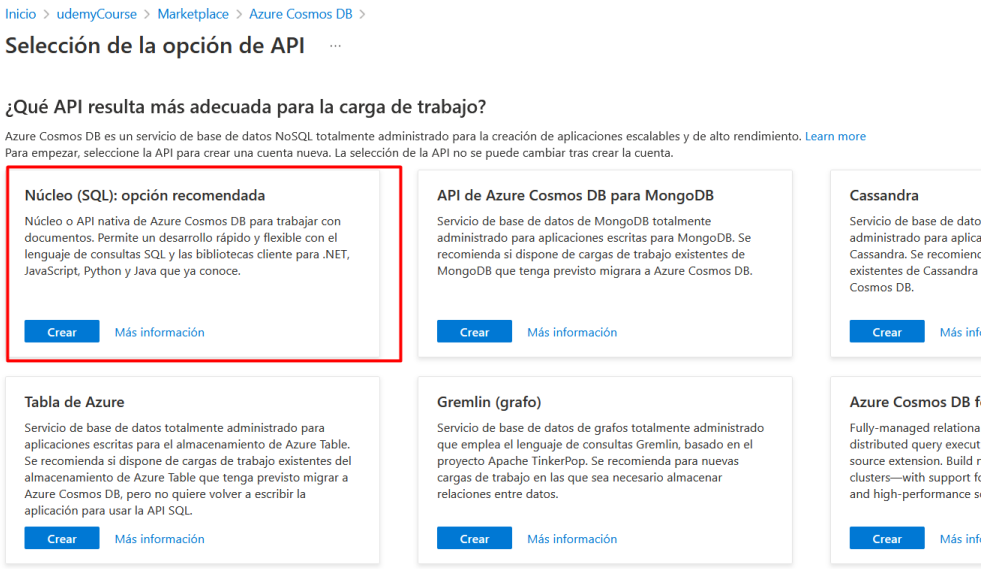
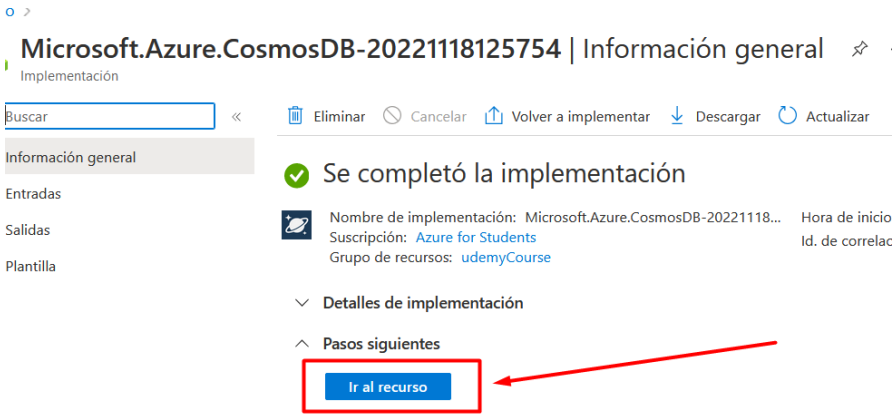
CASO PRACTICO – BENEFICIOS DE LA DESNORMALIZACION EN COSMOS DB

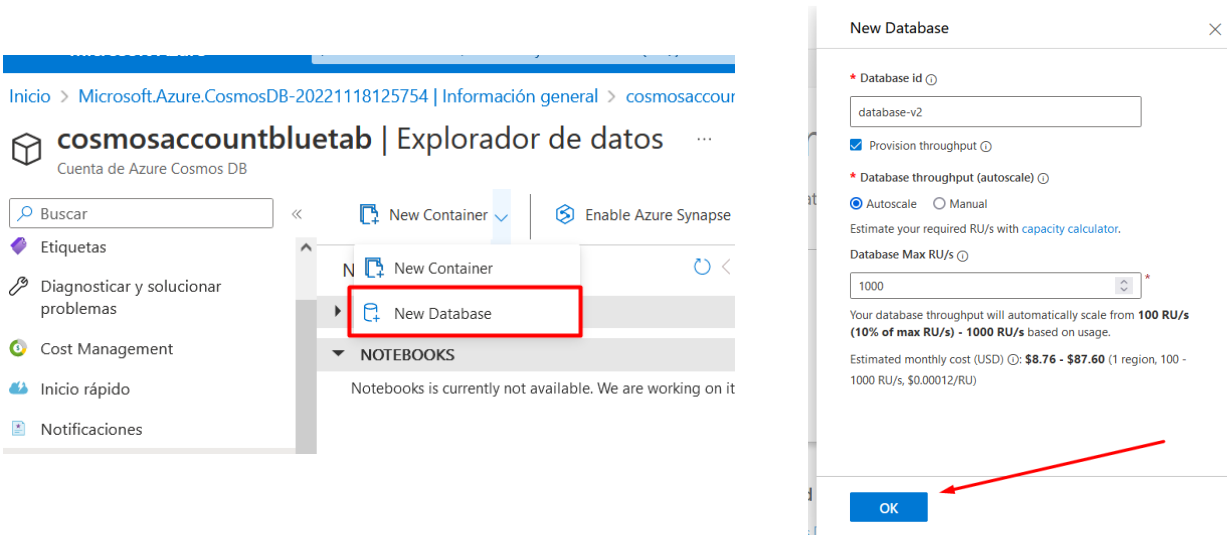
1-Creamos el recurso de CosmosDB, y seleccionamos la api con el se va a trabajar CORE SQL



