

데이터 베이스 시스템 01분반

프로젝트

소프트웨어학부

20194653

윤다인

1. 개발 환경 및 구조

개발 환경 – 파이썬 3.9, visual studio code 사용

main.py – 여기서 모든 API를 불러와 처리한다.

structure.py – 사용한 CLASS를 정의하였다. VLR, SLP, dataDict

API.py – 기능에 대한 함수를 정의하였다.

구조구현의 예시는 모두 아래의 형태를 띈다.

```
table_name = 'student'
col_names = ['ID', 'name', 'grade', 'dept']
col_types = ['c8', 'v16', 'c1', 'v17']
```

table_name 은 string으로, col_names 와 col_types는 같은 length의 list로 이루어져 있다.

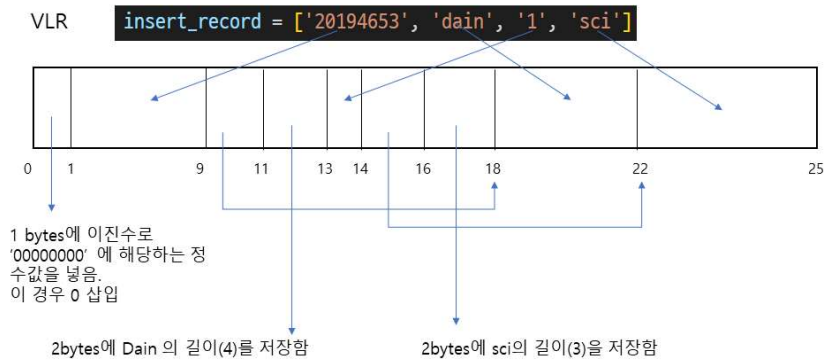
자세한 구현 결과값은 4. 결과 에서 확인 가능하다.

2. structure.py 필요한 클래스를 저장한 파일

1) variable length record (VLR)

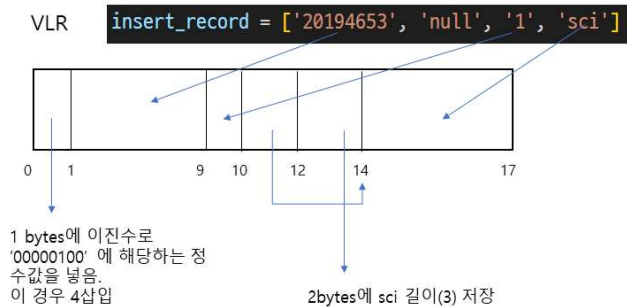
구현하고자 한 VLR의 구조는 다음과 같다

Null 이 없을 경우



Null bitmap
가장 앞에 둬으로써 이 record의
찾고자 하는 column 값이 null인
지 파악을 용이하게 하였다

Null 이 있는 경우 - 아예 공간을 차지하지 않도록 하였다. Null bitmap의 값만 바꾸어주었다.



Null bitmap
가장 앞에 둬으로써 이 record의
찾고자 하는 column 값이 null인
지 파악을 용이하게 하였다

<pre> ''' VLR : 최대 길이 42 bytes ''' VLR_LENGTH = 42 class VLR: def __init__(self): self.vlr = bytearray(VLR_LENGTH) self.tableName = '' self.nullbitmap = '' self.isNotNull=[] # 이 두개는 중복정보긴 함 self.colType = [] # self.colName = [] self.value = [] </pre>	<p>VLR의 최대 길이는 42 bytes로 정하였다.</p> <p>VLR class에 들어있는 정보는 bytearray 로 구현된 vlr 이 vlr 이 포함된 table 명, null bitmap, 해당 column의 값이 null인지 아닌지 판단하는 isNotNull이 있다.</p> <p>colType, colName, value 의 값은 초기화 할 때는 쓰이지 않고 record 검색 할 때 쓰인다.</p>
<pre> def makeVLR(self, tableName:str, insert_columns:list): # table이랑 insert column이랑 meta_data = dataDict() meta_data.getDict(tableName) col_type = meta_data.colType # c인지 v인지 # ===== null bitmap 에 들어갈 숫자 int 형으로 / null인지 아닌지 bool type의 list null_bitmap, tf = self.checkNotNull(insert_columns) # ===== 필요한 숫자 추출 ex) offset 시작 위치 //null값 varstart = 1 numNeed = [0] for coltype, insertcol, ttff in zip(col_type, insert_columns,tf): if ttff == False: continue # null이면 pass if coltype[0] == 'c' : varstart += int(coltype[1:]) elif coltype[0] == 'v' : numNeed.append(len(insertcol)) varstart += OFFSET numNeed[0] =varstart total_length = sum(numNeed) # print(total_length) # ===== self init 할 byte를 total_length 만큼 할당 bitmap = bytearray(total_length) bitmap[0:1] = null_bitmap.to_bytes(1,'big') </pre>	<p>makeVLR은 record 의 값 (string list)을 bytearray로 바꾸는 메서드이다.</p> <p>Null bitmap을 만드는 함수는 아래의 checkNull을 사용하였다.</p> <p>여기서 반환된 정수값을 1byte로 바꿔 [0:1]위치에 1byte로 저장한다.</p> <p>numNeed는 가변길이 정보를 저장하는 시작 위치, 저장할 각 value의 길이를 저장한다.</p>

```

# ===== null bitmap 제외한 [1:] 정보 저장
numNeedNum = 1
bit_start = 1

# null check 해야함 --> return 된 tf bool 리스트가지고
for coltype, insertcol, ttff in zip(col_type, insert_columns , tf):

    if ttff == False: continue # null 이면 아무것도 하지 말기

    if coltype[0] == 'c' :
        cvarlength = len(insertcol)
        bitmap[bit_start : bit_start + cvarlength] = bytearray(insertcol, encoding='UTF-8')
        bit_start += cvarlength

    elif coltype[0] == 'v':
        start = numNeed[0]
        vvarlength = numNeed[numNeedNum]
        bitmap[bit_start : bit_start + OFFSET//2] = start.to_bytes(2,'big') # 시작하는 곳
        bitmap[bit_start + OFFSET//2 : bit_start + OFFSET] = vvarlength.to_bytes(2,'big') # 변수 길이

        bitmap[start : start+ vvarlength] = bytearray(insertcol, encoding='UTF-8') # 뒤쪽에 정보 저장

        numNeed[0] += vvarlength
        numNeedNum += 1
        bit_start += OFFSET

self.vlr = bitmap # 이 bitmap으로 초기화

```

Null 값을 check 하여 null 인 경우는 공간을 차지하지 않게 하였다.

