1. Display all movies from the year ‘2015’

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015

}

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"actors": [

"Ranveer Singh",

"Deepika Padukon",

"Priyanka Chopra"

],

"Year": 2015,

"release\_date": "2015-12-18T00:00:00Z",

"plot": "Love story of the great Maratha warrior Peshwa Bajirao",

"genres": [

"Drama"

],

"image\_url": "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@.\_V1\_SX400\_.jpg",

"directors": [

"Sanjay Leela Bhansali"

],

"rating": 8.2,

"Title": "Bajirao Mastani",

"rank": 20,

"running\_time\_secs": 5215

},

{

"actors": [

"Shahrukh Khan",

"Kajol",

"Varun Dhawan",

"Kriti Sanon"

],

"Year": 2015,

"release\_date": "2015-12-18T00:00:00Z",

"plot": "Love story",

"genres": [

"Romance"

],

"image\_url": "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@.\_V1\_SX400\_.jpg",

"directors": [

"Rohit Shetty"

],

"rating": 7.2,

"Title": "Dilwale",

"rank": 13,

"running\_time\_secs": 5215

},

{

"actors": [

"Shahrukh Khan",

"Kajol",

"Varun Dhawan",

"Kriti Sanon"

],

"Year": 2015,

"release\_date": "2015-12-18T00:00:00Z",

"plot": "Love story",

"genres": [

"Romance"

],

"image\_url": "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@.\_V1\_SX400\_.jpg",

"directors": [

"Rohit Shetty"

],

"rating": 7.2,

"Title": "Neerja",

"rank": 13,

"running\_time\_secs": 5215

},

{

"actors": [

"Ranbir Kapoor",

"Deepika Padukon"

],

"Year": 2015,

"release\_date": "2015-11-27T00:00:00Z",

"plot": "Love",

"genres": [

"Romance"

],

"image\_url": "http://ia.media-imdb.com/images/N/O9ERWAU7FS797AJ7LU8HN09AMUP908RLlo5JF90EWR7LJKQ7@@.\_V1\_SX400\_.jpg",

"directors": [

"Imtiaz Ali"

],

"rating": 7.2,

"Title": "Tamasha",

"rank": 15,

"running\_time\_secs": 5215

}

],

"Count": 4,

"ScannedCount": 4

}

1. Retrieving only names of the movies from the year 2016

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2016

},

ProjectionExpression: "Title"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"Title": "\tFitoor"

},

{

"Title": "Kapoor & Sons"

},

{

"Title": "Neerja"

}

],

"Count": 3,

"ScannedCount": 3

}

1. Movies from 2015 and their title beginning with ‘T’

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id and begins\_with(Title, :letter)",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015,

":letter":'T'

},

ProjectionExpression: "Title,directors,actors"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"actors": [

"Ranbir Kapoor",

"Deepika Padukon"

],

"Title": "Tamasha",

"directors": [

"Imtiaz Ali"

]

}

],

"Count": 1,

"ScannedCount": 1

}

1. Movies of 2015 in which ‘Deepika Padukon’ acted

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015,

":actor":"Deepika Padukon"

},

ProjectionExpression: "Title",

FilterExpression: " contains(actors, :actor)"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"Title": "Bajirao Mastani"

},

{

"Title": "Tamasha"

}

],

"Count": 2,

"ScannedCount": 4

}

1. Movies of 2016 rated above 7

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2016,

":rate" : 7

},

ProjectionExpression: "Title",

FilterExpression: "rating > :rate"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"Title": "Neerja"

}

],

"Count": 1,

"ScannedCount": 3

}

1. Movies whose genres is not romance

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015,

":genres" : "Romance"

},

ProjectionExpression: "Title,genres",

FilterExpression: "not contains(genres,:genres)"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

{

"Items": [

{

"Title": "Bajirao Mastani",

"genres": [

"Drama"

]

}

],

"Count": 1,

"ScannedCount": 4

}

1. Movies whose rating is between 6-10

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2016,

":rate1" : 6,

":rate2" : 10

},

ProjectionExpression: "Title,rating",

FilterExpression: "rating between :rate1 and :rate2"

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

{

"Items": [

{

"rating": 6.2,

"Title": "Kapoor & Sons"

},

{

"rating": 8.2,

"Title": "Neerja"

}

],

"Count": 2,

"ScannedCount": 3

}

1. Find movies of rating 7.2 but limit query to 1

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015,

":rate" : 7.2

},

ProjectionExpression: "Title,rating",

FilterExpression: "rating = :rate",

Limit:1

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

{

"Items": [],

"Count": 0,

"ScannedCount": 1,

"LastEvaluatedKey": {

"Year": 2015,

"Title": "Bajirao Mastani"

}

}

1. Display all movies of 2015 but limit query to 2

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2015,

},

ProjectionExpression: "Title",

Limit:2

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

{

"Items": [

{

"Title": "Bajirao Mastani"

},

{

"Title": "Dilwale"

}

],

"Count": 2,

"ScannedCount": 2,

"LastEvaluatedKey": {

"Year": 2015,

"Title": "Dilwale"

}

}

1. Movies of 2016 where number of actors=2

var params = {

TableName: "Movie",

KeyConditionExpression: "#y = :year\_id ",

ExpressionAttributeNames: { // a map of substitutions for attribute names with special characters

'#y': 'Year'

},

ExpressionAttributeValues: {

":year\_id": 2016,

":howmany" : 2

},

ProjectionExpression: "Title ,actors",

FilterExpression: "size(actors) = :howmany",

};

docClient.query(params, function(err, data) {

if (err)

console.log(JSON.stringify(err, null, 2));

else

console.log(JSON.stringify(data, null, 2));

});

Result:

{

"Items": [

{

"actors": [

"Aditya Roy Kapoor",

"Katrina Kaif"

],

"Title": "\tFitoor"

}

],

"Count": 1,

"ScannedCount": 3

}