD0001 ~

2 데이터

각 테이블 소개

- TEST: 시험 정보






Table Name:

Schema: **teamproject**

Charset/Collation:





utf8mb4

utf8mb4_0900_ai_ci

Engine:

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 TestID	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 TestDomainID	CHAR(6)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 PublisherID	CHAR(8)	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 TestTimeLimit	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

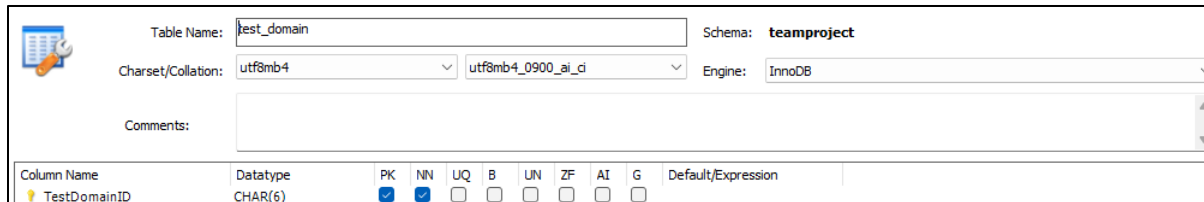
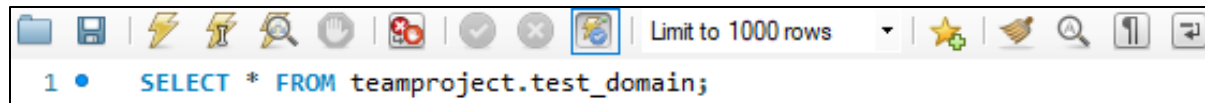
	TestID	TestDomainID	PublisherID	TestTimeLimit
▶	T0000001	TD0001	PS000001	5400
	T0000002	TD0001	PS000002	5400
	T0000003	TD0001	PS000003	5400
	T0000004	TD0001	PS000004	5400
	T0000005	TD0001	PS000005	5400
	T0000006	TD0001	PS000006	5400
	T0000007	TD0001	PS000007	5400
	T0000008	TD0001	PS000008	5400
	T0000009	TD0002	PS000009	5400
	T0000010	TD0002	PS000010	5400

- TestID : PK, 시험 종류, T000001~
- TestDomainID : FK, 시험 분야, TD0001~
- PublisherID : 문제 출제자(출판사), PS000001
- TestTimeLimit : 시험 제한 시간, 시간(초)

2 데이터

각 테이블 소개

- TEST_DOMAIN : 시험 도메인 정보



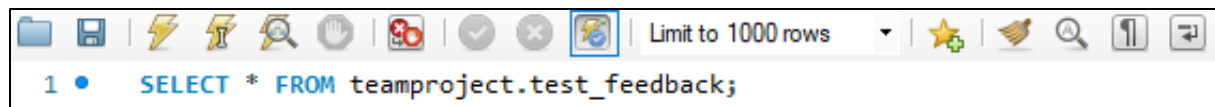
	TestDomainID
▶	TD0001
	TD0002
	TD0003
	TD0004
	TD0005

- TestDomainID : PK, 시험 분야, TD0001~

2 데이터

각 테이블 소개

- TEST_FEEDBACK : 사용자의 시험 응시 정보










Table Name:

Schema: **teamproject**

Charset/Collation:

Engine:

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 TESTID	CHAR(8)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 USERID	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 SCORE	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL
 RE_RANKING	BIGINT	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	'0'

	TESTID	USERID	SCORE	RE_RANKING
▶	T0000016	U00000001	65	8
	T0000016	U00000002	64	11
	T0000016	U00000003	59	18
	T0000016	U00000004	58	20
	T0000016	U00000005	71	1
	T0000016	U00000006	64	11
	T0000016	U00000007	65	8
	T0000016	U00000008	68	5
	T0000016	U00000009	59	18
	T0000016	U00000010	69	2

- TestID : PK, 시험 종류, T000001~
- UserID : PK, 사용자, U000000001~
- Score : 점수
- RE_Ranking : 등수