Type이 c인 값은 그대로 그대로 저장하고 type이 v인 값은 offset에 저장할 위치, 길이를 저장한 후 뒤에 길이만큼 따로 저장된다

```

def makeVLR_bytearray(self, bytes, tableName):

    metadata = dataDict()
    metadata.getDict(tableName)
    self.vlr = bytes
    self.colType = metadata.colType
    self.colName = metadata.colName

    # null 값 확인
    null_int = int.from_bytes(bytes[0:1], 'big')
    null_check_str = ('{0:08b}'.format(null_int))[-len(metadata.colType):] # 길이가 8인 스트림
    self.nullbitmap = null_check_str
    self.isNotNull = list(map(mapping, null_check_str))

    ...

    vlr의 byte array와 null 정보로 value를 string으로 가져오기
    ...

    record_pointer = 1 # null부터 시작

    for coltype, ttff in zip(self.colType, self.isNotNull):

        if ttff == False:
            self.value.append('null')
            continue # null 이면 아무것도 하지 말기

        if coltype[0] == 'c' :
            self.value.append(self.vlr[record_pointer: record_pointer + int(coltype[1:]).decode('utf-8')])
            record_pointer += int(coltype[1:])
        elif coltype[0] == 'v':
            stpt = int.from_bytes(self.vlr[record_pointer : record_pointer + OFFSET//2], 'big')
            length = int.from_bytes(self.vlr[record_pointer + OFFSET//2: record_pointer + OFFSET], 'big')

            self.value.append(self.vlr[stpt:stpt + length].decode('utf-8'))
            record_pointer += OFFSET

```

makeVLR_bytearray는 SLP에서 추출한 bytearray의 정보를 기반으로 그 record의 값(string list)로 바꾸는 makeVLR과 정반대의 기능을 하는 메소드이다.

```
def mapping(x):
    # null 이 아니면 0
    if x == '0' : return True
    else: return False
```

```
def checkNull(self, insert_columns):
    null_bitmap_string = ['0','0','0','0','0','0','0','0']
    for i,ins in enumerate(reversed(insert_columns)):
        if ins == 'null': null_bitmap_string[7-i] = '1'
        else: null_bitmap_string[7-i] = '0'

    bitmap_number = int("".join(null_bitmap_string ),2)

    # byte바꾸기 전 정수, null 값인지 [True, True, False, False] 두개 return
    tf = null_bitmap_string[-len(insert_columns):]
    ptf = list(map(mapping, tf))

    return bitmap_number , ptf
```

checkNull은
makeVLR 메서드에서
쓰이는 메서드이다. 1
바이트(0을 8개)로 초
기화 한 후 null 값이
있으면 1로 바꾼다.
그리고 이를 다시 int
로 바꾼 값과, 해당
column의 값이 null
인지 아닌지 T/F 로
Boolean list를 return
한다.

```
def printVLR(self): # VLR의 정보 print
    print(f'bytearray : {self.vlr}')
    print(f'bytearray length : {len(self.vlr)}')

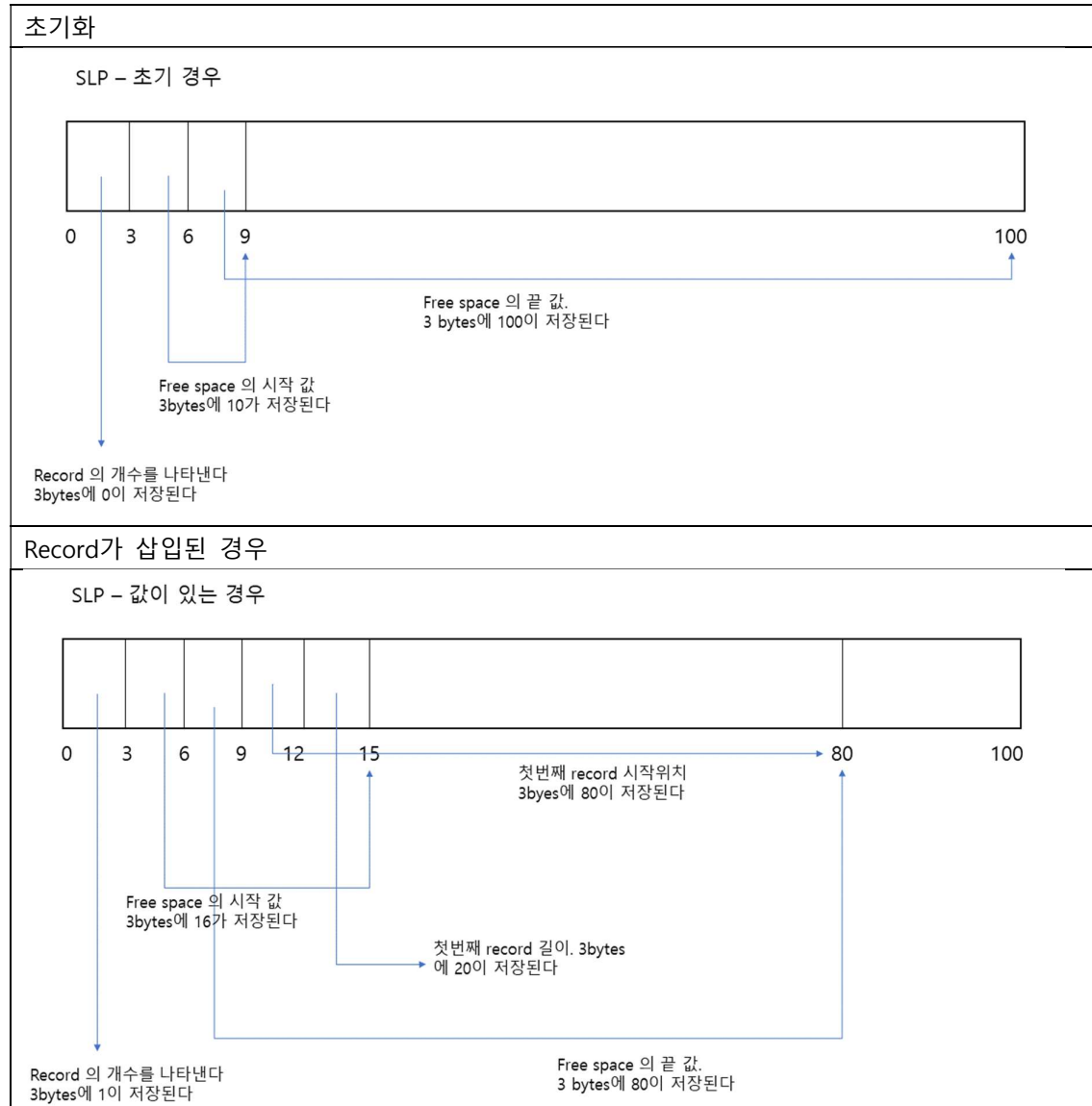
    print(f'null bit map : {self.nullbitmap}')
    print(f'is not null : {self.isNotNull}')

    print(f'colType : {self.colType}')
    print(f'colName : {self.colName}')
    print(f'value : {self.value}')
```

