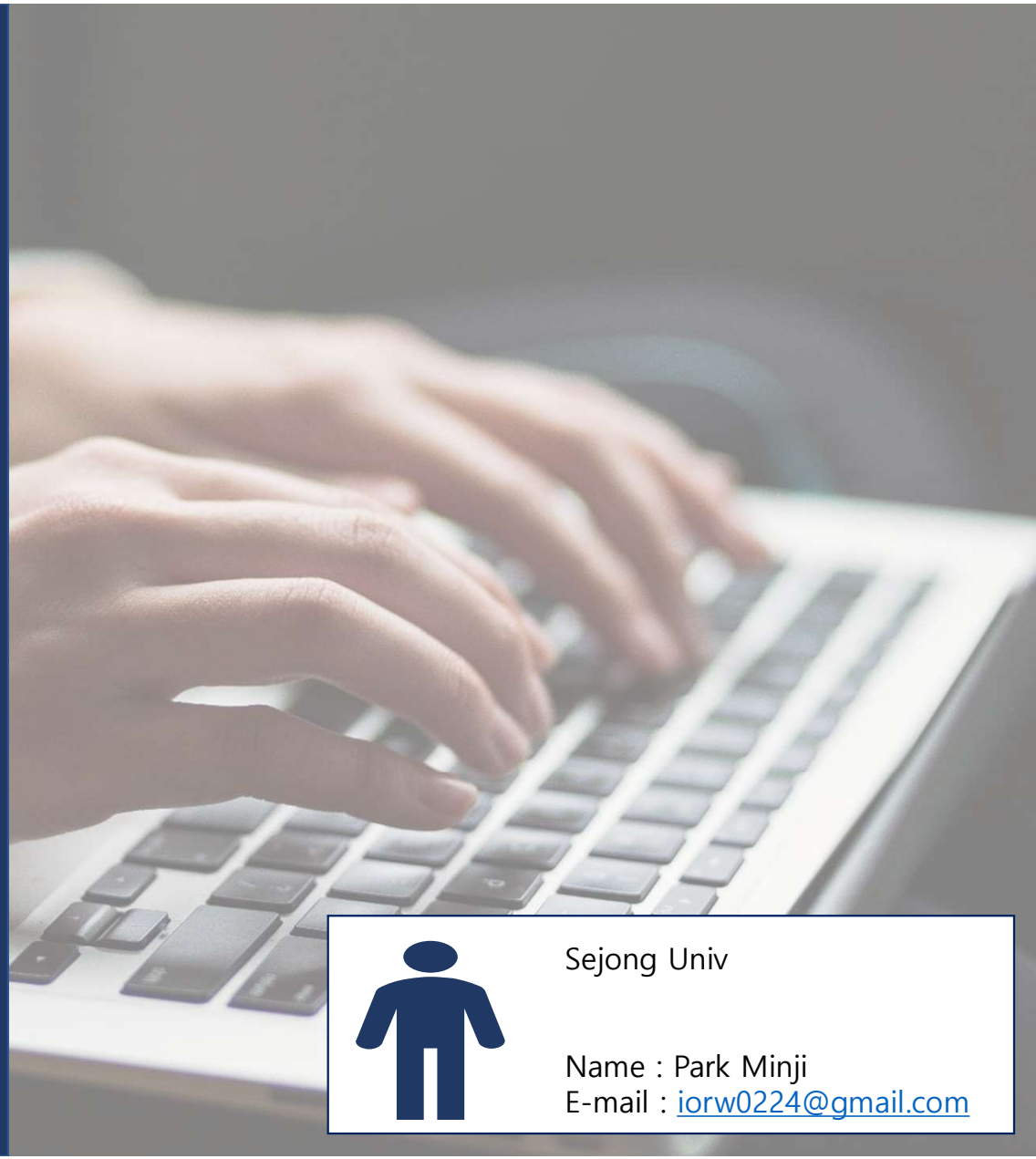
Berkeley DB for TinyIoT



Sejong Univ

Name : Park Minji
E-mail : iorw0224@gmail.com

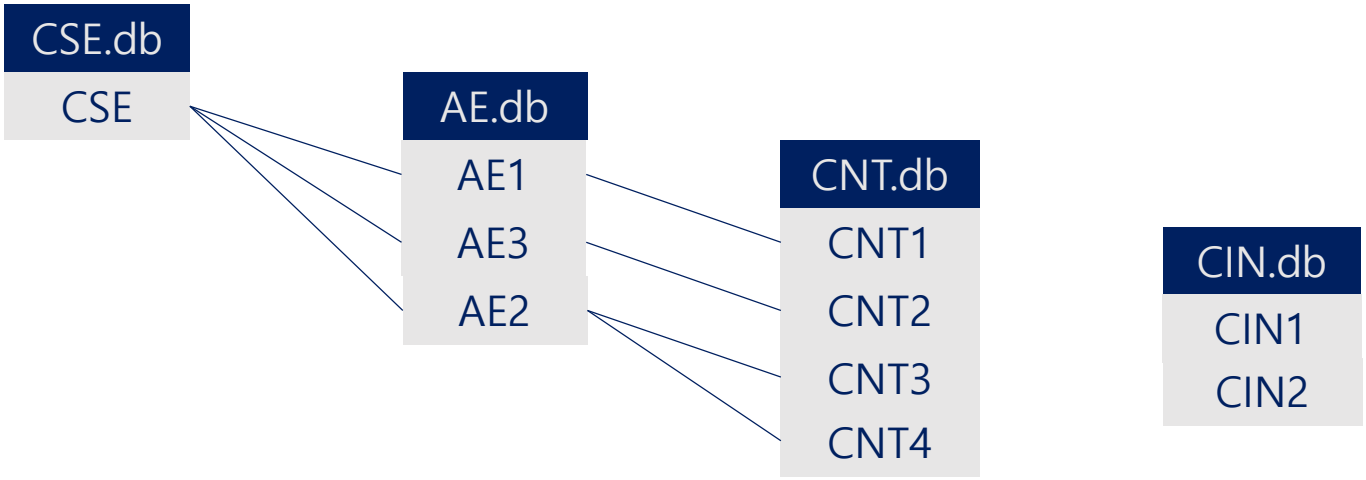
Berkeley DB 이번주 진행 상황

- Mapping Table ~~추가~~고민
- AE.db 안 모든 데이터를 반환하는 **Get_All_AE** 함수 구현
- 디버깅이나 에러 메세지에 대해 표준 에러로 출력(printf -> fprintf)
- Store 함수에서 NULL값이 들어오는 경우, default값으로 설정(int형은 -1, char* 형은 “”, bool형은 true)
- Get, Delete 함수에서 인자로 들어온 ri가 DB에 존재하지 않는 경우, NULL 반환

 Github

 Notion

Berkeley DB Resource Structure



Berkeley DB Mapping table

Mapping.db

Key	Value
CSE	TinyProject
AE	Sensor1
AE	Sensor3
AE	Sensor2
CNT	status1
CNT	status2
CNT	status3
CNT	status4
CIN	4-20220513093154147745
CIN	4-20210513093154147745

(uri) TinyProject/Sensor1/status1/4-20220513093154147745

CSE : TinyProject
AE : Sensor1
CNT : status1
CIN : 4-20220513093154147745

-> Uri에 입력된 값이 ri인가? Rn인가? 에 따라 mapping table 변경

Berkeley DB Store&Get CSE

- Use Cursor - **DB_KEYLAST** : Create a new record, as the last of the duplicate records for the supplied key.
- Set Flag - **DB_DUP** : Permit duplicate data items in the database. The ordering of duplicates in the database is determined by the order of insertion, unless the ordering is otherwise specified by use of a cursor operation or a duplicate sort function.