2 데이터

각 테이블 소개

- USER : 사용자 정보

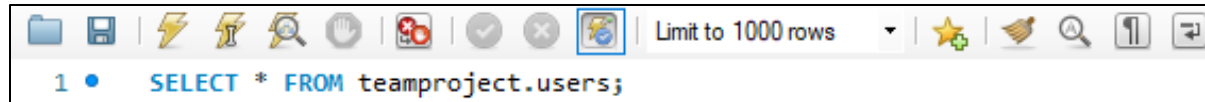


Table Name:	users	Schema:	teamproject							
Charset/Collation:	utf8mb4	utf8mb4_0900_ai_ci	Engine:	InnoDB						
Comments:										
Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
UserID	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

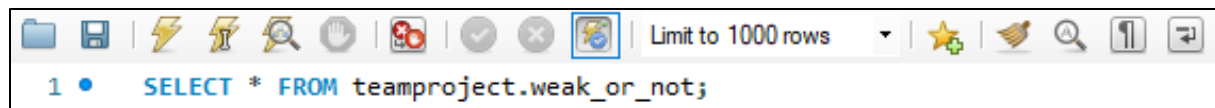
	UserID
▶	U00000001
	U00000002
	U00000003
	U00000004
	U00000005
	U00000006
	U00000007
	U00000008
	U00000009
	U00000010

- UserID : PK, 사용자, U00000001~

2 데이터

각 테이블 소개

- WEAK_OR_NOT : 사용자 별 취약유형 정보






Table Name:

Schema: **teamproject**

Charset/Collation:




utf8mb4

utf8mb4_0900_ai_ci

Engine:

InnoDB

Comments:

Column Name	Datatype	PK	NN	UQ	B	UN	ZF	AI	G	Default/Expression
 ProblemTypeID	CHAR(30)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 UserID	CHAR(9)	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
 WeakOrNot	INT	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	NULL

	ProblemTypeID	UserID	WeakOrNot
▶	건설안전기술	U00000001	0
	건설안전기술	U00000002	0
	건설안전기술	U00000003	0
	건설안전기술	U00000004	0
	건설안전기술	U00000005	0
	건설안전기술	U00000006	0
	건설안전기술	U00000007	0
	건설안전기술	U00000008	0
	건설안전기술	U00000009	0
	건설안전기술	U00000010	0

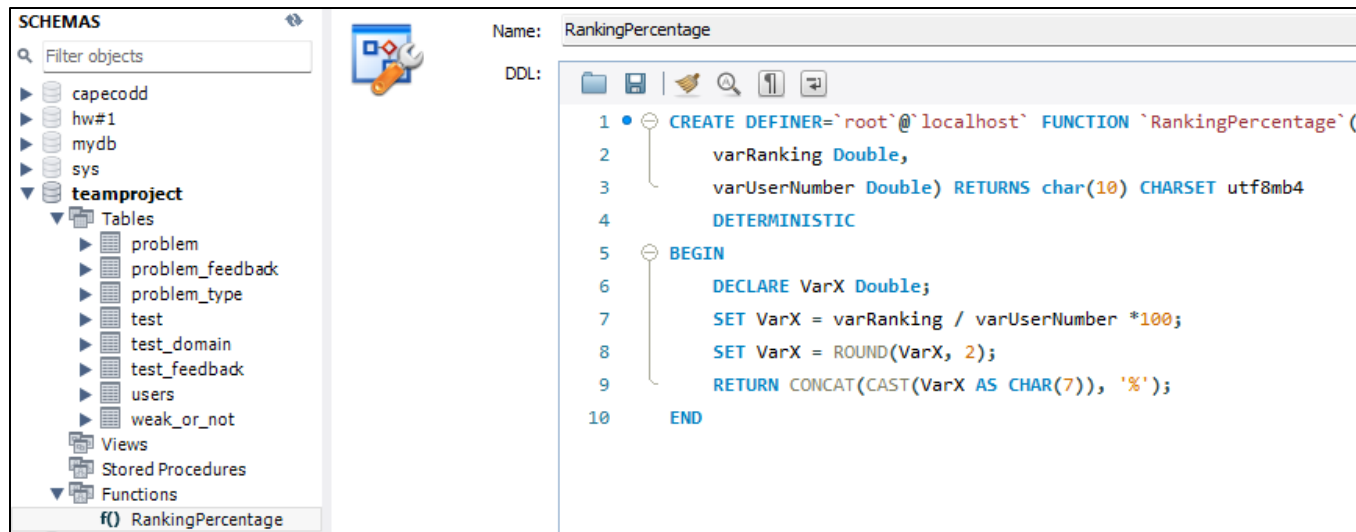
- ProblemTypeID : PK, 문제 종류, PT000001~
- UserID: PK, 사용자, U00000001~
- WeakOrNot : 취약유형인지 아닌지, 1이면 weak, 0이면 not weak

