CSCI E-103: Data Engineering for Analytics to Solve Business Challenges

제 6 강: BI 분석 및 데이터 시각화

Anindita Mahapatra Eric Gieseke
(강의 자료 기반 요약 및 재구성)

2025 년 가을 학기

▣ **강의명:** CSCI E-103: 재현 가능한 머신러닝

▣ **주차:** Lecture 06

▣ **교수명:** Anindita Mahapatra
Eric Gieseke

▣ **목적:** Lecture 06의 핵심 개념 학습

Contents

1 강의 개요

본 문서는 비즈니스 인텔리전스(BI) 분석 및 데이터 시각화의 핵심 개념을 다룹니다. 데이터 웨어하우스와 데이터 레이크의 전통적인 차이점에서 시작하여, 두 아키텍처의 장점을 결합한 **레이크하우스(Lakehouse)**의 필요성과 구조를 중점적으로 설명합니다.

또한 BI와 비즈니스 분석(BA)의 차이점을 명확히 하고, BI 분석가(BI Analyst)라는 핵심 페르소나와 이들의 주요 기술(SQL) 및 데이터 모델링(Star Schema 등) 방법을 배웁니다.

마지막으로 Databricks 플랫폼을 활용한 최신 BI 기능들, 특히 **Databricks SQL**, **Lakeview 대시보드**, 그리고 **Genie**와 같은 자연어 기반 AI 분석 도구의 작동 원리와 활용법을 실습 예제와 함께 살펴봅니다.

1.1 이 강의의 주요 질문 (Key Questions)

이 문서를 학습한 후, 다음 질문들에 답할 수 있어야 합니다.

- 비즈니스 인텔리전스(BI)란 무엇이며, 비즈니스 분석(BA) 및 AI와 어떻게 다른가?
- 데이터 웨어하우스(Data Warehouse)와 데이터 레이크(Data Lake)의 근본적인 차이점은 무엇인가?
- 왜 **레이크하우스(Lakehouse)**라는 새로운 아키텍처가 필요하게 되었는가?
- BI 데이터를 소비하는 주요 사용자 페르소나는 누구이며, 그들의 핵심 기술은 무엇인가? (힌트: SQL)
- BI 성능을 측정하는 핵심성과지표(KPI)인 **동시성(Concurrency)**과 **지연 시간(Latency)**은 무엇을 의미하는가?
- Databricks SQL 환경에서 **AI 함수** (예: `ai_query`)를 어떻게 활용할 수 있는가?
- **Genie**가 일반 챗봇(ChatGPT 등)과 달리 "환각(Hallucination)"을 일으키지 않는 이유는 무엇인가?

2 핵심 용어 정리

본격적인 학습에 앞서, 이번 강의에서 자주 등장하는 핵심 용어들을 정리합니다.

Table 1: *BI* 분석 및 데이터 저장소 핵심 용어

용어 (Term)	쉬운 설명	원어 (English)	비고
BI	기업이 더 나은 의사결정을 하도록 데이터를 수집, 분석, 시각화하는 기술. (과거 현재 "무엇"이 일어났는지)	Business Intelligence	리포트, 대시보드
BA	과거 데이터를 사용해 현재를 설명하고 미래를 예측하는 분석. ("왜" 일어났는지, "무엇"이 일어날지)	Business Analytics	통계, 예측 모델링
데이터 웨어하우스	정형화된(구조화된) 데이터만 저장하는, 빠르고 비싼 '데이터 도서관'. 스키마가 미리 정해짐.	Data Warehouse (DW)	Schema-on-Write
데이터 레이크	모든 종류(정형, 비정형)의 데이터를 원본 그대로 저장하는, 저렴하고 거대한 '데이터 차고'.	Data Lake (DL)	Schema-on-Read
레이크하우스	데이터 레이크의 저렴한 저장소 위에 데이터 웨어하우스의 성능/안정성(ACID)을 구현한 통합 아키텍처.	Lakehouse	DW + DL
메달리온 아키텍처	데이터를 3단계(Bronze: 원본, Silver: 정제, Gold: 집계/BI용)로 나누어 관리하는 파이프라인 구조.	Medallion Architecture	Bronze, Silver, Gold
데이터 사일로	데이터가 부서별, 시스템별로 고립되어 연결되지 않은 상태.	Data Silo	
데이터 스웜프	데이터 레이크가 거버넌스(관리) 없이 방치되어 쓸모없게 된 상태. '데이터 늪'.	Data Swamp	
데이터 연합	데이터를 물리적으로 이동(ETL)하지 않고, 원격 소스에서 직접 쿼리(읽기)하는 기술.	Data Federation	소유권(Ownership)이 없음.
뷰 (View)	쿼리 자체를 가상 테이블처럼 저장한 것. 호출 시점에 쿼리가 실행됨. (물리적 저장 X)	View	
구체화된 뷰	쿼리 결과를 물리적으로 미리 계산하여 저장해 둔 테이블. (성능 향상 목적)	Materialized View (MV)	
동시성	시스템이 동시에 몇 개의 쿼리(작업)를 처리할 수 있는지 나타내는 KPI.	Concurrency	
지연 시간	쿼리를 요청한 시점부터 결과가 반환될 때까지 걸리는 시간 (딜레이).	Latency	

3 핵심 개념 1: 데이터 저장소의 진화

BI를 이해하기 위해서는 먼저 BI가 사용하는 데이터가 어디에, 어떻게 저장되는지 알아야 합니다. 데이터 저장소는 크게 '데이터 웨어하우스'와 '데이터 레이크'로 나뉩니다.

