



Azure Service

---

# Migrate EDW to Azure SQL Data Warehouse

# Contents

1. 사전 준비 .....	3
1.1. 시나리오 개요 .....	3
1.2. 실습대상 .....	3
1.3. Azure 서비스 및 관련 제품 .....	3
1.4. 필수 조건 .....	3
2. 학습 목표 .....	3
2.1. 솔루션 아키텍처 .....	3
2.2. 대상 환경 정보 .....	4
3. Exercise 1: Configure Azure Services.....	5
3.1. Task 1: Create an Azure Data Factory v2 .....	5
3.2. Task 2: Create an Azure SQL Data Warehouse .....	6
3.3. Task 3: Create an Azure Storage Account.....	8
3.4. Task 4: Prepare Environment and Create Migration Accounts .....	10
4. Exercise 2: Data and schema preparation.....	13
4.1. Task 1: Prepare Azure SQL Data Warehouse and migrate schema .....	13
4.2. Task 2: Data Exporting .....	13
5. Exercise 3: Migrate the data to Azure SQL Data Warehouse.....	15
5.1. Task 1: Transfer your data to Azure .....	15
6. Exercise 4: Migrate to Data Factory v2 .....	19
6.1. Task 1: Copy data .....	19
6.2. Task 2: Add new data.....	30
6.3. Task 3: Create pipeline .....	31
7. Exercise 5: Visualize data with Power BI Desktop .....	36
7.1. Task 1: Install Power BI Desktop .....	36
7.2. Task 2: Query data with Power BI Desktop .....	39



## 1. 사전 준비

### 1.1. 시나리오 개요

Coho, 가전 제품에 중점을 둔 소매 회사인 Coho는 데이터 아키텍처를 현대화하고 있습니다. 이러한 노력의 핵심은 기존 엔터프라이즈 Data Warehouse를 클라우드로 마이그레이션 하는 것입니다.

### 1.2. 실습대상

- Database Administrators
- Database Developers
- Data Architects

### 1.3. Azure 서비스 및 관련 제품

- Azure SQL Data Warehouse
- Azure Data Factory v2
- Azure Storage
- Power BI

### 1.4. 필수 조건

- Azure Subscription

## 2. 학습 목표

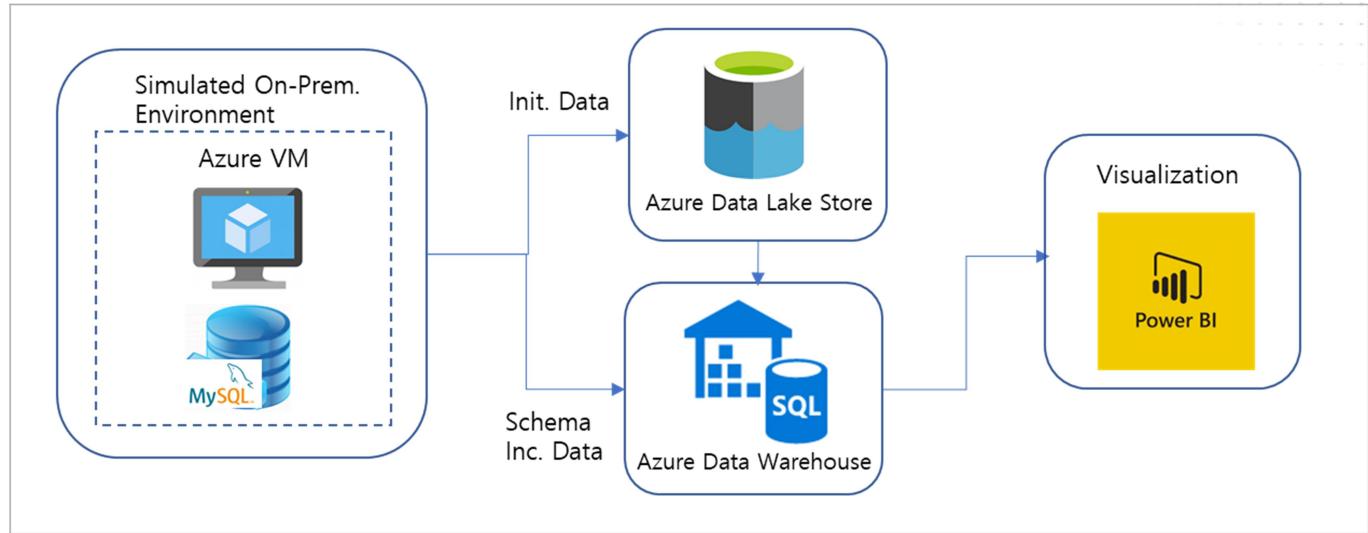
이번 실습에서는 온-프레미스 Data Warehouse를 Azure SQL Data Warehouse로 마이그레이션 하는 과정을 살펴볼 것입니다. 실습을 통해 사내 구축 환경의 데이터베이스에서 클라우드로의 효율적인 마이그레이션, Azure SQL Data Warehouse의 데이터 배포, ETL 작업을 Azure Data Factory에 연결하고 Azure SQL Data Warehouse에서 Ad-hoc 워크로드를 지원합니다.

실습이 끝나면 기존 온-프레미스 환경의 엔터프라이즈 Data Warehouse를 Azure SQL Data Warehouse로 마이그레이션하고 이를 클라우드 기반 및 온-프레미스, 데이터 소스와 통합하는 것을 보다 효과적으로 계획하고 구현할 수 있습니다.

### 2.1. 솔루션 아키텍처

솔루션의 아키텍처입니다. 기존 온-프레미스 환경의 엔터프라이즈 Data Warehouse를 클라우드로 마이그레이션 합니다. 현재 Data Warehouse를 조사하여 비호환성을 식별하고 온-프레미스 Data Warehouse에서 데이터를 내보낸 다음 Azure Blob 스토리지로 전송합니다. 그런 다음 Polybase를 사용

하여 데이터를 웨어하우스에 로드 합니다. 마지막으로, ETL을 Azure Data Factory로 마이그레이션 함으로써 웨어하우스를 통합할 수 있습니다.



## 2.2. 대상 환경 정보

이 연습에서는 이 실습을 위한 소스 환경을 배포합니다. 원본 환경은 Azure SQL Data Warehouse로 마이그레이션 할 기존 사내 구축 환경을 나타내도록 설계되었습니다.

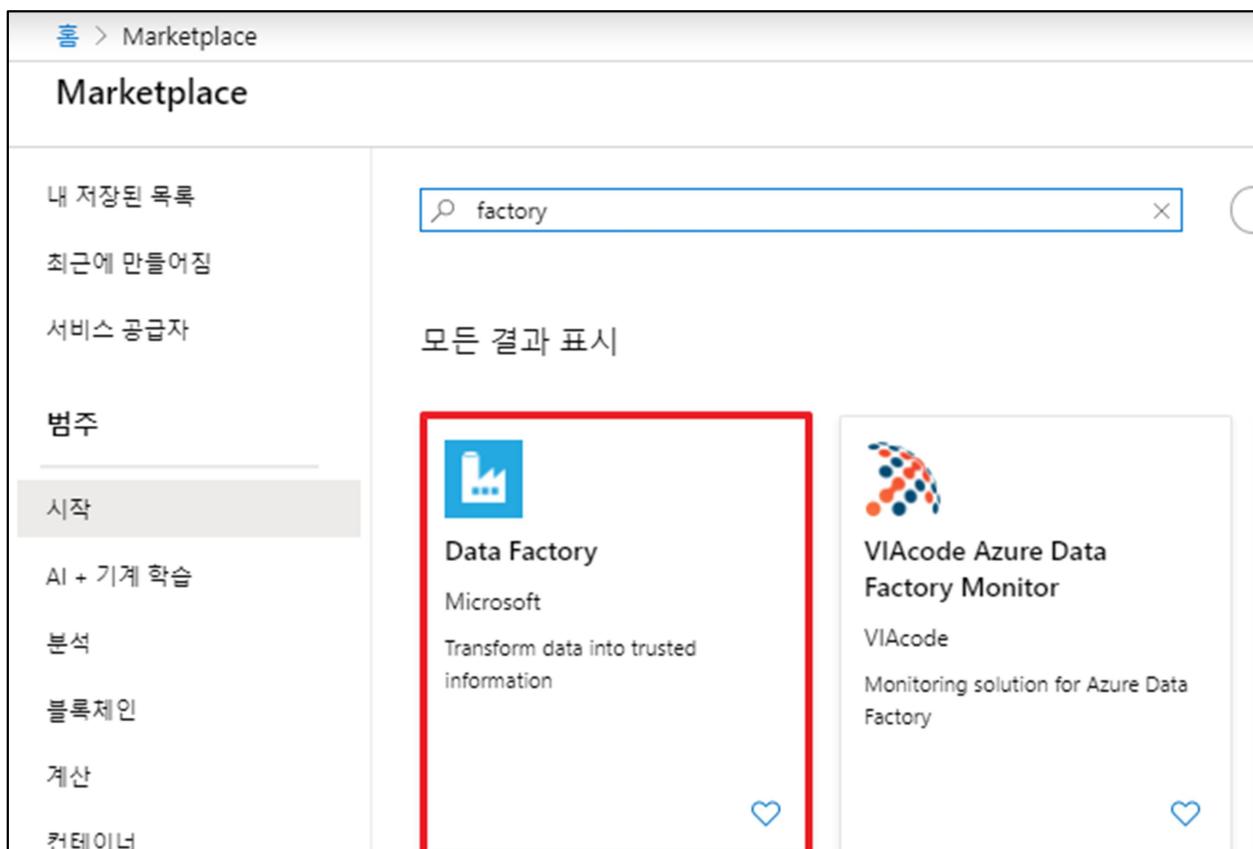
1. Azure Portal (<https://portal.azure.com>)에서 이 랩에 사용할 구독으로 로그인 했는지 확인합니다.
2. 배포 되는 환경은 MySQL DW 환경으로 VM 이미지로 제공 됩니다.

### 3. Exercise 1: Configure Azure Services

이 실습에서는 Azure Storage Account, Azure SQL Data Warehouse 및 Azure Data Factory V2를 생성하고 구성합니다. 이러한 서비스를 사용하여 기존 Data Warehouse 및 샘플 워크로드를 Azure로 마이그레이션 할 수 있습니다.

#### 3.1. Task 1: Create an Azure Data Factory v2

1. Azure 포탈로 이동하여 <https://portal.azure.com>에서 로그인합니다.
2. 리소스 만들기를 클릭하여 **Factory**를 입력, 검색하고 **Data Factory**를 선택합니다.



3. Data Factory 블레이드에서 만들기를 클릭합니다. 다음 정보를 입력하고 검토+만들기를 클릭한 후 만들기를 클릭합니다.
  - ✓ 이름: HolDf(Prefix) - Prefix는 사용자 별로 고유하게 입력.
  - ✓ 버전: V2
  - ✓ 구독: 본인의 구독 선택.
  - ✓ 리소스 그룹: 새로 만들기 > Hol-Day1-(Prefix) > Prefix는 사용자 별로 고유하게 각자 입력
  - ✓ 위치: 대한민국 중부(Korea Central)
  - ✓ Git사용 미 체크

홈 > 새로 만들기 > Marketplace > Data Factory > 새 data factory

### 새 data factory

---

이름 \*

 ✓

버전 ⓘ

구독 \*

리소스 그룹 \*

(신규) Hol-Day1-Prefix ✓

[새로 만들기](#)

위치 \*

GIT 사용 ⓘ



---

[만들기](#)

Data Factory 위치는 Data Factory의 메타데이터가 저장되고 파이프라인의 트리거가 시작되는 위치입니다. 한편, Data Factory는 다른 Azure 영역의 데이터 저장소 및 컴퓨팅 서비스에 액세스하여 데이터 저장소 간에 데이터를 이동하거나 컴퓨팅 서비스를 사용하여 데이터를 처리할 수 있습니다. 이 동작은 전 세계적으로 사용 가능한 통합 런타임을 통해 실현되어 데이터 컴플라이언스, 효율성 및 네트워크 송신 비용 절감을 보장합니다.

### 3.2. Task 2: Create an Azure SQL Data Warehouse

- 리소스 만들기를 선택하여 DW 입력 검색한 후 Azure Synapse Analytics를 선택합니다.

홈 > 새로 만들기 > Marketplace

Marketplace

내 저장된 목록

최근에 만들어짐

서비스 공급자

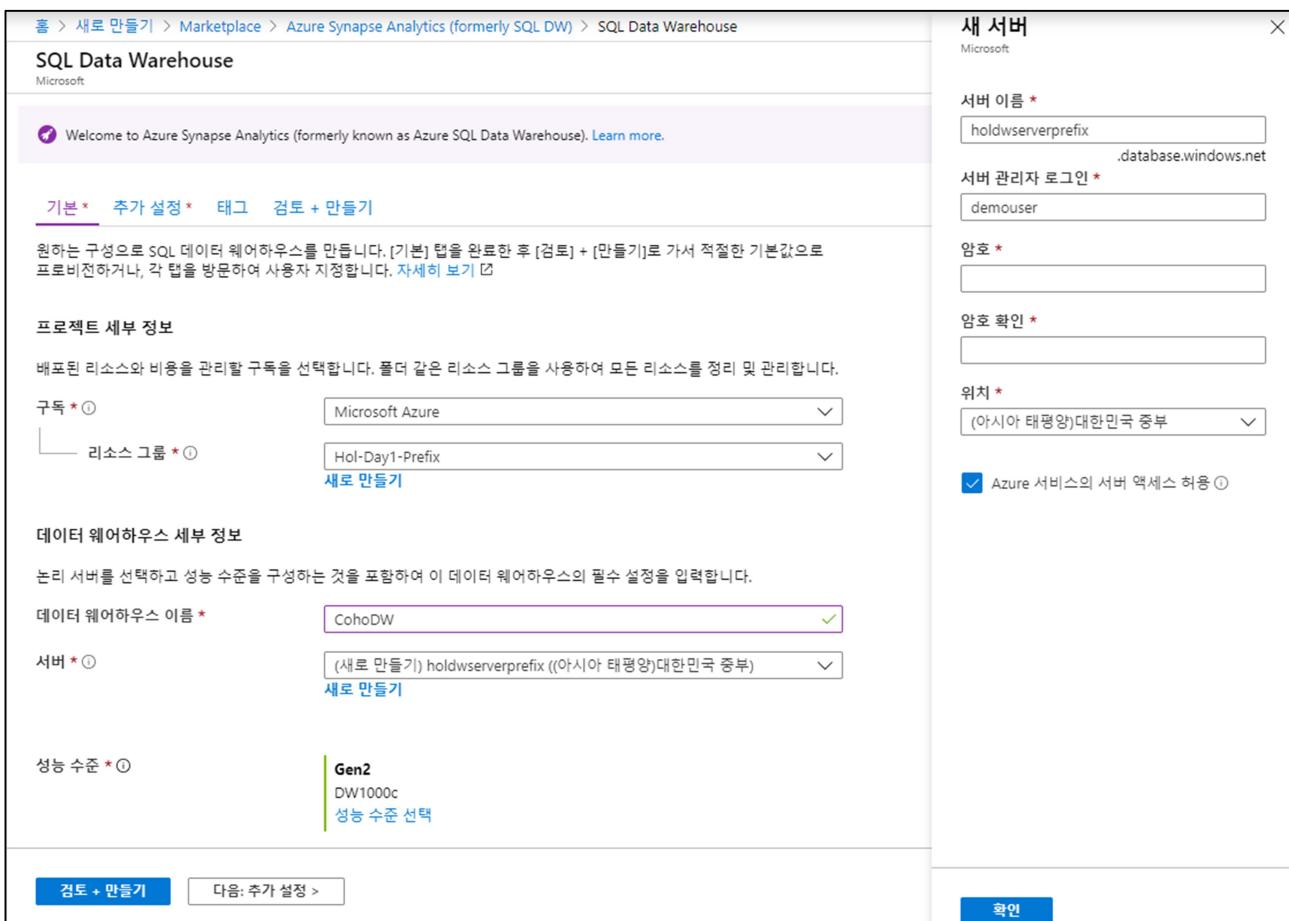
모든 결과 표시

검색창: DW

 <b>Azure Synapse Analytics</b> (formerly SQL DW) Microsoft Limitless analytics service with unmatched time to insight (Formerly SQL Data Warehouse)	 <b>Discovery Hub® with ADLS Gen2 and Azure SQL DW</b> Timextender Discovery Hub® with ADLS Gen2 and Azure SQL DW
--	---

