

2020 spring Database

PROJECT 1-2 TUTORIAL

Project 1-2

- Implementing DDL
 - Project 1-1에 기반하여 스키마를 저장하고 관리할 수 있는 기능을 구현 (create table, drop table, desc, show tables)
- 프로그램이 종료되어도 스키마와 데이터는 없어지면 안 됨
 - 즉, 여러분들의 스키마와 데이터는 파일에 저장되어야 합니다.

Berkeley DB

- Provided by Oracle
- <key, value> store/load
- Download from...
- <http://www.oracle.com/technetwork/database/database-technologies/berkeleydb/downloads/index.html>
- Berkeley DB Java Edition 7.5.11

Overview Downloads Documentation Community Learn More

Oracle Berkeley DB Downloads

Sign up for [Berkeley DB release announcements](#) so we can alert you to future releases and other Berkeley DB product updates.

The Oracle Berkeley DB versions downloaded from this site are licensed under open source terms. Oracle Berkeley DB is also made available for use under an Oracle commercial license. Please contact berkeleydb-info_us@oracle.com for more details on obtaining a commercial version of Oracle Berkeley DB.

Latest Production Releases

Berkeley DB 18.1 (18.1.32) [Previous Releases](#)

[change log](#)

- [Berkeley DB 18.1.32.tar.gz](#), with AES encryption (42M) [sha](#)
- [Berkeley DB 18.1.32.zip](#), with AES encryption (57M) [sha](#)
- [Berkeley DB 18.1.32_64.msi](#), with AES encryption (37M) [sha](#)

Berkeley DB Java Edition 7.5.11 [Previous Releases](#)

[change log](#)

- [Berkeley DB Java Edition je-7.5.11.tar.gz](#) (13M) [md5](#)
- [Berkeley DB Java Edition je-7.5.11.zip](#) (15M) [md5](#)

Berkeley DB XML 6.1.4 [Previous Releases](#)

[change log](#)

- [Berkeley DB XML 6.1.4.tar.gz](#) (84M) [md5](#)
- [Berkeley DB XML 6.1.4.zip](#) (105M) [md5](#)
- [Berkeley DB XML 6.1.4_x86.msi](#) (38M) [md5](#)
- [Berkeley DB XML 6.1.4_x64.msi](#) (39M) [md5](#)



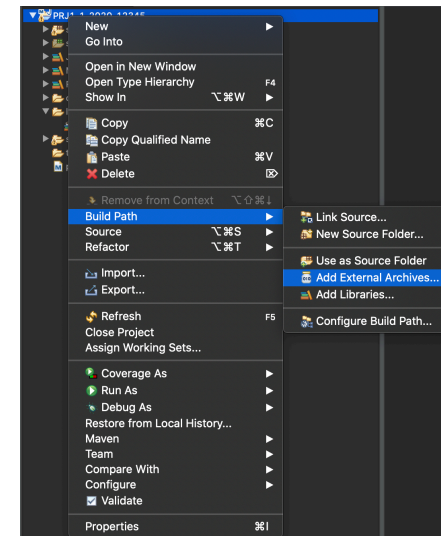
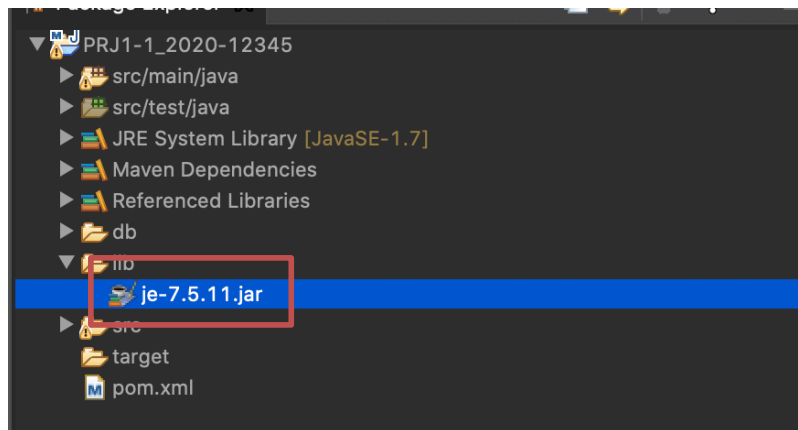
이름	수정한 날짜	유형	크기
ant	2017-10-31 오후...	파일 폴더	
dist	2017-10-31 오후...	파일 폴더	
docs	2019-04-07 오후...	파일 폴더	
examples	2019-04-07 오후...	파일 폴더	
lib	2019-04-07 오후...	파일 폴더	
src	2019-04-07 오후...	파일 폴더	
test	2019-04-07 오후...	파일 폴더	
build.xml	2017-10-31 오후...	XML 파일	95KB
build-common.xml	2017-10-31 오후...	XML 파일	1KB
FindBugsExclude.xml	2017-10-31 오후...	XML 파일	13KB
ivy.xml	2017-10-31 오후...	XML 파일	1KB
LICENSE	2017-10-31 오후...	파일	12KB
README	2017-10-31 오후...	파일	1KB

이름	수정한 날짜	유형	크기
je-7.5.11.jar	2017-10-31 오후...	Executable Jar File	3,210KB
JEJConsole.jar	2017-10-31 오후...	Executable Jar File	58KB
RepJEConsole.jar	2017-10-31 오후...	Executable Jar File	104KB



Lib 추가

- 반드시 사용한 jar 파일은 프로젝트 안의 lib폴더 안에 넣어서 제출하도록 합니다.
- Project right click-> Build Path -> Add External Archives -> select “je-7.5.11.jar” from lib folder