해당 VLR의 정보를
출력하는 함수이다

2) slotted page (SLP)

구현하고자 하는 SLP의 구조는 다음과 같다



```

SLP_LENGTH = 100
EACH_HEADER_SIZE = 3

class SLP:
    def __init__(self):
        bytemap = bytearray(SLP_LENGTH)
        # print(f'bytemap : {bytemap[:10]}, length : {len(bytemap)}')
        bytemap[0:EACH_HEADER_SIZE] = (0).to_bytes(EACH_HEADER_SIZE, 'big') # record 개수는 0으로 초기화
        bytemap[EACH_HEADER_SIZE:
                EACH_HEADER_SIZE*2] = (10).to_bytes(EACH_HEADER_SIZE, 'big') # free space의 start

```

최대 길이는 100bytes 로 정하였다.

각 헤더의 크기는 3으로 정하였다. (위 구조 사진에서 확인 가능)

```

def getSLP(self, tableName : str, slotNum : int):
    directory = os.getcwd() + '/table/' + tableName
    # tempSlp = SLP()#bytearray(EACH_HEADER_SIZE)
    with open(directory + '/slot'+str(slotNum)+'.bin', 'rb') as f:
        self.slp = bytearray(f.read())
        #print(self.slp)
    self.recordNum = int.from_bytes(self.slp[0:EACH_HEADER_SIZE], 'big')
    self.freespaceStart = int.from_bytes(self.slp[EACH_HEADER_SIZE:2*EACH_HEADER_SIZE], 'big')
    self.freespaceEnd = int.from_bytes(self.slp[2*EACH_HEADER_SIZE * self.recordNum :
        2*EACH_HEADER_SIZE * self.recordNum + EACH_HEADER_SIZE], 'big')
    self.freespaceRemain = self.freespaceEnd - self.freespaceStart

```

SLP의 정보를 가져오는 메소드이다. Table_name 과 slotNum 는 테이블의 몇번째 slot 의 정보를 불러올지에 대한 인자이다

```

def printSLP(self):
    print(f'record num : {self.recordNum}')
    print(f'freespace start : {self.freespaceStart}')
    print(f'freespace end : {self.freespaceEnd}')
    print(f'freespace remain : {self.freespaceRemain}')

```

SLP의 정보를 출력하는 메소드이다

3) Data Dictionary

<pre>class dataDict: def __init__(self): self.tableName = '' self.colName = [] self.colType = [] self.slotNum = 0 self.recordNum = 0</pre>		<p>초기화 부분이다.</p> <p>저장될 table 이름, 컬럼명, 각 컬럼별 타입 (varchar인지 char인지) 테이블의 slot 개수, 테이블의 record개수가 저장된다.</p>
<pre>def makeDict(self, table_name:str, col_name: list, col_type:list): directory = os.getcwd() + '/table/' + table_name try: if not os.path.exists(directory): os.makedirs(directory) else: pass except OSError: print(r'Error: Creating directory. ' + directory) exit() # meta data write with open(directory + '/meta_data.txt', 'w+') as f: f.write(table_name+'\n') for i,n in enumerate(col_name): f.write(n) if i != len(col_name)-1: f.write('/') f.write('\n') for q,t in enumerate(col_type): f.write(t) if q != len(col_type)-1: f.write('/') f.write('\n0') # total slot 개수 f.write('\n0') # total record 개수</pre>		<p>Table name, col name, col type을 입력값으로 받아 write 한다.</p> <p>이 메서드는 createTable에서 실행된다. 하나의 테이블은 하나의 폴더를 생성하고 하나의 폴더 안에 하나의 메타데이터가 하나의 txt로 저장된다</p>
<pre>def printDict(self): print(f'table name : {self.tableName}') print(f'columns : {self.colName}') print(f'columns type : {self.colType}') print(f'# of slots : {self.slotNum}') print(f'# of records : {self.recordNum}')</pre>		<p>메타데이터의 정보를 출력하는 함수이다.</p>
<pre>def getDict(self, table_name): directory = os.getcwd() + '/table/' + table_name with open(directory + '/meta_data.txt', 'r') as f: self.tableName = f.readline().rstrip() self.colName = f.readline().rstrip().split(sep="/") self.colType = f.readline().rstrip().split(sep="/") self.slotNum = int(f.readline().rstrip()) self.recordNum = int(f.readline().rstrip())</pre>		<p>dataDict()의 값을 불러오는 메서드이다.</p> <p>해당 table의 metadata의 값을 가져온다.</p>

```

def updateDict(self, tableName : str, updateSlot : bool, updateRecord:bool):
    metadata = dataDict()
    metadata.getDict(tableName)
    directory = os.getcwd() + '/table/' + tableName
    with open(directory + '/meta_data.txt', 'w+') as f:
        f.write(tableName+'\n')
        for i,n in enumerate(metadata.colName):
            f.write(n)
            if i != len(metadata.colName)-1: f.write('/')
        f.write('\n')
        for q,t in enumerate(metadata.colType):
            f.write(t)
            if q != len(metadata.colType)-1: f.write('/')
        if updateSlot : f.write('\n'+str(metadata.slotNum+1))
        else: f.write('\n'+str(metadata.slotNum))
        if updateRecord : f.write('\n'+str(metadata.recordNum+1))
        else: f.write('\n'+str(metadata.recordNum))

```

SLP에 변화가 생겼을 때
실행되는 메서드이다.

updateSlot의 값이 True
면 slotNum을 +1 하여
write 한다.

updateRecord가 True면
recordNum을 +1 하여
write한다

3. API.py 기능 파일

1) createTable – 테이블을 생성

```
def createTable(table_name:str, col_name : list, col_type:list):  
    metadata = dataDict()  
    metadata.makeDict(table_name,col_name, col_type)
```

Table name, col_name, col_type을 인자로 받아서 table을 생성한다. 인자의 정보들을 받아 data dictionary를 생성한다. 테이블을 처음 생성할 때는 SLP를 만들지 않게 했다.

