DP-420 Lab 16 (Measure Performance)

1 Hour Remaining

Instructions Resources Help  100%

Before You Start

As you follow the instructions in this pane, whenever you see a icon, you can use it to copy text from the instruction pane into the virtual machine interface. This is particularly useful to copy code; but bear in mind you may need to modify the pasted code to fix indent levels or formatting before running it!

1. If prompted, log into the **Student** account with the following local credentials:

|  |  |
| --- | --- |
| VM Username | Student |
| VM Password | Pa55w.rd |

1. If prompted to allow your PC to be discoverable, select **No**.
2. An Azure sandbox subscription is provided for this lab. You can use the following Azure subscription credentials to complete this lab:

|  |  |
| --- | --- |
| Email Address | Student-36970147@triplecrownlabsoutlook.onmicrosoft.com |
| Password | e!DheDB00! |

1. If you are interested in completing this lab using your own Azure subscription, sign up for a free trial at <https://azure.microsoft.com/free/>.

Measure performance for customer entities

In this exercise, you'll measure the difference for customer entities when you model entities as separate containers versus when you model for a NoSQL database by embedding entities in a single document.

Prepare your development environment

If you have not already cloned the lab code repository for **DP-420** to the environment where you're working on this lab, follow these steps to do so. Otherwise, open the previously cloned folder in **Visual Studio Code**.

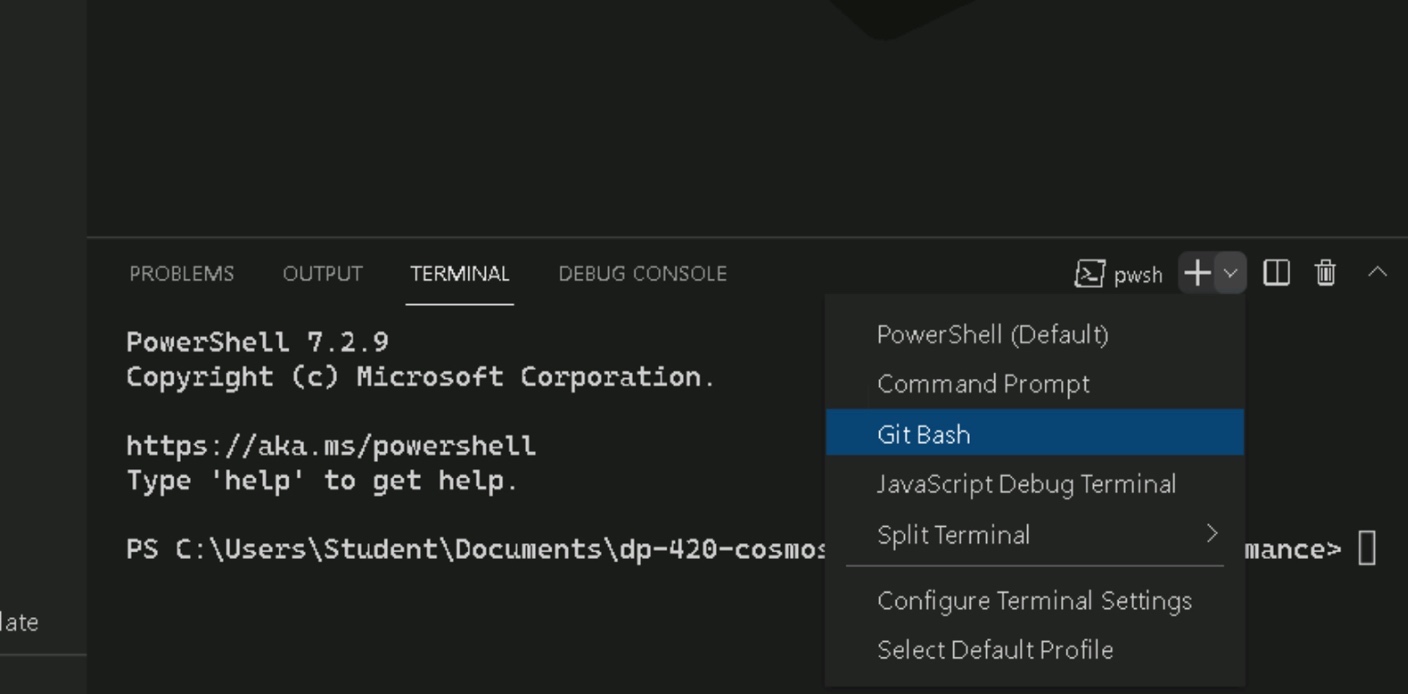
1. Start **Visual Studio Code**.

