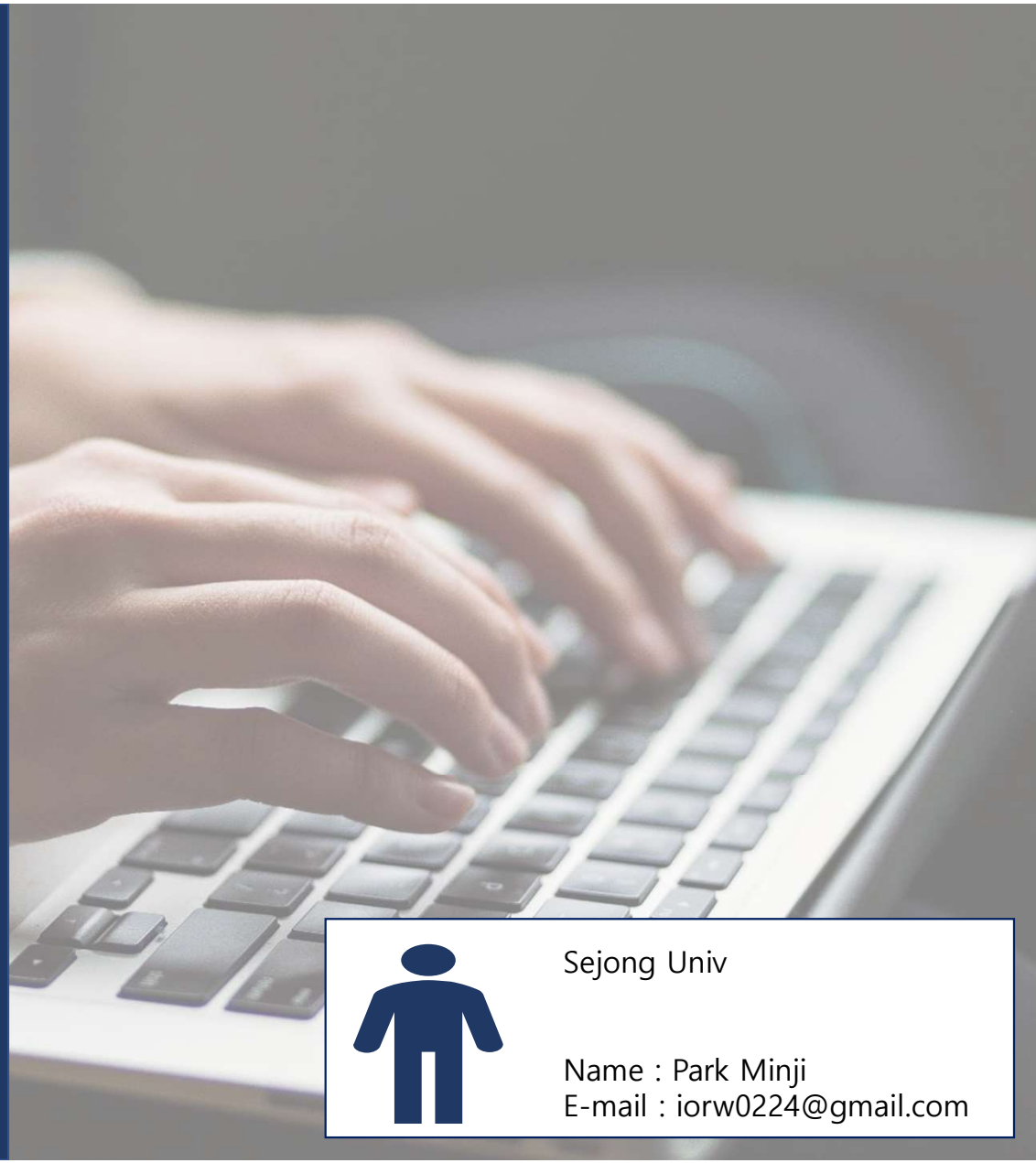
Berkeley DB for Tinylot



Sejong Univ

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

Berkeley DB

- 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

Permit duplicate data items in the database

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 ri="TAE3"

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

Del_AE(char* ri)

Cursor

Find "TAE3"
Index[1]

ri not found

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

Return delete object

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

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
```