

# Berkeley DB for TinyIoT



Sejong Univ

Name : Park Minji

E-mail : [iorw0224@gmail.com](mailto:iorw0224@gmail.com)

# Berkeley DB 이번주 진행 상황

---

- Delete\_CNT, Delete\_CIN 추가
- Get\_All\_CNT, Get\_All\_CIN 추가
- CNT 밑에 CNT 저장해보기 -> 가능
- **Get\_All\_AE** 함수 노드로 반환
- Get\_CSE의 인자(ri) 없이 반환 가능하게 수정
- **Mapping Table** 추가 (URI 와 Key값이 어떻게 맵핑되는지)



[Github](#)



[Notion](#)

# Berkeley DB

BerkeleyDB 함수들

Mapping  
Table

Store

CSE

AE

CNT

CIN

Delete

CSE

AE

CNT

CIN

Get

CSE

AE

CNT

CIN

Get\_All

CSE

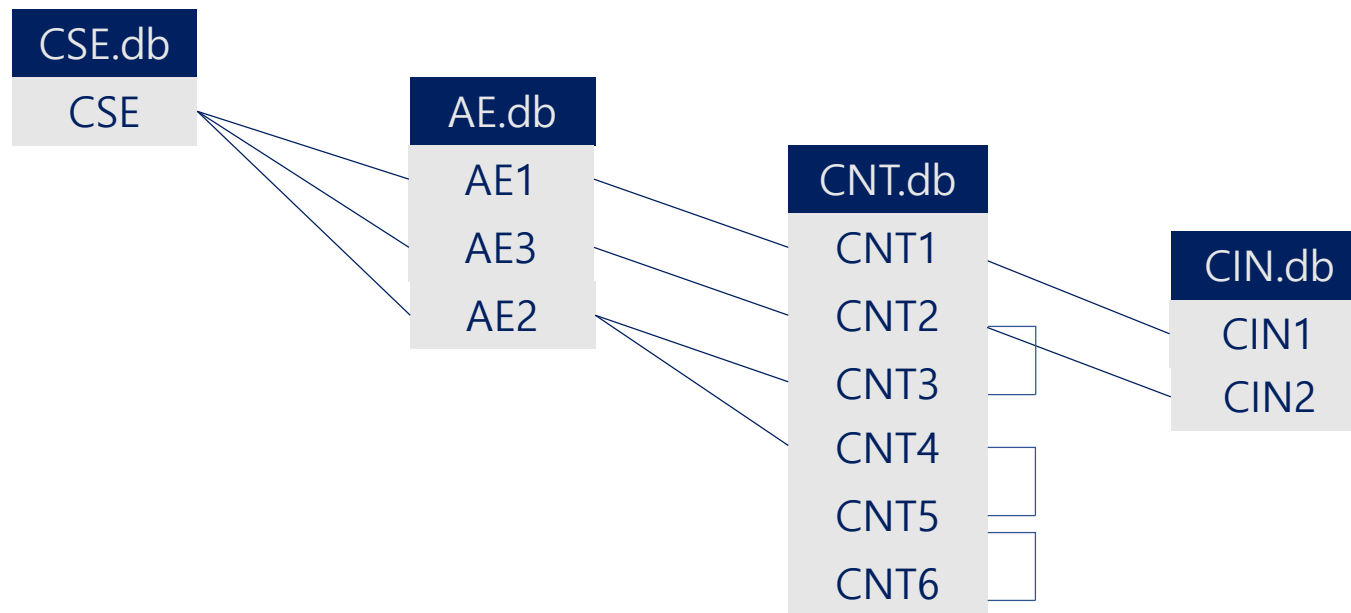
AE

CNT

CIN

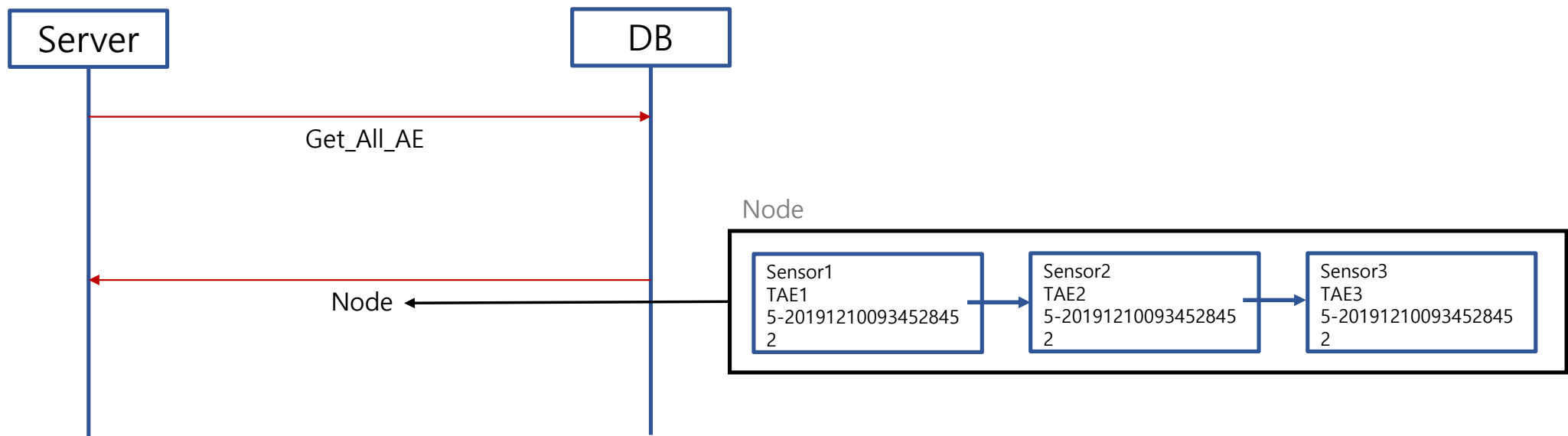
# Berkeley DB Resource Structure

CNT 밑에 CNT 저장 가능



# Berkeley DB Get\_All\_AE

Get\_All\_AE 함수에서 기존 이중 포인터로 반환하는 대신 노드로 반환하는 것으로 변경



## Berkeley DB Get\_CSE

---

CSE값은 1개이기 때문에 Get\_CSE 함수에서 기존의 인자(ri) 없이 반환 가능하게 수정

