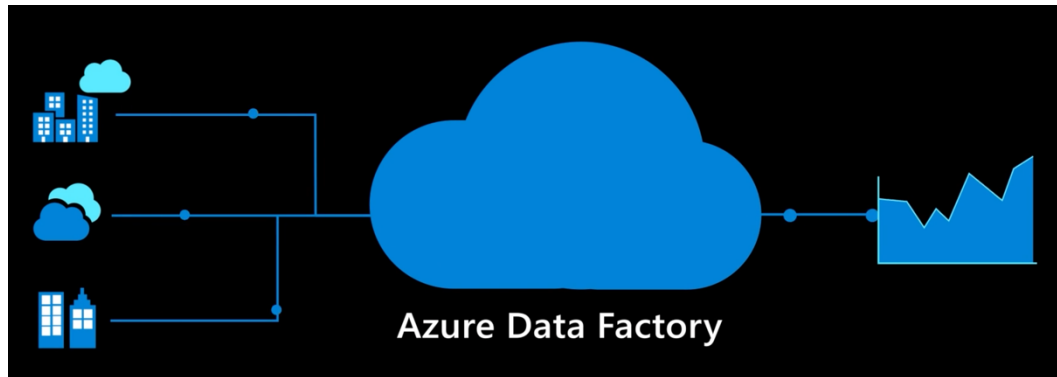


# Data Integration Using Azure Data Factory



Azure Data Factory 는 완전 관리형 서버리스 데이터 통합 서비스입니다. Azure data services, SaaS apps, big data sources, enterprise datawarehouses 등 90 개 이상의 커넥터에서 데이터를 수집하고, Azure Synapse Studio 를 사용하여 데이터를 통합 및 변환하는 파이프라인을 생성할 수 있습니다. 데이터 흐름의 시각화를 통해 직관적인 환경에서 ETL 및 ELT 프로세스를 code-free 로 작성할 수 있습니다.

택시 운행과 요금에 관한 TripData.csv 와 TripFares.csv 데이터를 샘플로 Azure Data Factory 에서 데이터를 통합하고 변환하는 파이프라인을 구축했습니다. 과정은 다음과 같습니다.

1. 데이터를 수집하기 위해 연결 정보를 포함하는 linked service 를 생성합니다.
2. Azure SQL Database 에 있는 데이터셋을 빅데이터 아нал리틱스 서비스인 Azure Data Lake Storage Gen2 로 이동하는 copy data activity 를 생성합니다.
3. 이 데이터셋을 변환하는 data flow activity 를 생성합니다.
4. 두 activity 를 연결하여, copy data activity 가 성공했을 때만 data flow activity 가 실행되도록 하는 파이프라인을 생성합니다.

## 구축 과정

### 1. linked service 생성

Azure SQL Database, Azure Synapse Analytics, Azure Data Lake Storage Gen2 와의 연결 정보를 포함하는 linked service 를 생성합니다. 연결할 서버, 데이터베이스, 로그인 정보 등을 포함합니다.




#### Linked services

Linked service defines the connection information to a data store or compute. [Learn more](#)

[+](#) New

Annotations : Any

Showing 1 - 3 of 3 items

| Name <span>↑↓</span>   | Type <span>↑↓</span>         |
|--|------------------------------|
|  ADLSGen2 | Azure Data Lake Storage Gen2 |
|  SQLDB    | Azure SQL Database           |
|  SQLDW    | Azure Synapse Analytics      |

### 2. Data ingest 를 위한 copy data activity 생성

Azure SQL Database 에 있는 데이터셋을 빅데이터 아날리티クス 서비스인 Azure Data Lake Storage Gen2 에 ingest 하는 copy data activity 를 생성합니다. 'IngestAndTransformTaxiData' 파이프라인을 생성한 뒤, source로 SQLDB linked service 와 복사할 테이블 이름을, sink 로 ADLSGen2 linked service 와 새로 생성될 경로를 지정합니다.

Azure SQL Database 에 데이터셋 taxi-data.bacpac 업로드

```
az sql db import --admin-password $sqlPassword --admin-user $sqlUser \  
--storage-key $StorageAccountKey --storage-key-type StorageAccessKey \  
--storage-uri https://dntwkzz79.blob.core.windows.net/adf-xptmxm/taxi-data.bacpac \  
--name $sqlDB --resource-group $resourceGroup --server $sqlServer
```

파이프라인 'IngestAndTransformTaxiData' 생성

The screenshot shows the Azure Data Factory pipeline editor. On the left, a tree view displays the 'Pipeline' section with 1 item, 'IngestAndTransformTaxiData', and the 'Dataset' section with 4 items, 'AggregatedTaxiData'. On the right, a search bar labeled 'Search activities' is present, followed by a dropdown menu for 'Move & transform' which includes 'Copy data' and 'Data flow'.

copy data activity 'IngestIntoADLS' 생성

SQLDB linked service 를 사용해서 SQL 데이터베이스의 TripData 를 대상으로 지정하는 source properties 를 지정합니다.

