

2025년 상반기 K-디지털 트레이닝

MySQL 기반 RDBMS 활용과 JDBC <모듈평가>

[KB] IT's Your Life

1 달성 목표

- MySQL 데이터베이스에 저장된 데이터를 SQL 쿼리로 조작하고 관리
- JDBC를 사용하여 Java 애플리케이션에서 SQL 쿼리를 실행하는 능력을 평가

2 평가 방법

- 참가자는 주어진 데이터베이스 스키마를 기반으로 SQL 쿼리를 작성 (1~3번 문항)
- JDBC를 통해 실행하는 Java 프로그램을 개발 (4~6번 문항)

3 기술 요구사항

- SQL: SELECT, INSERT, UPDATE, DELETE, 함수, 조인, 서브쿼리 등의 쿼리 작성 능력
- JDBC: 데이터베이스 연결, 쿼리 실행, 결과 처리 방법의 이해와 구현

평가답안 제출 요령

- 평가와 관련된 파일은 공유 폴더에서 다운로드 받습니다.

[공유폴더 다운로드 경로](#) ← (클릭)

제출 파일명 및 제출 형태

1. 제출 파일명 안내

1) SQL 쿼리문

- 문제 1, 2, 3번 : 각 번호별 sql문 작성 후 **00회차_성명.txt** 에 sql문 답안 작성
- 공유 폴더의 **00회차_성명.txt** 양식 활용요망

2) JAVA 코드

- 문제 4, 5, 6번 : 각 번호별 java 부분 코드 작성 후 프로젝트 파일명 **module2.zip** 으로 압축
- IntelliJ에서 프로그램을 완성 후 정답 부분만 **00회차_성명.txt** 양식에 복사하여 제출함

2. **00회차_성명.txt**과 **module2.zip** 파일을 모두 zip으로 묶어서 하나의 파일로 제출

(00회차_성명.txt 양식에 1-6번 문항 답안 모두 작성)

- 최종 제출파일명 : **00회차_성명.zip**

— 전체 6개의 문항(1번 ~ 6번)

1~3번 문항. SQL(mysql8 기준) → 00회차_성명.txt 필수제출

- 문제 1번 : SQL기본
- 문제 2번 : SQL함수
- 문제 3번 : 조인, 서브쿼리

4~6번 문항. JDBC프로그래밍 → 00회차_성명.txt + module2.zip 필수제출

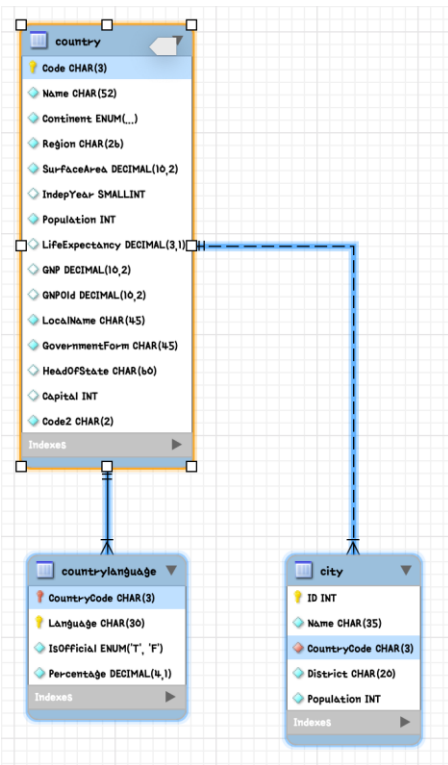
- 문제 4번 : insert 기능 구현
- 문제 5번 : selectList 기능 구현
- 문제 6번 : selectOne 기능 구현

— 작업 순서 정리

1. root 계정으로 로그인합니다.
2. 평가데이터 생성 : 평가에 사용될 데이터는 주어진 스크립트 파일(world2.sql)로 Database, table을 생성하고, 데이터를 삽입하여 준비합니다.
3. 문제 1~3번에 대한 답안 SQL문을 작성합니다.(00회차_성명.txt 작성양식 활용)
4. 주어진 자바 프로젝트 압축 파일(module2.zip)을 다운로드후 압축을 해제하여 프로젝트를 엽니다.
5. 문제 4~6번은 프로젝트 내 있는 각 파일들을 열어 insert기능과 select기능, UI기능을 각각 구현합니다.
 - mySQL JDBC Driver는 이미 gradle로 의존성 설정이 되어 있습니다.