범주: 시작

- AI + 기계 학습
- 분석
- 클록체인
- 계산
- 컨테이너

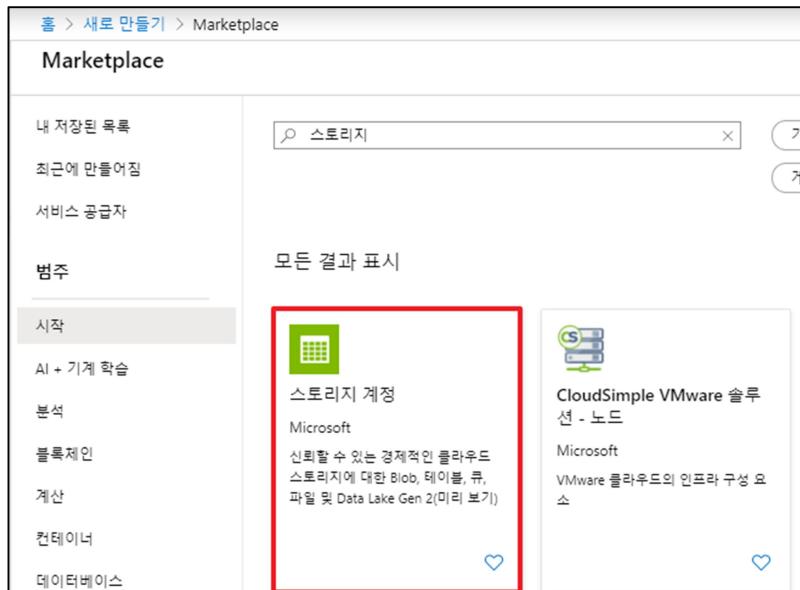
2. Azure Synapse Analytics 블레이드에서 만들기를 클릭하고 아래 내용을 입력합니다.
    - ✓ 구독: 본인의 구독 선택.
    - ✓ 리소스 그룹: 이전에 생성한 리소스 선택 -> Hol-Day1-(Prefix)
    - ✓ 데이터 웨어하우스 이름: CohoDW
  
  3. 서버에서 새로 만들기를 클릭하고 -> 아래 내용 입력 후 -> 확인
    - ✓ 서버 이름: holdwserver(prefix)
    - ✓ 서버 관리자 로그인: demouser
    - ✓ 암호: Demo@pass123
    - ✓ 위치: 대한민국 중부(Korea Central)
    - ✓ Azure 서비스의 서버 액세스 허용: 체크함
- 
4. 성능 수준 선택을 클릭하고 Gen2 탭을 선택한 다음 성능을 400cDWU로 설정하고 적용을 클릭합니다.



5. SQL Data Warehouse 블레이드에서 검토+만들기를 클릭한 후 만들기를 클릭합니다.

### 3.3. Task 3: Create an Azure Storage Account

1. 리소스 만들기 -> 스토리지를 검색하고-> 스토리지 계정을 선택합니다.



2. 저장소 계정 블레이드에서 만들기를 클릭합니다. 다음 정보를 지정하고 검토 + 작성성을 클릭합니다.

- ✓ 구독: 본인의 구독 선택
- ✓ 리소스 그룹: 이전에 생성한 리소스 선택 -> Hol-Day1-(Prefix)
- ✓ 스토리지 계정 이름: holstorage(prefix).
- ✓ 위치: 대한민국 중부(Korea Central).
- ✓ 성능: 표준

- ✓ 계정 종류: StorageV2 (범용 v2)
- ✓ Replication: LRS(로컬 중복 스토리지)
- ✓ 액세스 계층: 핫

홈 > 새로 만들기 > Marketplace > 스토리지 계정 > 스토리지 계정 만들기

### 스토리지 계정 만들기

[기본 사항](#) [네트워킹](#) [고급](#) [태그](#) [검토 + 만들기](#)

Azure Storage는 가용성, 보안, 내구성, 확장성 및 중복성이 뛰어난 클라우드 스토리지를 제공하는 Microsoft 관리 서비스입니다. Azure Blob(개체), Azure Data Lake Storage Gen2, Azure Files, Azure 큐 및 Azure 테이블을 포함합니다. 스토리지 계정의 비용은 사용량 및 아래에서 선택한 옵션에 따라 다릅니다. [자세한 정보](#)

**프로젝트 정보**

배포된 리소스와 비용을 관리할 구독을 선택합니다. 풀더 같은 리소스 그룹을 사용하여 모든 리소스를 정리 및 관리합니다.

구독 \* Microsoft Azure  
리소스 그룹 \* Hol-Day1-Prefix [새로 만들기](#)

**인스턴스 정보**

기본 배포 모델은 최신 Azure 기능을 지원하는 Resource Manager입니다. 대신 클래식 배포 모델을 사용하여 배포하도록 선택할 수 있습니다. [클래식 배포 모델 선택](#)

스토리지 계정 이름 \* holstorageprefix  
위치 \* (아시아 태평양) 대한민국 중부  
성능 표준  
계정 종류 StorageV2(범용 v2)  
복제 LRS(로컬 중복 스토리지)  
액세스 계층(기본값) 핫

[검토 + 만들기](#) < 이전 [다음: 네트워킹 >](#)

3. 검토+만들기를 클릭하고 탭에서 구성 선택 사항을 확인하고 만들기를 선택합니다.

4. 새 저장소 계정이 완성되면 저장소로 이동하여 컨테이너를 클릭합니다.

holstorageprefix

스토리지 계정

검색(Ctrl+ /) 리소스 그룹 (변경) 상태  
개요 활동 로그 기분: 사용 가능  
액세스 제어(IAM) 위치 대한민국 중부  
태그 구독 (변경) 구독 ID  
문제 진단 및 해결 구조화되지 않은 데이터를 위한 REST 기반의 개체 스토리지  
데이터 전송 이벤트  
Storage Explorer(미리 보기)  
설정 액세스 키 파일 공유  
지역에서 복제 서비스 SMB 및 NFS 파일 공유  
CORS 자세한 정보  
구성 큐  
암호화 테이블  
공유 액세스 서명 트래픽에 따라 효과적으로 업 확장  
방화벽 및 가상 네트워크 자세한 정보

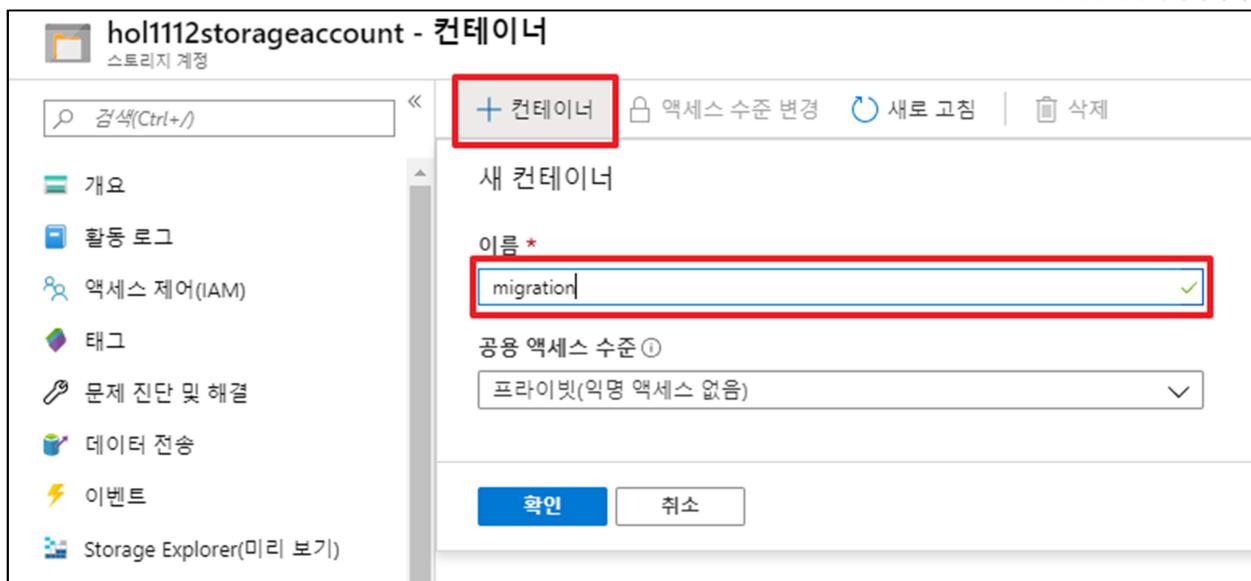
**컨테이너** 구조화되지 않은 데이터를 위한 REST 기반의 개체 스토리지 [자세한 정보](#)

**파일 공유** 서비스 SMB 및 NFS 파일 공유 [자세한 정보](#)

**테이블** 테이블 형식 데이터 스토리지 [자세한 정보](#)

**큐** 트래픽에 따라 효과적으로 업 확장 [자세한 정보](#)

- + 컨테이너 버튼을 클릭합니다.
- 새 컨테이너 블레이드에서 migration을 입력하고 OK를 클릭합니다.



### 3.4. Task 4: Prepare Environment and Create Migration Accounts

- 스토리지 계정에서 설정의 액세스 키를 클릭합니다.



- 스토리지 계정 이름과 key1의 키를 복사해서 텍스트 파일에 저장해둡니다.

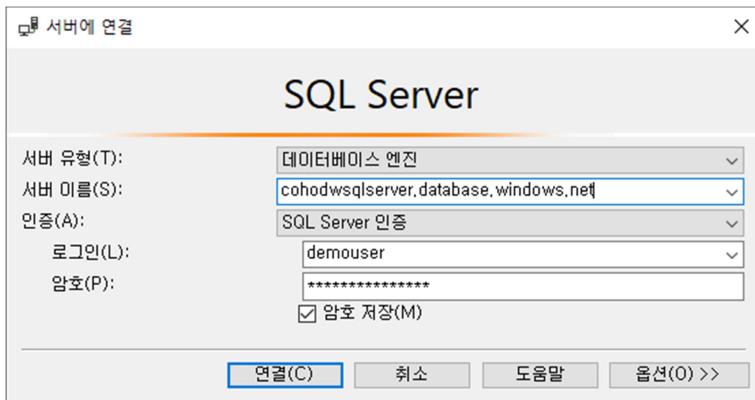
The screenshot shows the Azure Storage Explorer interface for a storage account named 'holstorageprefix'. The left sidebar has a tree view with 'Storage Explorer(미리 보기)' selected under '설정'. Under '액세스 키', there are two entries: 'key1' and 'key2'. Each entry has a 'Key' field containing a long hex string and a 'ConnectionString' field containing a URL. Red boxes highlight the 'ConnectionString' fields for both keys.

### 3. 리소스 그룹에서 CohoDW SQL Data Warehouse를 선택하고 서버 이름을 복사합니다.

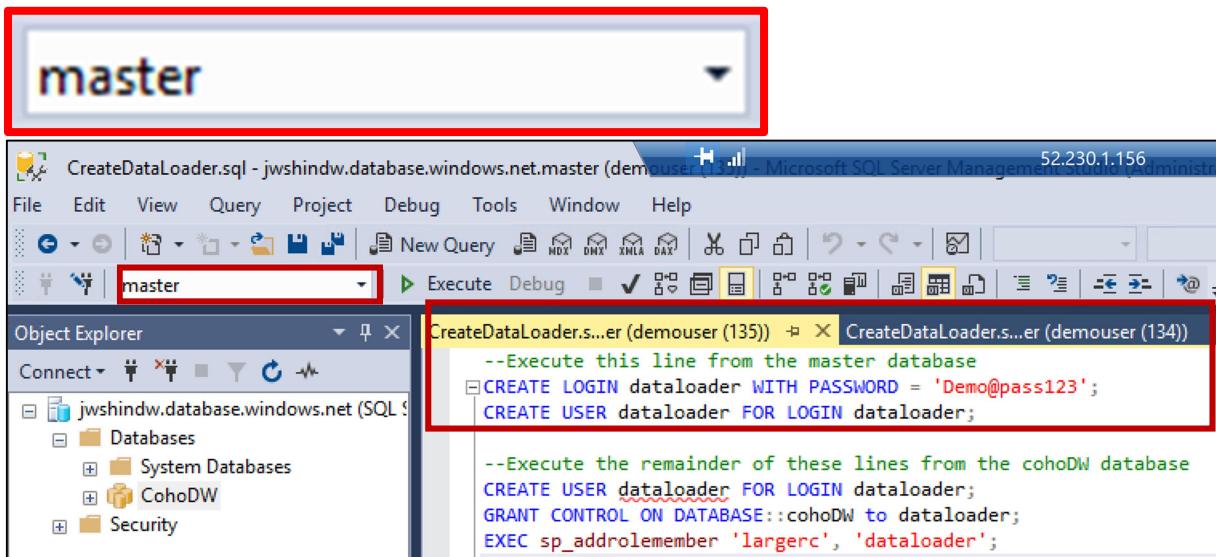
The screenshot shows the Azure portal resource details page for 'CohoDW(holdwserverprefix/CohoDW)'. The left sidebar shows 'Azure Synapse Analytics' under '리소스 그룹 (변경)'. The main pane displays various resource details, including '서버 이름' which is set to 'holdwserverprefix.database.windows.net'. A red box highlights this field.

4. 각자에게 할당된 VM에 접속합니다.
5. 다음으로 데이터 로드 작업을 수행할 수 있는 특별 계정을 생성하려고 합니다. 이 계정은 더 큰 리소스 클래스에 추가됩니다. 기본적으로 모든 계정은 처음에는 smallrc 리소스 클래스에 있습니다. 더 큰 리소스 클래스에 계정을 추가하면 쿼리를 실행하는 동안 계정이 더 많은 메모리를 사용할 수 있으므로 데이터 로드 및 유지 관리 태스크와 같은 작업에 더 효율적입니다.
6. C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\CreateDataLoader.sql 파일을 엽니다. 해당 파일은 SQL Server Management Studio의 sql 스크립트 파일입니다.
7. **SQL Server 인증과 demouser 계정과 암호를 사용하여 Azure SQL Data Warehouse에 연결합니다.**
  - ✓ 서버유형: 데이터베이스 엔진
  - ✓ 서버이름: Data Warehouse 서버 명
  - ✓ 인증: SQL Server 인증

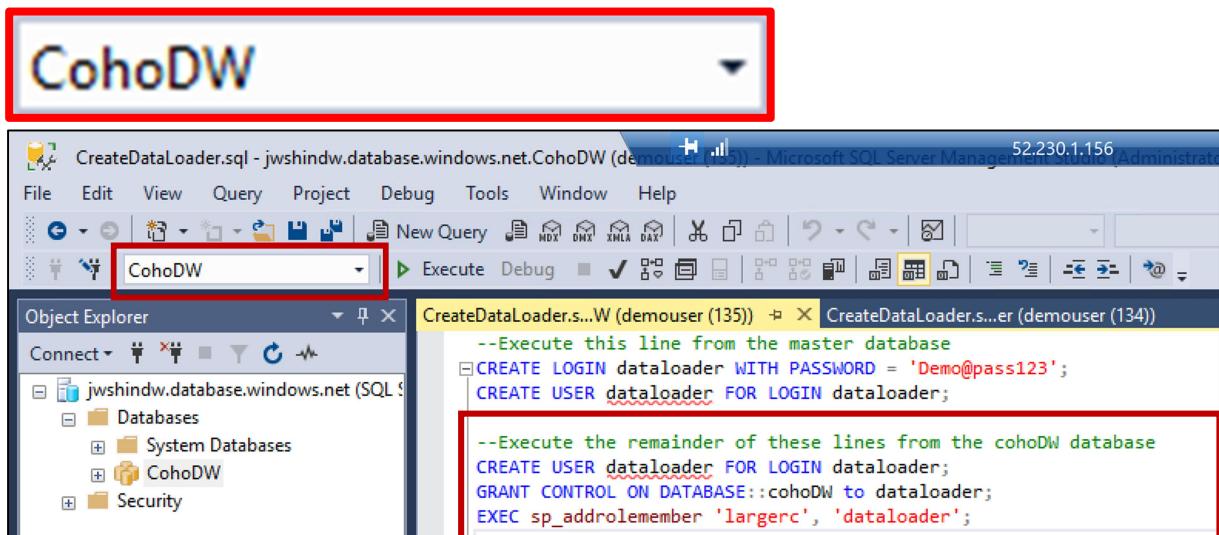
- ✓ 로그인: demouser
- ✓ 암호: Demo@pass123



8. 데이터베이스가 **Master**로 선택되어있는지 확인 후 처음 2라인을 실행하여 User를 생성합니다.