Struct CSE

```
typedef struct {
  char* ct;
  char* lt;
  char* rn;
  char* ri;
  char* pi;
  char* csi;
  int ty;
} CSE;
```

```
cse.rn = "TinyProject";
cse.ri = "5-20191210093452845";
cse.pi = "NULL";
cse.ty = 5;
cse.ct = "20191210T093452";
cse.lt = "20191210T093452";
cse.csi = "/Tiny Project2";
```

Store_CSE(CSE* cse_object)

CSE.db

Key	Value
csi	/Tiny Project2
ct	20191210T093452
lt	20191210T093452
pi	NULL
ri	5-20191210093452845
rn	TinyProject
ty	5

Struct CSE

```
csi : /Tiny Project2
ct : 20191210T093452
lt : 20191210T093452
pi : NULL
ri : 5-20191210093452845
rn : TinyProject
ty : 5
```

Get_CSE(char* ri)

Berkeley DB Store&Get AE

Struct AE

```
typedef struct {
  char* et;
  char* ct;
  char* lt;
  char* rn;
  char* ri;
  char* pi;
  char* api;
  char* aei;
  int ty;
  bool rr;
} AE;
```

AE1

```
rn = "Sensor1";
ty = 2;
pi = "5-20191210093452845";
ri = "TAE1";
ct = "20220513T083900";
lt = "20220513T083900";
et = "20240513T083900";
api = "tinyProject1";
rr = true;
aei = "TAE1";
```

AE3

```
rn = "Sensor3";
ty = 2;
pi = "5-20191210093452845";
ri = "TAE3";
ct = "20220513T083900";
lt = "20220513T083900";
et = "20240513T083900";
api = "tinyProject3";
rr = true;
aei = "TAE3";
```