📝 If you are not already familiar with the Visual Studio Code interface, review the [Get Started guide for Visual Studio Code](https://labclient.labondemand.com/Instructions/60e7542a-581a-4447-afe6-9e607efec35a?rc=10" \t "_blank)

1. Open the command palette and run **Git: Clone** to clone the https://github.com/microsoftlearning/dp-420-cosmos-db-dev GitHub repository in a local folder of your choice.

💡 You can use the **CTRL+SHIFT+P** keyboard shortcut to open the command palette.

1. Once the repository has been cloned, open the local folder you selected in **Visual Studio Code**.
2. In **Visual Studio Code**, in the **Explorer** pane, browse to the **16-measure-performance** folder.
3. Open the context menu for the **16-measure-performance** folder and then select **Open in Integrated Terminal** to open a new terminal instance.
4. If the terminal opens as a **Windows Powershell** terminal, open a new **Git Bash** terminal.



💡 To open a **Git Bash** terminal, on the right hand side of the the terminal menu,click on the pulldown besides the **+** sign and choose *Git Bash*.

1. In the **Git Bash terminal**, run the following commands. The commands open a browser window to connect to the azure portal where you will use the provided lab credentials, run a script that creates a new Azure Cosmos DB account, and then build and start the app you use to populate the database and complete the exercises. *Once you have entered the provided credential for the azure account, the build may take 15-20 minutes to complete, so it might be a good time to get some coffee or tea*.

"C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe" -m pip install pip-system-certs

az login

cd 16-measure-performance

bash init.sh

dotnet add package Microsoft.Azure.Cosmos --version 3.22.1

dotnet build

dotnet run --load-data

🡪 수정: azuredeploy.json 의 location > default 값을 ‘westus’ 로 수정

A screen shot of a computer

Description automatically generated

1. Close the integrated terminal.

Measure performance of entities in separate containers

In Database-v1, data is stored in individual containers. In that database, run queries to get the customer, customer address, and customer password. Review the request charge for each of those queries.

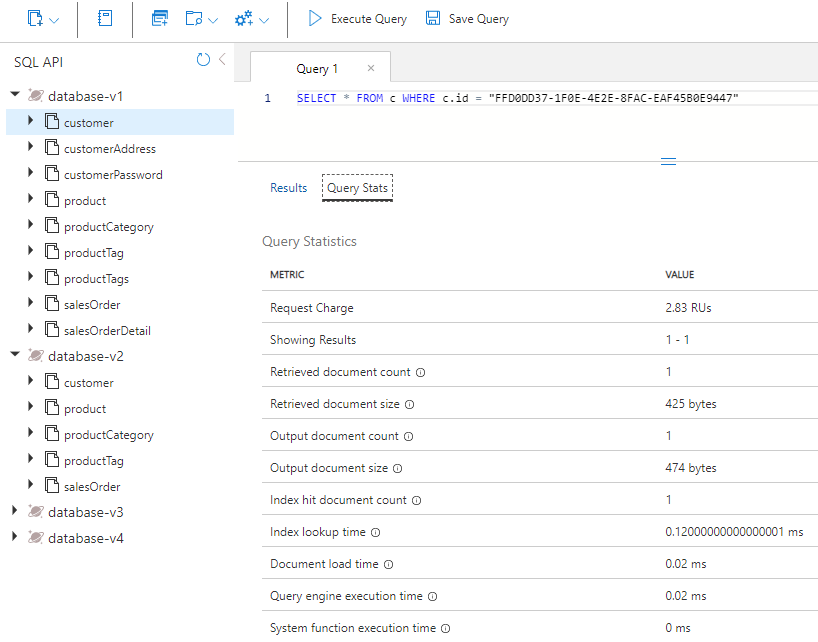
Query for customer entity

In Database-v1, run a query to get the customer entity and review the request charge.