9. 데이터베이스를 **cohoDW**로 선택한 후 나머지 라인을 실행합니다.

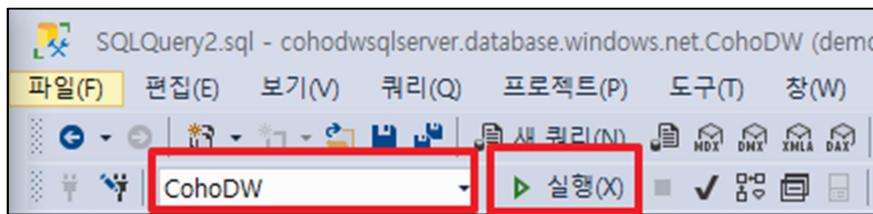


#### 4. Exercise 2: Data and schema preparation

이 실습에서는 Data Warehouse를 Azure SQL Data Warehouse로 마이그레이션을 하기 위하여 Azure SQL Data Warehouse에 테이블을 생성하고 원본 Data Warehouse에서 데이터를 추출합니다.

##### 4.1. Task 1: Prepare Azure SQL Data Warehouse and migrate schema

1. C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\CreateDWTables.sql 파일을 엽니다.
2. 데이터베이스 Master에 있다면 CohoDW로 변경하고 실행 버튼을 클릭하여 스크립트를 실행합니다. 그러면 기본 옵션을 사용하여 Clustered Columnstore Indexing 및 ROUNDROBIN 배포를 생성할 수 있습니다.



3. 다음 쿼리를 실행하여 테이블이 생성되었는지 확인합니다. 반환되는 행이 33개여야 합니다.

결과 메시지								
	name	object_id	principal...	schema_id	parent_object_id	type	type_desc	create_date
1	FactInternetSales	1986106116	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:29....
2	FactInternetSalesReas...	2002106173	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:29....
3	FactResellerSales	2018106230	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:29....
4	FactSalesQuota	2034106287	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:30....
5	FactSurveyResponse	2050106344	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:30....
6	ProspectiveBuyer	2066106401	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:31....
7	SalesByCategory	2082106458	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:31....
8	SalesByCustomer	2098106515	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:31....
9	SalesByDate	2114106572	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:32....
10	SalesByRegion	2130106629	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:32....
11	vDMPrep	2146106686	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:32....
12	vTargetMail	14623095	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:33....
13	AggregateSales	1650104919	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:21....
14	DatabaseLog	1666104976	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:21....
15	DimAccount	1682105033	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:22....
16	DimCurrency	1698105090	NULL	1	0	U	USER_TABLE	2019-11-13 09:49:22....

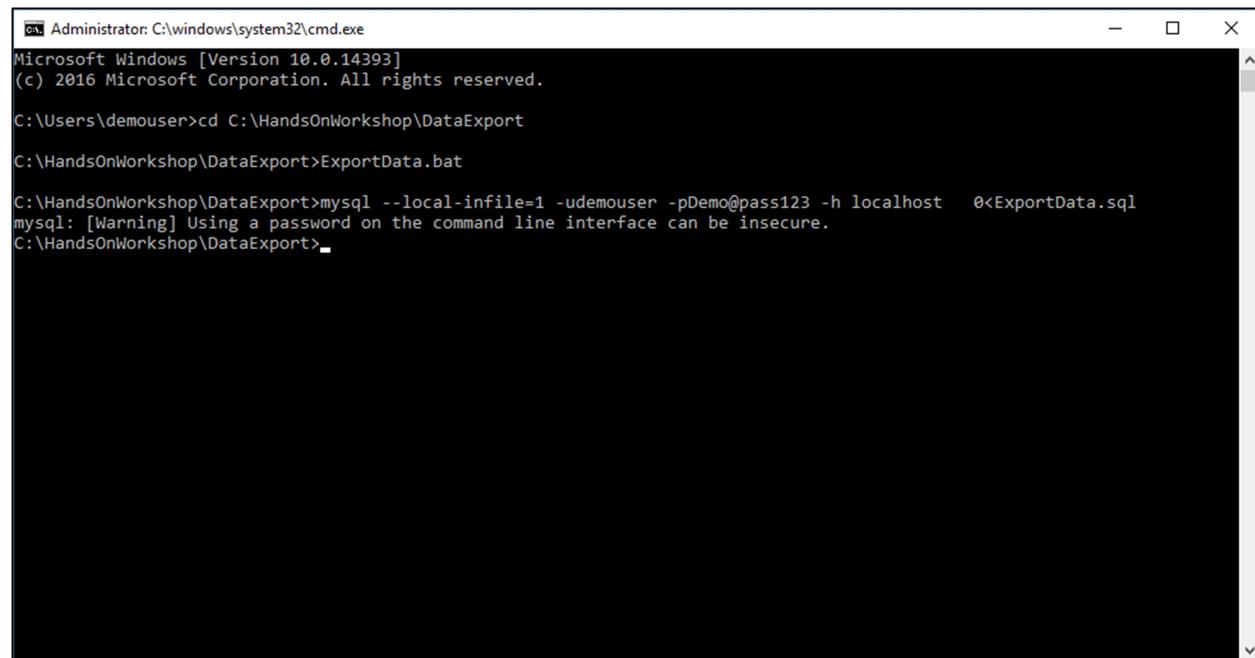
이 실습은 기존 Data Warehouse의 데이터를 SQL Data Warehouse로 마이그레이션 하는데 집중합니다. 데이터를 가져와서 Azure 저장소 계정에 업로드 합니다. 그런 다음 Polybase를 통해 데이터를 가져옵니다.

##### 4.2. Task 2: Data Exporting

1. 명령 프롬프트를 열고 C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\DataExport 폴더로 이동합니다.

2. **DataExport.bat** 파일을 실행합니다.

이 파일은 기존 MySQL DW의 데이터를 테이블 별로 txt파일로 Export 하는 파일입니다.



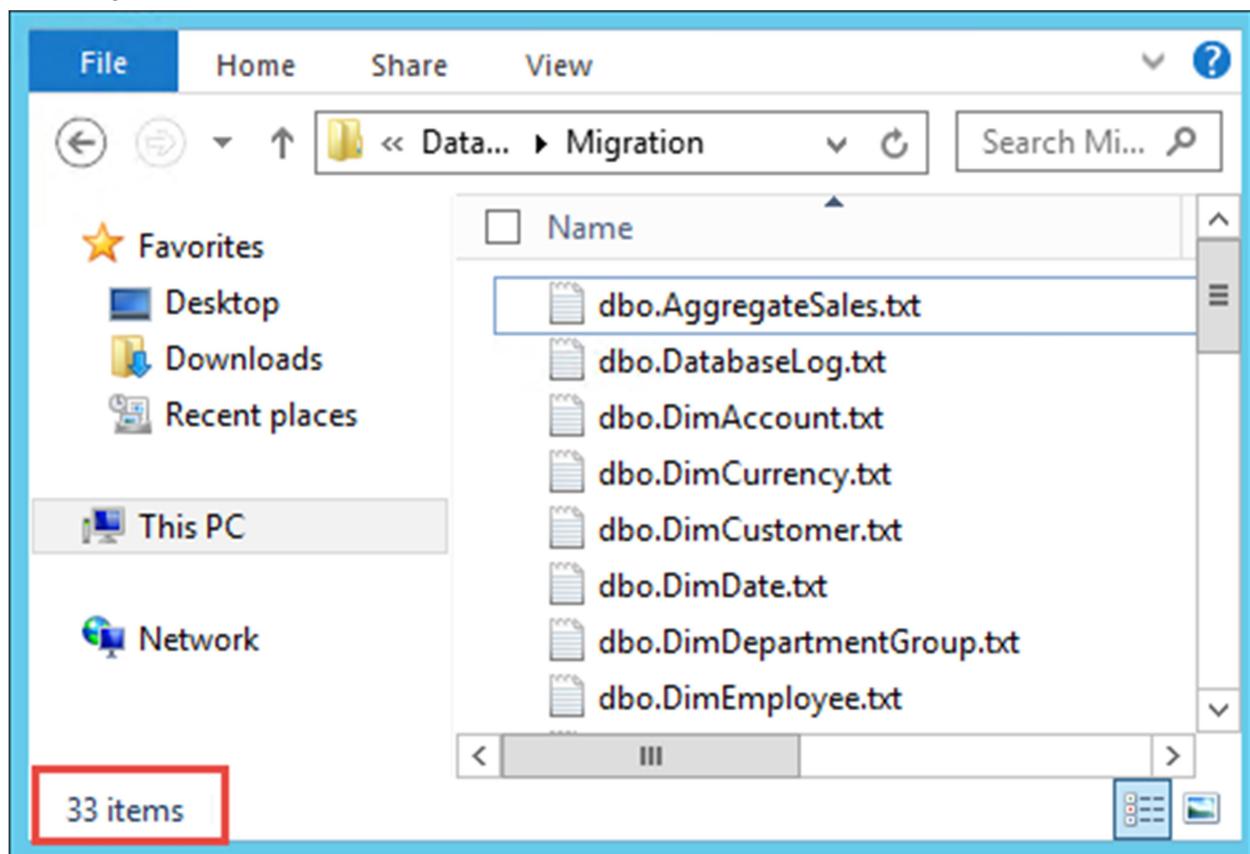
```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\demouser>cd C:\HandsOnWorkshop\DataExport

C:\HandsOnWorkshop\DataExport>ExportData.bat

C:\HandsOnWorkshop\DataExport>mysql --local-infile=1 -udemouser -pDemo@pass123 -h localhost <ExportData.sql
mysql: [Warning] Using a password on the command line interface can be insecure.
C:\HandsOnWorkshop\DataExport>
```

3. C:\Migration 폴더를 열어보면 33개의 파일이 생성된 것을 확인할 수 있습니다.



## 5. Exercise 3: Migrate the data to Azure SQL Data Warehouse

이 실습에서는 앞서 추출한 원본 Data Warehouse 데이터를 AzureCopy를 이용하여 Azure Storage에 업로드하고, 업로드 한 파일을 외부 테이블 형식으로 정의하여 읽어 들어 데이터를 이관합니다.

### 5.1. Task 1: Transfer your data to Azure

- 컴퓨터에서 <http://aka.ms/downloadazcopy> 경로로 연결하여 Microsoft Azure Storage 도구의 최신 버전을 다운로드 하여 설치합니다. 기본설정으로 진행하시면 됩니다.
- 명령 프롬프트를 열고 C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy 폴더로 이동합니다.  
이전에 저장한 스토리지 계정 이름과 키로 다음의 노란 부분의 명령을 업데이트하고 실행시켜 데이터 파일을 Azure로 복사하기 시작합니다 (모든 텍스트는 단일 명령입니다).

```
AzCopy /Source:"C:\Migration"  
/Dest:https://<YourStorageAccount>.blob.core.windows.net/migration  
/DestKey:<YourStorageAccountKey> /pattern:*.txt /NC:2
```

<예시>

```
AzCopy /Source:"C:\Migration"  
/Dest:https://jwshinstrg.blob.core.windows.net/migration  
/DestKey:DG6WPCGsFji1Fwv7+FGX20tiApjDpMunCAemgw6CArzbt8V0Mo/Cgx62YXGVJt85A4tsjTUn  
ufBfzY21lqgAXg== /pattern:*.txt /NC:2
```

```
C:\Windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.18362.418]  
(c) 2019 Microsoft Corporation. All rights reserved.  
C:\Users\rangho>cd C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy  
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>AzCopy /Source:"C:\Migration" /Dest:https://cohostorage.blob.core.windows.net/migration /DestKey:jlapa/Dem48bOP0HG18a96eurJEQmTMkEN4odCnwyX1GxPuFyMKHgIMF5+T801WoE+Ky69YaGkCGxBWm3x7x+w== /pattern:*.txt /NC:2
```

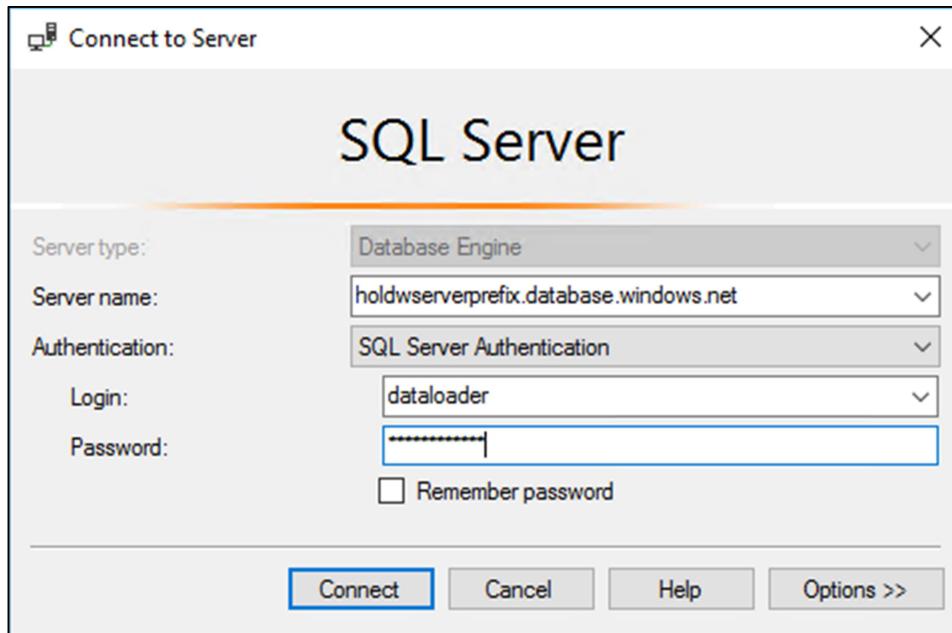
- 33 개의 파일이 모두 성공적으로 전송되었는지 확인합니다.

```
Administrator: C:\windows\system32\cmd.exe  
Microsoft Windows [Version 10.0.14393]  
(c) 2016 Microsoft Corporation. All rights reserved.  
C:\Users\demouser>cd C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy  
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>AzCopy /Source:"C:\Migration" /Dest:https://holstorageprefix.blob.core.windows.net/migration /DestKey:ZZb7JWutvKQhyt9wT0DsP0azjMh0YG5e/QFqBSg3WrdbSzvG2mnHIvQ7sIE8+8UGuXNBMF/LKc+vqWAR50uZg== /pattern:*.txt /NC:2  
Finished 33 of total 33 file(s).  
[2019/11/17 06:54:17] Transfer summary:  
-----  
Total files transferred: 33  
Transfer successfully: 33  
Transfer skipped: 0  
Transfer failed: 0  
Elapsed time: 00.00:00:05  
C:\Program Files (x86)\Microsoft SDKs\Azure\AzCopy>
```

4. Azure Portal에서 저장소 계정으로 이동 후 컨테이너를 선택하여 Export한 파일들이 migration 컨테이너에 있는지 확인합니다.

이름	수정한 날짜	액세스 ...	Blob 유형	크기	임대 단계
dbo.AggregateSales.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	2.08 MiB	사용 가능
dbo.DatabaseLog.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	37.39 KiB	사용 가능
dbo.DimAccount.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	5.41 KiB	사용 가능
dbo.DimCurrency.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	2.05 KiB	사용 가능
dbo.DimCustomer.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	4.26 MiB	사용 가능
dbo.DimDate.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	112.38 KiB	사용 가능
dbo.DimDepartmentGroup.txt	2019. 11. 13. 오후 7:3...	핫(유주)	블록 Blob	172 B	사용 가능