2) insertColumn – record 삽입

```
def insertRecord(table_name:str, col_name : list):  
    metadata = dataDict()  
    metadata.getDict(table_name)  
    slotnum = metadata.slotNum  
    print(f'now slot num : {slotnum}')  
    directory = os.getcwd() + '/table/' + table_name  
  
    record = VLR()  
    record.makeVLR(table_name, col_name)  
  
    updateSlot = False  
    updateRecord = False  
    newSlotNum = 0  
  
    if slotnum == 0 : # 해당 table에 slot이 아예 존재하지 않을 때 -> 새로 1개를 만들어야 한다  
        newSlotNum = 1  
        # slp 초기화  
        newslp = SLP()  
        # 개수 update  
        newslp.slp[0:EACH_HEADER_SIZE] = (1).to_bytes(EACH_HEADER_SIZE, 'big') # 해당 slp의 record 개수 1개가 됨  
        # start of freespace UPDATE  
        newslp.slp[EACH_HEADER_SIZE: EACH_HEADER_SIZE*2] = (EACH_HEADER_SIZE * 4).to_bytes(EACH_HEADER_SIZE, 'big') # free space 시작  
        # 새로운 record 삽입  
        newslp.slp[-len(record.vlr):] = record.vlr # newslp.slp[newslp.freespaceEnd-len(record.vlr):newslp.freespaceEnd] = record.vlr  
        # 삽입한 record의 offset - 시작위치 알려주는  
        newslp.freespaceEnd -= len(record.vlr)  
        newslp.slp[EACH_HEADER_SIZE*2 : EACH_HEADER_SIZE*3] = (newslp.freespaceEnd).to_bytes(EACH_HEADER_SIZE, 'big')  
        # 삽입한 record의 offset - length 알려줌  
        newslp.slp[EACH_HEADER_SIZE*3 : EACH_HEADER_SIZE*4] = (len(record.vlr)).to_bytes(EACH_HEADER_SIZE, 'big')  
  
        updateSlot = True  
        updateRecord = True
```

Table의 metadata를 가져와 해당 폴더에 SLP의 개수를 가져온다. 그리고 record를 vlr 형태로 만든다.

SLP의 개수가 0이면 새로 SLP를 만든 후 값을 넣는다. 새로 만드니 record개수는 1개가 되고 위의 class 의 SLP 의 구조대로 차례차례 정보를 삽입한다.

이 경우 SLP의 개수도 1개가 되고 record의 개수도 1개 늘었으므로 updateSlot과 updateRecord가 모두 True 값이 된다.

```

else: # 하나라도 있을 때
    newslp = SLP()
    newslp.getSLP(table_name, slotnum) # 가장 마지막에 있는 slot 가져온다
    if newslp.freespaceRemain < len(record.vlr): # 길이 부족하면
        newSlotNum = slotnum + 1
        ...
        *****slot 새로 만들어야함 그리고 삽입*****
        ...
        # slp 초기화 새로 만들어야 하므로!
        newslp = SLP()
        # 개수 update
        newslp.slp[0:EACH_HEADER_SIZE] = (1).to_bytes(EACH_HEADER_SIZE, 'big') # 새로 만든 slp의 record 개수 1개가 됨
        # start of freespace UPDATE
        newslp.slp[EACH_HEADER_SIZE*2: EACH_HEADER_SIZE*2] = (EACH_HEADER_SIZE * 4).to_bytes(EACH_HEADER_SIZE, 'big') # free space 시작
        # 새로운 record 삽입
        newslp.slp[-len(record.vlr):] = record.vlr # newslp.slp[newslp.freespaceEnd-len(record.vlr):newslp.freespaceEnd] = record.vlr
        # 삽입한 record의 offset - 시작위치 알려주는
        newslp.freespaceEnd -= len(record.vlr)
        newslp.slp[EACH_HEADER_SIZE*2: EACH_HEADER_SIZE*3] = (newslp.freespaceEnd).to_bytes(EACH_HEADER_SIZE, 'big')
        # 삽입한 record의 offset - length 알려줌
        newslp.slp[EACH_HEADER_SIZE*3: EACH_HEADER_SIZE*4] = (len(record.vlr)).to_bytes(EACH_HEADER_SIZE, 'big')
        updateSlot = True
        updateRecord = True
        pass
    else: # record가 들어갈 자리가 충분히 있다면 load한 slp에 record 넣으면 됨
        newSlotNum = slotnum
        # 개수 update done
        newslp.slp[0:EACH_HEADER_SIZE] = (newslp.recordNum+1).to_bytes(EACH_HEADER_SIZE, 'big') # record 개수 + 1 해서 저장
        # start of freespace UPDATE done
        newslp.slp[EACH_HEADER_SIZE*2: EACH_HEADER_SIZE*2] = (2 * EACH_HEADER_SIZE * (newslp.recordNum + 2)).to_bytes(EACH_HEADER_SIZE, 'big') # free space 시작
        # 새로운 record 삽입 done
        newslp.slp[newslp.freespaceEnd-len(record.vlr):newslp.freespaceEnd] = record.vlr
        # 삽입한 record의 offset - 시작위치 알려주는
        newslp.freespaceEnd -= len(record.vlr)
        newslp.slp[EACH_HEADER_SIZE*2 * (newslp.recordNum+1) :
                    EACH_HEADER_SIZE*2 * (newslp.recordNum+1) + EACH_HEADER_SIZE] = (newslp.freespaceEnd).to_bytes(EACH_HEADER_SIZE, 'big')
        # 삽입한 record의 offset - length 알려줌
        newslp.slp[EACH_HEADER_SIZE*2 * (newslp.recordNum+1) + EACH_HEADER_SIZE :
                    EACH_HEADER_SIZE*2 * (newslp.recordNum+1) + EACH_HEADER_SIZE * 2] = (len(record.vlr)).to_bytes(EACH_HEADER_SIZE, 'big')
        updateSlot = False
        updateRecord = True

```

그 테이블(폴더)에 SLP가 하나라도 있는 경우이다. Slotnum을 받아와 가장 마지막 slot을 연다. 그 slot에 새로운 record를 넣을 공간이 있는지 확인한다. 만약 모자라면 새로운 SLP를 하나 생성하여 위와 같은 경우(SLP가 0인 경우)를 반복한다. 이 경우 또한 updateSlot과 updateRecord 값이 모두 True이다.