1. In a new web browser window or tab, navigate to the Azure portal (portal.azure.com).
2. Sign into the portal using the Microsoft credentials associated with your subscription.
3. On the Azure portal menu, or from the **Home** page, select **Azure Cosmos DB**.
4. Select the Azure Cosmos DB account with the name that starts with **cosmicworks**.
5. Select **Data Explorer** on the left side.
6. Expand **Database-v1**.
7. Select the **Customer** container.
8. At the top of the screen, select **New SQL Query**.
9. Copy and paste the following SQL text and then select **Execute Query**.

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. Select the **Query Stats** tab and note the request charge of 2.83.



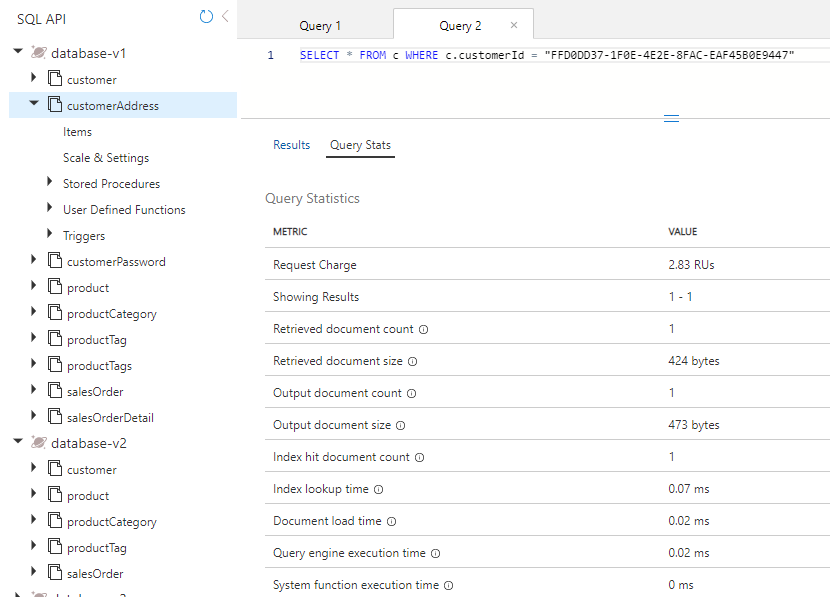
Query for customer address

Run a query to get the customer address entity and review the request charge.

1. Select the **CustomerAddress** container.
2. At the top of the screen, select **New SQL Query**.
3. Copy and paste the following SQL text and then select **Execute Query**.

SELECT \* FROM c WHERE c.customerId = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. Select the **Query Stats** tab and note the request charge of 2.83.



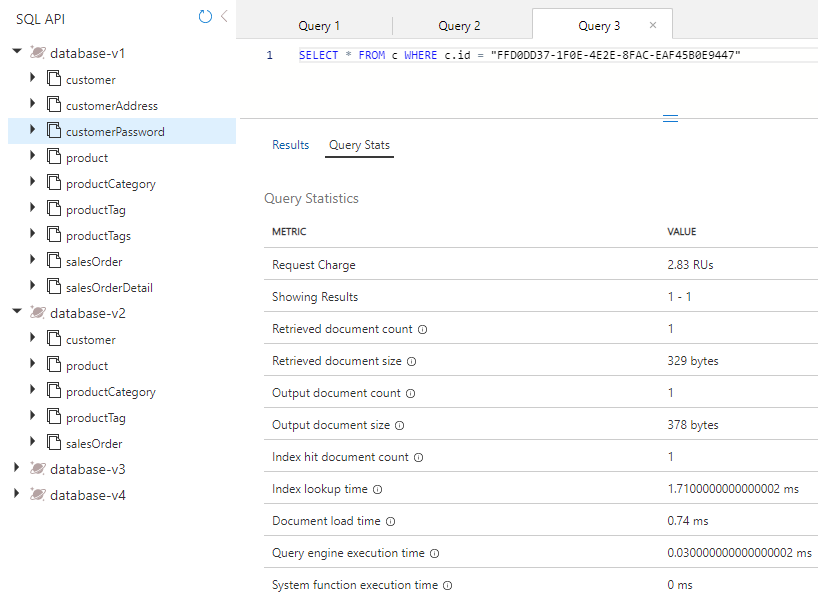
Query for customer password

Run a query to get the customer password entity and review the request charge.