5. SQL Server Management Studio를 열고 위에서 만들었던 dataloader 계정을 사용하여 SQL Data Warehouse의 CohoDW 데이터베이스에 연결합니다.



6. 스토리지 이름과 Key부분을 변경한 후 다음을 실행하여 액세스 저장소 키를 마이그레이션 저장소 계정에 저장하는데 사용할 데이터베이스 자격 증명을 만듭니다. 이전에 입력한 암호가 올바른지 확인합니다.

```

CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Demo@pass123';

CREATE DATABASE SCOPED CREDENTIAL MigrationCredential
WITH IDENTITY = '<YourStorageAccountName>', SECRET = '<YourStorageAccountKey>'
  
```

```

SQLQuery8.sql - jws...(dataloader (159))* ⇨ X SQLQuery7.sql - jw...W (demouser (324))*      SQLQuery6.sql - jw...W (demouser (144))*
CREATE MASTER KEY ENCRYPTION BY PASSWORD = 'Demo@pass123';

CREATE DATABASE SCOPED CREDENTIAL MigrationCredential
WITH IDENTITY = 'jwshinstrg', SECRET = 'DG6WPCGsFji1Fvw7+FGX20tiApjDpMunCAemgw6CArzbt8V0Mo/Cgx62YXGVJt'

```

7. 다음 쿼리를 실행하여 외부 데이터 원본을 만듭니다. 외부 데이터 소스는 데이터의 위치와 데이터에 액세스하는데 사용되는 권한 정보를 정의합니다. 값을 자신의 스토리지 이름과 키로 바꾸십시오.

```

CREATE EXTERNAL DATA SOURCE MigrationStor WITH (TYPE = HADOOP,
LOCATION=
'wasbs://migration@<YourStorageAccountName>.blob.core.windows.net',
CREDENTIAL = MigrationCredential);

```

```

SQLQuery8.sql - jws...(dataloader (159))* ⇨ X SQLQuery7.sql - jw...W (demouser (324))**
CREATE EXTERNAL DATA SOURCE MigrationStor WITH (TYPE = HADOOP,
LOCATION=
'wasbs://migration@jwshinstra.blob.core.windows.net',
CREDENTIAL = MigrationCredential);

```

8. 다음 쿼리를 실행하여 외부 파일 형식을 만듭니다. 외부 파일 형식은 외부 저장소 및 해당 레이아웃을 정의합니다.

```

CREATE EXTERNAL FILE FORMAT MigrationFiles WITH(FORMAT_TYPE = DelimitedText,
FORMAT_OPTIONS (FIELD_TERMINATOR = '|'));

```

9. SQL Server Management Studio에서 C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles>CreateExternalTables.sql 파일을 열고 Azure SQL Data Warehouse CohoDW 데이터베이스에 연결되어 있는지 확인합니다.
10. 이 파일은 테이블에 대한 모든 외부 테이블 정의를 포함하고 위에서 작성한 외부 데이터 소스 및 외부 파일 형식을 직접 활용합니다. 실행을 클릭하여 외부 테이블을 작성합니다.
11. 다음 코드를 실행하여 33 개의 테이블이 만들어 졌는지 확인합니다.

```

SELECT * FROM SYS.TABLES WHERE is_external = 1

```

12. SQL Server Management Studio에서 C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\LoadData.sql 파일을 엽니다.

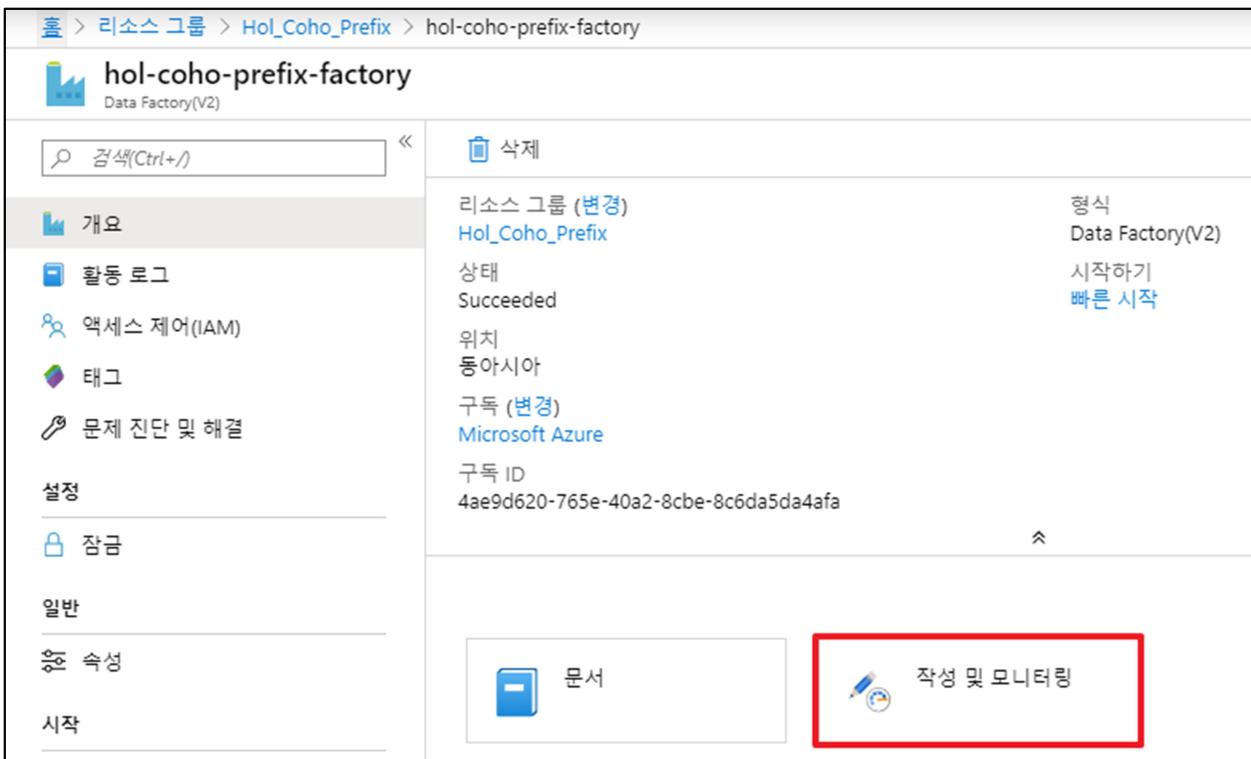
13. 이 파일의 명령은 이전 단계에서 정의한 외부 테이블을 통해 Azure 저장소에 저장된 데이터 파일에서 직접 추출된 데이터를 삽입합니다. 실행을 클릭하여 데이터로드를 시작합니다.
14. 데이터가 업로드 된 후, 테이블의 데이터를 선택하여 성공 여부를 확인할 수 있습니다. 프로덕션 환경에서는 훨씬 더 철저한 데이터 유효성 검사 프로세스를 수행하게 됩니다.

## 6. Exercise 4: Migrate to Data Factory v2

이 실습에서는 Azure Data Factory에서의 Copy data와 Pipeline 기능을 이용하여 특정 테이블의 데이터를 이관해보고 이후 원본 테이블에서 추가된 데이터를 주기적으로 가져오도록 작업합니다.

### 6.1. Task 1: Copy data

1. 이관대상 테이블을 생성하기 위해 SQL Server Management Studio에서 C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\CreateTransactionTables.sql를 엽니다
2. 데이터베이스 컨텍스트가 Master에 있다면 CohoDW로 변경하고 실행 버튼을 클릭하여 스크립트를 실행합니다
3. Azure 포털로 이동합니다. Azure 포털에서 Hol-Day1-(Prefix) 리소스 그룹으로 이동하여 이전에 만들어놓은 Azure Data Factory를 열고 작성 및 모니터링 타일을 클릭합니다.



The screenshot shows the Azure Data Factory V2 blade. On the left, there is a sidebar with the following items:

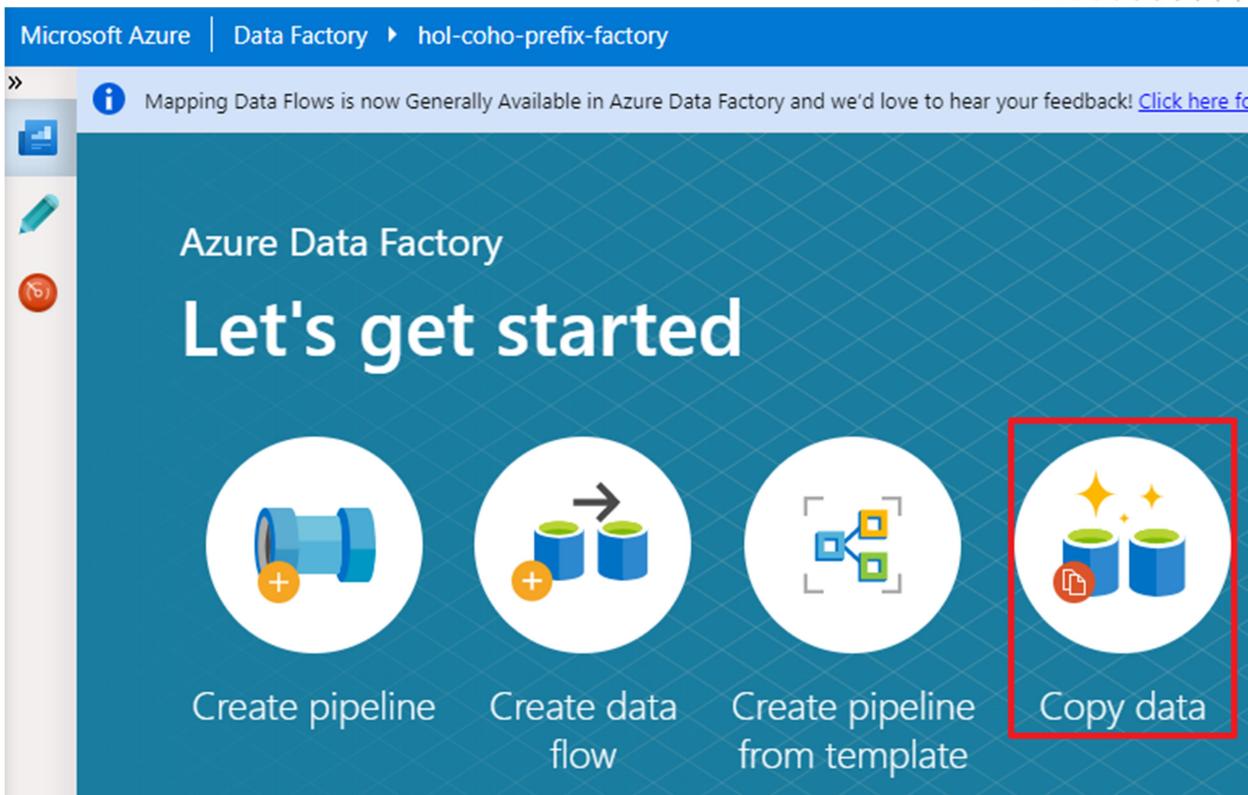
- 검색(Ctrl+ /)
- 개요 (selected)
- 활동 로그
- 액세스 제어(IAM)
- 태그
- 문제 진단 및 해결
- 설정
- 잠금
- 일반
- 속성
- 시작

The main content area displays the following details for the 'hol-coho-prefix-factory' Data Factory (V2):

리소스 그룹 (변경) Hol_Coho_Prefix	형식 Data Factory(V2)
상태 Succeeded	시작하기 빠른 시작
위치 동아시아	
구독 (변경) Microsoft Azure	
구독 ID 4ae9d620-765e-40a2-8cbe-8c6da5da4afa	

At the bottom right, there are two tiles: '문서' (Document) and '작성 및 모니터링' (Create and Monitor), with the latter being highlighted by a red box.

4. Data Factory 포털 홈 화면에서 Copy data를 선택합니다.

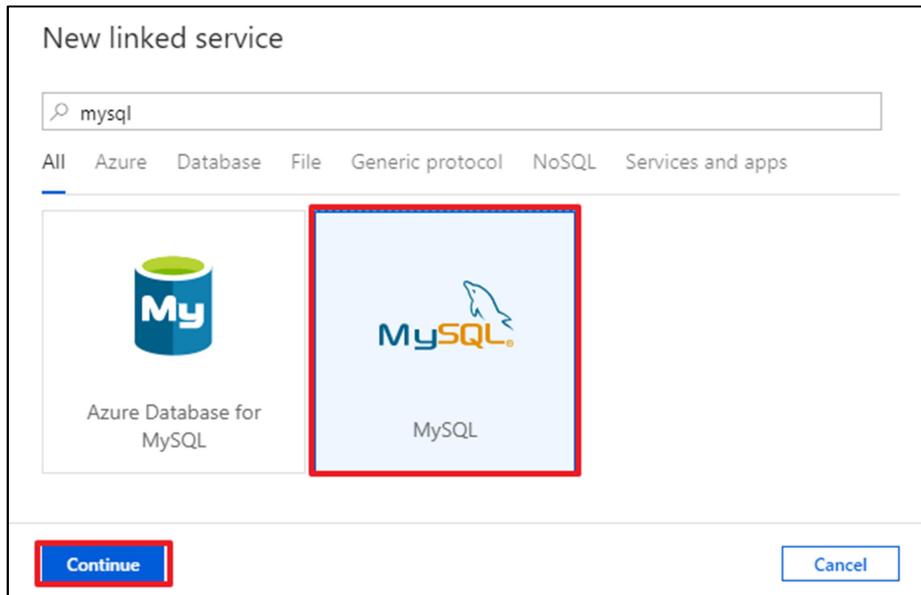


5. 아래 정보를 입력한 후 **Next**를 클릭합니다.

- ✓ Task name : CopyMySQL(임의입력)
- ✓ Task cadence or task schedule: Run once now

The screenshot shows the 'Properties' step of the 'Copy data' wizard. The left sidebar lists steps 1 through 6. Step 2, 'Source', is selected and expanded, showing 'Connection' (selected) and 'Dataset'. Step 3, 'Destination', is also expanded, showing 'Connection' (selected) and 'Dataset'. The main panel displays the 'Properties' configuration screen. The 'Task name' field contains 'CopyMySQL' and is highlighted with a red box. The 'Task cadence or task schedule' section shows two radio buttons: 'Run once now' (selected) and 'Run regularly on schedule'. At the bottom, there are 'Previous' and 'Next' buttons, with 'Next' being highlighted with a red box.

6. Destsource data store 화면에서 **+ Create new connection** 을 클릭한 후 MySQL을 선택한 후 Continue 버튼을 클릭합니다..



7. 아래 내용을 입력한 후 **Test connection**을 클릭하여 연결을 확인한 후 **Create** 버튼을 클릭합니다.
- ✓ Name: MySQL(임의입력)
  - ✓ Server name: 지급받은 VM의 IP
  - ✓ Port: 3306
  - ✓ Database name: cohoOLTP
  - ✓ User name: demouser
  - ✓ Password: Demo@pass123

## New linked service (MySQL)

Name \*

MySqlserver

Description

Connect via integration runtime \*

AutoResolveIntegrationRuntime



Connection string

Azure Key Vault

Server name \*

52.141.20.191

Port

3306

Database name \*

cohoOLTP

User name \*

mysql

Password

Azure Key Vault

Password \*

.....

SSL mode

Preferred



Connection successful

Create

Back

Test connection

Cancel

8. 만들어진 MySQL을 선택한 후 **Next**를 클릭합니다.

## Source data store

Specify the source data store for the copy task. You can use an existing data store connection or specify a new data store.