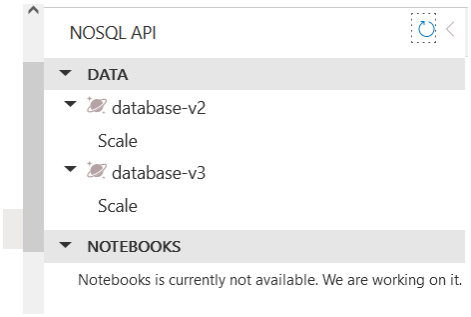
2.-Seleccionamos el grupo de recursos, asignamos el nombre, y creamos el recurso. Al terminar de crearse seleccionamos go to resource.



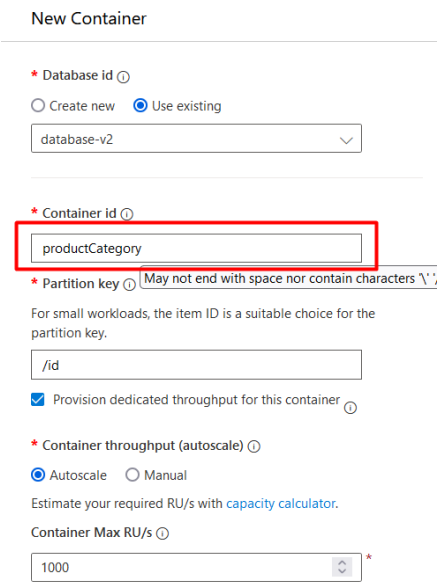
3.- Creamos la base de datos, nos vamos a la opción data explorer

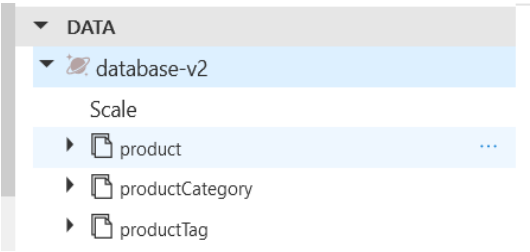


4.- Luego que ya tenemos creamos las bases de datos, refrescamos y nos aparecerá las dos bases de datos.

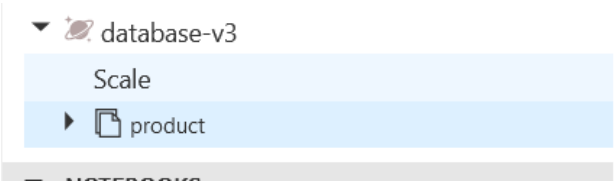


5.- Creando los contenedores database-v2 y verificamos que se haya creado. Repetimos los mismos pasos para producto y productTag