```
CSE* Get_CSE(char* ri);
```



```
CSE* Get_CSE();
```

# Berkeley DB

Mapping table – Mapping.c //uri로 받은 rn을 ri로 매핑해 Mapping.db에 저장하고, 노드로 반환하는 함수

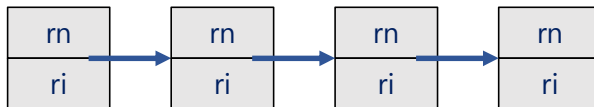
함수 원형

```
Node* Mapping(char uri[]);
```

인자 : 문자열 uri

<URI : TinyProject/Sensor1/status2/status3>

반환형태 : Node



출력값

```
TinyProject : 5-20191210093452845  
Sensor1 : TAE1  
status2 : 3-20210513093154147745  
status3 : 3-20200513093154147745
```

//찾는 rn에 해당하는 ri를 반환하는 함수

```
char* Find_ri(char* database, char* rn)
```

Find\_ri("CSE.db", TinyProject) → 5-20191210093452845

Find\_ri("AE.db", Sensor1) → TAE1

Find\_ri("CNT.db", status2) → 3-2021051309315414147745

Find\_ri("CNT.db", status3) → 3-2020051309315414147745

//Mapping.db에 키,값 형태로 저장하는 함수

```
int Store_map(char* key_str, char* data_str)
```