All Azure Database File Generic protocol NoSQL Services and apps

All Filter by name + Create new connection

MySqlserver

Previous Next

9. CardTransaction 테이블을 체크한 후 Next 버튼을 클릭합니다.

Copy data

Properties One time copy

Source Connection Dataset

Destination Connection Dataset

Settings

Summary

Deployment

Select tables from which to copy the data or use a custom query.

You can select multiple tables, or you can provide single custom query.

EXISTING TABLES USE QUERY

Filter by name... Refresh

Select all

`Address`

`BuildVersion`

`CardTransaction`

`Customer`

`CustomerAddress`

Showing 16 out of 16 items (1 selected)

transactionID accountID transactionAmountUSD transactionAmount transactionCurrencyCode localHour

47F3C9CC- 2948-4069- B7BC- 260361539AAD	A914800993377925	86	86	USD	9
A949C89R-					

Previous Next

10. Destination data store 화면에서 + Create new connection 을 클릭한 후 Azure Synapse Analytics (formerly SQL DW)을 선택한 후 Continue 버튼을 클릭합니다.

## New linked service

DW

All Azure Database File Generic protocol Services and apps

Azure Synapse Analytics (formerly SQL DW)

Continue Cancel

11. 아래 내용을 입력한 후 **Test connection**을 클릭하여 연결을 확인한 후 **Create** 버튼을 클릭합니다..

- ✓ Name: **AzureSynapseAnalytics**(임의 입력)
- ✓ Azure subscription: **본인의 구독**
- ✓ Server name: **Azure SQL Data Warehouse** 서버 명
- ✓ Database name: **cohodW**
- ✓ Authentication type: **SQL authrntication**
- ✓ User name: **demouser**
- ✓ Password: **Demo@pass123**

## New linked service (Azure Synapse Analytics (formerly SQL...)

Name \*

Description

Connect via integration runtime \*

Connection string

Azure Key Vault

Account selection method

From Azure subscription

Enter manually

Azure subscription

Server name \*

Database name \*



Authentication type \*



User name \*

Password

Azure Key Vault

Password \*

Connection successful

Create

Back

Test connection

Cancel

12. 만들어진 Destination을 선택한 후 Next를 클릭합니다.

## Destination data store

Specify the destination data store for the copy task. You can use an existing data store connection or specify a new data store.

All

Azure

Database

File

Generic protocol

NoSQL

Services and apps

All

Filter by name

+ Create new connection

AzureSynapseAnalytics

Previous

Next

13. Table mapping에서 **dbo.CardTransaction**을 선택한 후 **Next** 버튼을 클릭한다.

## Table mapping

For each table you have selected to copy in the source data store, select a corresponding table in the destination data store or specify the stored procedure to run at the destination.

Source

Destination

'CardTransaction'

→ **dbo.CardTransaction**

Skip column mapping for all tables

Previous

Next

14. Column mapping 내용을 확인한 후 **Next** 버튼을 클릭합니다.

**Column mapping**

Choose how source and destination columns are mapped

Table mappings (1)		Column mappings	
Source	Destination	Source	Destination
'CardTransaction'	dbo.CardTransaction	transactionID (String)	transactionID (nvarchar)
		accountID (String)	accountID (nvarchar)
		transactionAmountUS...	transactionAmountUS...
		transactionAmount (D...)	transactionAmount (fl...)
		transactionCurrencyCo...	transactionCurrencyCo...
		localHour (String)	localHour (nvarchar)
		transactionIPaddress (...)	transactionIPaddress (...)
		ipState (String)	ipState (nvarchar)

Azure Synapse Analytics (formerly SQL DW) sink properties

Pre-copy script

Write batch timeout

**Previous** **Next**

15. Staging account linked service의 **+ New**를 선택합니다..

## Settings

More options for data movement

**Fault tolerance settings**

Fault tolerance Abort activity on first incompatible row

**Performance settings**

Enable staging

**Staging settings**

Staging account linked service Select... **+ New**

16. 다음 내용을 입력, **Test connection**을 클릭하여 연결을 확인한 후 **Create** 버튼을 클릭합니다.

- ✓ Name: **AzureBlobStorage**(임의입력)
- ✓ Account selection method: **From Azure subscription**

- ✓ Azure subscription: 본인의 구독
- ✓ Storage account name: 스토리지 계정 명

**New linked service**

Name \* AzureBlobStorage

Description

Connect via integration runtime \* AutoResolveIntegrationRuntime

Authentication method

Account key

Connection string	Azure Key Vault
-------------------	-----------------

Account selection method

From Azure subscription       Enter manually

Azure subscription

Microsoft Azure (4ae9d620-765e-40a2-8cbe-8c6da5da4afa)

Storage account name \* cohostorage

Additional connection properties

+ New

Test connection

To linked service       To file path

If the identity you use to access the data store only has permission to subdirectory instead of the entire account, specify the path to test connection. Please make sure your self-hosted integration runtime is higher than version 4.0 if connecting via self-hosted integration runtime.

Annotations

✓ Connection successful

Test connection      Cancel

17. **Next** 버튼을 클릭합니다.

**Settings**

More options for data movement

◀ Fault tolerance settings

Fault tolerance Abort activity on first incompatible row ⓘ

◀ Performance settings

Enable staging  ⓘ

◀ Staging settings

Staging account linked service  ⓘ Test connection ⌂ Edit + New

Storage Path  ⓘ

Enable Compression  ⓘ

◀ Advanced settings

Allow PolyBase  ⓘ

Reject type Value ⓘ

Reject value 0

Use type default

Data integration unit Auto ⓘ

You will be charged # of used DIUs \* copy duration \* \$0.25/DIU-hour. Local currency and separate discounting may apply per subscription type. [Learn more](#)

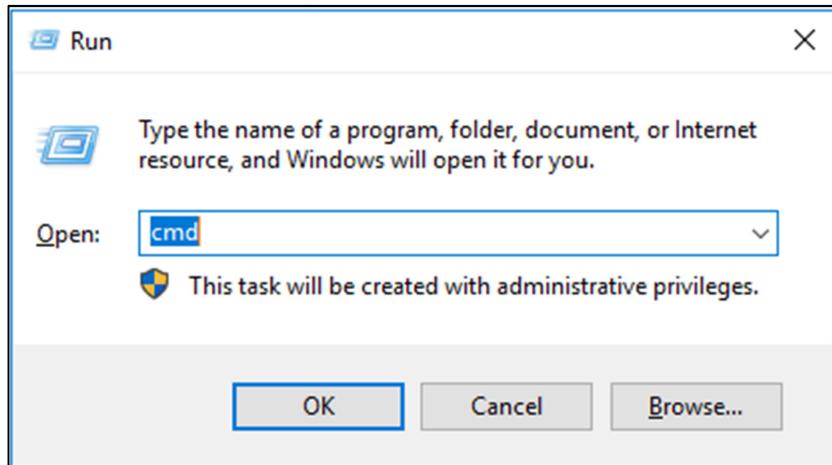
⌂

18. Summary 내용을 확인한 후 **Next** 버튼을 클릭합니다.
19. 진행상황을 확인한 후 완료되면 **Finish** 버튼을 클릭하여 이관 작업을 마무리 합니다.

The diagram illustrates the data flow: MySQL → Azure Blob Storage → Azure Synapse Analytics (formerly SQL DW). Below the diagram, the text "Deployment complete" is displayed, followed by a list of successful steps: "Validate copy runtime environment", "Creating datasets", "Creating pipelines", and "Running pipelines". A note below states: "Datasets and pipelines have been created. You can now monitor and edit the copy pipelines or click finish to close the copy wizard." At the bottom, there are "Edit pipeline" and "Monitor" buttons, and a "Finish" button highlighted with a red box.

## 6.2. Task 2: Add new data

- 데이터 이관을 마친 원본 테이블에 추가 데이터를 등록하기 위해 다음을 진행합니다.
- 명령 프롬프트를 실행시킵니다.



- mysql -u demouser -p 입력한 후 패스워드는 Demo@pass123을 입력합니다.
- SET GLOBAL local\_infile = 1; 을 실행합니다.
- show variables like '%infile%'; 을 실행하여 local\_infile 값이 ON 되어있는지 확인합니다.
- exit를 입력하여 빠져나옵니다.

```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\demouser>mysql -u demouser -p
Enter password: *****
Welcome to the MySQL monitor. Commands end with ; or \g.
Your MySQL connection id is 18
Server version: 8.0.18 MySQL Community Server - GPL

Copyright (c) 2000, 2019, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show variables like '%infile%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile | OFF   |
+-----+-----+
1 row in set (0.01 sec)

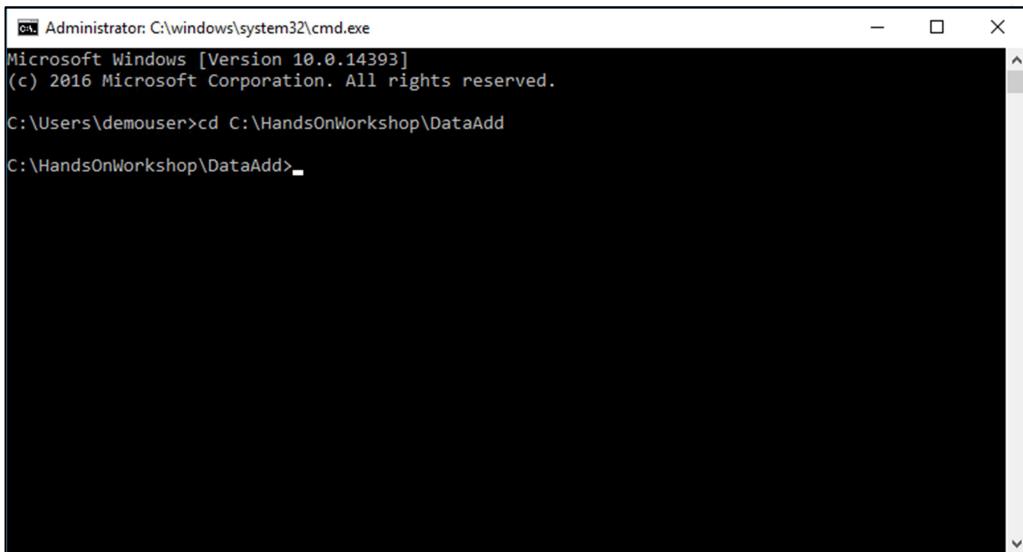
mysql> SET GLOBAL local_infile = 1;
Query OK, 0 rows affected (0.00 sec)

mysql> show variables like '%infile%';
+-----+-----+
| Variable_name | Value |
+-----+-----+
| local_infile | ON    |
+-----+-----+
1 row in set (0.00 sec)

mysql> exit
Bye

C:\Users\demouser>
```

7. C:\CloudScaleAnalytics-master\Hands on Lab\Day1\LabFiles\DataAdd 경로로 이동합니다.

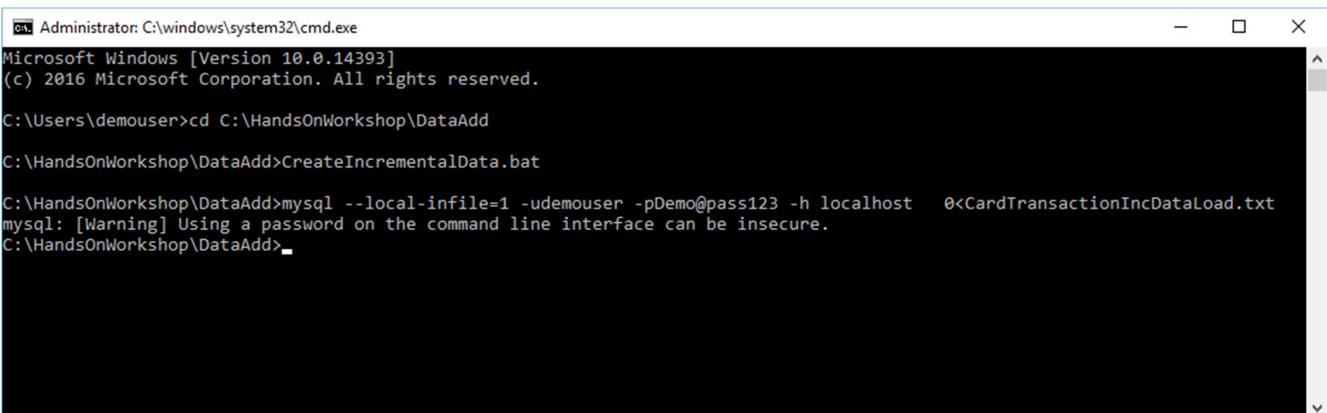


```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\demouser>cd C:\HandsOnWorkshop\DataAdd

C:\HandsOnWorkshop\DataAdd>
```

8. CreateIncrementalData.bat 파일을 실행시킵니다.



```
Administrator: C:\windows\system32\cmd.exe
Microsoft Windows [Version 10.0.14393]
(c) 2016 Microsoft Corporation. All rights reserved.

C:\Users\demouser>cd C:\HandsOnWorkshop\DataAdd

C:\HandsOnWorkshop\DataAdd>CreateIncrementalData.bat

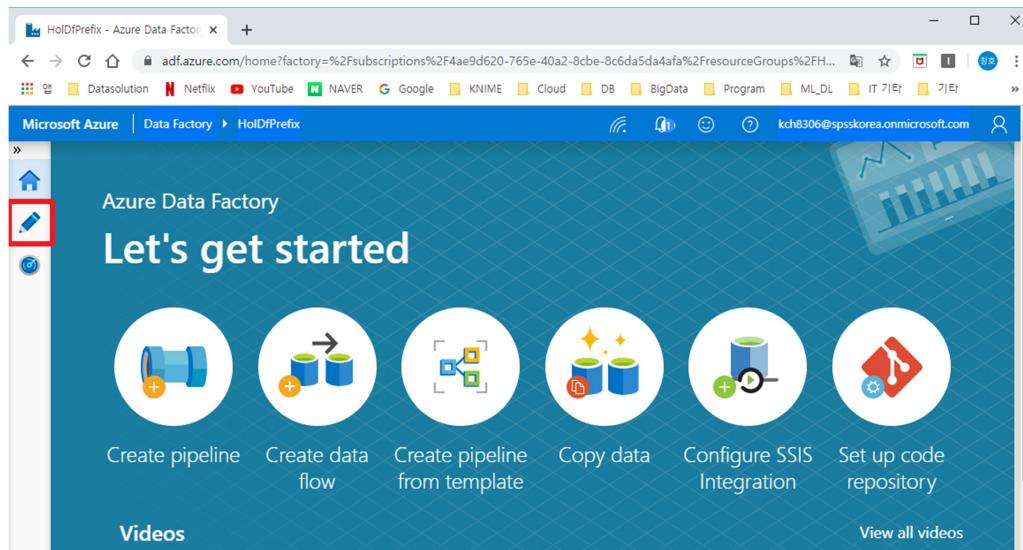
C:\HandsOnWorkshop\DataAdd>mysql --local-infile=1 -udemouser -pDemo@pass123 -h localhost <CardTransactionIncDataLoad.txt
mysql: [Warning] Using a password on the command line interface can be insecure.

C:\HandsOnWorkshop>DataAdd>
```

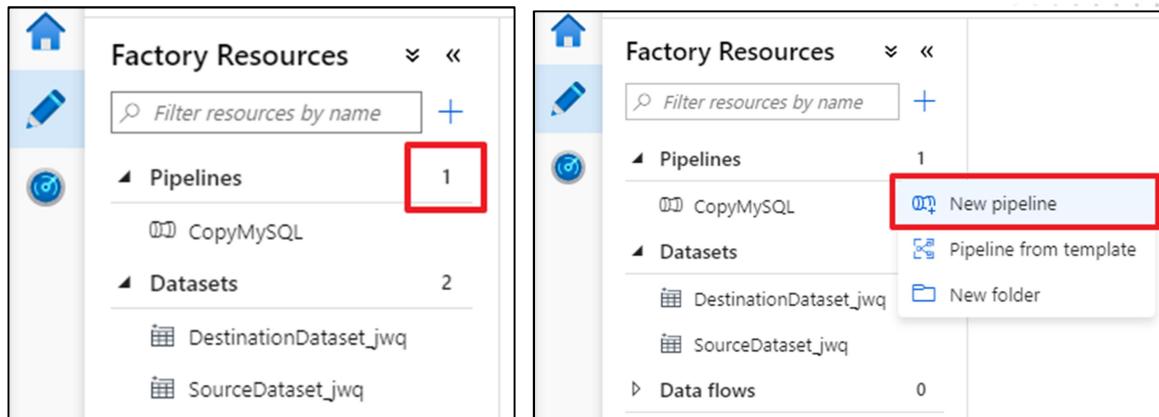
9. 위 작업이 완료되면 원본 MySQL의 CardTransaction 테이블에 추가로 데이터가 쌓이게 됩니다.