### Set properties

Name

Linked service \*

[Edit connection](#)

Table name 


☐ Edit



Import schema  
☒ From connection/store ☐ None

ADLSGen2 linked service 를 사용해서 Azure Data Lake Storage Gen2 의 특정 경로를 대상으로 하는 sink properties 를 지정합니다.

### Set properties

Name

Linked service \*  
 

File path  
 /  /   | 

First row as header ☒

Import schema  
☐ From connection/store ☐ From sample file ☒ None

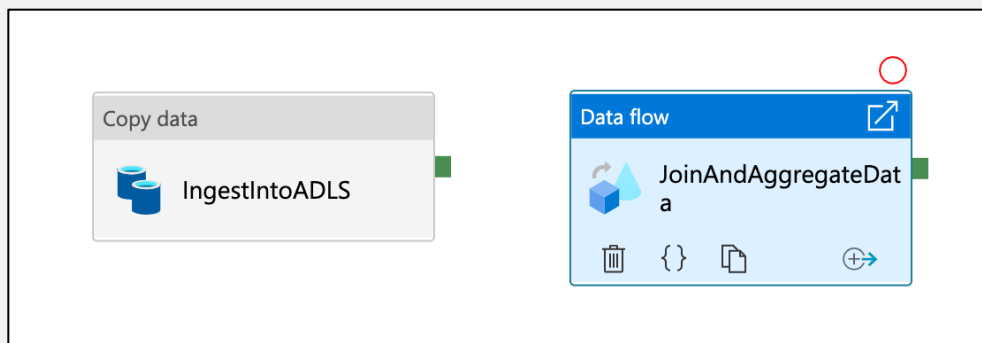
▼ Advanced

[Open this dataset](#) for more advanced configuration with parameterization.

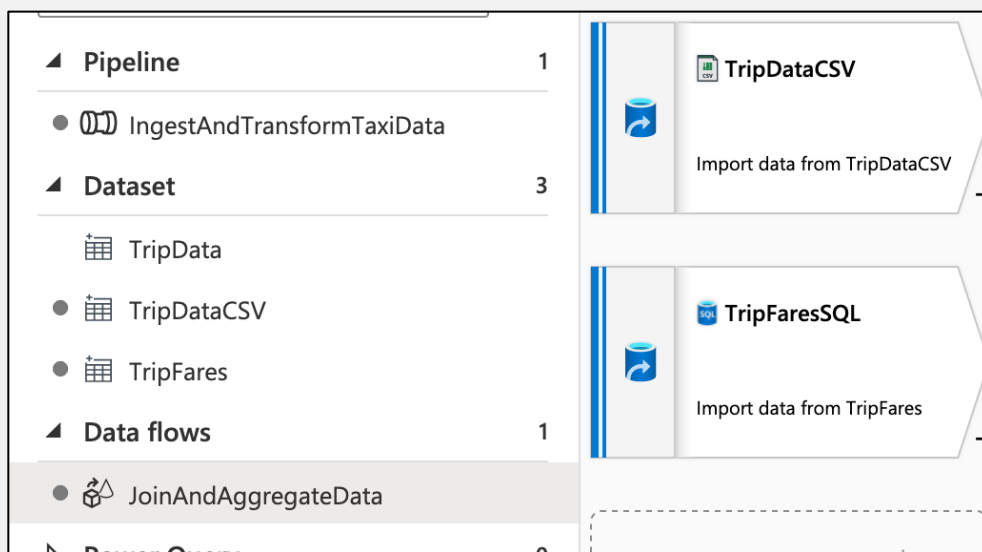
### 3. Mapping data flows 서비스를 사용한 데이터 변환

Mapping data flows 는 코딩 없이 데이터 변환 로직을 생성하게 하고 spark 클러스터에서 실행되는 Azure Data Factory 서비스입니다. 파이프라인에 Mapping data flows 를 사용하는 data flow activity 를 추가하고, 두 개의 데이터셋을 조건에 따라 inner join 하고 aggregate 하도록 하였습니다. 변환된 데이터는 아날리틱스 서비스인 Azure Synapse Analytics 의 SQL pool 에 로드됩니다.

