



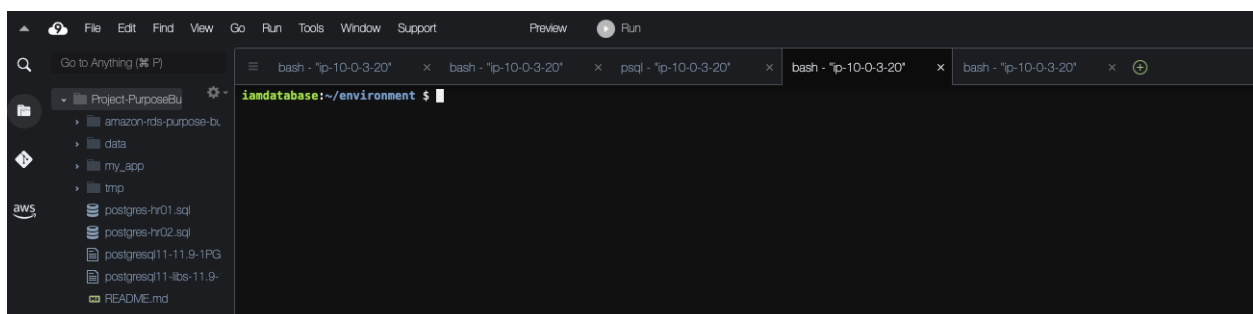
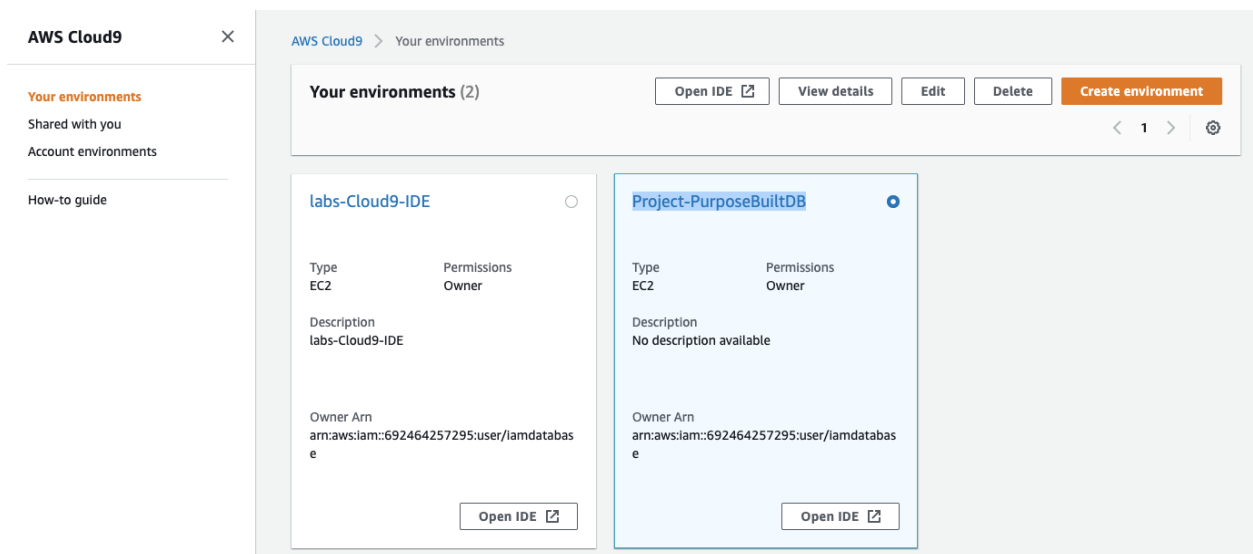
## 03) Dynamodb 에서 입력,수정,삭제,조회 수행

