## Structure for storing tables partition key and sort key in single table

| Table Name | Partition Key | Sort Key |
|---|---|---|
| Organization | ORG#org_id | #METADATA# org_id |
| Project | ORG# org_id | PRO#project-type#project_id |
| Employee | ORG#org_id | EMP#employee_id |

## Defining Relationship

- Here One Organization can have multiple projects, which is already defined. So, one to many relationships is defined.
- Each employee is positioned under one organization that means one to many relationships is defined.
- We need to define many to many relationships between project and employee table. So, the partition key and sort key could be

| Table Name | Partition Key | Sort Key |
|---|---|---|
| Organization | ORG#org_id#PRO#project_id | ORG#org_id#EMP#employee_id |

To get the project an employee is part of we need to define an inverted index as GSI

Index Name: Project Employee Index

| Table Name | Partition Key | Sort Key |
|---|---|---|
| Organization | ORG#org_id#EMP#employee_id | ORG#org_id#PRO#project_id |

For filtering organization, project, employee name we need to define another GSI which is called as GSI Overloading

Index Name: Filter by name

| Table Name | Partition Key | Sort Key |
|---|---|---|
| Organization | ORG#org_id | Data #org_name<br>Or,<br>Data #project_name<br>Or,<br>Data#employee_name |