프로젝트1-2 3줄요약

- ① DB 연결
- ② <key, value> DB에 넣기
- ③ <key, value> DB에서 찾아서 가져오기

① DB 연결

- *.jj 파일의 `PARSER_BEGIN ~ PARSER_END` 부분에서 java 코드를 이용하여 초기화하는 부분에서 **je-7.5.11.jar** 파일에 있는 class들을 import 합니다.

```
// import
import com.sleepycat.je.Database;
import com.sleepycat.je.DatabaseException;
import com.sleepycat.je.DatabaseConfig;
import com.sleepycat.je.Cursor;

import com.sleepycat.je.Environment;
import com.sleepycat.je.EnvironmentConfig;
```

① DB 연결

- DB 선언과 열기

http://docs.oracle.com/cd/E17277_02/html/java/com/sleepycat/je/DatabaseConfig.html

```
// Environment & Database define

Environment myDbEnvironment = null;
Database myDatabase = null;

/* OPENING DB */

// Open Database Environment or if not, create one.
EnvironmentConfig envConfig = new EnvironmentConfig();
envConfig.setAllowCreate(true);
myDbEnvironment = new Environment(new File("db/"), envConfig);

// Open Database or if not, create one.
DatabaseConfig dbConfig = new DatabaseConfig();
dbConfig.setAllowCreate(true);
dbConfig.setSortedDuplicates(true);
myDatabase = myDbEnvironment.openDatabase(null, "sampleDatabase", dbConfig);
```


중요!!



- 오른쪽 빨간 네모 부분이 로컬에 DB폴더 안에 파일로 key, value들을 저장하는 경로입니다.
- 이상한 자기 컴퓨터 내의 경로 넣지 말고, 프로젝트 안에 db 폴더를 생성하여 그 폴더를 DB파일 폴더로 사용하고 같이 제출해주세요.

① DB 연결

- DB 닫기

```
/* CLOSING DB */  
if (myDatabase != null) myDatabase.close();  
if (myDbEnvironment != null) myDbEnvironment.close();
```

② <key, value> DB에 넣기

```
Cursor cursor = null;
DatabaseEntry key;
DatabaseEntry data;

try {
    cursor = myDatabase.openCursor(null, null);
    key = new DatabaseEntry("key".getBytes("UTF-8"));
    data = new DatabaseEntry("value".getBytes("UTF-8"));
    cursor.put(key, data);
} catch (DatabaseException de){
} catch (UnsupportedEncodingException e) {
    e.printStackTrace();
}
```

Cursor



key	Value
k1	V1
k2	V2
k3	V3
...	...

넣고자 하는 string을 byte data 형태로 변형

③ <key, value> 찾아서 가져오기

```
DatabaseEntry foundKey = new DatabaseEntry();
DatabaseEntry foundData = new DatabaseEntry();

cursor.getFirst(foundKey, foundData, LockMode.DEFAULT); Cursor를 key, value 첫 pair부터

do {
    String keyString = new String(foundKey.getData(), "UTF-8");
    String dataString = new String(foundData.getData(), "UTF-8");
} while ( cursor.getNext(foundKey, foundData, LockMode.DEFAULT) == OperationStatus.SUCCESS );
```

하나씩 다음으로 옮기면서 Data를 가져옴.

요약

- Berkeley DB를 이용하여 Schema를 파일에 저장
- Schema를 <key, value> 형태로 저장
 - 설계는 여러분들의 몫!
 - 어떻게 설계했는지 보고서에 명시할 것!
- **자세한 설명은 프로젝트 문서, Message 문서, About Referential Constraint 문서를 참고**

공지

- 프로젝트에 관한 전반적인 것 관련 문의는 마감 전 월요일 (5/11)까지 받습니다.
 - 또한 프로젝트 전반적인 것 외 본인의 코드 구현에 따른 문의는 형평성을 위해 답변이 원칙적으로 불가능합니다.
 - Jar 파일 정상작동 여부 필히 확인
 - 1-1과 마찬가지로 etl 공지사항에 중요 QA 공지 예정
 - 주어진 소스 코드는 참고용 (수정 가능성 존재)
-
- 1-1 성적 공개 금일(4/20) 오후 2시
 - 성적 문의 기간은 금주 수요일(4/22)까지

[2. Database Environments](#)

[Opening Database Environments](#)

[Multiple Environments](#)
[Multiple Environment Subdirectories](#)
[Configuring a Shared Cache for Multiple Environments](#)

[Closing Database Environments](#) [Environment Properties](#)

[The EnvironmentConfig Class](#)
[EnvironmentMutableConfig](#)

[Environment Statistics](#)
[Database Environment Management Example](#)

[II. Programming with the Base API](#)

[7. Databases](#)

[Opening Databases](#)

[Deferred Write Databases](#)
[Temporary Databases](#)
[Closing Databases](#)

[Database Properties](#)
[Administrative Methods](#)
[Database Example](#)

[8. Database Records](#)

[Using Database Records](#) [Reading and Writing Database Records](#)

[Writing Records to the Database](#)
[Getting Records from the Database](#)
[Deleting Records](#)
[Data Persistence](#)

[Using the BIND APIs](#)

[Numerical and String Objects](#)
[Serializable Complex Objects](#)
[Custom Tuple Bindings](#)

[Using Comparators](#)

[Writing Comparators](#)
[Setting Comparators](#)

[Database Record Example](#)

[9. Using Cursors](#)

[Opening and Closing Cursors](#) [Getting Records Using the Cursor](#)

[Disk Ordered Cursors](#)
[Searching for Records](#)
[Working with Duplicate Records](#)

[Putting Records Using Cursors](#)
[Deleting Records Using Cursors](#)
[Replacing Records Using Cursors](#)
[Cursor Example](#)

더 자세한 내용은

http://docs.oracle.com/cd/E17277_02/html/GettingStartedGuide/index.html