1. Select the **CustomerPassword** container.
2. At the top of the screen, select **New SQL Query**.
3. Copy and paste the following SQL text and then select **Execute Query**.

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. Select the **Query Stats** tab and note the request charge of 2.83.



Add up the request charges

Now that we've run all of our queries, let's add up all of the Request Unit costs for them.

| **Query** | **RU/s cost** |
| --- | --- |
| Customer | 2.83 |
| Customer Address | 2.83 |
| Customer Password | 2.83 |
| **Total RU/s** | **8.49** |

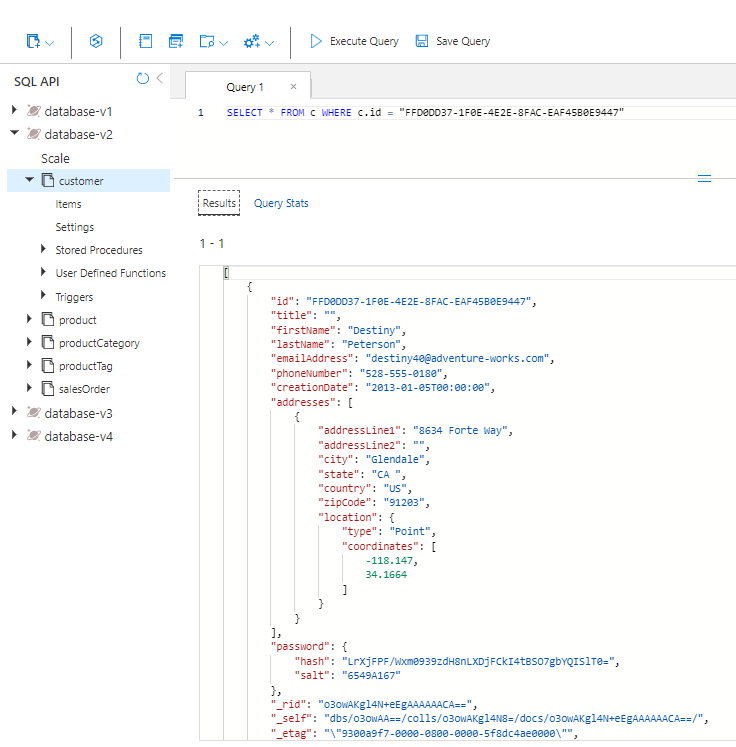
Measure performance of embedded entities

Now we're going to query for the same information but with the entities embedded in a single document.

1. Select the **Database-v2** database.
2. Select the **Customer** container.
3. Run the following query.

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. Observe that the data coming back is now a hierarchy of customer, address, and password data.



1. Select **Query Stats**. Note the request charge of 2.83, versus the 8.49 RU/s for the three queries that you ran earlier.

Compare the performance of the two models

When you compare the RU/s for each query that you ran, you see that the last query where the customer entities are in a single document is much less expensive than the combined cost for running the three queries independently. The latency for returning this data is lower because the data is returned in a single operation.

When you're searching for a single item and know the partition key and ID of the data, you can retrieve this data via a *point-read* by calling ReadItemAsync() in the Azure Cosmos DB SDK. A point-read is even faster than our query. For the same customer data, the cost is just 1 RU/s, which is a nearly threefold improvement.

Wrapping up

Clean up

The Azure sandbox automatically cleans up your resources when you're finished with this module.

When you're working in your own subscription, it's a good idea at the end of a project to identify whether you still need the resources you created. Resources left running can cost you money. You can delete resources individually or delete the resource group to delete the entire set of resources.

More information

For more information about Azure Cosmos DB, see the [Azure Cosmos DB documentation](https://docs.microsoft.com/azure/cosmos-db/" \t "_blank).

Return to Microsoft Learn

Now that you have completed the exercise, return to Microsoft Learn to complete the knowledge check and earn points for completing this module.

