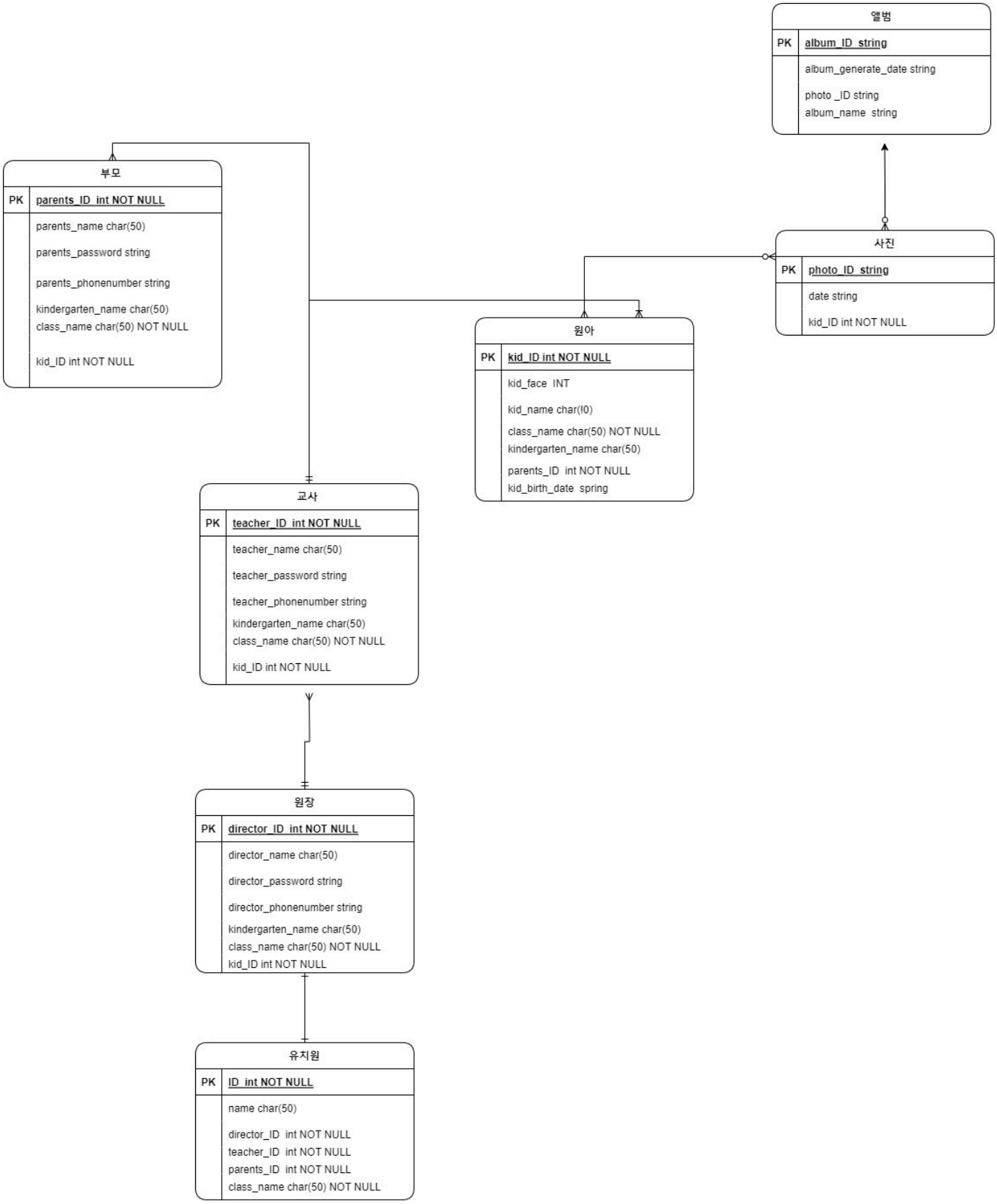
ERD


앨범
PK
album_ID string
album_generate_date string
photo _ID string
album_name string
부모
PK
parents_ID int NOT NULL
parents_name char(50)
parents_password string
parents_phonenumber string
kindergarten_name char(50)
class_name char(50) NOT NULL
kid_ID int NOT NULL
사진
PK
photo_ID string
date string
kid_ID int NOT NULL
원아
PK
kid_ID int NOT NULL
kid_face INT
kid_name char(!0)
class_name char(50) NOT NULL
kindergarten_name char(50)
parents_ID int NOT NULL
kid_birth_date spring
교사
PK
teacher_ID int NOT NULL
teacher_name char(50)
teacher_password string
teacher_phonenumber string
kindergarten_name char(50)
class_name char(50) NOT NULL
kid_ID int NOT NULL
원장
PK
director_ID int NOT NULL
director_name char(50)
director_password string
director_phonenumber string
kindergarten_name char(50)
class_name char(50) NOT NULL
kid_ID int NOT NULL
유치원
PK
ID int NOT NULL
name char(50)
director_ID int NOT NULL
teacher_ID int NOT NULL
parents_ID int NOT NULL
class_name char(50) NOT NULL

AI 속도

1장 검색

```
소요 시간: 0.1675281524658203 초
```

```
소요 시간: 0.2837493419647217 초
```

45장 검색

```
소요 시간: 9.403565883636475 초
```

AI 비용

그룹	API*	처음 100만 개의 이미지	다음 4백만 개의 이미지	다음 3,000만 개의 이미지	3,500만 개 초과 이미지
그룹 1	AssociateFaces CompareFaces DisassociateFaces IndexFaces SearchFacesbyImage SearchFaces SearchUsersByImage SearchUsers	0.0012 USD	0.00096 USD	0.00072 USD	0.00048 USD
그룹 2	DetectFaces DetectModerationLabels** DetectLabels*** DetectText RecognizeCelebrities DetectPPE	0.0012 USD	0.00096 USD	0.00072 USD	0.0003 USD
	이미지 속성****	0.0009 USD	0.00072 USD	0.00054 USD	0.000225 USD

```python
import boto3
import os
import time

# AWS 계정 정보 및 리전 설정
aws_access_key_id = 'access_key'
aws_secret_access_key = 'secret_access_key'
region_name = 'ap-northeast-2'
collection_id = 'l'

# Amazon Rekognition 클라이언트 생성
rekognition_client = boto3.client('rekognition', aws_access_key_id=aws_access_key_id,
                                  aws_secret_access_key=aws_secret_access_key, region_name=region_name)