작업 순서 1) 평가 데이터 생성

- ERD에 맞게 스크립트(world2.sql)를 불러와 실행하여 Database와 table을 생성하시오.



world2.sql의 내용

```
create database world2;
use world2;
```

```
CREATE TABLE `country` (
  `Code` char(3) NOT NULL DEFAULT "",
  `Name` char(52) NOT NULL DEFAULT "",
  `Continent` enum('Asia','Europe','North America','Africa','Oceania','Antarctica','South America')
    NOT NULL DEFAULT 'Asia',
  `Region` char(26) NOT NULL DEFAULT "",
  `SurfaceArea` decimal(10,2) NOT NULL DEFAULT '0.00',
  `IndepYear` smallint DEFAULT NULL,
  `Population` int NOT NULL DEFAULT '0',
  `LifeExpectancy` decimal(3,1) DEFAULT NULL,
  `GNP` decimal(10,2) DEFAULT NULL,
  `GNPOld` decimal(10,2) DEFAULT NULL,
  `LocalName` char(45) NOT NULL DEFAULT "",
  `GovernmentForm` char(45) NOT NULL DEFAULT "",
  `HeadOfState` char(60) DEFAULT NULL,
  `Capital` int DEFAULT NULL,
  `Code2` char(2) NOT NULL DEFAULT "",
  PRIMARY KEY (`Code`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

world2.sql의 내용

```
CREATE TABLE `countrylanguage` (  
  `CountryCode` char(3) NOT NULL DEFAULT "",  
  `Language` char(30) NOT NULL DEFAULT "",  
  `IsOfficial` enum('T','F') NOT NULL DEFAULT 'F',  
  `Percentage` decimal(4,1) NOT NULL DEFAULT '0.0',  
  PRIMARY KEY (`CountryCode`, `Language`),  
  KEY `CountryCode` (`CountryCode`),  
  CONSTRAINT `countryLanguage_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)  
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

```
CREATE TABLE `city` (  
  `ID` int NOT NULL AUTO_INCREMENT,  
  `Name` char(35) NOT NULL DEFAULT "",  
  `CountryCode` char(3) NOT NULL DEFAULT "",  
  `District` char(20) NOT NULL DEFAULT "",  
  `Population` int NOT NULL DEFAULT '0',  
  PRIMARY KEY (`ID`),  
  KEY `CountryCode` (`CountryCode`),  
  CONSTRAINT `city_ibfk_1` FOREIGN KEY (`CountryCode`) REFERENCES `country` (`Code`)  
) ENGINE=InnoDB AUTO_INCREMENT=4080 DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_0900_ai_ci;
```

world2.sql의 내용

```
INSERT INTO `country`
(Code, Name, Continent, Region, SurfaceArea, IndepYear, Population, LifeExpectancy, GNP, GNPOld, LocalName, GovernmentForm, HeadOfState, Capital, Code2)
VALUES
('TKM', 'Turkmenistan', 'Asia', 'Southern and Central Asia', 488100.00, 1991, 4459000, 60.9, 4397.00, 2000.00, 'Türkmenostan', 'Republic', 'Saparmurat Niyazov', 3419, 'TM'),
('TUV', 'Tuvalu', 'Oceania', 'Polynesia', 26.00, NULL, 12000, 66.3, 6.00, NULL, 'Tuvalu', 'Constitutional Monarchy', 'Elisabeth II', 3424, 'TV'),
('PNG', 'Papua New Guinea', 'Oceania', 'Melanesia', 462840.00, 1975, 4807000, 63.1, 4988.00, NULL, 'Papua Niu Gini', 'Constitutional Monarchy', 'Elisabeth II', 2884, 'PG'),
('TZA', 'Tanzania', 'Africa', 'Eastern Africa', 883749.00, 1961, 33517000, 52.3, 8005.00, 7388.00, 'Tanzania', 'Republic', 'Benjamin William Mkapa', 3306, 'TZ'),
('NPL', 'Nepal', 'Asia', 'Southern and Central Asia', 147181.00, 1769, 23930000, 57.8, 4768.00, 4837.00, 'Nepal', 'Constitutional Monarchy', 'Gyanendra Bir Bikram', 2729, 'NP'),
('CHE', 'Switzerland', 'Europe', 'Western Europe', 41284.00, NULL, 7160400, 79.6, 264478.00, 256092.00, 'Schweiz/Suisse/Svizzera/Svizra', 'Federation', 'Adolf Ogi', 3248, 'CH'),
('EGY', 'Egypt', 'Africa', 'Northern Africa', 1001449.00, 1922, 68470000, 63.3, 82710.00, 75617.00, 'Misr', 'Republic', 'Hosni Mubarak', 608, 'EG'),
('GMB', 'Gambia', 'Africa', 'Western Africa', 11295.00, 1965, 1305000, 53.2, 320.00, 325.00, 'The Gambia', 'Republic', 'Yahya Jammeh', 904, 'GM'),
('OMN', 'Oman', 'Asia', 'Middle East', 309500.00, 1951, 2542000, 71.8, 16904.00, 16153.00, 'Uman', 'Monarchy (Sultanate)', 'Qabus ibn Sa'id', 2821, 'OM'),
('MDG', 'Madagascar', 'Africa', 'Eastern Africa', 587041.00, 1960, 15942000, 55.0, 3750.00, 3545.00, 'Madagasikara/Madagascar', 'Federal Republic', 'Didier Ratsiraka', 2455, 'MG');
```

```
INSERT INTO `countrylanguage` (CountryCode, Language, IsOfficial, Percentage) VALUES
('TKM', 'Turkmen', 'T', 85.0),
('TUV', 'Tuvaluan', 'T', 96.5),
('PNG', 'Hiri Motu', 'F', 15.0),
('TZA', 'Swahili', 'T', 90.0),
('NPL', 'Nepali', 'T', 80.0),
('CHE', 'German', 'T', 65.0),
('CHE', 'French', 'F', 22.5),
('EGY', 'Arabic', 'T', 99.0),
('GMB', 'English', 'F', 20.0),
('OMN', 'Arabic', 'T', 75.0),
('MDG', 'Malagasy', 'T', 85.0);
```

```
INSERT INTO `city` (Name, CountryCode, District, Population) VALUES
('Ashgabat', 'TKM', 'Ahal', 1000000),
('Funafuti', 'TUV', 'Funafuti', 6000),
('Port Moresby', 'PNG', 'National Capital', 364125),
('Dar es Salaam', 'TZA', 'Dar es Salaam', 4364541),
('Kathmandu', 'NPL', 'Bagmati', 1000000),
('Zurich', 'CHE', 'Zurich', 402762),
('Geneva', 'CHE', 'Geneva', 201818),
('Cairo', 'EGY', 'Cairo', 9900000),
('Banjul', 'GMB', 'Banjul', 35701),
('Muscat', 'OMN', 'Muscat', 797000),
('Antananarivo', 'MDG', 'Analamanga', 1391433);
```