DP-420 실습 16(성능 측정)

59 Minutes Remaining

지침 자원 돕다  100%

시작하기 전에

이 창의 지침을 따르면 아이콘이 나타날 때마다 이를 사용하여 지침 창의 텍스트를 가상 머신 인터페이스에 복사할 수 있습니다. 이는 코드를 복사할 때 특히 유용합니다. 하지만 실행하기 전에 들여쓰기 수준이나 형식을 수정하기 위해 붙여넣은 코드를 수정해야 할 수도 있다는 점을 명심하세요!

1. 메시지가 표시되면 다음 로컬 자격 증명을 사용하여 **학생 계정에 로그인합니다.**

|  |  |
| --- | --- |
| VM 사용자 이름 | 학생 |
| VM 비밀번호 | Pa55w.rd |

1. PC를 검색할 수 있도록 허용하라는 메시지가 표시되면 **아니요 를** 선택합니다 .
2. 이 랩에는 Azure 샌드박스 구독이 제공됩니다. 다음 Azure 구독 자격 증명을 사용하여 이 랩을 완료할 수 있습니다.

|  |  |
| --- | --- |
| 이메일 주소 | Student-36970147@triplecrownlabsoutlook.onmicrosoft.com |
| 비밀번호 | 에!DheDB00! |

1. [자신의 Azure 구독을 사용하여 이 랩을 완료하려면 https://azure.microsoft.com/free/](https://azure.microsoft.com/free/) 에서 무료 평가판에 등록하세요 .

고객 엔터티의 성과 측정

이 연습에서는 엔터티를 별도의 컨테이너로 모델링할 때와 단일 문서에 엔터티를 포함하여 NoSQL 데이터베이스에 대해 모델링할 때 고객 엔터티의 차이를 측정합니다.

개발 환경 준비

**DP-420** 용 랩 코드 저장소를 이 랩에서 작업 중인 환경에 아직 복제하지 않은 경우 다음 단계에 따라 복제하십시오. **그렇지 않으면 Visual Studio Code** 에서 이전에 복제한 폴더를 엽니다 .

1. **Visual Studio Code를** 시작합니다 .

📝 Visual Studio Code 인터페이스에 아직 익숙하지 않은 경우 [Visual Studio Code 시작 가이드를 검토하세요.](https://labclient.labondemand.com/Instructions/60e7542a-581a-4447-afe6-9e607efec35a?rc=10" \t "_blank)

1. 명령 팔레트를 열고 **Git: Clone을** 실행하여 원하는 로컬 폴더에 GitHub 리포지토리를 복제합니다 .https://github.com/microsoftlearning/dp-420-cosmos-db-dev

💡 **CTRL+SHIFT+P** 키보드 단축키를 사용하여 명령 팔레트를 열 수 있습니다.

1. 리포지토리가 복제되면 **Visual Studio Code** 에서 선택한 로컬 폴더를 엽니다 .
2. **Visual Studio Code** 의 **탐색기** 창 에서 **16-measure-performance** 폴더를 찾습니다 .
3. **16-measure-performance** 폴더 의 컨텍스트 메뉴를 연 다음 **통합 터미널에서 열기를** 선택하여 새 터미널 인스턴스를 엽니다.
4. 터미널이 **Windows Powershell** 터미널로 열리면 새 **Git Bash** 터미널을 엽니다.

**💡 Git Bash** 터미널을 열려면 터미널 메뉴 오른쪽에서 **+** 기호 옆의 풀다운을 클릭하고 *Git Bash를* 선택합니다 .

1. **Git Bash 터미널** 에서 다음 명령을 실행합니다. 이 명령은 브라우저 창을 열어 제공된 랩 자격 증명을 사용할 Azure Portal에 연결하고, 새 Azure Cosmos DB 계정을 만드는 스크립트를 실행한 다음, 데이터베이스를 채우고 연습을 완료하는 데 사용하는 앱을 빌드 및 시작합니다. . *Azure 계정에 대해 제공된 자격 증명을 입력하면 빌드가 완료되는 데 15~20분 정도 걸릴 수 있으므로 커피나 차를 마시기에 좋은 시간이 될 수 있습니다* .
2. "C:\Program Files (x86)\Microsoft SDKs\Azure\CLI2\python.exe" -m pip install pip-system-certs
3. az login
4. cd 16-measure-performance
5. bash init.sh
6. dotnet add package Microsoft.Azure.Cosmos --version 3.22.1
7. dotnet build

dotnet run --load-data

1. 통합 터미널을 닫습니다.