Record가 들어갈 길이가 된다면, 마지막 slot에 삽입한다. 이 경우 slot의 개수는 늘어나지 않으므로 updateSlot은 False, updateRecord의 값만 True가 된다.

```

# print(f'newly updated slp : {newslp.slp}')
with open(directory + '/slot' + str(newSlotNum) + '.bin', 'wb+') as f:
    f.write(newslp.slp)

metadata.updateDict(table_name, updateSlot, updateRecord) # meta data update

```

새로운 SLP를 테이블(폴더)에 write 한다. updateSlot과 updateRecord 값을 가지고 dataDict 메서드의 updateDict을 실행시켜 dataDict 또한 update 된다.

3) findRecord – 조건에 맞는 record 검색

```
def findRecord(select : str, table_name : str, where : str, target : str):
    print(where, target)
    metadata = dataDict()
    metadata.getDict(table_name)
    result_vlr = [] # return 할 최종 값

    # metadata col_name에서 몇번째의 col 값인지 확인해야함
    try:
        where_index = metadata.colName.index(where) # dept 의 index
    except:
        return result_vlr

    if select == '*': select_index = -1 # 전부일때
    else:
        try: select_index = metadata.colName.index(select) # 'name'의 index
        except: return result_vlr

    if metadata.slotNum == 0 : return result_vlr # slot 없으면 바로 return
```

Select 은 보고자 하는 column 값,
table_name은 보고자 하는 table, where =target일 경우를 뜻하는 것이다.

최종 결과값으로 result_vlr 의 array를 return 한다

```
for slot_num in range(1,metadata.slotNum+1): # 있는 slp 전부 확인
    print(f'{slot_num}th SLP')
    tempSlp = SLP()
    tempSlp.getSLP(table_name, slot_num) # slot 불러오기
    for i in range(1, tempSlp.recordNum + 1): # slot에 있는 record(vlr)를 하나씩 꺼내기
        ...
        1. record 꺼내기
        ...
        recordStartPoint = int.from_bytes(tempSlp.slp[ (2* EACH_HEADER_SIZE) * i :
                                                         (2* EACH_HEADER_SIZE) * i + EACH_HEADER_SIZE], 'big')
        recordLength = int.from_bytes(tempSlp.slp[ (2* EACH_HEADER_SIZE) * i + EACH_HEADER_SIZE :
                                                         (2* EACH_HEADER_SIZE) * i + EACH_HEADER_SIZE + EACH_HEADER_SIZE], 'big')
        record = tempSlp.slp[recordStartPoint:recordStartPoint+recordLength] # slot_num 번째의 i 번째 record
        ...
        2. 꺼낸 record bytearray를 VLR로 만들어 정보 가져오기
        ...
        # ===== class 사용 =====
        SLPreord = VLR()
        SLPreord.makeVLR_bytearray(record, table_name) # 1에 해당하는 record를 만드는 것
        #SLPreord.printVLR()

        # print(SLPreord.value[where_index])
        # print(SLPreord.isNotNull)
        ...
        3. target 값 비교
        ...
        if SLPreord.isNotNull[where_index] != False and SLPreord.value[where_index] == target:
            if select_index == -1: result_vlr.append(SLPreord.value)
            else: result_vlr.append(SLPreord.value[select_index])

    print(f'record 개수 : {len(result_vlr)} , {result_vlr}')
    return result_vlr
```

(outer for loop) 해당 table의 slp를 모두 확인한다.

(inner for loop)각 slp에서 record의 정보를 가져온다.

Where=target 조건을 check 한다
만약 select 의 값이 * 면 record 전체를 append 하고 아니면 select 에 맞는 값을 append 한다

4) findColumn – 해당 table의 column들 검색

```
def findColumn(table_name: str):  
    meta_data = dataDict()  
    meta_data.getDict(table_name)  
    return meta_data.colName
```

Tablename을 인자로 받으면 그 table에 있는 컬럼명을 다 반환한다

4. 결과

```
table_name = 'student'
col_names = ['ID', 'name', 'grade', 'dept']
col_types = ['c8', 'v16', 'c1', 'v17']
```

table_name: table 이름

col_names: table의 column명

col_types: table의 column들의 각 type

(c는 고정길이, v는 가변길이이다. c8인 경우, 길이가 8인 고정길이의 값을 가진다. v16은 최대 16 길이를 가진다)

실행 방법

main.py 파일에서 table_names, col_names, col_types 를 설정한다.

그리고 아래의 코드를 실행한다. (insertRecord 는 insert_record의 list 필요)

```
createTable(table_name,col_names,col_types)
```

```
insert_record = ['20194653', 'dain', '1', 'sci']
insertRecord(table_name, insert_record)
```

```
res = findRecord(select = '*',
                  table_name = table_name,
                  where = 'dept',
                  target = 'sci')
if (len(res) == 0) : print('query not found')
```