### 6.3. Task 3: Create pipeline

1. Data Factory 포털 홈 화면에서 우측의 연필모양의 아이콘을 클릭합니다.



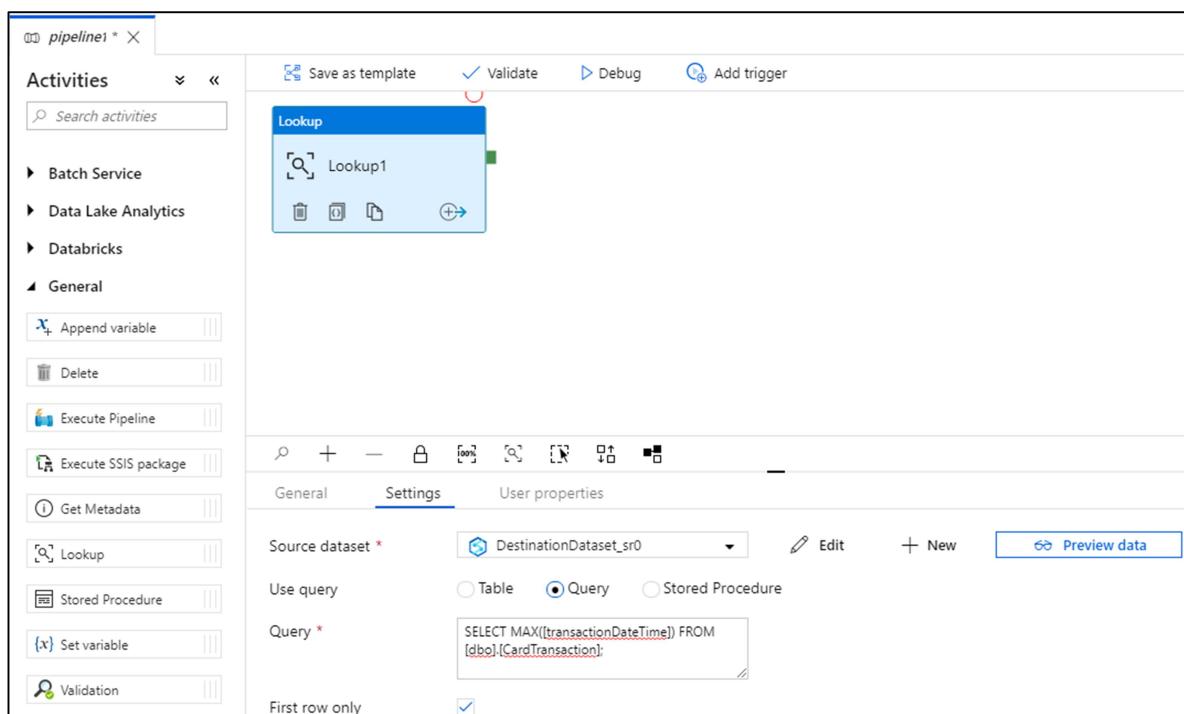
2. Pipeline의 오른쪽의 숫자에 마우스를 가져가서 클릭 후 New pipeline을 선택합니다.



3. Activities > General > Lookup을 드래그해서 옮긴 후 하단 Setting탭에서 다음을 입력합니다.

- ✓ Source dataset: DestinationDataset\_sr0
- ✓ Use query: Query
- ✓ Query: 

```
SELECT MAX([transactionDateTime]) AS transactionDateTime FROM [dbo].[CardTransaction];
```
- ✓ First row only: Checked

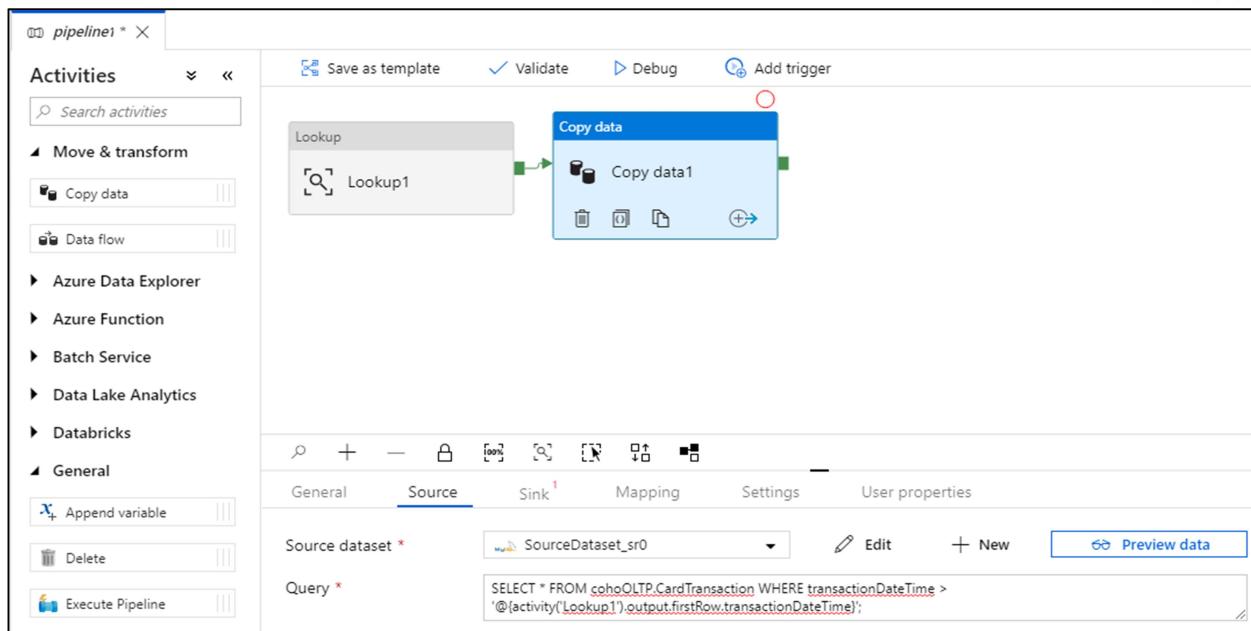


4. Activities > Move & transform > Copy data를 드래그해서 옮겨 Lookup과 Copy data를 연결한 후

Source 탭에서 다음을 입력합니다.

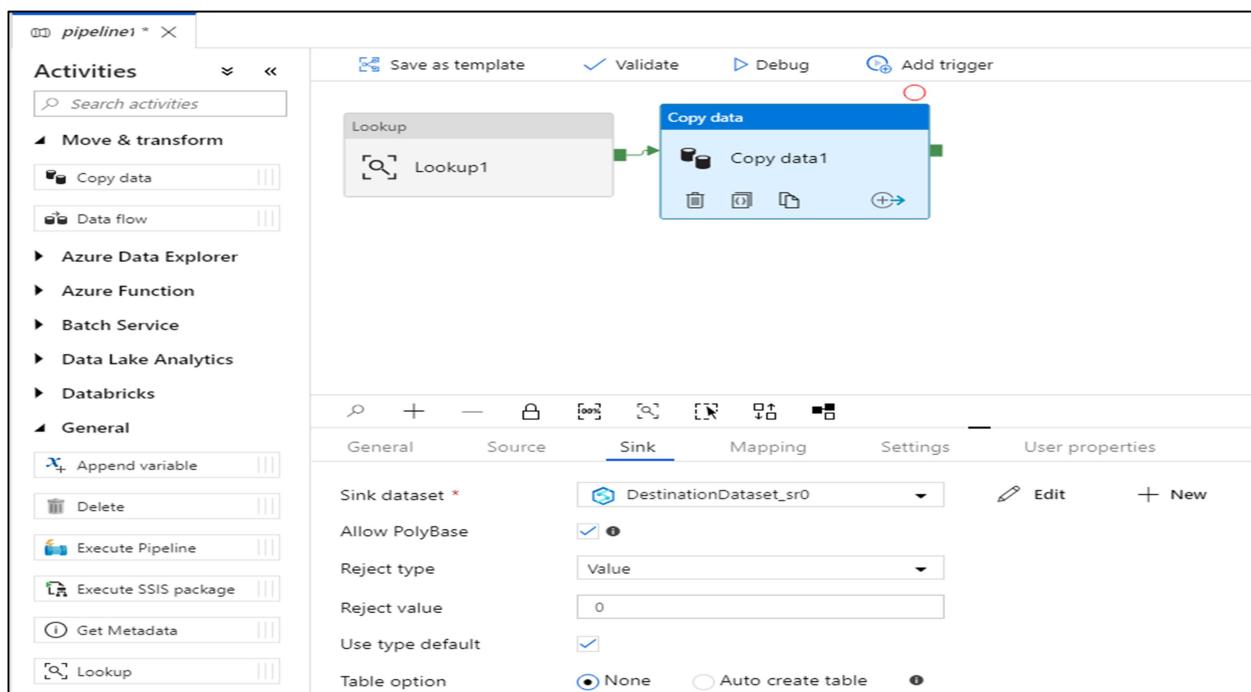
- ✓ Source dataset: SourceDataset\_sr0
- ✓ Query: 

```
SELECT * FROM cohoOLTP.CardTransaction WHERE transactionDateTime > '@{activity('Lookup1').output.firstRow.transactionDateTime}';
```

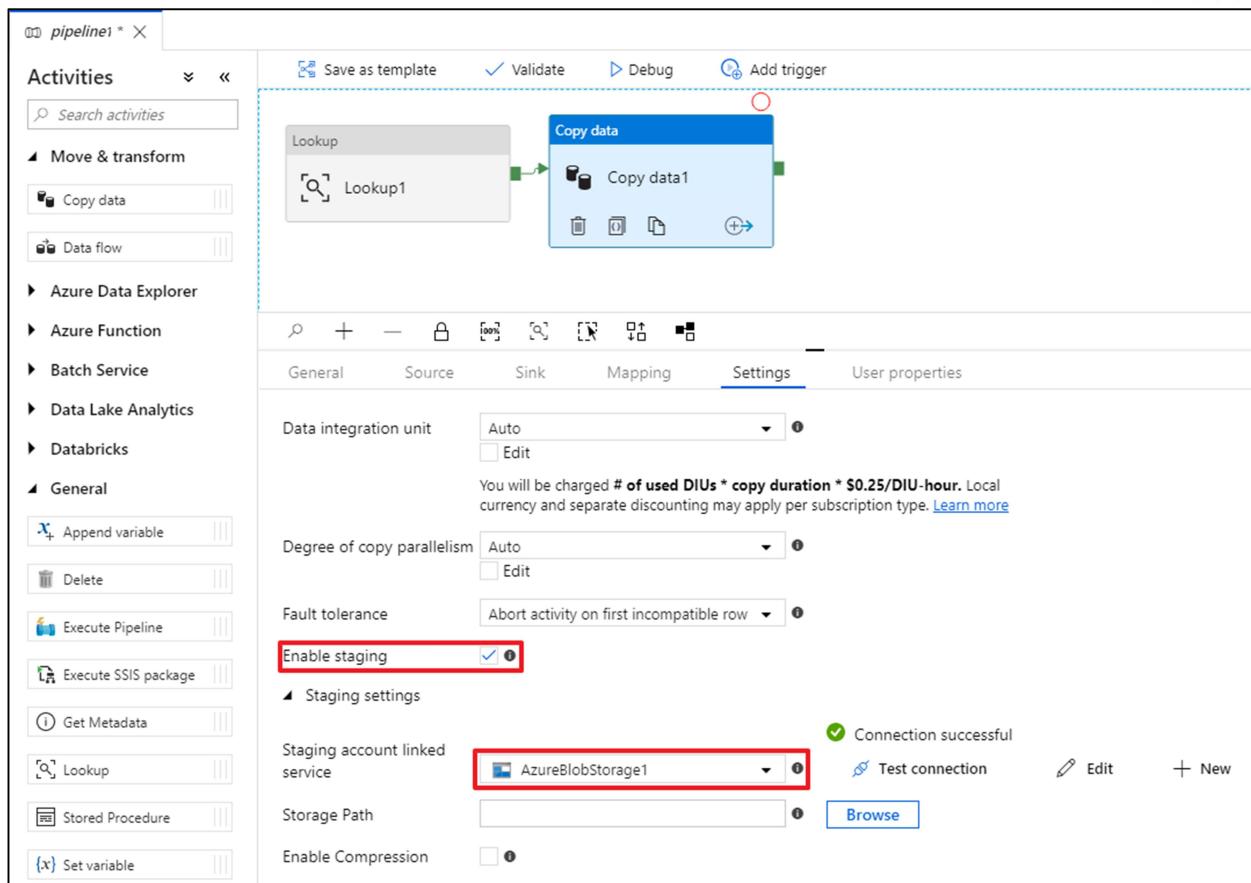


5. Sink 탭에서 다음을 입력합니다.

- ✓ Sink dataset: DestinationDataset\_sr0



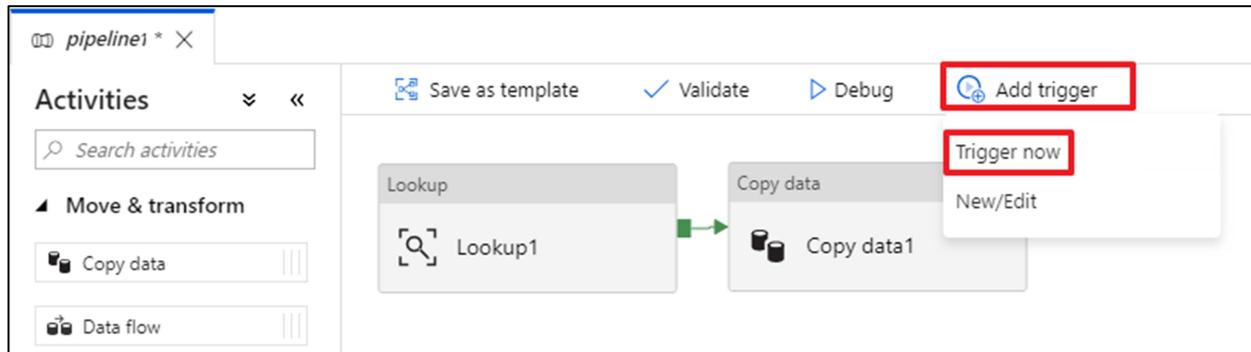
6. Setting 탭에서 Enable staging을 체크한 후 AzureBlobStorage1을 선택합니다.



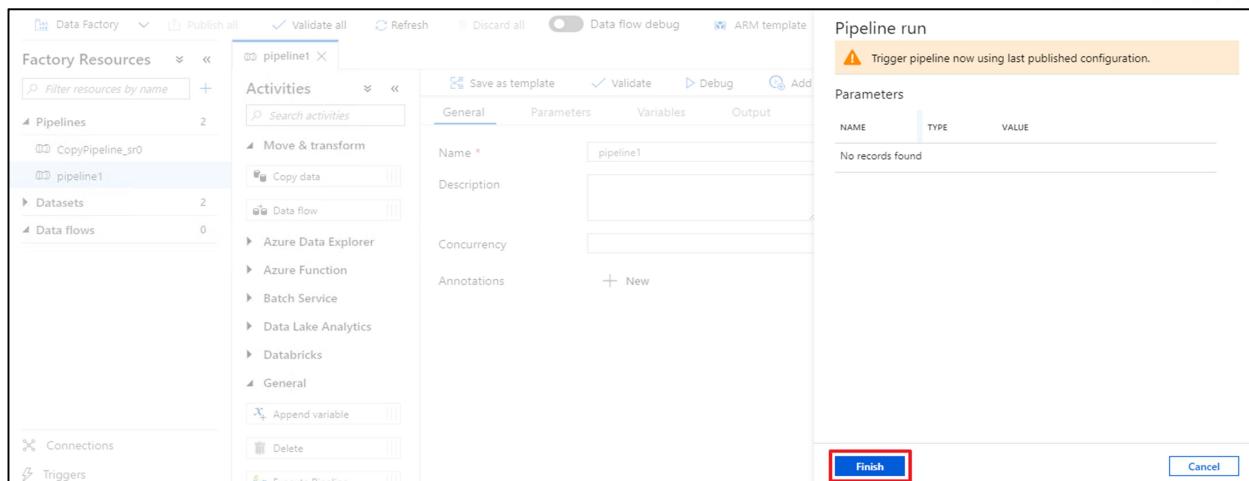
7. 상단의 Publish all 버튼을 클릭합니다.