3

MySQL

MySQL 1: Fuction

- RankingPercentage: 유저의 등수를 상위 X%로 알려주는 함수 구현



	RankingPercentage(2, 10)
▶	20%

총 10명의 Users 중에서 2번째
등수를 가진 User의 Ranking
percentage는 20%

MySQL

결과

3

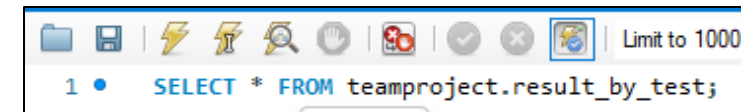
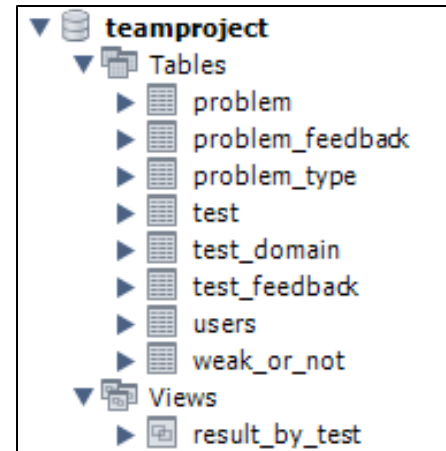
MySQL

MySQL 2: VIEW

- result_by_test: VIEW를 생성하여 시험별 성적과 등수 오름차순 정렬로 확인하기

```
1      # 시험별 성적과 등수 오름차순 정렬 VIEW
2
3  ●   CREATE VIEW RESULT_BY_TEST AS
4      SELECT TestID, UserID, Score, RE_RANKING
5      FROM test_feedback
6      WHERE TestID = 'T0000001'    -- @@@@ 시험ID 입력
7      ORDER BY RE_RANKING;
```

MySQL



	TestID	UserID	Score	RE_RANKING
▶	T0000001	U00000021	65	0

결과

3

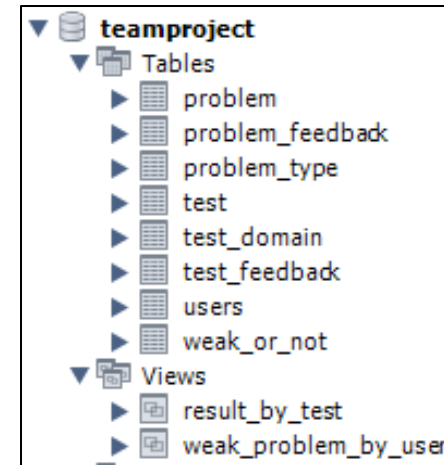
MySQL

MySQL 2: VIEW

- weak_problem_by_user: VIEW를 생성하여 유저 별 특정 시험 분야의 취약 문제 유형 확인

```
10 # 특정 시험 분야와 유저별 취약 문제 유형 VIEW
11
12 • CREATE VIEW WEAK_PROBLEM_BY_USER AS
13 SELECT USERID, PROBLEMTYPEID
14 FROM weak_or_not
15 WHERE USERID = 'U000000001' AND      -- @@@@ 유저ID 입력
16        WEAKORNOT = 1 AND
17        ProblemTypeID IN (SELECT PROBLEMTYPEID FROM PROBLEM_TYPE
18                           WHERE TESTDOMAINID = 'TD0001');    -- @@@@ 시험ID 입력
```

MySQL



1 • SELECT * FROM teamproject.weak_problem_by_user;