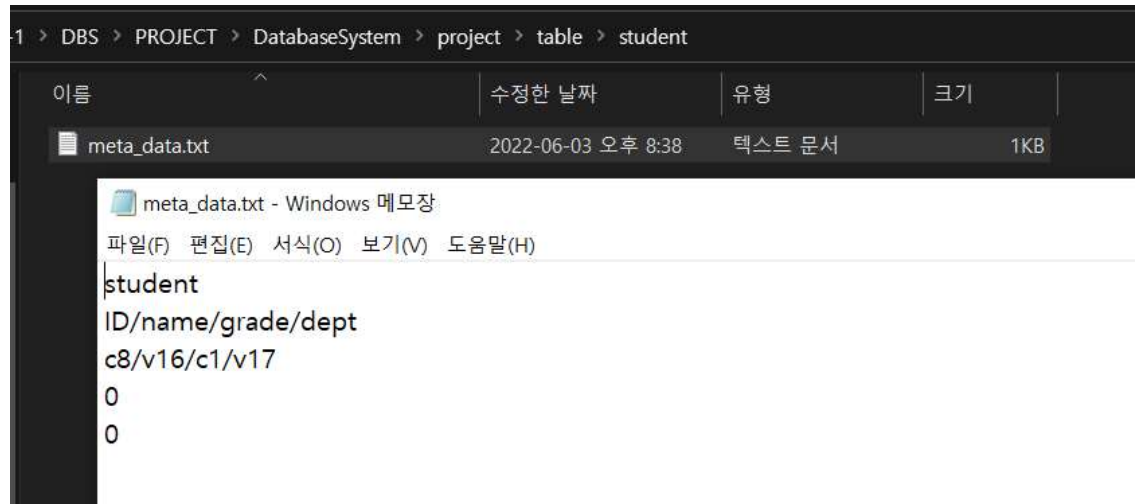
```
print(findColumn(table_name))
```

1) table creation

하나의 테이블 당 하나의 폴더와 meta_data.txt 생성

```
if __name__ == '__main__':  
  
    table_name = 'student'  
    col_names = ['ID', 'name', 'grade', 'dept']  
    col_types = ['c8', 'v16', 'c1', 'v17']  
  
    # metadata의 데이터 확인  
    metadata = dataDict()  
    # SLP  
    slp = SLP()  
  
    # 1 create table  
    print("=====create table=====")  
    createTable(table_name,col_names,col_types)  
    print(f'create table "{table_name}"')
```

```
C:\Windows\System32\cmd.exe  
  
C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py  
=====create table=====  
create table "student"  
# of SLP:0 , # of Record:0  
table name : student  
columns : ['ID', 'name', 'grade', 'dept']  
columns type : ['c8', 'v16', 'c1', 'v17']  
# of slots : 0  
# of records : 0
```

폴더와 meta_data.txt 파일이 생성되었다

2) record insertion

```
# 2-1 record insert - slot page 의 사이즈에 딱 맞는 경우
print("=====insert record 1=====")
insert_record = ['20194653', 'dain', '1', 'sci']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name, metadata.slotNum)
slp.printSLP()

print("=====insert record 2=====")
insert_record = ['12345678', 'null', '1', 'sci']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name, metadata.slotNum)
slp.printSLP()

print("=====insert record 3=====")
insert_record = ['00000000', 'yoon', '3', 'null']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name, metadata.slotNum)
slp.printSLP()
```

```
# 하나를 더 추가하면 slot의 개수가 늘어난다.
print("=====insert record 4=====")
insert_record = ['3333333', 'dd', '1', 'biology']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name, metadata.slotNum)
slp.printSLP()

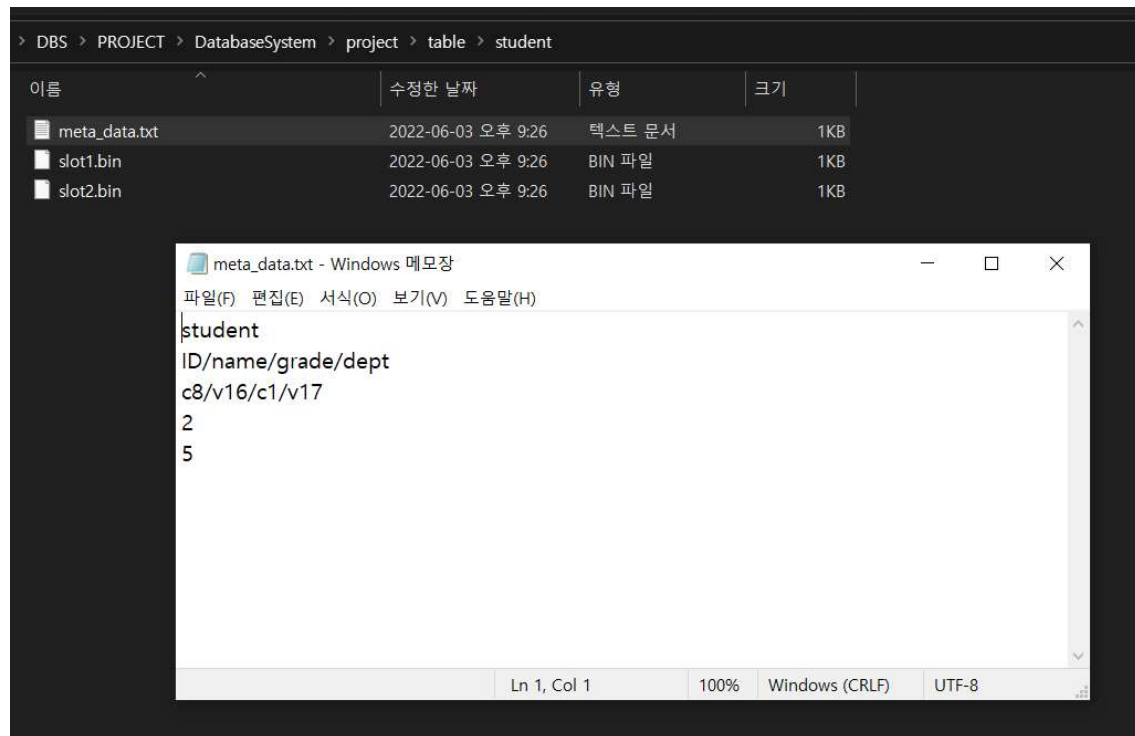
print("=====insert record 5=====")
insert_record = ['13131313', 'null', '1', 'sci']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name, metadata.slotNum)
slp.printSLP()
```

```
C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
=====insert record 1=====
# of SLP:1 , # of Record:1
record num : 1
freespace start : 12
freespace end : 75
freespace remain : 63
=====insert record 2=====
# of SLP:1 , # of Record:2
record num : 2
freespace start : 18
freespace end : 58
freespace remain : 40
=====insert record 3=====
# of SLP:1 , # of Record:3
record num : 3
freespace start : 24
freespace end : 40
freespace remain : 16
=====insert record 4=====
# of SLP:2 , # of Record:4
record num : 1
freespace start : 12
freespace end : 73
freespace remain : 61
=====insert record 5=====
# of SLP:2 , # of Record:5
record num : 2
freespace start : 18
freespace end : 56
freespace remain : 38

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>
```

Slot 값이 늘어난 것을 확인할 수 있다



Bin 파일(slot)이 늘어난 것과 meta_data.txt 의 값이 변경되었다.

