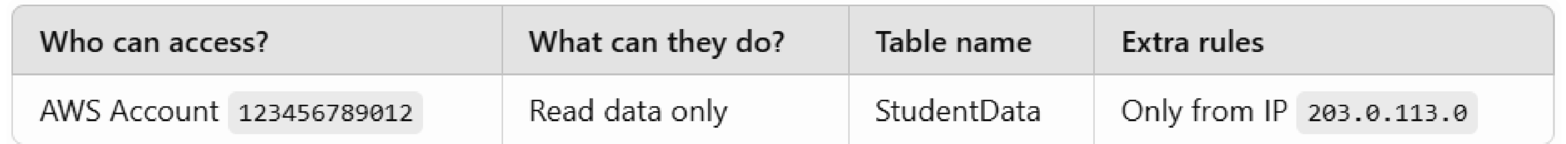
Resource-based policy

- 1 Introduction
  - 1 If you have a DynamoDB table and want to control who can access or who cannot access the table or index, you need to create a resource-based policy
  - 2 This policy is directly attached to the table and works like a rulebook that defines
    - 1 Who — The AWS accounts, users, or services allowed or denied access
    - 2 What — The actions they can perform, like GetItem, PutItem, or Query
    - 3 Conditions — Extra rules, like allowing access only from a specific IP address or within a certain time
  - 3 Example
    - 1 Imagine you have a table called "StudentData."
    - 2 You want to allow someone from another AWS account (e.g., Account ID: 123456789012) to read data from this table but not make any changes
    - 3 You create a resource-based policy and attach it to the table
    - 4 The policy might say — Hey DynamoDB, let account 123456789012 read data from the 'StudentData' table but only if they're accessing from a specific IP address
- 2 Key Features
  - 1 Resource-level Permissions — Unlike IAM policies, which apply to users, groups, or roles, resource-based policies are attached directly to DynamoDB tables or indexes
  - 2 Cross-account Access — You can grant permissions to AWS principals in other accounts without needing to grant full access to your own account
  - 3 Fine-grained Control — You can specify granular permissions, such as allowing certain actions (e.g., dynamodb:GetItem) or conditions (e.g., accessing the table from a specific IP)
- 3 Example of a Resource-based Policy

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Principal": {
        "AWS": "arn:aws:iam::123456789012:root"
      },
      "Action": "dynamodb:Query",
      "Resource": "arn:aws:dynamodb:us-east-1:111122223333:table/StudentData",
      "Condition": {
        "IpAddress": {
          "aws:SourceIp": "203.0.113.0/24"
        }
      }
    }
  ]
}
```

  - 1 Effect — Defines whether to allow or deny the specified action (Allow in this case)
  - 2 Principal — Specifies the AWS account or user granted the permission
  - 3 Action — Specifies the DynamoDB actions the principal can perform (e.g., dynamodb:Query)
  - 4 Resource — Specifies the ARN of the resource (a specific DynamoDB table in this case)
  - 5 Condition — Adds additional restrictions, like limiting access based on IP address
  - 6 Visual Example



Who can access?	What can they do?	Table name	Extra rules
AWS Account `123456789012`	Read data only	StudentData	Only from IP `203.0.113.0`