Patch Support in the Cosmos DB Spark Connector

Support in Cosmos DB

https://docs.microsoft.com/en-us/azure/cosmos-db/partial-document-update

Add	`Add` performs one of the following, depending on the target path: • If the target path specifies an element that does not exist, it is added. • If the target path specifies an element that already exists, its value is replaced. • If the target path is a valid array index, a new element will be inserted into the array at the specified index. It shifts existing elements to the right. • If the index specified is equal to the length of the array, it will append an element to the array. Instead of specifying an index, you can also use the - character. It will also result in the element being appended to the array. **Note**: Specifying an index greater than the array length will result in an error.
Set	`Set` operation is similar to `Add` except in the case of Array data type - if the target path is a valid array index, the existing element at that index is updated.
Replace	`Replace` operation is similar to `Set` except it follows *strict* replace only semantics. In case the target path specifies an element or an array that does not exist, it results in an error.
Remove	`Remove` performs one of the following, depending on the target path: • If the target path specifies an element that does not exist, it results in an error. • If the target path specifies an element that already exists, it is removed. • If the target path is an array index, it will be deleted and any elements above the specified index are shifted one position to the left. **Note**: Specifying an index equal to or greater than the array length would result in an error.
Increment	• This operator increments a field by the specified value. It can accept both positive and negative values. If the field does not exist, it creates the field and sets it to the specified value.

Limitations

https://docs.microsoft.com/en-us/azure/cosmos-db/partial-document-update-faq

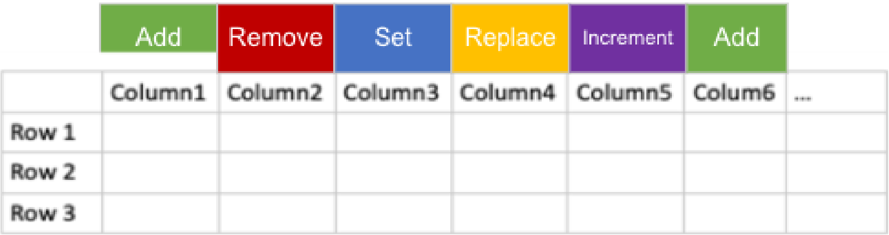
- Max 10 patch operations that can be executed for single document patch.

- Partial document update is not supported for system-generated properties like _id, _ts, _etag, _rid, _self
- Partial document update is not supported for id, {partitionKey}
- Increment operation type only support for types which can be converted to long or double.

Support in Cosmos DB Spark Connector

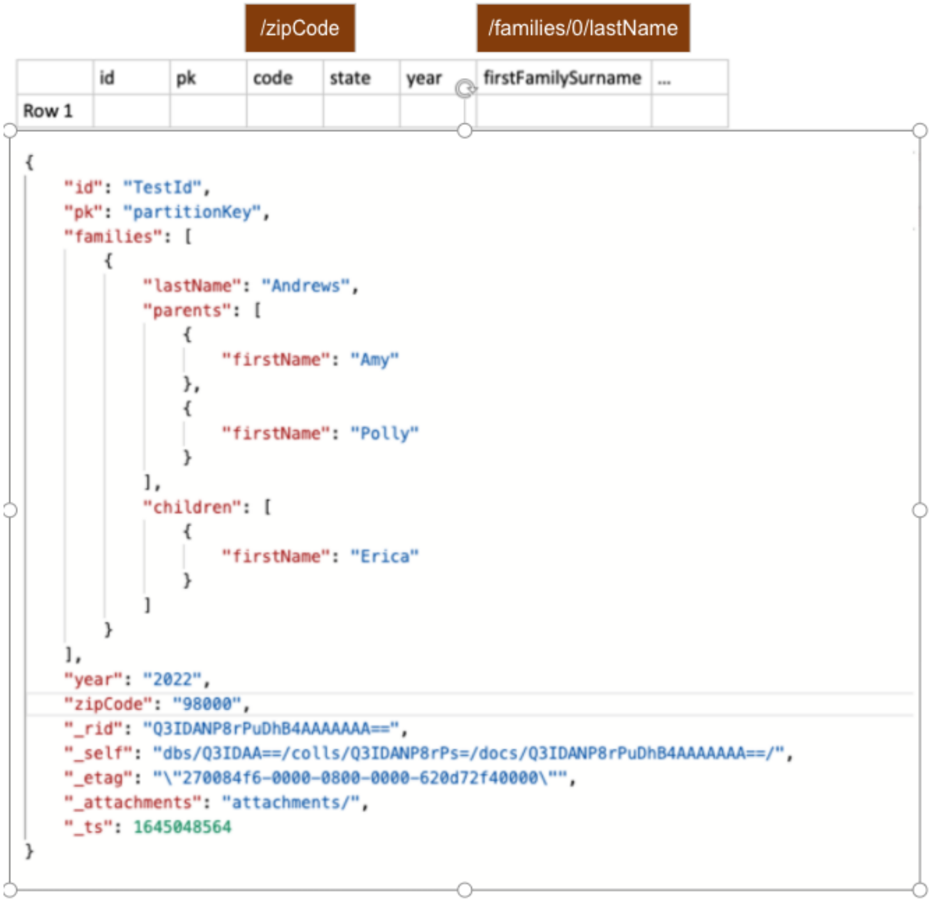
Requirements

- Cosmos DB spark connector will support all the patch operations supported in Cosmos DB, which is Add, Set, Replace, Remove, Increment
- Each column will be able to use different patch operations



