

## COSMOSDB TASKS

### 1.Question :

SELECT \* FROM c WHERE ARRAY\_CONTAINS(c.tags, 'database')

The screenshot shows the Azure Cosmos DB query editor interface. The query editor has a toolbar with icons for saving, executing, and refreshing. The query text is: `SELECT * FROM c WHERE ARRAY_CONTAINS(c.tags, 'database')`. The left sidebar shows the database structure: `database1` (Scale) and `container1` (Items, Settings, Stored Procedures, User Defined Functions, Triggers). The bottom section shows the query results under the 'Results' tab, displaying a single document with the following JSON structure:

```
{
  "isbn": 6677,
  "downloadable": true,
  "no_of_reviews": 4,
  "tags": [
    "database",
    "cloud"
  ]
}
```

### 2.Question:

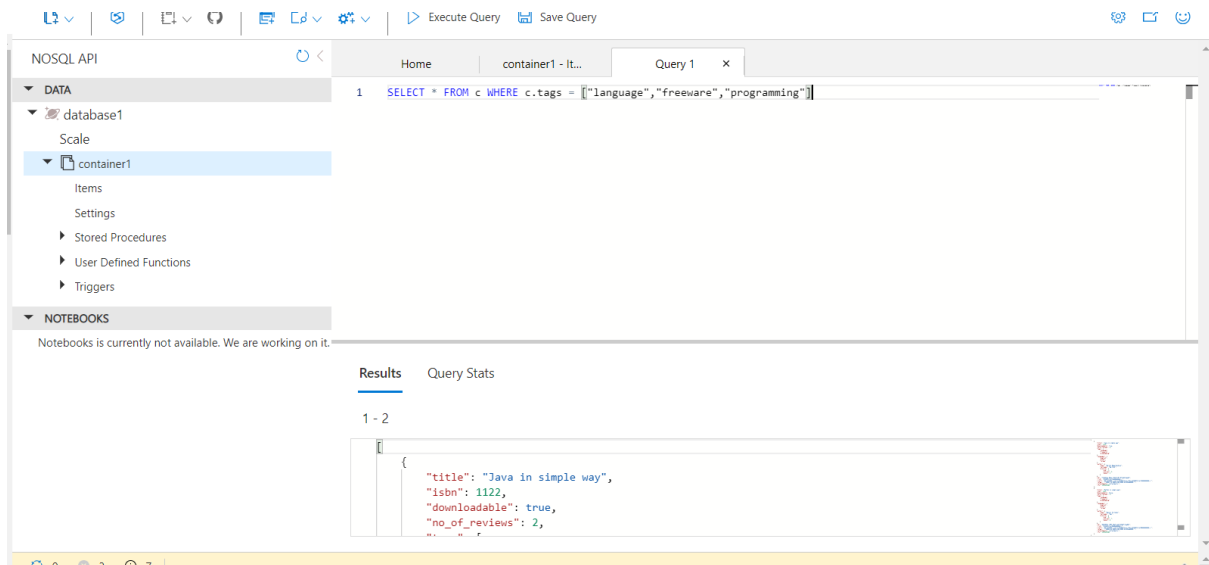
SELECT \* FROM c WHERE c.author.profile.courses = 2

The screenshot shows the Azure Cosmos DB query editor interface. The query editor has a toolbar with icons for saving, executing, and refreshing. The query text is: `SELECT * FROM c WHERE c.author.profile.courses = 2`. The left sidebar shows the database structure: `database1` (Scale) and `container1` (Items, Settings, Stored Procedures, User Defined Functions, Triggers). The bottom section shows the query results under the 'Results' tab, displaying a single document with the following JSON structure:

```
{
  "title": "Java in simple way",
  "isbn": 1122,
  "downloadable": true,
  "no_of_reviews": 2,
  "author": {
    "profile": {
      "courses": 2
    }
  }
}
```

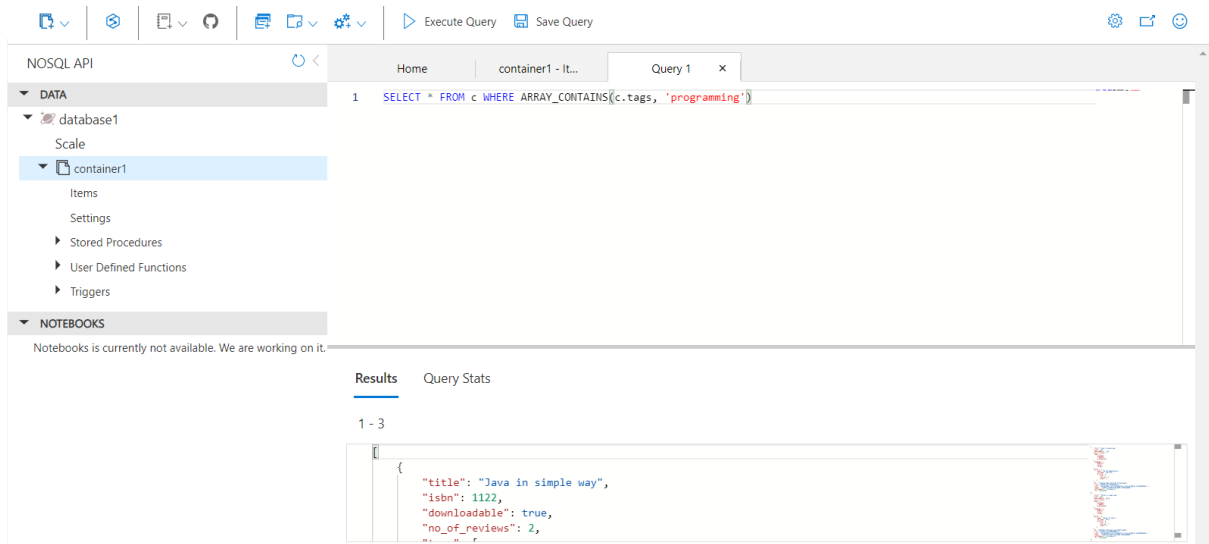
### 3.Question:

