Homework 2: Cosmos DB 요약 노트 📝

이 문서는 NoSQL 과제 2의 전체 과정을 단계별로 요약한 설명서입니다.

🎯 최종 목표

Azure Data Factory를 사용해 웹에 있는 영화 데이터(JSON)를 Azure Cosmos DB로 가져온 후, SQL 쿼리를 실행하여 데이터를 분석하고 결과를 제출합니다.

⚙️ 1단계: Cosmos DB 데이터베이스 설정

가장 먼저 데이터를 담을 공간을 설정해야 합니다.

1. Cosmos DB **계정 생성**: Azure Portal에서 Cosmos DB for NoSQL 계정을 새로 생성합니다.
2. **데이터베이스 및 컨테이너 생성**: 아래의 설정값을 정확하게 입력하여 생성합니다.
    - **데이터베이스 ID**: `omdsmod4`
    - **컨테이너 ID**: `movies`
    - **파티션 키** (Partition Key): `/status`
    - **처리량** (Throughput): 수동(Manual) 400 RU/s (무료 등급)

🚚 2단계: Data Factory로 데이터 로드

Data Factory 파이프라인을 만들어 원본 데이터를 Cosmos DB로 복사합니다.

1. **새 파이프라인 생성**: Homework 1에서 사용했던 Data Factory에 Homework 2를 위한 새 파이프라인을 만듭니다.
2. **데이터 복사 작업 설정**:
    - **원본** (Source): 아래의 TMDB 데이터셋 URL을 사용합니다.
        - 데이터 소스 형식: HTTP
        - URL: `https://mod4.blob.core.windows.net/hw2/tmdb_5000_movies.json`
    - **싱크** (Sink): 1단계에서 생성한 Cosmos DB의 `movies` 컨테이너를 목적지로 지정합니다.
3. **파이프라인 실행**: 파이프라인을 실행하여 데이터 복사를 완료합니다.

🔍 3단계: 데이터 쿼리 및 확인

Data Explorer를 사용해 데이터가 올바르게 들어왔는지 두 개의 쿼리로 확인합니다.

1. **쿼리 1: "artificial intelligence" 키워드 검색**
    - 아래 쿼리를 실행합니다.

```
SELECT c.title
FROM c
JOIN p IN c.keywords
```

```
WHERE p.name = "artificial intelligence"
```

- **확인**: 결과로 **26개**의 영화가 반환되어야 합니다.

2. **쿼리 2: "Dentsu" 제작사 영화 검색**
   - 위 쿼리 구조를 수정하여 `production_companies` 배열에서 "Dentsu"를 찾도록 변경합니다.

```
SELECT c.title
FROM c
JOIN p IN c.production_companies
WHERE p.name = "Dentsu"
```

- **확인**: 결과로 **12개**의 영화가 반환되어야 합니다.

📸 4단계: 최종 제출물

과제 증명을 위해 아래 항목을 제출합니다.

- **스크린샷 2개**: 위에서 실행한 **두 쿼리의 결과 화면**을 각각 캡처합니다.
- **중요**: 스크린샷의 오른쪽 상단에 **본인의 Azure 계정 정보가 반드시 보여야 합니다**.