작업 순서 2) 문제 1번 문항

① 문제 1번(SQL기본 - SQL문 이용하여 작성)

1-1. city테이블에 대한 전체 내용을 출력

1-2. countrylanguage테이블에서 CountryCode가 'CHE'인 Language, Percentage컬럼 출력

1-3. city테이블에 다음 항목 데이터를 삽입

- 아이디 : 자동 증가
- 도시 이름: Cairo
- 국가 코드: EGY
- 지구(District): Cairo Governorate
- 인구: 9,500,000

1-4. city 테이블에서 CountryCode가 'PNG'인 모든 행의 Name 열 값을 'GoodSite'로 변경

1-5. country 테이블에서 Name 열을 기준으로 내림차순으로 정렬하여 모든 컬럼을 출력

② 문제 2번(SQL함수 - SQL문 이용하여 작성)

- 2-1. city테이블에서 Name 컬럼의 데이터를 모두 대문자로 변경하여 출력
- 2-2. city테이블에서 ID 컬럼과 CountryCode 컬럼의 데이터를 결합하여 “4080-TKM”와 같이 출력
- 2-3. city테이블에서 District 컬럼 데이터(영문)의 글자수를 출력
- 2-4. city테이블에서 Population 컬럼의 평균값을 출력
- 2-5. country테이블에서 Continent별 최대값 GNP값 출력
(출력 항목은 Continent 명, GNP 최대값. 출력 순서는 GNP 최대값의 오름차순)

작업 순서 2) 문제 3번 문항

③ 문제 3번(조인, 서브쿼리 - SQL문 이용하여 작성)

(괄호 안에 명시되어있는 구문을 반드시 사용해야함)

3-1. city 테이블과 country 테이블을 CountryCode 컬럼과 Code 컬럼 기준으로 조인하고,
city의 Name과 country의 Name을 출력 하시오. (inner join문 이용)

3-2. country테이블에서 모든 국가명을 출력하고 각 국가에서 사용되는 언어를 함께 출력
(사용 언어가 없는 경우 NULL 표시, outer join문 이용)

3-3. city테이블에서 최대 인구를 가진 도시의 이름을 출력
(서브쿼리, MAX()함수 이용)

작업 순서 3) 프로젝트 오픈_문제 4,5,6번 환경세팅

- module2.zip 프로젝트 압축 파일을 다운로드 받아 압축 해제 후, 프로젝트를 오픈
- IntelliJ에서 프로그램을 완성 후 정답 부분만 00회차_성명.txt 에 복사하여 제출함

작업 순서 4) 문제 4, 5, 6번 실행 화면

- 문제 4번 실행 화면(Main.java), 세부 처리 조건은 소스코드의 주석으로 명시되어 있음.

```
1) City 추가, 2) City 목록 보기, 3) 상세 보기, 4) 종료
선택>> 1
삽입처리 호출 완료됨.
```