파이프라인 'IngestAndTransformTaxiData'에 data flow activity 'JoinAndAggregateData' 추가



변환할 데이터셋 TripDataCSV 와 TripFaresSQL 을 변환 소스로 импорт



두 개의 데이터셋을 AND 조건을 따라 inner join 변환

The diagram illustrates the configuration of an inner join operation. Two input streams, **TripDataCSV** (Import data from TripDataCSV) and **TripFaresSQL** (Import data from TripFares), are connected to a central join node labeled **InnerJoinWithTripFares**. The join node indicates it has **Columns: 25 total**. Below the diagram, the **Join settings** tab is active, showing the following configuration:

- Output stream name \***: InnerJoinWithTripFares [Learn more](#)
- Left stream \***: TripDataCSV
- Right stream \***: TripFaresSQL
- Join type \***: Inner (selected)

The join type options are: Full outer, Inner, Left outer, Right outer, and Custom (cross).

| Left: TripDataCSV's column |    | Right: TripFaresSQL's column |
|----------------------------|----|------------------------------|
| abc medallion              | == | abc medallion                |
| abc hack_license           | == | abc hack_license             |
| abc vendor_id              | == | abc vendor_id                |
| abc pickup_datetime        | == | abc pickup_datetime          |

두 개의 데이터셋을 'payment\_type' column 을 기준으로 통합하여 평균 요금과 운행 총 거리 데이터 산출

**Aggregate settings** | Optimize | Inspect | Data preview ● | Description

Output stream name \*  [Learn more](#)

Incoming stream \*

**Group by** | Aggregates

| Columns                                       | Name as                                   |
|---|---|
| <input type="text" value="abc payment_type"/> | <input type="text" value="payment_type"/> |

**Group by** | **Aggregates**

Grouped by: payment\_type

[+ Add](#) [Clone](#) [Delete](#) [Open expression builder](#)

| <input type="checkbox"/> Column              | Expression  |
|--|---|
| <input type="checkbox"/> average_fare        | <input type="text" value="avg(toInteger(total_amount))"/> 1.2 <a href="#">+</a> <a href="#">🗑️</a>  |
| <input type="checkbox"/> total_trip_distance | <input type="text" value="sum(toInteger(trip_distance))"/> 121 <a href="#">+</a> <a href="#">🗑️</a> |

변환한 데이터를 로드할 sink 로 Azure Synapse Analytics SQL pool 지정

**Set properties**

Name: AggregatedTaxiData

Linked service: SQLDW

☐ Select from existing table ☒ Create new table

Schema and table name: dbo . AggregateTaxiData

> Advanced

Output stream name: SQLDWSink

Incoming stream: AggregateByPaymentType

데이터 변환 결과

Inner join 한 결과입니다. 두 개의 데이터셋에서 조건으로 지정한 medallion, hack\_license, vendor\_id, pickup\_datetime 열이 모두 일치하는 데이터가 하나의 행으로 조인되었습니다.

| Join settings               | Optimize   | Inspect                     | Data preview | Description                 |
|-----------------------------|------------|-----------------------------|--------------|-----------------------------|
| Number of rows              | INSERT 100 | UPDATE 0                    | DELETE 0     | UPSERT 0                    |
|                             | LOOKUP 0   | ERROR 0                     | TOTAL 203    |                             |
| Refresh                     | Typecast   | Modify                      | Map drifted  | Statistics                  |
|                             |            |                             |              | Remove                      |
| medallion                   | abc        | medallion                   | abc          | hack_license                |
| 7E94181F851247ACE580CA73... |            | 7E94181F851247ACE580CA73... |              | AC433CD9F60ED257513ED36...  |
| A0DEAEC3D5592AE94B87635...  |            | A0DEAEC3D5592AE94B87635...  |              | 0C8B8F7DBFBFA590CBE10177... |
| 696321779D687411F2E5DF69... |            | 696321779D687411F2E5DF69... |              | 23F9C64B453AE29002A74E46... |
| 31872DBC7C48642A6C91817...  |            | 31872DBC7C48642A6C91817...  |              | AAE55F0C9A69D5AAE1D0D6F...  |