8. Add trigger를 클릭하여 Trigger now를 선택합니다.



9. 좌측 Pipeline run에서 Finish 버튼을 클릭합니다.



10. 좌측의 Monitor를 클릭하면 실행중인 파이프라인의 상태가 확인 가능합니다.

Pipeline Name	Actions	Run Start	Duration	Triggered By	Status
pipeline1		11/17/2019, 9:25:20 AM	00:00:12	Manual trigger	
CopyPipeline_sr0		11/17/2019, 8:06:04 AM	00:00:33	Manual trigger	

11. 작업이 모두 끝난 후 Refresh 버튼을 클릭하면 완료된걸 확인할 수 있습니다.

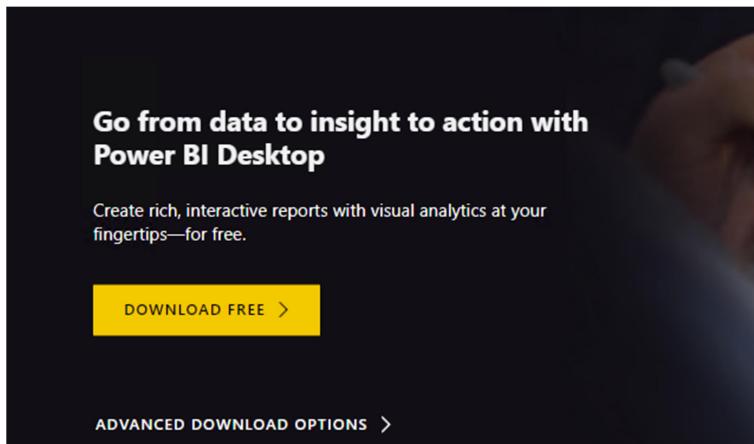
Pipeline Name	Actions	Run Start	Duration	Triggered By	Status
pipeline1		11/17/2019, 9:25:20 AM	00:00:37	Manual trigger	
CopyPipeline_sr0		11/17/2019, 8:06:04 AM	00:00:33	Manual trigger	

## 7. Exercise 5: Visualize data with Power BI Desktop

이 실습에서는 Power BI Desktop과의 통합을 설정합니다.

### 7.1. Task 1: Install Power BI Desktop

1. 웹 브라우저에서 Power BI Desktop 다운로드 페이지 (<https://powerbi.microsoft.com/en-us/desktop/>)로 이동합니다.
2. 페이지 중간에서 Download Free 링크를 선택합니다.



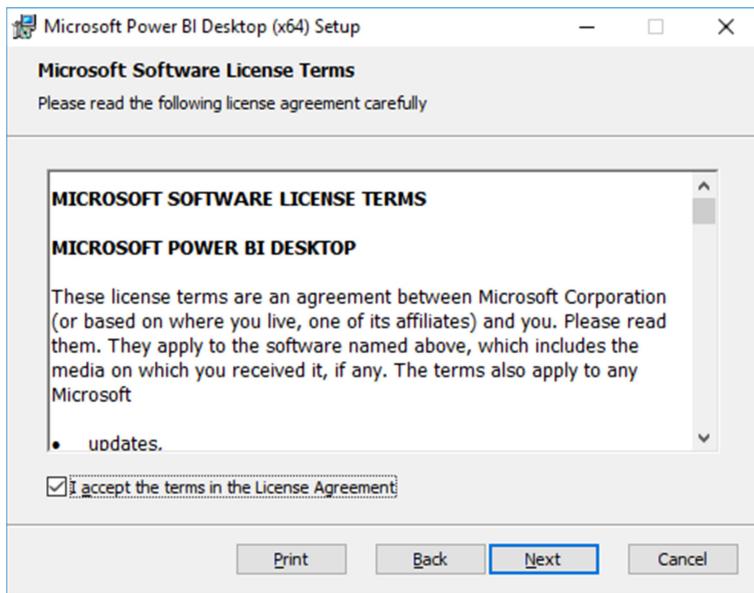
3. 설치파일을 실행합니다.



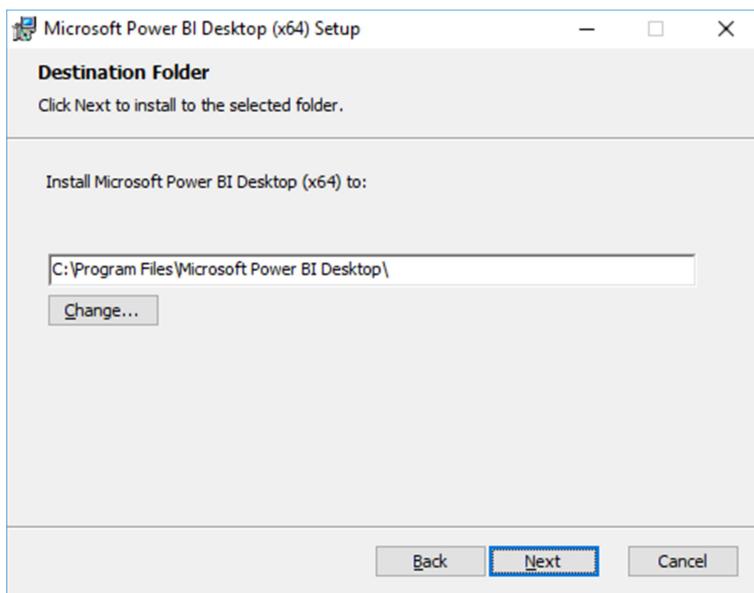
4. Welcome 화면에서 다음을 클릭합니다.



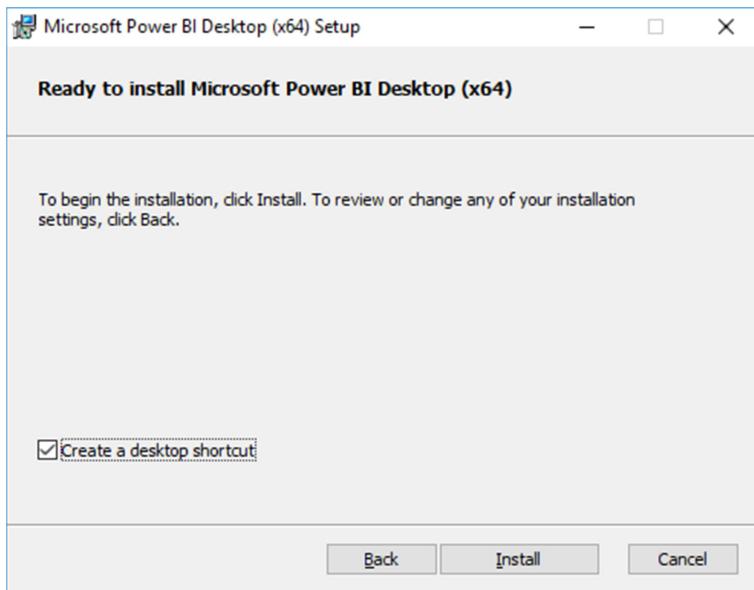
5. 사용권 계약에 동의하고 다음을 선택합니다.



6. 지정 폴더를 디폴트로 두고 다음을 선택합니다.



7. 바탕 화면 바로 가기 만들기 상자가 선택되어 있는지 확인하고 설치를 선택합니다.

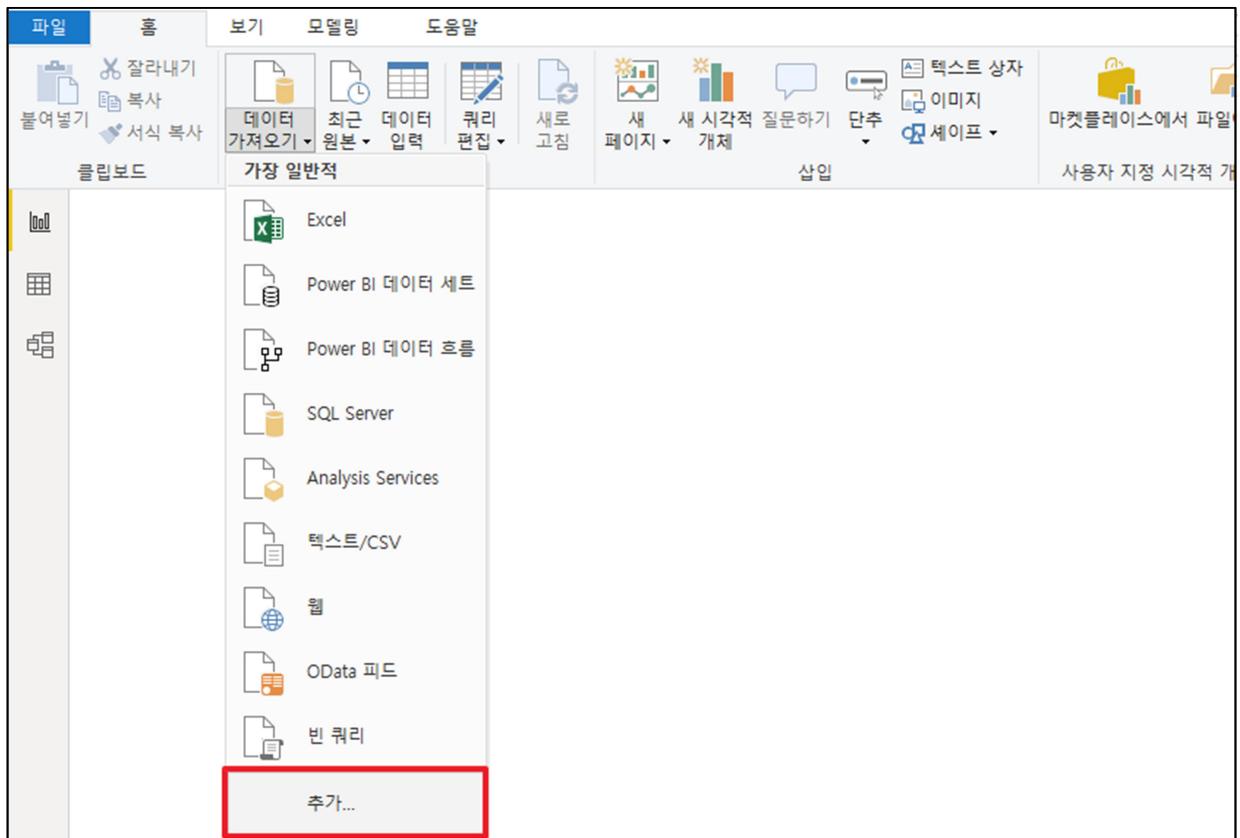


8. Microsoft Power BI Desktop 시작이 선택되어 있는지 확인하고 마침을 선택합니다.

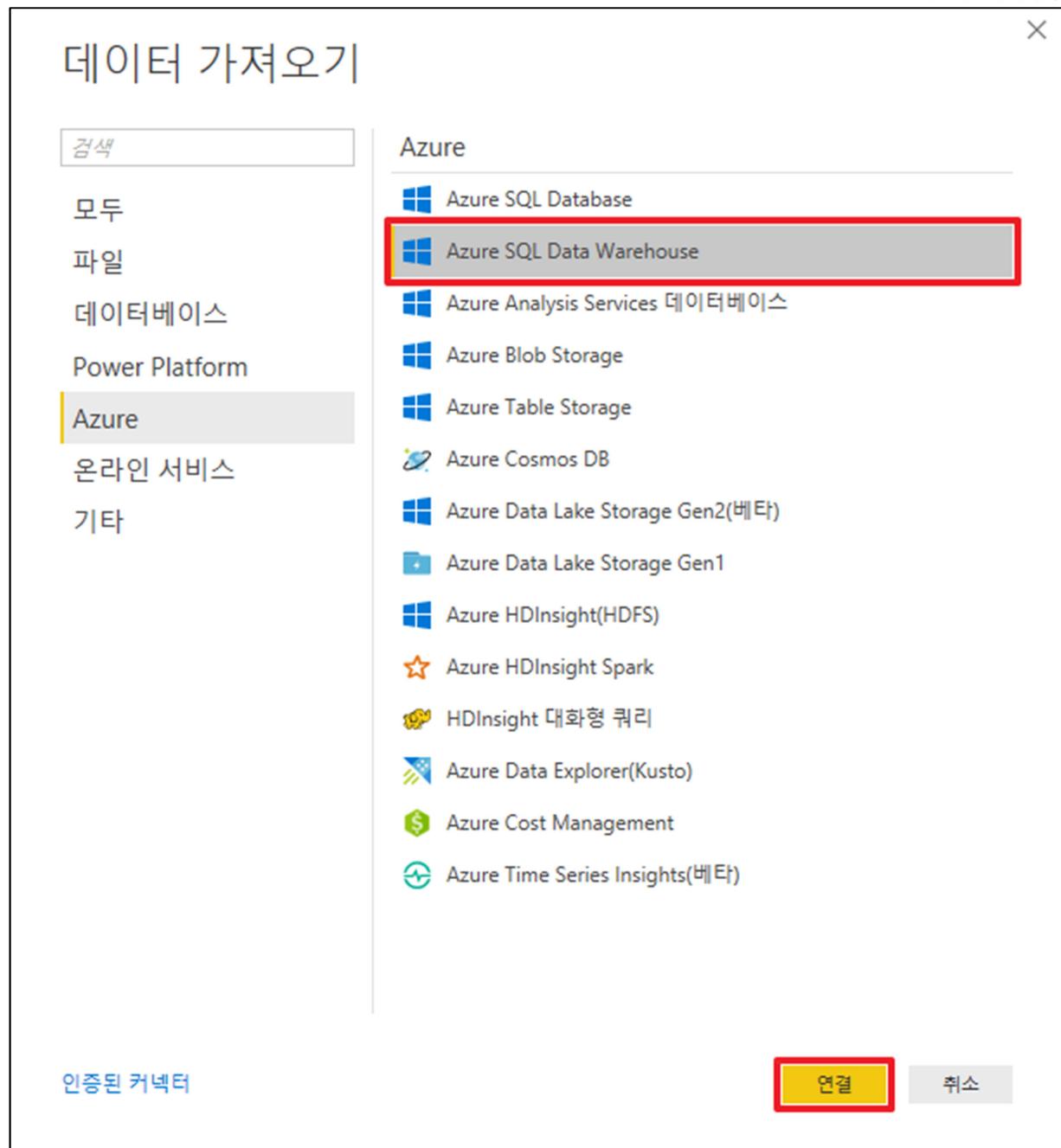


## 7.2. Task 2: Query data with Power BI Desktop

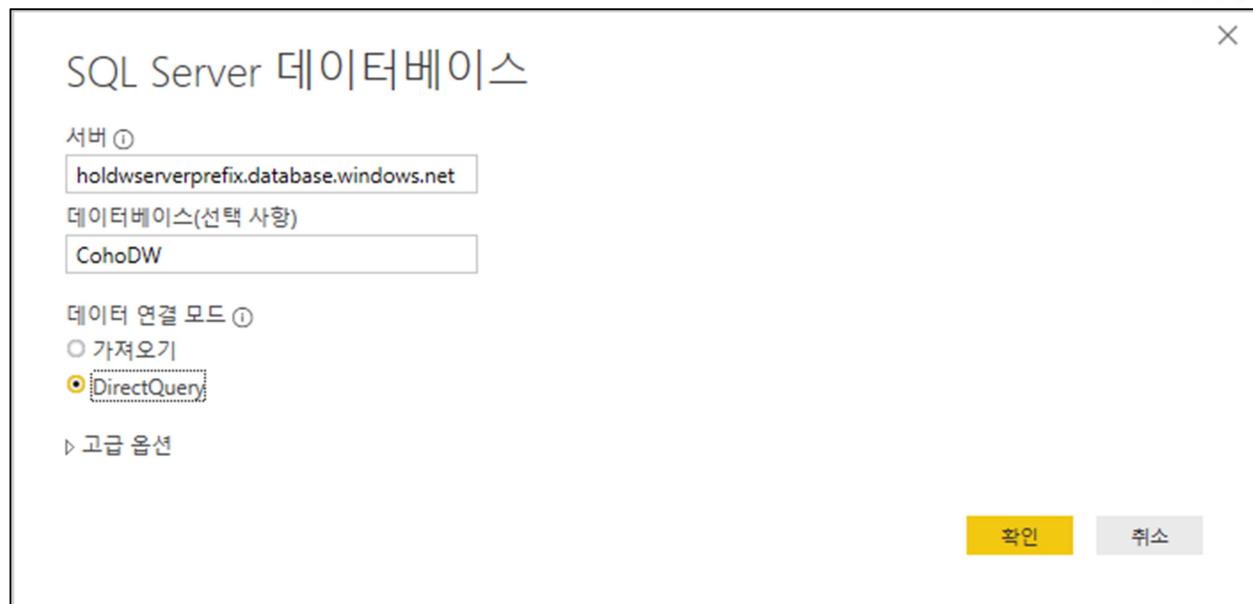
1. Power BI에서 데이터 가져오기 버튼을 클릭한 후 추가...을 선택합니다..