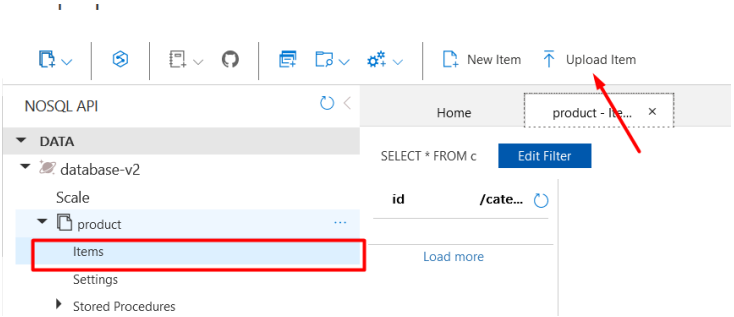




6.- Creamos los contenedores en database-v3,



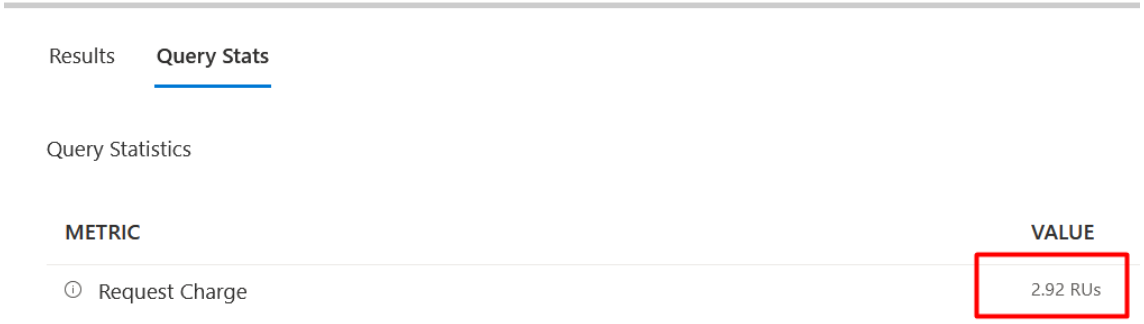
7.- Nos redirigimos al contenedor producto del database-v2 y upload item



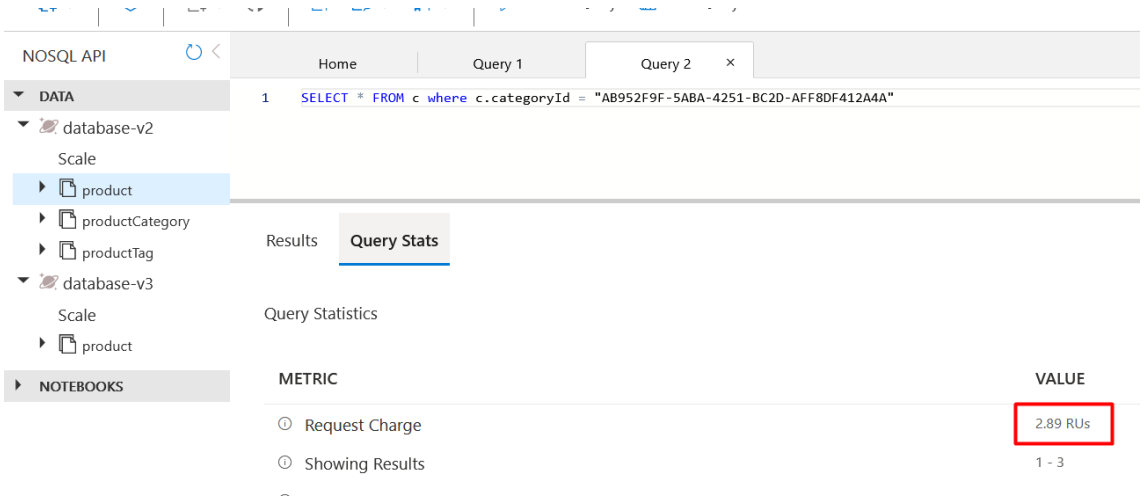
8.- Consulta a base de datos normalizada