3) search record

특정 col을 출력

```
res = findRecord(select = 'ID',
                  table_name = table_name,
                  where = 'grade',
                  target = '1')
if (len(res) == 0) : print('query not found')
```

```
C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
record 개수 : 3 , ['20194653', '12345678', '13131313']

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>
```

전체를 출력

```
res = findRecord(select = '*',
                  table_name = table_name,
                  where = 'grade',
                  target = '1')
if (len(res) == 0) : print('query not found')
```

```
C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
record 개수 : 3 , [['20194653', 'dain', '1', 'sci'], ['12345678', 'null', '1', 'sci'], ['13131313', 'null', '1', 'sci']]

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>
```

```
res = findRecord(select = '*',
                  table_name = table_name,
                  where = 'grade',
                  target = '2')
if (len(res) == 0) : print('query not found')
```

```
C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
record 개수 : 0 , []
query not found
```

4) table columns

```
print(f'columns for table"{table_name}" {findColumn(table_name)}')
```

```
C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py  
columns for table"student" ['ID', 'name', 'grade', 'dept']
```

5) 한 번에 실행

```
File Edit Selection View Go Run Terminal Help
mainpy M APIpy M structurepy M

1 from API import *
2 from structure import *
3
4
5 if __name__ == '__main__':
6
7     table_name = 'student'
8     col_names = ['ID','name','grade','dept']
9     col_types = ['c8','v16','c1','v17']
10
11 # metadata
12 metadata = datablct()
13 # SLP
14 slp = SLP()
15
16 # 1 create table
17 print("=====create table=====")
18 createTable(table_name,col_names,col_types)
19 print("create table "+table_name)
20 metadata.getTable(table_name)
21 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
22 metadata.getTable()
23
24 # 2 record insert
25 # slot page 외 사이즈에 딱 맞는 경우
26 print("=====insert record 1=====")
27 insert_record = ['20194653', 'dain', '1', 'sci']
28 insertRecord(table_name, insert_record)
29 metadata.getTable(table_name)
30 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
31 slp.getSLP(table_name,metadata.slotNum)
32 slp.printSLP()
33
34 print("=====insert record 2=====")
35 insert_record = ['12345678', 'null', '1', 'sci']
36 insertRecord(table_name, insert_record)
37 metadata.getTable(table_name)
38 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
39 slp.getSLP(table_name,metadata.slotNum)
40 slp.printSLP()
41
42 print("=====insert record 3=====")
43 insert_record = ['00000000', 'yoon', '3', 'null']
44 insertRecord(table_name, insert_record)
45 metadata.getTable(table_name)
46 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
47 slp.getSLP(table_name,metadata.slotNum)
48 slp.printSLP()
49
50 # 하나를 더 추가하면 slot의 계수가 늘어났다.
51 print("=====insert record 4=====")
52 insert_record = ['3333333', 'dd', '1', 'biology']
53 insertRecord(table_name, insert_record)
54 metadata.getTable(table_name)
55 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
56 slp.getSLP(table_name,metadata.slotNum)
57 slp.printSLP()
58
59 print("=====insert record 5=====")
60 insert_record = ['13131313', 'null', '1', 'sci']
61 insertRecord(table_name, insert_record)
62 metadata.getTable(table_name)
63 print("# of SLP:(metadata.slotNum) , # of Record:(metadata.recordNum)")
64 slp.getSLP(table_name,metadata.slotNum)
65 slp.printSLP()
66
67 # 3. find record
68 # ===== 특정 column =====
69 res = findRecord(select = 'ID',
70                 table_name = table_name,
71                 where = 'grade',
72                 target = '1')
73 if (len(res) == 0) : print('query not found')
74
75 # ===== wiscp =====
76 res = findRecord(select = '*',
77                 table_name = table_name,
78                 where = 'grade',
79                 target = '1')
80 if (len(res) == 0) : print('query not found')
81
82
83 # 4. find columns
84 print("columns for table["+table_name+"] (findColumn(table_name))")
```


C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py

=====create table=====

create table "student"

of SLP:0 , # of Record:0

table name : student

columns : ['ID', 'name', 'grade', 'dept']

columns type : ['c8', 'v16', 'c1', 'v17']

of slots : 0

of records : 0

=====insert record 1=====

of SLP:1 , # of Record:1

record num : 1

freespace start : 12

freespace end : 75

freespace remain : 63

=====insert record 2=====

of SLP:1 , # of Record:2

record num : 2

freespace start : 18

freespace end : 58

freespace remain : 40

=====insert record 3=====

of SLP:1 , # of Record:3

record num : 3

freespace start : 24

freespace end : 40

freespace remain : 16

=====insert record 4=====

of SLP:2 , # of Record:4

record num : 1

freespace start : 12

freespace end : 73

freespace remain : 61

=====insert record 5=====

of SLP:2 , # of Record:5

record num : 2

freespace start : 18

freespace end : 56

freespace remain : 38

record 개수 : 3 , ['20194653', '12345678', '13131313']

record 개수 : 3 , [['20194653', 'dain', '1', 'sci'], ['12345678', 'null', '1', 'sci'], ['13131313', 'null', '1', 'sci']]

columns for table"student" ['ID', 'name', 'grade', 'dept']

6) 다른 예시 : people 테이블

```
table_name = 'people'
col_names = ['ID', 'name', 'grade', 'dept', 'age']
col_types = ['c8', 'v16', 'c1', 'v15', 'c2']

# metadata의 데이터 확인
metadata = dataDict()
# SLP
slp = SLP()

# 1 create table
# print("=====create table=====")
createTable(table_name, col_names, col_types)
print(f'create table "{table_name}"')

metadata.getDict(table_name)

print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
metadata.printDict()
```