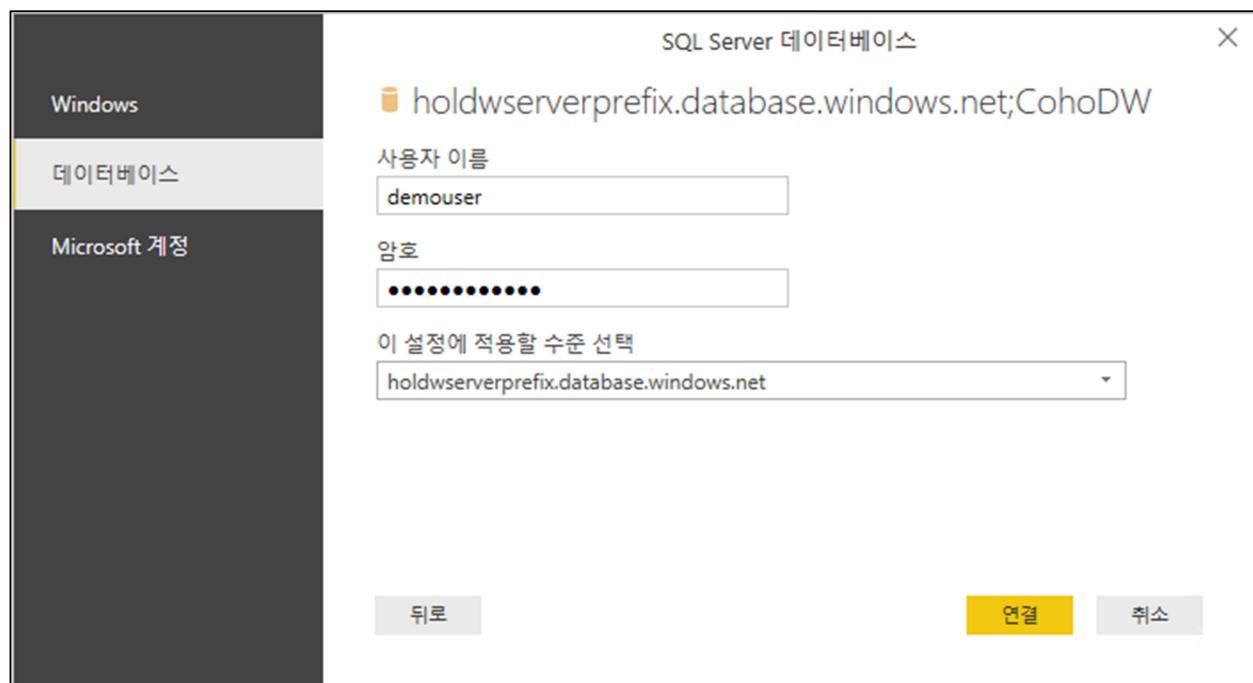
2. 데이터 가져오기 창에서 왼쪽 목록에서 Azure를 선택합니다. 그런 다음 Azure SQL Data Warehouse를 선택하고 연결을 클릭합니다.



3. SQL Server 데이터베이스 화면에서 서버의 이름과 데이터베이스를 입력한 다음 확인을 클릭합니다.



4. 데이터베이스를 선택한 후 Data Warehouse의 사용자 이름과 암호를 입력한 후 **연결**을 클릭합니다.



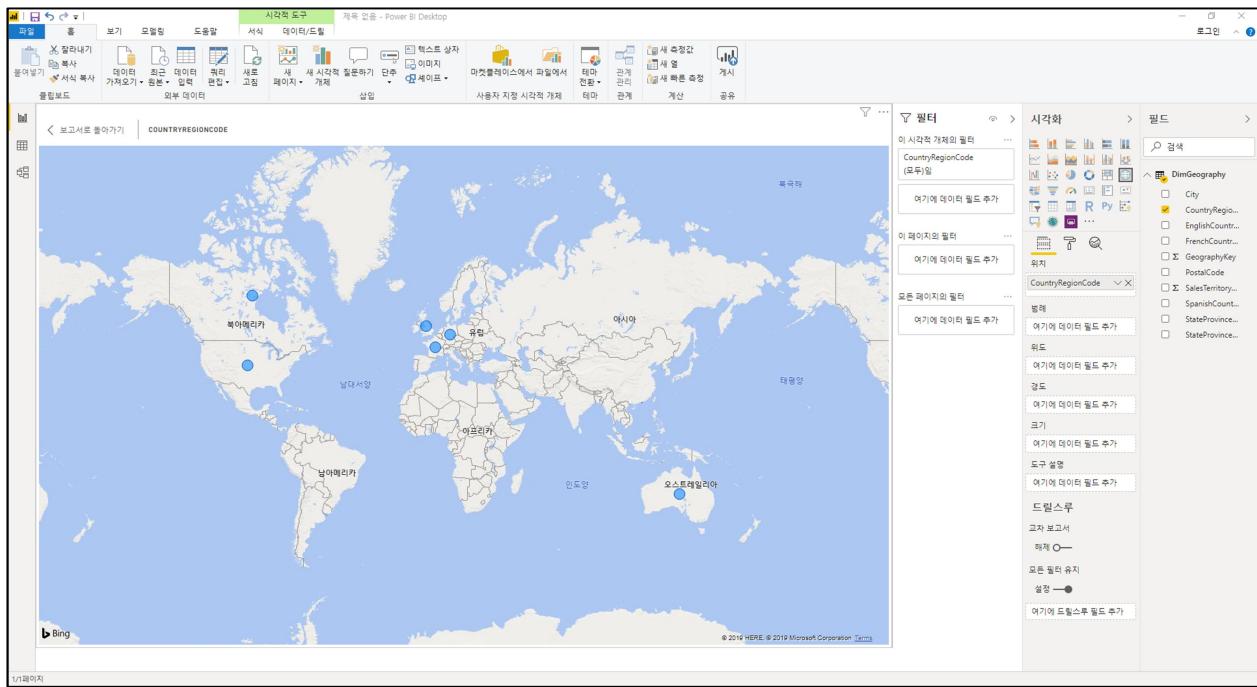
5. DimGeography 테이블을 선택한 후 **로드** 버튼을 클릭합니다.

탐색 창

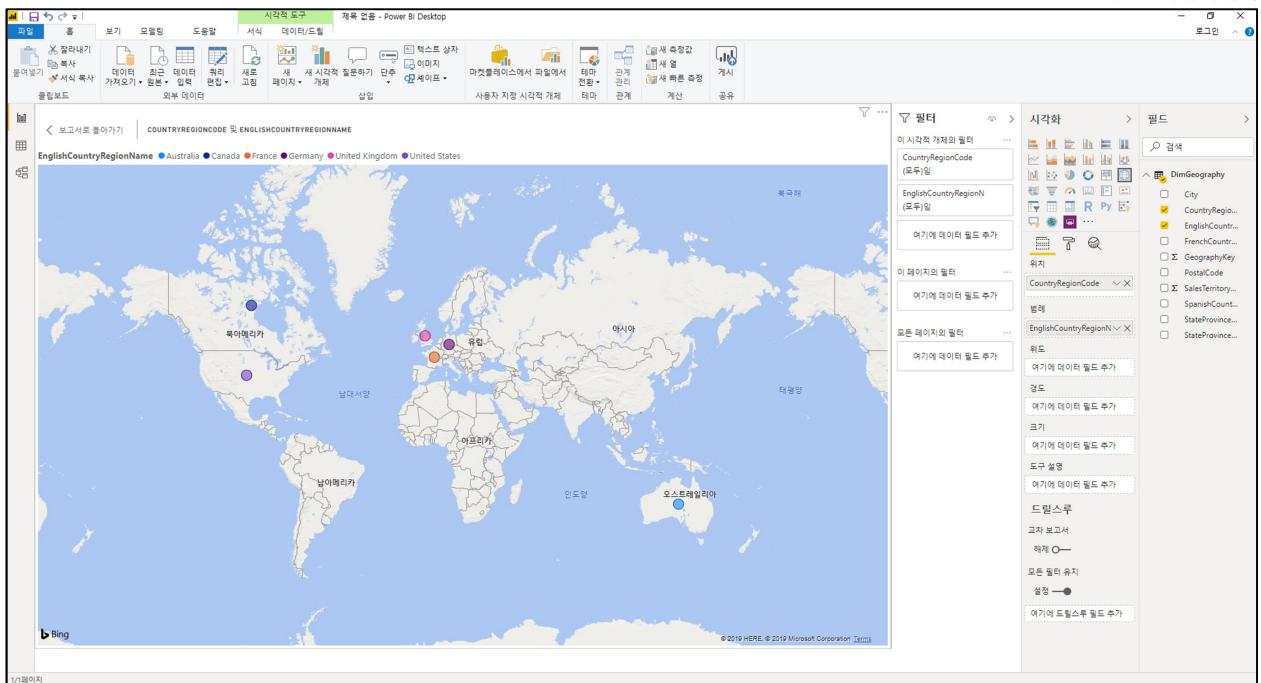
The screenshot shows the Power BI Data Explorer interface. On the left, there is a tree view of tables under '표시 옵션'. One node, 'DimGeography', is highlighted with a red border and has a checked checkbox next to it. On the right, the 'DimGeography' table is displayed in a grid format with columns: GeographyKey, City, StateProvinceCode, and StateProvinceName. The table contains 17 rows of data. At the bottom right of the interface, there are three buttons: '로드' (Load) with a yellow background, '데이터 변환' (Data Transform), and '추소' (Close). A vertical scroll bar is visible on the right side of the table grid.

GeographyKey	City	StateProvinceCode	StateProvinceName
10	Newcastle	NSW	New South Wales
22	East Brisbane	QLD	Queensland
228	Basingstoke Hants	ENG	England
514	Cheektowaga	NY	New York
95	Toronto	ON	Ontario
295	Baldwin Park	CA	California
487	Saint Louis	MO	Missouri
565	La Vergne	TN	Tennessee
119	Grevenbroich	BY	Bayern
475	Branch	MN	Minnesota
3	Darlinghurst	NSW	New South Wales
192	Paris	75	Seine (Paris)
482	Ferguson	MO	Missouri
138	Berlin	HH	Hamburg

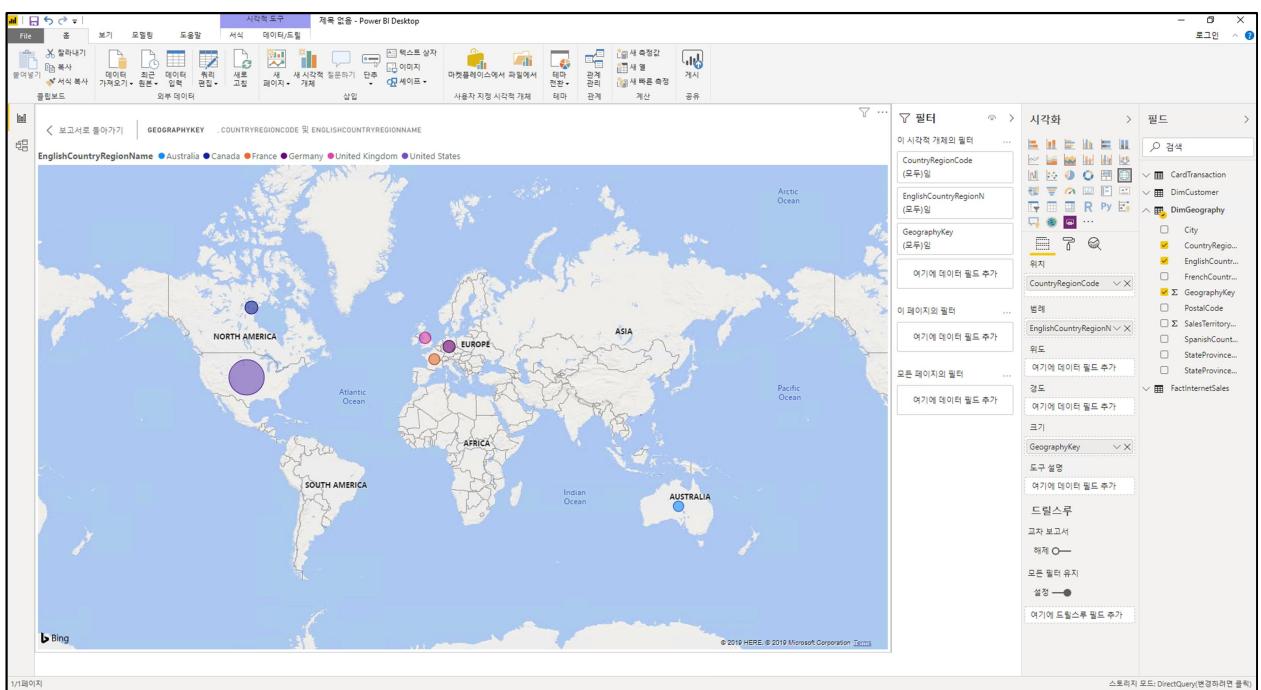
6. 오른쪽에 있는 필드 블레이드에서 DimGeography 테이블을 확장하고 CountryRegionCode 옆에 있는 상자를 선택합니다. Power BI는 자리 데이터를 인식하고 Map 시각화가 자동으로 시작됩니다.



7. 이 원이 실제로 무엇을 의미하는지에 대해 좀 더 구체적인 세부 사항을 보고 싶습니다. 국가/지역을 식별하는 범례를 추가해 봅시다. Legend 아래에서 EnglishCountryRegionName 필드를 드래그합니다.



8. 원의 크기를 변경시켜 보겠습니다. 수치를 가진 필드인 GeographyKey를 드래그해서 Map에 추가 해봅니다.



9. 화면 왼쪽 상단에 있는 저장 버튼을 클릭하고 보고서 이름을 countryregion로 지정한 다음 저장을 클릭합니다.

실습이 완료되었습니다. 이번 실습을 통해 온-프레미스 Data Warehouse를 Azure SQL Data Warehouse로 마이그레이션 하는 과정에 대해 알아보았습니다

- ✓ Data Warehouse 마이그레이션 계획, 사내 구축 환경의 데이터베이스에서 클라우드로의 효율적인 마이그레이션
- ✓ Azure SQL Data Warehouse의 데이터 배포, ETL 작업을 Azure Data Factory에 연결