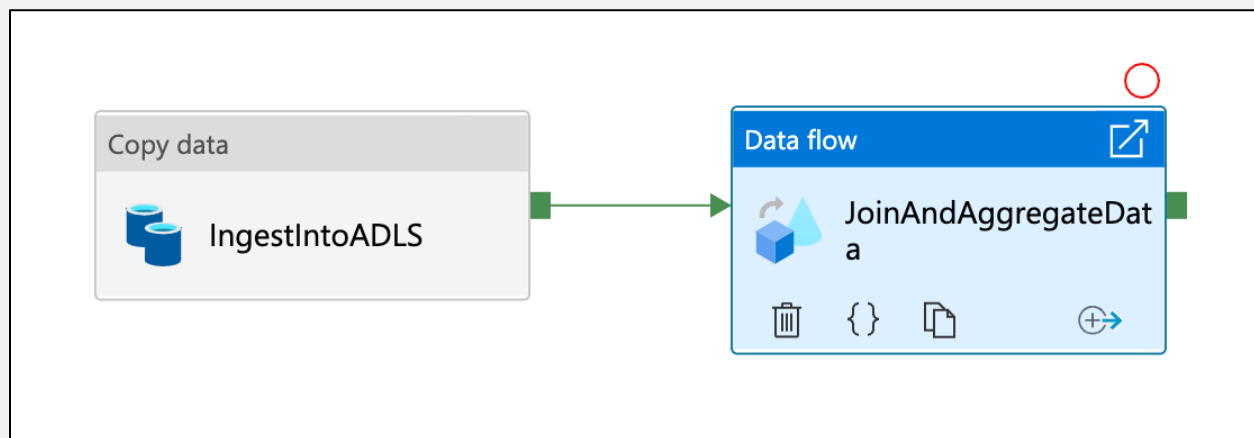


Aggregate 한 결과입니다. CSH, CRD 두 개의 payment\_type 열을 기준으로 평균 요금과 운행 총 거리 데이터가 산출되었습니다.

| Aggregate settings   | Optimize           | Inspect    | Data preview ●      |                       |
|--|--------------------|------------|---------------------|-----------------------|
| Number of rows + INSERT 2 * UPDATE 0 ✕ DELETE 0 *+ UPSERT 0 🔍 LOOKUP 0 ✕ ERROR |                    |            |                     |                       |
| ↻ Refresh  | Typecast ▾         | ↻ Modify ▾ | 🔄 Map drifted       | 📊 Statistics ✕ Remove |
| ↕ payment_type abc   | average_fare 1.2   |            | total_trip_distance |                       |
| + CSH  | 10.787037037037036 |            | 229                 |                       |
| + CRD  | 16.2               |            | 160                 |                       |

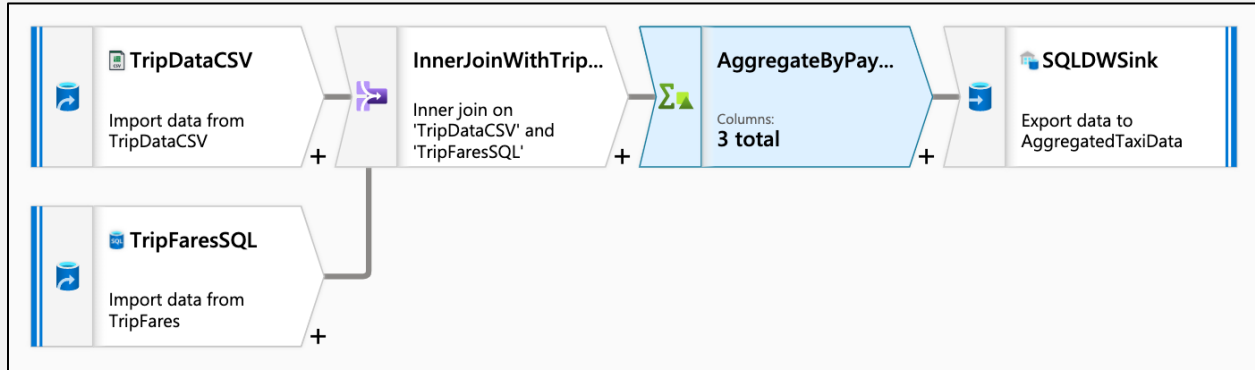
## 4. 최종 Azure Data Factory 파이프라인

Pipeline 'IngestAndTransformTaxiData'



Copy data activity 'IngestIntoADLS' 성공시에 data flow activity 'JoinAndAggregateData'가 실행되는 파이프라인입니다.

### Data flow activity 'JoinAndAggregateData'



TripDataCSV 와 TripFaresSQL 두개의 데이터셋을 가지고 inner join, aggregate 변환한 뒤 최종적으로 Azure Synapse Analytics 에 저장하는 data flow activity 입니다.