	Add	Remove	Set	Replace	Increment	Add	
	Column1	Column2	Column3	Column4	Column5	Colum6	...
Row 1							
Row 2							
Row 3							

- Each column will be able to redefine its own mapping path in Cosmos DB item/document



	id	pk	code	state	year	firstFamilySurname	...
Row 1							

```
{
    "id": "TestId",
    "pk": "partitionKey",
    "families": [
        {
            "lastName": "Andrews",
            "parents": [
                {
                    "firstName": "Amy"
                },
                {
                    "firstName": "Polly"
                }
            ],
            "children": [
                {
                    "firstName": "Erica"
                }
            ]
        }
    ],
    "year": "2022",
    "zipCode": "98000",
    "_rid": "Q3IDANP8rPuDhB4AAAAAAA==",
    "_self": "dbs/Q3IDAA==/colls/Q3IDANP8rPs=/docs/Q3IDANP8rPuDhB4AAAAAAA==/",
    "_etag": "\"270084f6-0000-0800-0000-620d72f40000\"",
    "_attachments": "attachments/",
    "_ts": 1645048564
}
```

- The update to each row should be atomic
- Patch should be supported for both bulk mode and point write mode

Changes

https://aka.ms/azure-cosmos-spark-3-config

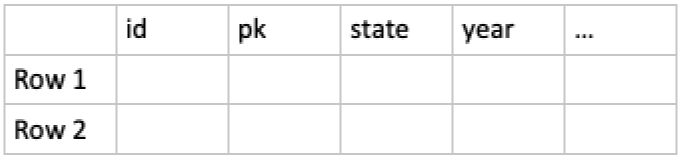
Config property name	Default	
spark.cosmos.write.strategy	ItemOverwrite	ItemOverwrite ItemAppend ItemDelete … ItemPatch
spark.cosmos.write.patch.defaultOperationType	Replace	None Add Set Replace Remove Increment
spark.cosmos.write.patch.columnConfigs	None	Details in below

spark.cosmos.write.patch.filter	None	Details in below

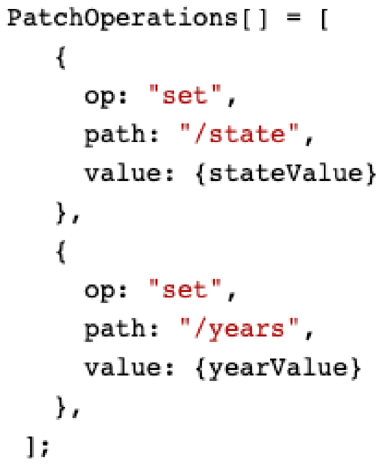
- ***spark.cosmos.write.patch.defaultOperationType***

As we have mentioned above, each column can use a different patch operation. In case all or most of the columns are using the same patch operation, this config gives a convenient way to set it up.

For columns included in the data frame, but not defined in "spark.cosmos.write.patch.operationConfigs", it will use defaultOperationType as the patch operation type, and use column name as the mapping path.



	id	pk	state	year	...
Row 1					
Row 2					

spark.cosmos.write.patch.defaultOperationType -> None	PatchOperation[] = []
spark.cosmos.write.patch.defaultOperationType -> Set (Same rule applies to Add, Replace, Remove, Increment)	 PatchOperations[] = [{ op: "set", path: "/state", value: {stateValue} }, { op: "set", path: "/years", value: {yearValue} },];

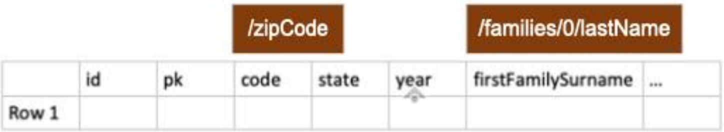
- ***spark.cosmos.write.patch.columnConfigs***

Customer can use this config to give more granular definition of the patch operation type and mapping path for a certain column or columns.