DB 정렬 기준 – 사전순(숫자-대문자-소문자 순서), 길이순

```
[Display] Mapping.db  
4-20210513093154147745 : 4-20210513093154147745  
Sensor1 : TAE1  
TinyProject : 5-20191210093452845  
status2 : 3-20210513093154147745  
status3 : 3-20200513093154147745
```

# Berkeley DB Mapping table – Finde\_ri 관련 고민할 부분

//찾는 rn에 해당하는 ri를 반환하는 함수

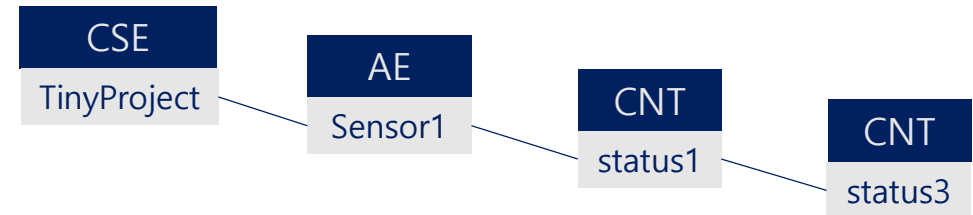
```
char* Find_ri(char* database, char* rn)
```

Find\_ri("CSE.db", TinyProject) → 5-20191210093452845

Find\_ri("AE.db", Sensor1) → TAE1

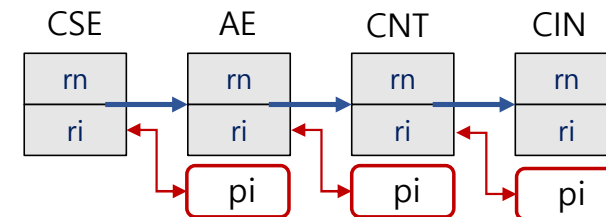
Find\_ri("CNT.db", status2) → 3-2021051309315414147745

Find\_ri("CNT.db", status3) → 3-2020051309315414147745



계층적 구조 표현을 보다 정확하게 하기 위해

=> Finde\_pi함수를 추가해 반환 받은 pi를 siblingLeft의 ri와 비교할 필요?





# Berkeley DB Mapping table

URI : TinyProject/Sensor1/status2/4-20210513093154147745

CSE

AE

CNT

CNT or CIN



CIN에서 rn 탐색 후

Mapping.db

Key(rn)	Value(ri)
TinyProject	5-20191210093452845
Sensor1	TAE1
status2	3-20210513093154147745
4-20210513093154147745	4-20210513093154147745

CSE  
AE  
CNT  
CIN

값이 없으면 NULL 반환 -> CNT에서 rn 탐색  
값이 있으면 해당 rn의 ri 반환

CNT 밑에 CNT 가능

URI : TinyProject/Sensor1/status2/status3

CSE

AE

CNT

CNT or CIN

Mapping.db

Key(rn)	Value(ri)
TinyProject	5-20191210093452845
Sensor1	TAE1
status2	3-20210513093154147745
status3	3-20200513093154147745

CSE  
AE  
CNT  
CNT

# Berkeley DB Mapping table을 어떻게 효율적으로 이용할 수 있는가?

이미 한번 Mapping한 오브젝트는 Mapping.db에 저장되어 또 불러오지 않게 수정할 예정

URI : TinyProject/Sensor1/status2/4-20210513093154147745  
CSE AE CNT CNT or CIN

URI : TinyProject/Sensor1/status2/status3  
CSE AE CNT CNT or CIN

새로운 오브젝트

Mapping.db

	Key(rn)	Value(ri)
CSE	TinyProject	5-20191210093452845
AE	Sensor1	TAE1
CNT	status2	3-20210513093154147745
CNT	status3	3-20200513093154147745
CIN	4-20210513093154147745	4-20210513093154147745

# Berkeley DB

## 다음주 수정할 사항

---

- Mapping table 에서 일정 부분 공간 절약이 되는 부분 고민
- Get\_All\_CSE 구현, Delete\_CSE 구현



[Github](#)



[Notion](#)