### [1] 선결 조건

- 앞의 단계 완료

### [2] python 이용 데이터 적재

1. AWS Management console → 돋보기에서 cloud9 입력 및 이동 → 새화면 AWS cloud9 에서 Project-PurposeBuiltDB 선택 → open IDE클릭



## 2. python 용 boto3 설치

```
# shell에서 수행
cd ~/environment/builder202204/script
sh installBoto3.sh
```

## 3. 데이터 적재

```
table 생성 script 만들기
-----
# shell에서 수행
# aws dynamodb delete-table --table-name employee_dept # 테이블 삭제시에만 수행

aws dynamodb create-table --table-name employee_dept \
--attribute-definitions AttributeName=employee_id,AttributeType=S AttributeName=email,AttributeType=S \
--key-schema AttributeName=employee_id,KeyType=HASH AttributeName=email,KeyType=RANGE \
--provisioned-throughput ReadCapacityUnits=10,WriteCapacityUnits=10 \
--tags Key=workshop-dynamodb,Value=employee_dept

aws dynamodb wait table-exists --table-name employee_dept

이후 employee_dept 테이블 생성 확인

python 프로그램, 데이터파일 복사 및 적재
-----
# shell에서 수행
cd ~/environment/builder202204/script
python load_employee_dept_v5.6.py
```

## 4. 적재된 데이터를 조회 → AWS Management console 돋보기에서 dynamodb 입력 → 새화면 dynamoDB에서 좌측 tables클릭

The screenshot shows the AWS Management console interface for DynamoDB. On the left, there's a sidebar with navigation options: Dashboard, Tables (selected), Update settings, Explore items, PartiQL editor, and Backups. The main area is titled 'DynamoDB > Tables' and displays a table named 'employee\_dept'. The table's status is 'Active', and it has a partition key of 'employee\_id (S)' and a sort key of 'email (S)'. The read capacity mode is 'Provisioned (10)'. There are buttons for 'Create table', 'Delete', and 'Actions' at the top right of the table list.

## 5. employee\_dept 클릭 → 우측 상단의 상단 Explore table items 클릭시 아래와 같이 적재된 데이터를 확인 → ( 데이터가 보이지 않는 경우 화면 우측 중간 run 클릭)

DynamoDB

Dashboard
Tables
Update settings
Explore items
PartiQL editor
Backups
Exports to S3
Reserved capacity

DAX
Clusters
Subnet groups
Parameter groups
Events

Tell us what you think
Return to the previous console experience
Density settings

DynamoDB > Items > employee\_dept

Tables (1)

Any table tag

Find tables by table name

employee\_dept

employee\_dept

Autopreview

Actions

Create Item

Update table settings

Scan/Query items

Expand to query or scan items.

Items returned (50)

employee...	email	departm...	departm...	fi
141	TRAJS	50	Shipping	T
195	VJONES	50	Shipping	V
120	MWEISS	50	Shipping	M
168	LOZER	80	Sales	L
103	AHUNOLD	60	IT	A
102	LDEHAAN	90	Executive	L
204	HBAER	70	Public Relat...	H

## [3] 조회

### 1. 간단한 건 별 조회

```
# shell에서 수행
# 대상 테이블 스캔
aws dynamodb scan --table-name employee_dept --return-consumed-capacity TOTAL

#사번 100번 인 직원정보인 직원정보를 조회 (## query 이용 전체 컬럼조회 )
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em
#사번 100번 인 직원정보인 직원정보를 조회 (## query 이용 특정 컬럼조회 )
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em
#사번 100번 인 직원정보인 직원정보를 조회 (## get-item 이용 전체 컬럼조회 / pk해당하는 partition_key, sort_key 모두 지정 필요 )
aws dynamodb get-item --table-name employee_dept --key '{ "employee_id" : { "S": "100" } , "email": { "S": "SKING" } }' --return-consumed

#이름의 first_name ='Steven' 이고 last_name = 'King' 인 직원정보를 조회(## scan 이용 전체 컬럼조회 )
aws dynamodb scan --table-name employee_dept --filter-expression 'first_name = :first_name and last_name = :last_name ' --expression-attr
#이름의 first_name ='Steven' 이고 last_name = 'King' 인 직원정보를 조회(## scan 이용 특정 컬럼조회 )
aws dynamodb scan --table-name employee_dept --filter-expression 'first_name = :first_name and last_name = :last_name ' --expression-attr
```