AE2

```
rn = "Sensor2";
ty = 2;
pi = "5-20191210093452845";
ri = "TAE2";
ct = "20220513T083900";
lt = "20220513T083900";
et = "20240513T083900";
api = "tinyProject2";
rr = true;
aei = "TAE2";
```

Store_AE(AE* ae_object)

```
Store_AE(&ae1)
Store_AE(&ae3)
Store_AE(&ae2)
```

AE.db

Key	Value
aei	TAE1
aei	TAE3
aei	TAE2
api	tinyProject1
api	tinyProject3
api	tinyProject2
ct	20220513T083900
ct	20220513T083900
ct	20220513T083900
et	20240513T083900
et	20240513T083900
et	20240513T083900
lt	20220513T083900
lt	20220513T083900
lt	20220513T083900
pi	5-20191210093452845
pi	5-20191210093452845
pi	5-20191210093452845
ri	TAE1
ri	TAE3
ri	TAE2
rn	Sensor1
rn	Sensor3
rn	Sensor2
rr	true
rr	true
rr	true
ty	2
ty	2
ty	2

Cursor → ri TAE3

Find "TAE3"
Index[1]

Get_AE(char* ri)

Struct AE3

```
[Get AE] ri = TAE3
ri : TAE3
rn : Sensor3
pi : 5-20191210093452845
et : 20240513T083900
aei : TAE3
api : tinyProject3
ct : 20220513T083900
lt : 20220513T083900
rr : true
ty : 2
```

Berkeley DB Delete AE [ri="TAE3"]

AE.db

Del_AE(char* ri)

Key	Value
aei	TAE1
aei	TAE3
aei	TAE2
api	tinyProject1
api	tinyProject3
api	tinyProject2
ct	20220513T083900
ct	20220513T083900
ct	20220513T083900
et	20240513T083900
et	20240513T083900
et	20240513T083900
lt	20220513T083900
lt	20220513T083900
lt	20220513T083900
pi	5-20191210093452845
pi	5-20191210093452845
pi	5-20191210093452845
ri	TAE1
ri	TAE3
ri	TAE2
rn	Sensor1
rn	Sensor3
rn	Sensor2
rr	true
rr	true
rr	true
ty	2
ty	2
ty	2

Cursor

Find "TAE3"
Index[1]

Return delete object

```
[Delete Object]
ri : TAE3
rn : Sensor3
pi : 5-20191210093452845
et : 20240513T083900
aei : TAE3
api : tinyProject3
ct : 20220513T083900
lt : 20220513T083900
rr : true
```

ri not found

```
[Delete AE] ri = TAE3
ri : TAE3 Not Found
```

AE.db

```
[Delete AE] ri = TAE3
[Display] AE.db
aei : TAE1
aei : TAE2
api : tinyProject1
api : tinyProject2
ct : 20220513T083900
ct : 20220513T083900
et : 20240513T083900
et : 20240513T083900
lt : 20220513T083900
lt : 20220513T083900
pi : 5-20191210093452845
pi : 5-20191210093452845
ri : TAE1
ri : TAE2
rn : Sensor1
rn : Sensor2
rr : true
rr : true
ty : 2
ty : 2
```

Berkeley DB Exception handling

Store 함수에서 NULL 값이 들어온 경우

➡ **Default값으로 지정**

```
cs : 2
cs : -1
```

int형은 -1

```
con : ON
con :
```

char* 형은 "" (공백)

```
rr : true
rr : true
```

bool형은 true

Get 함수에서 존재하지 않는 ri를 호출한 경우

```
[Get AE] ri = TAE3
Not Found
Return NULL
```

NULL 반환

Delete 함수에서 존재하지 않는 ri를 삭제하려는 경우

```
[Delete AE] ri = TAE3
ri : TAE3 Not Found
```

NULL 반환

Berkeley DB Get_All_AE

```
[Get All AE]
ri : TAE1
rn : Sensor1
pi : 5-20191210093452845
et : 20240513T083900
aei : TAE1
api : tinyProject1
ct : 20220513T083900
lt : 20220513T083900
rr : true
ty : 2
ri : TAE3
rn : Sensor3
pi : 5-20191210093452845
et : 20240513T083900
aei : TAE3
api : tinyProject3
ct : 20220513T083900
lt : 20220513T083900
rr : true
ty : 2
ri : TAE2
rn : Sensor2
pi : 5-20191210093452845
et : 20240513T083900
aei : TAE2
api : tinyProject2
ct : 20220513T083900
lt : 20220513T083900
rr : true
ty : 2
```

AE.db 안 모든 데이터 반환

Berkeley DB 다음주 수정할 사항

- Mapping table
- AE.db 반환형에 오브젝트 개수


Github


Notion