별도의 컨테이너에 있는 엔터티의 성능 측정

Database-v1에서는 데이터가 개별 컨테이너에 저장됩니다. 해당 데이터베이스에서 쿼리를 실행하여 고객, 고객 주소 및 고객 비밀번호를 가져옵니다. 각 쿼리에 대한 요청 요금을 검토하세요.

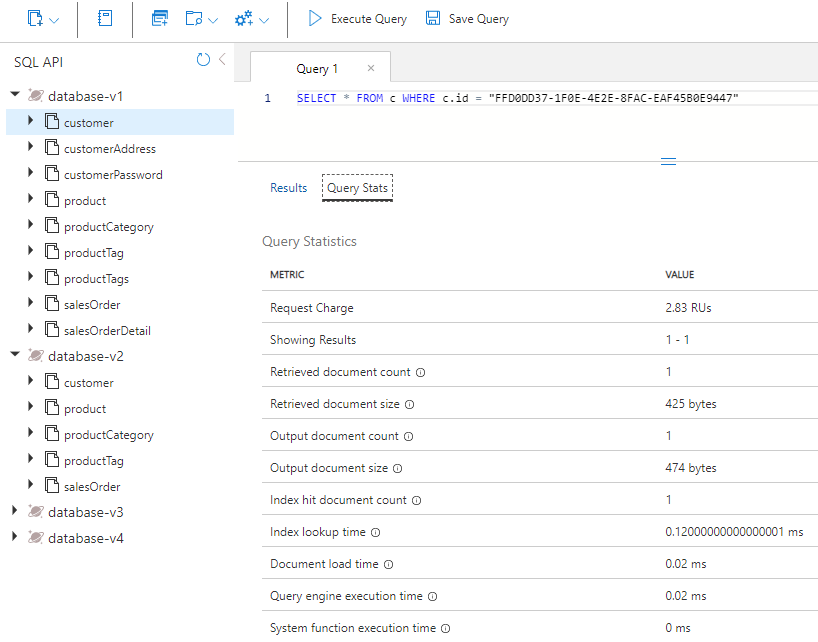
고객 엔터티에 대한 쿼리

Database-v1에서 쿼리를 실행하여 고객 엔터티를 가져오고 요청 요금을 검토합니다.

1. 새 웹 브라우저 창 또는 탭에서 Azure Portal( )로 이동합니다.portal.azure.com
2. 구독과 연결된 Microsoft 자격 증명을 사용하여 포털에 로그인합니다.
3. Azure Portal 메뉴 또는 **홈** 페이지 에서 **Azure Cosmos DB 를** 선택합니다 .
4. **cosmicworks** 로 시작하는 이름을 가진 Azure Cosmos DB 계정을 선택합니다 .
5. 왼쪽에서 **데이터 탐색기를** 선택합니다 .
6. **Database-v1 을** 확장합니다 .
7. **고객** 컨테이너를 선택합니다 .
8. 화면 상단에서 **New SQL Query 를** 선택합니다 .
9. 다음 SQL 텍스트를 복사하여 붙여넣은 다음 **쿼리 실행을** 선택합니다 .

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. **쿼리 통계** 탭을 선택 하고 요청 요금이 2.83인지 확인하세요.