- 문제 5번 실행 화면(Main.java), 세부 처리 조건은 소스코드의 주석으로 명시되어 있음.

```
1) City 추가, 2) City 목록 보기, 3) 상세 보기, 4) 종료
선택>> 2
추출할 도시수>> 5
전체 도시수 : 5
----- 도시 목록 -----
아이디>> 4087,   국가코드>> EGY,   지역명>> Cairo,   인구수>> 9900000
아이디>> 4083,   국가코드>> TZA,   지역명>> Dar es Salaam,   인구수>> 4364541
아이디>> 4090,   국가코드>> MDG,   지역명>> Analamanga,   인구수>> 1391433
아이디>> 4080,   국가코드>> TKM,   지역명>> Ahal,   인구수>> 1000000
아이디>> 4084,   국가코드>> NPL,   지역명>> Bagmati,   인구수>> 1000000
```

- 문제 6번 실행 화면(Main.java), 세부 처리 조건은 소스코드의 주석으로 명시되어 있음.

```
1) City 추가, 2) City 목록 보기, 3) 상세 보기, 4) 종료
선택>> 3
추출할 도시 ID>> 4087
아이디>> 4087,   국가코드>> EGY,   지역명>> Cairo,   인구수>> 9900000
```

작업 순서 4) 문제 4번 문항

④ 문제 4번(City.java, CityDao.java내의 insert())메서드 기능 구현

(java파일 내 주석에 명시된 요구사항에 따라 “구현한 코드”를 답안 파일에 복사해서 넣으시오.

application.properties의 db 서버 주소 및 root 계정 비밀번호는 자신의 컴퓨터에 맞게 수정하세요.)

4-1. city테이블의 상세 정보를 참고하여 City.java를 완성하시오.

(Lombok 사용, 기본 생성자, 전체 매개변수 생성자, Getter, Setter, toString, Builder 등 사용)

Name: city										
Column	Datatype	PK	NN	UQ	B...	UN	ZF	AI	G	Default / Expression
ID	INT	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
Name	CHAR(35)	<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"/>	''
CountryCode	CHAR(3)	<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"/>	''
District	CHAR(20)	<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"/>	''
Population	INT	<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"/>	'0'

4-2. 다음 데이터를 city테이블에 저장할 수 있는 CityDao.java내의 insert()메서드를 완성하시오.

(단, ID는 자동증가하는 컬럼이므로, ID는 화면과 다른 값으로 삽입될 수 있다.)

ID	Name	CountryCode	District	Population
4093	Win	OMN	Center	1000

⑤ 문제 5번(Main.java, CityDao.java내의 select()메서드 기능 구현)

(java파일 내 주석에 명시된 요구사항에 따라 “구현한 코드”를 답안 파일에 복사해서 넣으시오.)

5-1. city테이블에 인구수로 내림차순 정렬한 후 입력받은 추출 건 수 만큼 검색할 수 있도록 CityDao.java내의 selectList()메서드를 완성하시오.

5-2. 검색된 결과를 CityDao.java로 부터 반환 받아 다음과 같이 화면에 출력하는 Main의 printCity()를 완성하시오.
(printCity()는 한 행을 출력하는 메서드임)

```
1) City 추가, 2) City 목록 보기, 3) 상세 보기, 4) 종료
선택>> 2
추출할 도시수>> 5
전체 도시수 : 5
----- 도시 목록 -----
아이디>> 4087,   국가코드>> EGY,   지역명>> Cairo,   인구수>> 9900000
아이디>> 4083,   국가코드>> TZA,   지역명>> Dar es Salaam,   인구수>> 4364541
아이디>> 4090,   국가코드>> MDG,   지역명>> Analamanga,   인구수>> 1391433
아이디>> 4080,   국가코드>> TKM,   지역명>> Ahal,   인구수>> 1000000
아이디>> 4084,   국가코드>> NPL,   지역명>> Bagmati,   인구수>> 1000000
```

⑥ 문제 6번(Main.java, CityDao.java내의 selectOne()메서드 기능 구현)

(java파일 내 주석에 명시된 요구사항에 따라 “구현한 코드”를 답안 파일에 복사해서 넣으시오.)

6-1. City의 id값을 전달받았을 때 해당 City를 찾아 반환하는 CityDao의 selectOne() 메서드를 완성하시오.
(데이터가 없는 경우 null을 리턴함)

6-2. 검색된 결과를 CityDao.java로 부터 반환 받아 다음과 같이 화면에 출력하는 Main의 showOne()을 완성하세요.
(printCity()를 이용하여 출력함, 데이터가 없는 경우 "해당 도시가 없습니다." 출력)

```
1) City 추가, 2) City 목록 보기, 3) 상세 보기, 4) 종료
선택>> 3
추출할 도시 ID>> 4087
아이디>> 4087,   국가코드>> EGY,   지역명>> Cairo,   인구수>> 9900000
```

수고하셨습니다!