SELECT \* FROM c WHERE c.tags = ["language","freeware","programming"]



### 4.Question:

SELECT \* FROM c WHERE ARRAY\_CONTAINS(c.tags, 'programming')



### Question 5:

SELECT \* FROM c WHERE ARRAY\_CONTAINS(c.languages, 'telugu')

The screenshot shows the NOSQL API interface. The left sidebar has a 'DATA' section with 'database1' expanded, showing 'container1' selected. The main area displays a query: `SELECT * FROM c WHERE ARRAY_CONTAINS(c.languages, 'telugu')`. Below the query, the 'Results' tab is active, showing a single result: `{ "title": "Linux in simple way", "isbn": 6677, "downloadable": false, "no_of_reviews": 1, ... }`. The 'Query Stats' tab is also visible.

### Question 6:

SELECT \* FROM c

The screenshot shows the NOSQL API interface. The left sidebar has a 'DATA' section with 'database1' expanded, showing 'container1' selected. The main area displays a query: `SELECT * FROM c`. Below the query, the 'Results' tab is active, showing a single result: `{ "title": "Linux in simple way", "isbn": 6677, "downloadable": false, "no_of_reviews": 1, ... }`. The 'Query Stats' tab is also visible.

### Question 7:

SELECT VALUE COUNT(1) FROM c

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The 'container1' folder is selected. The main area shows a query editor with the text '1 SELECT VALUE COUNT(1) FROM c'. Below the editor, the 'Results' tab is active, displaying '1 - 1' and a single result: a JSON array containing the number 7.

```
1 SELECT VALUE COUNT(1) FROM c
```

```
[ 7 ]
```

### Question 8:

SELECT TOP 1 \* FROM c

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The 'container1' folder is selected. The main area shows a query editor with the text '1 SELECT TOP 1 \* FROM c'. Below the editor, the 'Results' tab is active, displaying '1 - 1' and a single result: a JSON object representing a book.

```
1 SELECT TOP 1 * FROM c
```

```
{  "title": "Linux in simple way",  "isbn": 6677,  "downloadable": false,  "no_of_reviews": 1,
```

### Question 9:

```
SELECT * FROM c WHERE ARRAY_CONTAINS(c.tags, 'programming') OR c.no_of_reviews > 3
```

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main panel shows a query editor with the query: `SELECT * FROM c WHERE ARRAY_CONTAINS(c.tags, 'programming') OR c.no_of_reviews > 3`. Below the query editor, the 'Results' tab is active, showing a single document: `{ "title": "Java in simple way", "isbn": 1122, "downloadable": true, "no_of_reviews": 2, ... }`.

### Question 10:

```
SELECT * FROM c WHERE c.downloadable = true OR c.no_of_reviews < 3 OR c.author.profile.books >= 2
```

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main panel shows a query editor with the query: `SELECT * FROM c WHERE c.downloadable = true OR c.no_of_reviews < 3 OR c.author.profile.books >= 2`. Below the query editor, the 'Results' tab is active, showing a single document: `{ "title": "Linux in simple way", "isbn": 6677, "downloadable": false, "no_of_reviews": 1, ... }`.

### Question 11:

`SELECT * FROM c WHERE c.no_of_reviews > 3`

The screenshot shows the NOSQL API interface. The left sidebar has a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main area displays the query: `1 SELECT * FROM c WHERE c.no_of_reviews > 3`. Below the query, the 'Results' tab is active, showing a single JSON document: `{ "title": "Python in simple way", "isbn": 1234, "downloadable": false, "no_of_reviews": 5 }`. The 'Query Stats' tab is also visible.

### Question 12:

`SELECT * FROM c WHERE c.no_of_reviews != 3`

The screenshot shows the NOSQL API interface. The left sidebar has a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main area displays the query: `1 SELECT * FROM c WHERE c.no_of_reviews != 3`. Below the query, the 'Results' tab is active, showing a single JSON document: `{ "title": "Linux in simple way", "isbn": 6677, "downloadable": false, "no_of_reviews": 1 }`. The 'Query Stats' tab is also visible.

### Question 13:

SELECT \* FROM c WHERE c.no\_of\_reviews >= 3 AND c.downloadable = true

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main area shows a query editor with the query: `SELECT * FROM c WHERE c.no_of_reviews >= 3 AND c.downloadable = true`. The 'Results' tab is active, showing a single document: `{ "title": "MongoDB in simple way", "isbn": 6677, "downloadable": true, "no_of_reviews": 4, ... }`. The status bar at the bottom indicates 'Successfully fetched 2 item for container container1'.

### Question 14:

SELECT \* FROM c WHERE ARRAY\_CONTAINS(c.tags, 'database') OR ARRAY\_CONTAINS(c.tags, 'programming')

The screenshot shows the NOSQL API interface. The left sidebar displays a tree view with 'DATA' expanded, showing 'database1' and 'container1'. The main area shows a query editor with the query: `SELECT * FROM c WHERE ARRAY_CONTAINS(c.tags, 'database') OR ARRAY_CONTAINS(c.tags, 'programming')`. The 'Results' tab is active, showing a single document: `{ "title": "Java in simple way", "isbn": 1122, "downloadable": true, "no_of_reviews": 2, ... }`. The status bar at the bottom indicates 'Successfully fetched 2 item for container container1'.