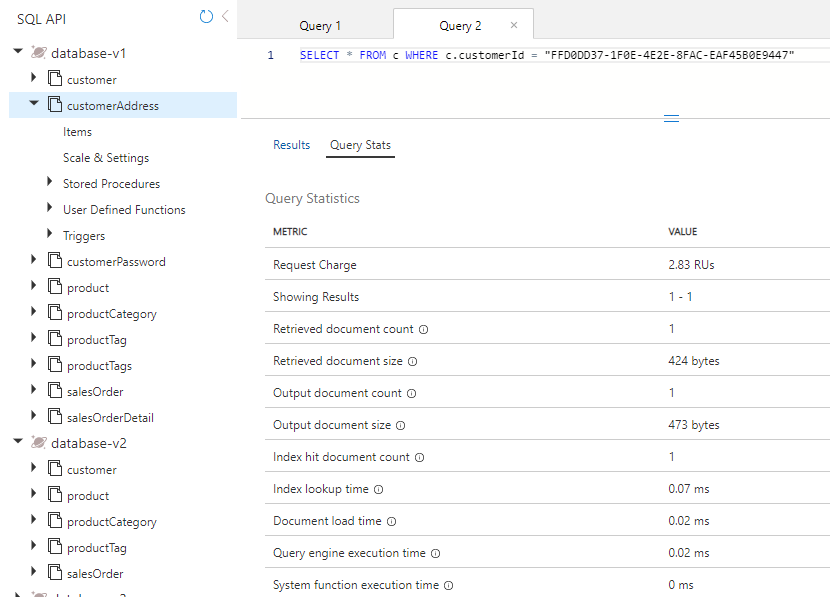
고객 주소 쿼리

쿼리를 실행하여 고객 주소 엔터티를 가져오고 요청 요금을 검토하세요.

1. **CustomerAddress** 컨테이너를 선택합니다 .
2. 화면 상단에서 **New SQL Query 를** 선택합니다 .
3. 다음 SQL 텍스트를 복사하여 붙여넣은 다음 **쿼리 실행을** 선택합니다 .

SELECT \* FROM c WHERE c.customerId = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. **쿼리 통계** 탭을 선택 하고 요청 요금이 2.83인지 확인하세요.



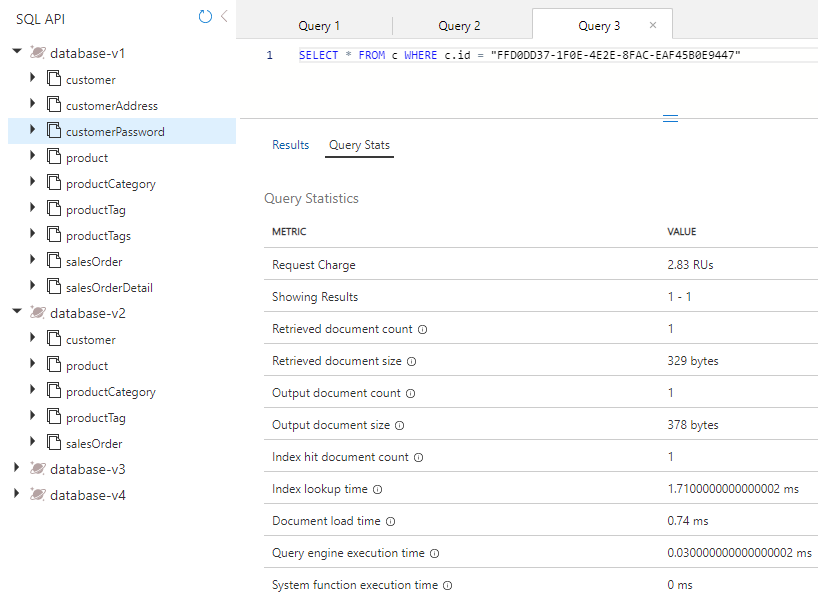
고객 비밀번호 조회

쿼리를 실행하여 고객 비밀번호 엔터티를 가져오고 요청 요금을 검토하세요.

1. **CustomerPassword** 컨테이너를 선택합니다 .
2. 화면 상단에서 **New SQL Query 를** 선택합니다 .
3. 다음 SQL 텍스트를 복사하여 붙여넣은 다음 **쿼리 실행을** 선택합니다 .

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. **쿼리 통계** 탭을 선택 하고 요청 요금이 2.83인지 확인하세요.



요청 비용을 합산하세요.