### 2. Group by 조회

```
# 각 부서별 salary 합계 와 직원수
바로 수행하는 명령은 없으며 별도 프로그램을 작성하거나 hiveQL 을 사용
```

## [4] 입력/수정/삭제

### 1. 입력

```
# shell에서 수행
## 사번 300 번 신규사원 없음 확인
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em

# 사번 300 번 신규사원 입력
aws dynamodb put-item --table-name employee_dept \
  --item '{
    "employee_id": { "S": "300" },
    "first_name": { "S": "joon" },
    "last_name": { "S": "park" },
    "email": { "S": "parkjoon" },
    "phone_number": { "S": "010.1234.5678" },
    "hire_date": { "S": "2003-06-17 00:00:00" },
    "job_id": { "S": "AC_ACCOUNT" },
    "salary": { "N": "1000.00" },
    "manager_id": { "S": "205" },
    "department_id": { "S": "110" }
  }' --return-consumed-capacity TOTAL

#사번 300 번 신규사원 insert 확인 (## query 이용 전체 컬럼조회 )
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em
```

### 2. 수정

```
# shell에서 수행
# 사번 300 번 사번의 변경전 직원 정보 조회
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em

# 사번 300 번 사번의 salary 를 1000 -> 9999.9 로 update ( pk해당하는 partition_key, sort_key 모두 지정 필요 )
aws dynamodb update-item --table-name employee_dept \
  --key '{ "employee_id" : { "S": "300" }, "email": { "S": "parkjoon" } }' \
  --update-expression "SET salary = :newsalary" \
  --expression-attribute-values '{ ":newsalary" : { "N": "9999.9" } }' \
  --return-consumed-capacity TOTAL

# 사번 300 번 사번의 변경후 직원 정보 조회
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em
```

### 3. 삭제

```
# shell에서 수행
# 사번 300 번 사번의 delete 전 직원 정보 조회
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em

# 사번 300 번 사번의 delete
aws dynamodb delete-item --table-name employee_dept --key '{ "employee_id" : { "S": "300" }, "email": { "S": "parkjoon" } }'
```

```
# 사번 300 번 사번의 delete 후 직원 정보 조회
aws dynamodb query --table-name employee_dept --key-condition-expression 'employee_id = :employee_id' --expression-attribute-values '{ ":em
```

## [5] 테이블, 어트리뷰트(=컬럼) 정보

1. AWS Management console 돋보기에서 dynamodb 입력 → 새화면 DynamoDB → 좌측 메뉴 에서 'Tables' 선택 → 우측 화면에서 'employee\_dept' 선택하여 partition key, sort key를 확인

The screenshot shows the AWS Management Console interface for the 'employee\_dept' table. The left sidebar contains the 'DynamoDB' menu with options like 'Dashboard', 'Tables', 'Update settings', 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', and 'Reserved capacity'. The main content area displays the 'employee\_dept' table details, including the 'General information' tab. The partition key is 'employee\_id (String)' and the sort key is 'email (String)'.

2. 우측 상단의 'Explore table items' 클릭 → 화면 하단에 여러 ITEM(=레코드)에 따라 다양한 어트리뷰트(=컬럼)을 확인

The screenshot shows the AWS Management Console interface for the 'employee\_dept' table items. The left sidebar contains the 'DynamoDB' menu with options like 'Dashboard', 'Tables', 'Update settings', 'Explore items', 'PartiQL editor', 'Backups', 'Exports to S3', and 'Reserved capacity'. The main content area displays the 'employee\_dept' table details, including the 'Scan/Query items' section. The 'Scan' button is highlighted, and the 'Items returned (106)' section shows a list of items with columns: employee\_id, email, department, first\_name, hire\_date, and job\_id.