	USERID	PROBLEMTYPEID
▶	U000000001	건축계획

결과

MySQL 3: INSERT, UPDATE

- 새로운 21번 유저가 들어와, 새로운 시험을 봤을 때 업데이트 되는 정보들을 조정

21번 유저가 1번 시험을 본경우 시험 정보가 problem_feedback에 저장

21번 유저 생성하여 User에 저장

```
3 • INSERT INTO USERS VALUES('U00000021');
```

```
6 • INSERT INTO PROBLEM_FEEDBACK
7 VALUES
8 ( "P0000001" , "U00000021" , 1 , 60.46497947 ) ,
9 ( "P0000002" , "U00000021" , 0 , 68.56974824 ) ,
10 ( "P0000003" , "U00000021" , 1 , 66.18977062 ) ,
11 ( "P0000004" , "U00000021" , 1 , 54.84809585 ) ,
12 ( "P0000005" , "U00000021" , 1 , 53.66806588 ) ,
13 ( "P0000006" , "U00000021" , 1 , 49.62748421 ) ,
14 ( "P0000007" , "U00000021" , 0 , 65.49899124 ) ,
15 ( "P0000008" , "U00000021" , 0 , 46.04169016 ) ,
16 ( "P0000009" , "U00000021" , 1 , 73.11515826 ) ,
17 ( "P0000010" , "U00000021" , 1 , 56.26865842 ) ,
18 ( "P0000011" , "U00000021" , 0 , 64.7681075 ) ,
19 ( "P0000012" , "U00000021" , 1 , 56.07500984 ) ,
20 ( "P0000013" , "U00000021" , 1 , 49.47099068 ) ,
21 ( "P0000014" , "U00000021" , 1 , 66.51386844 ) ,
22 ( "P0000015" , "U00000021" , 1 , 131.7551697 ) ,
23 ( "P0000016" , "U00000021" , 1 , 125.147758 ) ,
24 ( "P0000017" , "U00000021" , 1 , 121.4118183 ) ,
25 ( "P0000018" , "U00000021" , 0 , 157.6911464 ) ,
26 ( "P0000019" , "U00000021" , 1 , 116.0928751 ) ,
27 ( "P0000020" , "U00000021" , 0 , 111.3085691 ) ,
28 ( "P0000021" , "U00000021" , 1 , 136.053946 ) ,
```

```
85 ( "P0000078" , "U00000021" , 0 , 116.9282947 ) ,
86 ( "P0000079" , "U00000021" , 1 , 125.0645527 ) ,
87 ( "P0000080" , "U00000021" , 0 , 111.7056461 ) ,
88 ( "P0000081" , "U00000021" , 1 , 115.6708529 ) ,
89 ( "P0000082" , "U00000021" , 1 , 134.801349 ) ,
90 ( "P0000083" , "U00000021" , 1 , 123.493495 ) ,
91 ( "P0000084" , "U00000021" , 1 , 106.3178183 ) ,
92 ( "P0000085" , "U00000021" , 1 , 121.2992638 ) ,
93 ( "P0000086" , "U00000021" , 0 , 130.1244128 ) ,
94 ( "P0000087" , "U00000021" , 1 , 130.1575721 ) ,
95 ( "P0000088" , "U00000021" , 1 , 106.7563859 ) ,
96 ( "P0000089" , "U00000021" , 1 , 113.5873629 ) ,
97 ( "P0000090" , "U00000021" , 0 , 119.4984477 ) ,
98 ( "P0000091" , "U00000021" , 0 , 119.2126789 ) ,
99 ( "P0000092" , "U00000021" , 1 , 117.4087721 ) ,
100 ( "P0000093" , "U00000021" , 0 , 116.3656377 ) ,
101 ( "P0000094" , "U00000021" , 1 , 119.0264116 ) ,
102 ( "P0000095" , "U00000021" , 0 , 65.69105394 ) ,
103 ( "P0000096" , "U00000021" , 1 , 64.87422699 ) ,
104 ( "P0000097" , "U00000021" , 0 , 62.96511307 ) ,
105 ( "P0000098" , "U00000021" , 1 , 58.43087281 ) ,
106 ( "P0000099" , "U00000021" , 1 , 62.22905599 ) ,
107 ( "P0000100" , "U00000021" , 1 , 66.01392698 ) ;
```

MySQL 3: INSERT, UPDATE

- 해당 시험의 등수와 점수분포를 갱신하고, 추가적으로 해당 시험의 각 문제 정답률과 각 문제 풀이 시간의 평균과 표준편차를 갱신

유저정보, 점수 추가 및 랭킹 업데이트

```

109 • SET @cnt = 0;
110
111 • SELECT count(*) INTO @cnt
112   from problem_feedback
113   where userId = 'U00000021' and RightWrong = '1';
114
115   ### STEP1 : test_feedback 테이블 : 유저정보, 점수 추가
116 • INSERT INTO test_feedback
117   VALUES('T0000001', 'U00000021', @cnt, 0); -- @@@@ 유저ID, 시험ID 수동 기입 @@@@
118
119   ### STEP2 : test_feedback : 랭킹 업데이트
120 • CREATE VIEW RE_RANKING AS -- UPDATE 구문 FROM 절에서 자기 참조 안돼서 그냥 뷰 만들음
121   SELECT *, RANK() OVER (ORDER BY SCORE DESC) AS NEW_RANKING
122   FROM test_feedback
123   WHERE TESTID = 'T0000001'; -- @@@@ 시험 ID 수동 기입 @@@@