이제 모든 쿼리를 실행했으므로 해당 쿼리에 대한 요청 단위 비용을 모두 합산해 보겠습니다.

| **질문** | **RU/초 비용** |
| --- | --- |
| 고객 | 2.83 |
| 고객 주소 | 2.83 |
| 고객 비밀번호 | 2.83 |
| **총 RU/초** | **8.49** |

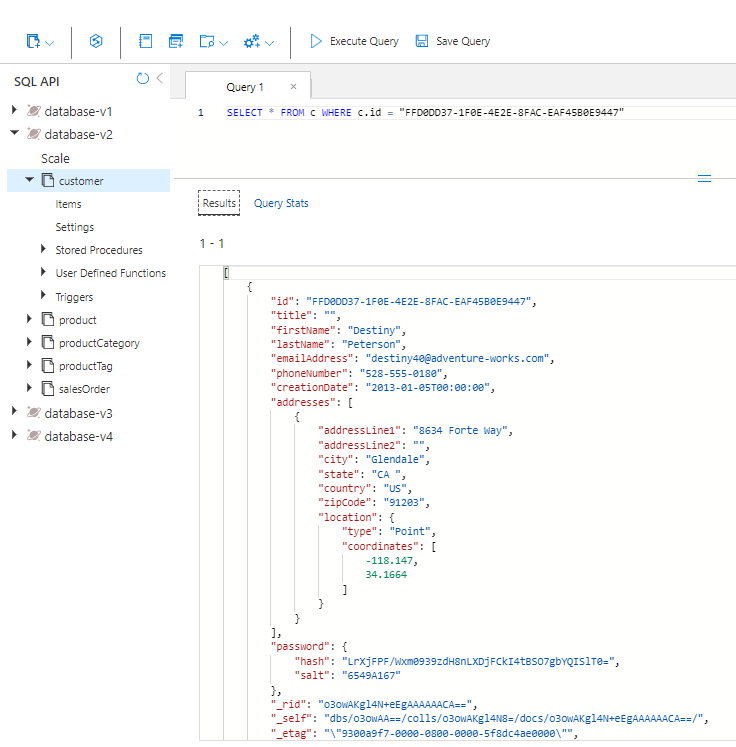
내장된 엔터티의 성능 측정

이제 동일한 정보를 쿼리하지만 단일 문서에 엔터티가 포함되어 있습니다.

1. **Database-v2** 데이터베이스를 선택합니다 .
2. **고객** 컨테이너를 선택합니다 .
3. 다음 쿼리를 실행합니다.

SELECT \* FROM c WHERE c.id = "FFD0DD37-1F0E-4E2E-8FAC-EAF45B0E9447"

1. 돌아오는 데이터는 이제 고객, 주소 및 비밀번호 데이터의 계층 구조입니다.



1. **쿼리 통계를** 선택합니다 . 이전에 실행한 세 가지 쿼리에 대한 요청 요금은 8.49RU/s와 비교하여 2.83입니다.

두 모델의 성능 비교

실행한 각 쿼리의 RU/s를 비교하면 고객 엔터티가 단일 문서에 있는 마지막 쿼리가 세 쿼리를 독립적으로 실행하는 데 드는 총 비용보다 훨씬 저렴하다는 것을 알 수 있습니다. 데이터가 단일 작업으로 반환되므로 이 데이터 반환에 대한 대기 시간이 더 짧습니다.

단일 항목을 검색하고 데이터의 파티션 키와 ID를 알고 있는 경우 Azure Cosmos DB SDK를 호출하여 *지점 읽기를 통해 이 데이터를 검색할 수 있습니다.*포인트 읽기는 쿼리보다 훨씬 빠릅니다. 동일한 고객 데이터의 경우 비용은 1RU/s에 불과하며 이는 거의 3배 향상된 것입니다.ReadItemAsync()

마무리

정리하다

이 모듈을 마치면 Azure 샌드박스가 자동으로 리소스를 정리합니다.

자신의 구독으로 작업하는 경우 프로젝트가 끝날 때 생성한 리소스가 여전히 필요한지 확인하는 것이 좋습니다. 리소스를 계속 실행하면 비용이 발생할 수 있습니다. 리소스를 개별적으로 삭제하거나 리소스 그룹을 삭제하여 전체 리소스 집합을 삭제할 수 있습니다.

추가 정보

Azure Cosmos DB에 대한 자세한 내용은 [Azure Cosmos DB 설명서를](https://docs.microsoft.com/azure/cosmos-db/" \t "_blank) 참조하세요 .

Microsoft Learn으로 돌아가기

이제 연습을 완료했으므로 Microsoft Learn으로 돌아가 지식 확인을 완료하고 이 모듈 완료에 대한 포인트를 획득하세요.