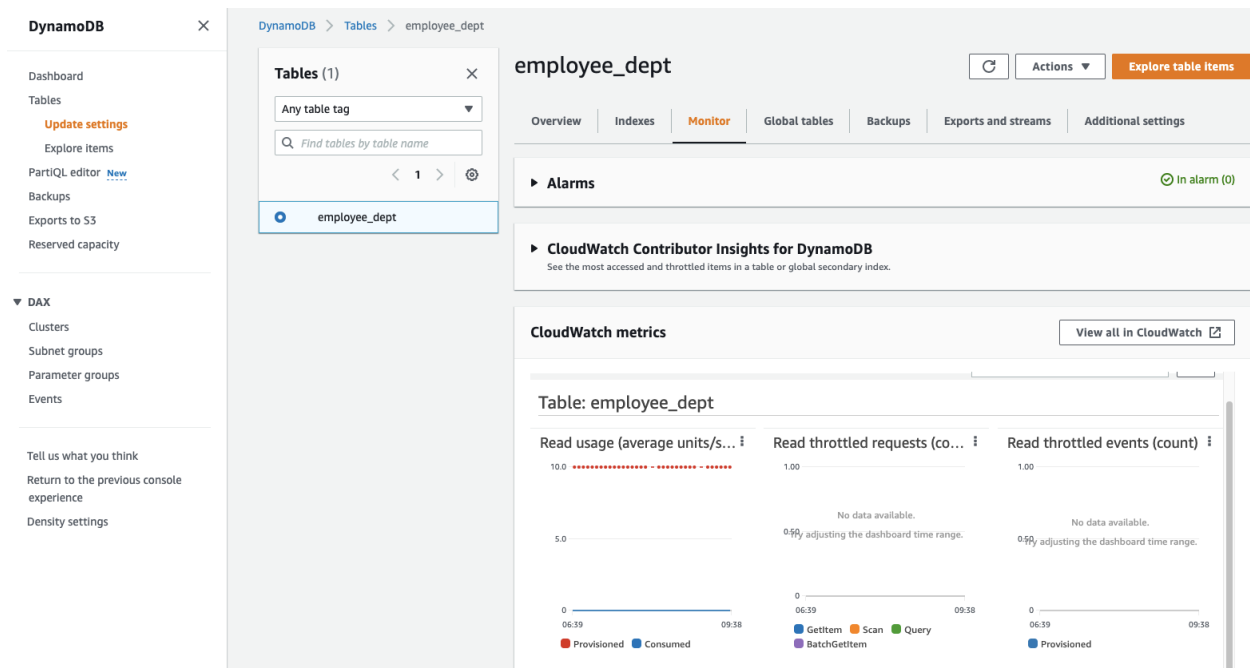
employee_id	email	department	first_name	hire_date	job_id
141	TRAJS	Shipping	Trenna	2003-10-1...	ST_CLERK
195	VJONES	Shipping	Vance	2007-03-1...	SH_CLERK
120	MWEISS	Shipping	Matthew	2004-07-1...	ST_MAN

3. 터미널에서 아래 명령을 수행하여 'employee\_dept' 테이블의 partition key, sort key를 확인

```
# 셸에서 수행
aws dynamodb describe-table --table-name employee_dept
```

## [6] 모니터링


1. DynamoDB 화면 → 좌측 메뉴에서 'Tables' 선택 → 우측 화면에서 'employee\_dept' 선택 → 상단의 'Monitor' 탭을 선택하여 다양한 지표로 모니터링



## 참고자료

- 집계성 쿼리
  - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/EMRforDynamoDB.Tutorial.QueryDataInDynamoDB>  
Query the Data in the DynamoDB Table
- modeling , table design
  - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/bp-relational-modeling.html> DynamoDB 모델링 제언  
Best Practices for Modeling Relational Data in DynamoDB

- <https://www.slideshare.net/awskorea/learn-how-to-model-and-operate-a-serverlessbased-database> dynamodb 의 간단한 설계 방안
- <https://www.youtube.com/watch?v=8rEsuvdL17s> 효과적인 NoSQL (Elasticache / DynamoDB) 디자인 및 활용 방안 - 최유희 솔루션즈 아키텍트(AWS), 최홍식 솔루션즈 아키텍트(AWS)
- insert, update, delete
  - [https://docs.aws.amazon.com/ko\\_kr/amazondynamodb/latest/developerguide/Expressions.UpdateExpressions.html](https://docs.aws.amazon.com/ko_kr/amazondynamodb/latest/developerguide/Expressions.UpdateExpressions.html) 업데이트 식
  - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/Expressions.ConditionExpressions.html> Conditional update, delete
- 테이블, 어트리뷰트 정보
  - <https://docs.aws.amazon.com/amazondynamodb/latest/developerguide/WorkingWithItems.html>
  - <https://amazon-dynamodb-labs.com/hands-on-labs/explore-cli/cli-read-data.html> 명령어

- 
- 수고하셨습니다. 다음 챕터로 이동하세요 —>  [04\) DocumentDB 의 환경구성](#)