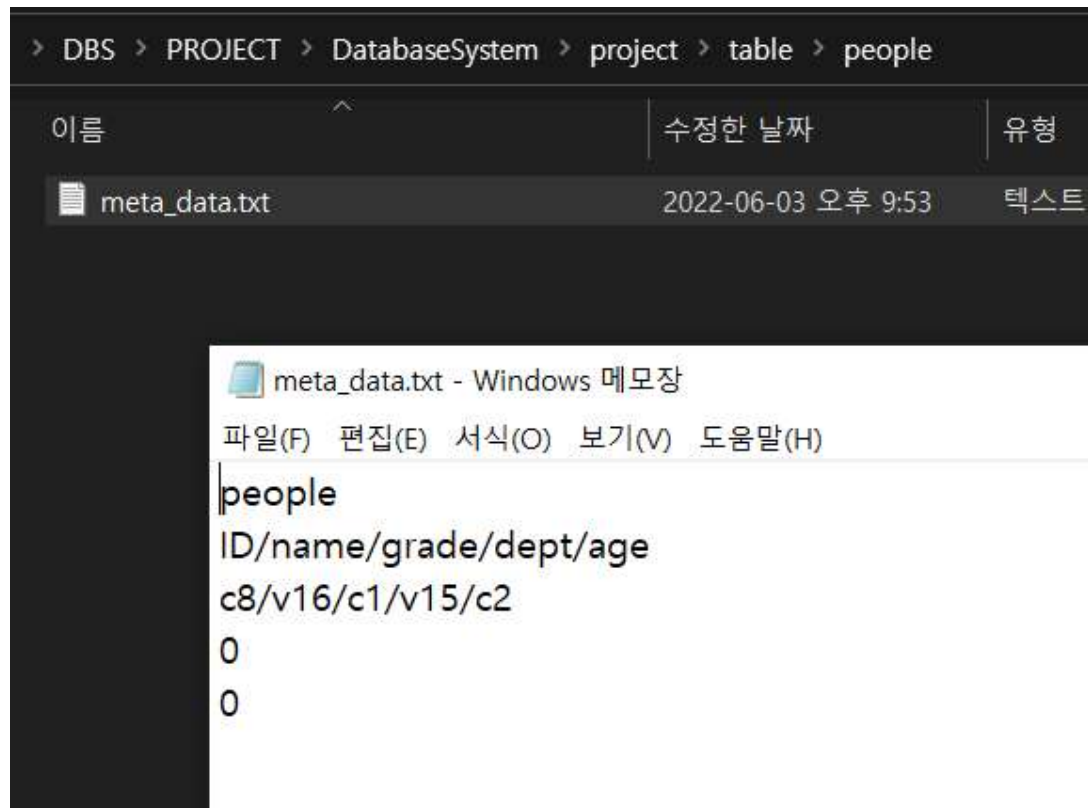
```
C:\Windows\System32\cmd.exe

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
create table "people"
# of SLP:0 , # of Record:0
table name : people
columns : ['ID', 'name', 'grade', 'dept', 'age']
columns type : ['c8', 'v16', 'c1', 'v15', 'c2']
# of slots : 0
# of records : 0

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>
```

1 > DBS > PROJECT > DatabaseSystem > project > table		
이름	수정한 날짜	유형
people	2022-06-03 오후 9:52	파일 폴더
student	2022-06-03 오후 9:38	파일 폴더

people 생성



meta_data.txt 생성

```

# 2-1 record insert - slot page 의 사이즈에 딱 맞는 경우
print("=====insert record 1=====")
insert_record = ['20194653', 'dain', '1', 'sci','20']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name,metadata.slotNum)
slp.printSLP()

print("=====insert record 2=====")
insert_record = ['12345678', 'null', '1', 'sci','23']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name,metadata.slotNum)
slp.printSLP()

print("=====insert record 3=====")
insert_record = ['00000000', 'yoon', '3', 'null','20']
insertRecord(table_name, insert_record)
metadata.getDict(table_name)
print(f'# of SLP:{metadata.slotNum} , # of Record:{metadata.recordNum}')
slp.getSLP(table_name,metadata.slotNum)
slp.printSLP()

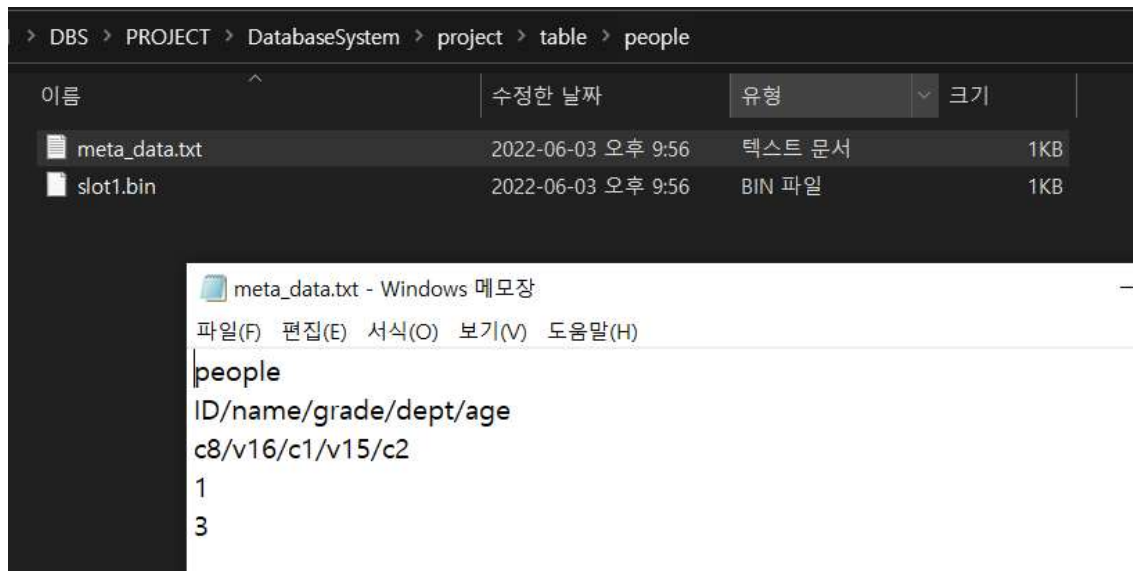
```

C:\Windows\System32\cmd.exe

```

C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
=====insert record 1=====
# of SLP:1 , # of Record:1
record num : 1
freespace start : 12
freespace end : 73
freespace remain : 61
=====insert record 2=====
# of SLP:1 , # of Record:2
record num : 2
freespace start : 18
freespace end : 54
freespace remain : 36
=====insert record 3=====
# of SLP:1 , # of Record:3
record num : 3
freespace start : 24
freespace end : 34
freespace remain : 10

```



Meta_data.txt 값이 업데이트. Slot 수와 record 수 update.

```
res = findRecord(select = 'ID',
                 table_name = table_name,
                 where = 'age',
                 target = '20')
if (len(res) == 0) : print('query not found')
```

```
C:\Users\yoondain\Desktop\00학교\4-1\DBS\PROJECT\DatabaseSystem\project>python main.py
record 개수 : 2 , ['20194653', '00000000']
```