productCategory

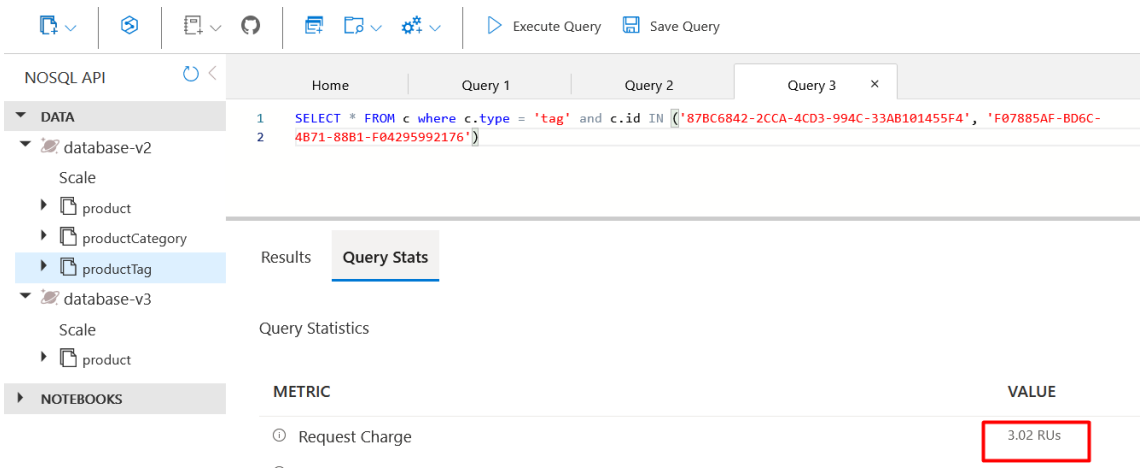
```
SELECT * FROM c where c.type = 'category' and c.id = "AB952F9F-5ABA-4251-BC2D-AFF8DF412A4A"
```

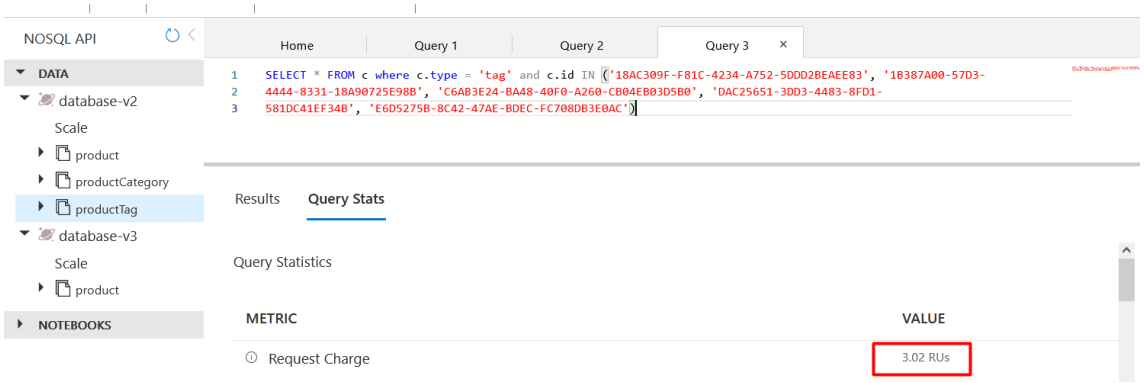


product



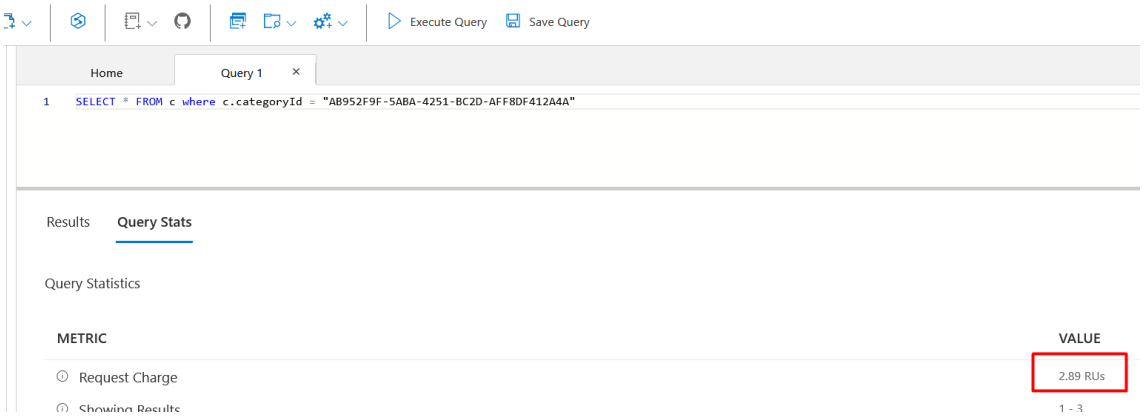
productTag





Requerimiento		RU
Req 1	---	2.92
Req 2	---	2.89
Req 3	---	3.02
Req 4	---	3.02
		10.85

9.- Consulta a base de datos no normalizada



Resumen consumo RU/s para la versión database-v3

Diferencia entre las distintas versiones: 10.85 -2.89 =7.96 (RU/s de ahorro)