# 얼굴 컬렉션 생성
rekognition_client.create_collection(CollectionId=collection_id)

# 얼굴 등록 함수
def register_faces(image_folder, person_id):
    faces = []
    for i in range(1, 6):
        image_path = os.path.join(image_folder, f"{person_id}{i}.jpg")
        with open(image_path, 'rb') as image_file:
            response = rekognition_client.index_faces(
                CollectionId=collection_id,
                Image={'Bytes': image_file.read()},
                ExternalImageId=person_id,
                DetectionAttributes=['ALL']
            )
            faces.extend(response['FaceRecords'])
    return faces

# 각각의 얼굴 등록
for person_id in ['a', 'b', 'c', 'd', 'e', 'f', 'g', 'h', 'i']:
    faces = register_faces('./dataset', person_id)
    print(f"Registered {len(faces)} faces for person {person_id}")


# 사진에서 얼굴 분류 함수
def classify_faces(image_path):
    with open(image_path, 'rb') as image_file:
        start_time = time.time()  # 시작 시간 기록
        response = rekognition_client.search_faces_by_image(
            CollectionId=collection_id,
            Image={'Bytes': image_file.read()},
            MaxFaces=1
        )
        end_time = time.time()  # 종료 시간 기록
        elapsed_time = end_time - start_time  # 경과 시간 계산
        if response['FaceMatches']:
            matched_face = response['FaceMatches'][0]
            external_image_id = matched_face['Face']['ExternalImageId']
            similarity = matched_face['Similarity']
            print(f"Face in {image_path} belongs to person {external_image_id} with similarity {similarity}%")
        else:
            print(f"No matching face found in {image_path}")

# 시간 측정 시작
start_time = time.time()

# 여러 사진에서 얼굴 분류
image_folder_to_classify = './unknown'
for image_file in os.listdir(image_folder_to_classify):
    if image_file.lower().endswith(('.png', '.jpg', '.jpeg')):
        image_path = os.path.join(image_folder_to_classify, image_file)
        classify_faces(image_path)

# 시간 측정 종료 및 출력
end_time = time.time()
elapsed_time = end_time - start_time
print(f"소요 시간: {elapsed_time} 초")
```