```

풀이시간 평균 업데이트

```

131 • UPDATE test_feedback
132   SET RE_RANKING = (SELECT NEW_RANKING FROM RE_RANKING
133                     WHERE test_feedback.USERID = RE_RANKING.USERID);
134   -- -----
135   #STEP 3 : PROBLEM TABLE : 풀이시간 평균 업데이트
136
137 • CREATE TABLE PROBLEM_IN_TEST AS -- UPDATE 구문은 자기참조 방지하므로 새로운 테이블 생성
138   SELECT PROBLEMID, TESTID
139   FROM PROBLEM;
140
141 • UPDATE PROBLEM
142   SET SolvingTimeAVG = (SELECT AVG(SolvingTime) FROM problem_feedback
143                         WHERE PROBLEM.PROBLEMID = PROBLEM_FEEDBACK.PROBLEMID AND
144                             PROBLEMID in (SELECT problemid FROM PROBLEM_IN_TEST
145                                             where TestID = NEW_TEST)); -- @@@@시험ID 입력

```

3

MySQL

MySQL 3: INSERT, UPDATE

- 해당 시험의 등수와 점수분포를 갱신하고, 추가적으로 해당 시험의 각 문제 정답률과 각 문제 풀이 시간의 평균과 표준편차를 갱신

풀이시간 표준편차, 정확도 업데이트

Weak_or_not 업데이트

```

147 # STEP4 : PROBLEM TABLE : 풀이시간 표준편차 업데이트
148 • UPDATE PROBLEM
149 SET SolvingTimeStd = (SELECT STD(SolvingTime) FROM problem_feedback
150 WHERE PROBLEM.PROBLEMID = PROBLEM_FEEDBACK.PROBLEMID AND
151 PROBLEMID in (SELECT problemid FROM PROBLEM_IN_TEST
152 where TestID = NEW_TEST)); -- @@@@시험ID 입력
153 -----
154 # STEP5 : PROBLEM TABLE : 정확도 업데이트
155 • CREATE VIEW USERNUM_BY_TEST AS
156 SELECT TESTID, COUNT(USERID) AS NUM
157 FROM TEST_FEEDBACK
158 GROUP BY TESTID;
159
160 • UPDATE PROBLEM
161 SET Accuracy = (SELECT COUNT(RIGHTWRONG)/(SELECT NUM FROM usernum_by_test WHERE TESTID = NEW_TEST)
162 WHERE PROBLEM_FEEDBACK AND
163 problem_feedback.RIGHTWRONG = '1');

```

```

165 # STEP6 : Weak_or_not 업데이트
166 -- 일단 weak_or_not값 배제하고 row 인서트
167
168 • INSERT INTO WEAK_OR_NOT
169 VALUES ('건축계획', NEW_U_TEST, 0),
170         ('건축시공', NEW_U_TEST, 0),
171         ('건축설비', NEW_U_TEST, 0),
172         ('건축구조', NEW_U_TEST, 0),
173         ('건축법규', NEW_U_TEST, 0);
174
175 -- WEAK_OR_NOT 업데이트
176
177 • UPDATE WEAK_OR_NOT
178 SET WEAKORNOT = 1
179 WHERE (SELECT COUNT(RIGHTWRONG) FROM PROBLEM_FEEDBACK, PROBLEM
180 WHERE ((USERID = NEW_U_TEST AND RIGHTWRONG = 0)
181 OR
182 (USERID = NEW_U_TEST AND
183 PROBLEM.PROBLEMID = PROBLEM_FEEDBACK.PROBLEMID AND
184 PROBLEM_FEEDBACK.SOLVINGTIME > PROBLEM.SOLVINGTIMEAVG + PROBLEM.SOLVINGSTD))
185 AND
186 PROBLEM.PROBLEMTYPEID = WEAK_OR_NOT.PROBLEMTYPEID
187 GROUP BY PROBLEM.PROBLEMTYPEID
188 ) > 65 ;

```

감사합니다

2022-2 산업정보관리론

- 24조 -