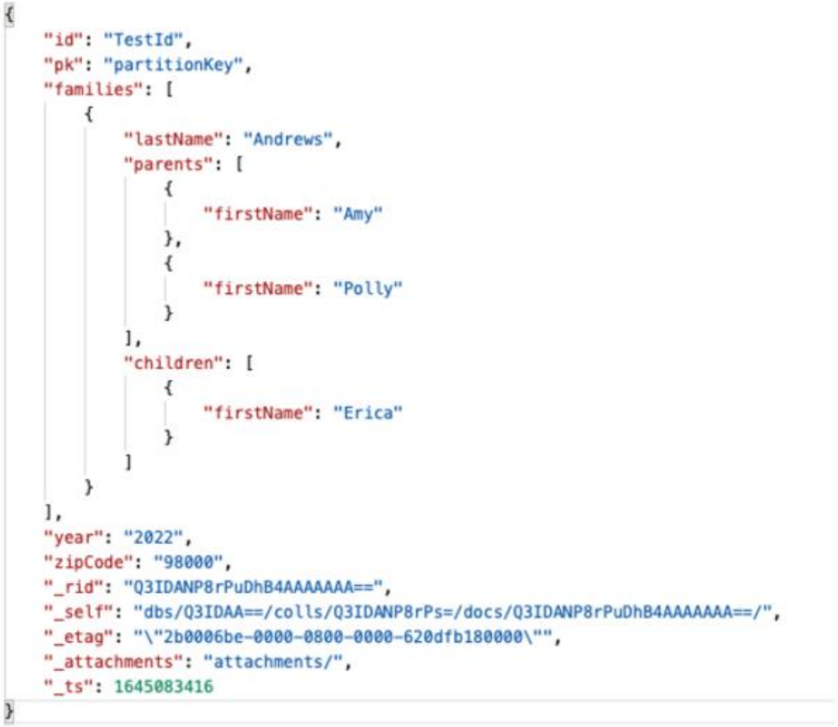
There are two patterns allowed.

col({colName}).op({operationName})

col({colName}).path({mappingPath}). op({operationName})



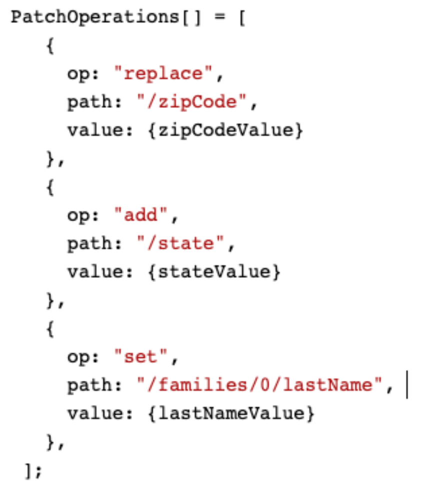
	id	pk	code	state	year	firstFamilySurname	...
Row 1							



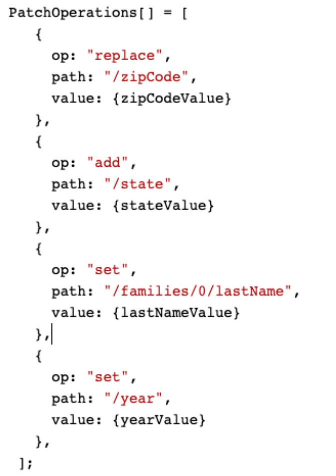
```
{
    "id": "TestId",
    "pk": "partitionKey",
    "families": [
        {
            "lastName": "Andrews",
            "parents": [
                {
                    "firstName": "Amy"
                },
                {
                    "firstName": "Polly"
                }
            ],
            "children": [
                {
                    "firstName": "Erica"
                }
            ]
        }
    ],
    "year": "2022",
    "zipCode": "98000",
    "_rid": "Q3IDANP8rPuDhB4AAAAAAA==",
    "_self": "dbs/Q3IDAA==/colls/Q3IDANP8rPs=/docs/Q3IDANP8rPuDhB4AAAAAAA==/",
    "_etag": "\"2b0006be-0000-0800-0000-620dfb180000\"",
    "_attachments": "attachments/",
    "_ts": 1645083416
}
```

spark.cosmos.write.patch.defaultOperationType -> None

Spark.cosmos.write.patch.columnConfigs -> [col(code).path(/zipCode).op(replace), col(state).op(add), col(firstFamilySurname).path(/families/0/lastName).op(set)]



```
PatchOperations[] = [
   {
     op: "replace",
     path: "/zipCode",
     value: {zipCodeValue}
   },
   {
     op: "add",
     path: "/state",
     value: {stateValue}
   },
   {
     op: "set",
     path: "/families/0/lastName",
     value: {lastNameValue}
   },
 ];
```

spark.cosmos.write.patch.defaultOperationType -> Set
 	
Spark.cosmos.write.patch.columnConfigs ->
[col(code).path(/zipCode).op(replace),
col(state).op(add),
col(firstFamilySurname).path(/families/0/lastName).op(set)]
	
	PatchOperations[] = [
	    {
	      op: "replace",
	      path: "/zipCode",
	      value: {zipCodeValue}
	    },
	    {
	      op: "add",
	      path: "/state",
	      value: {stateValue}
	    },
	    {
	      op: "set",
	      path: "/families/0/lastName",
	      value: {lastNameValue}
	    },
	    {
	      op: "set",
	      path: "/year",
	      value: {yearValue}
	    },
	  ];

- ***spark.cosmos.write.patch.filter***

Conditional patch : https://docs.microsoft.com/en-us/azure/cosmos-db/partial-document-update-getting-started

For example in the above case, you only want to apply the partial updates when it is 2022. Note: this filter will be applied for each row.

“spark.cosmos.write.patch.filter”: “from f where f.year = ‘2022’”