3.1 데이터 웨어하우스 (Data Warehouse) vs. 데이터 레이크 (Data Lake)

두 개념은 데이터를 저장하는 목적과 방식에서 근본적인 차이가 있습니다.

비유로 이해하기: 도서관 vs. 차고

- **데이터 웨어하우스(DW)는 '잘 정리된 도서관'입니다.** 도서관에는 오직 '책'(정형 데이터)만 받습니다. 책을 받으면 사서가 즉시 분류하고(Schema-on-Write), 정해진 책장(테이블)에 꽂습니다. 덕분에 특정 책을 찾을 때(BI 쿼리) 매우 빠르고 정확합니다. 하지만 비디오테이프나 사진(비정형 데이터)은 보관할 수 없고, 도서관을 짓고 유지하는 데 비용이 많이 듭니다.
- **데이터 레이크(DL)는 '모든 것을 쌓아두는 차고'입니다.** 차고에는 책, 사진, 비디오, 고장 난 자전거 등 모든 것(정형/비정형 데이터)을 원본 그대로 던져 넣을 수 있습니다. 저장 공간은 매우 저렴합니다. 나중에 무언가 필요할 때(Schema-on-Read) 차고를 뒤져서 찾아야 하므로 시간이 오래 걸립니다. 관리를 안 하면 쓰레기장, 즉 **데이터 스웜프(Data Swamp)**가 되기 쉽습니다.

다음은 데이터 레이크와 웨어하우스의 기술적 장단점을 비교한 표입니다.

Table 2: 데이터 레이크와 데이터 웨어하우스 비교

2*측면 (Dimension)	데이터 레이크 (Data Lake)		데이터 웨어하우스 (Data Warehouse)	
	장점 (Pro)	**단점 (Con)**	**장점 (Pro)**	**단점 (Con)**
스토리지	모든 파일 타입 지원 (Open-format)	데이터 품질이 낮을 수 있음 파일 수준의 접근 제어	신뢰성 높은 데이터 세분화된 접근 제어	주로 정형 데이터만 지원 특정 벤더 종속적 포맷
컴퓨팅	매우 경제적임 (스토리지/컴퓨트 분리)	운영 복잡성이 높음	사용하기 쉬움 높은 동시성, 낮은 지연시간	확장 시 비용이 많이 듦 (스토리지/컴퓨트 결합)
소비	풍부한 도구 생태계 (ML, AI, DS)	BI 사용 사례에 최적화 안됨	SQL에 최적화 (BI)	ML, 스트리밍 사용 제한

3.2 아키텍처의 한계와 레이크하우스 (Lakehouse)의 등장

전통적으로 기업들은 두 시스템을 함께 사용하려 했습니다.

1. **1세대 (DW Only):** 정형 데이터만 ETL을 통해 DW에 적재. BI, 리포트에만 사용.
2. **2세대 (Two-tier: Lake + DW):** 모든 데이터를 DL에 저장. 이 중 필요한 정형 데이터만 다시 ETL을 통해 DW로 복사하여 BI에 사용. ML/DS는 DL을 사용.

2세대 (Two-tier) 아키텍처의 문제점

모든 데이터를 DL에 저장하고 BI용 데이터만 DW로 복사하는 2세대 방식은 여러 문제를 야기했습니다.

- **데이터 중복:** 똑같은 데이터가 Lake와 Warehouse에 이중으로 저장되어 비용이 낭비됩니다.

- **복잡성 증가:** Lake와 DW라는 두 개의 시스템을 별도로 관리하고 동기화해야 합니다.
- **데이터 최신성 문제:** Lake의 원본 데이터가 DW로 복사(ETL)되는 데 시간이 걸려, BI 사용자는 항상 최신 데이터를 보지 못할 수 있습니다.

이러한 문제를 해결하기 위해 **레이크하우스(Lakehouse)** 아키텍처가 등장했습니다.

레이크하우스(Lakehouse)란? 데이터 레이크의 저렴하고 유연한 개방형 스토리지(예: S3, ADLS) 위에, 데이터 웨어하우스의 핵심 기능(ACID 트랜잭션, 데이터 거버넌스, 빠른 쿼리 성능)을 제공하는 **단일 통합 아키텍처**입니다. (예: Databricks Delta Lake)

레이크하우스의 장점:

- **단일 시스템 (Simplicity):** 데이터 복제나 별도 시스템 관리가 필요 없습니다.
- **모든 워크로드 지원:** BI, 리포팅, 데이터 사이언스(DS), 머신러닝(ML) 등 모든 작업을 **동일한 단일 데이터 소스**에서 직접 수행할 수 있습니다.
- **비용 효율성:** 웨어하우스의 성능을 레이크의 저렴한 비용으로 달성합니다.
- **최신성:** 데이터가 한 곳에만 있으므로(Single Source of Truth), BI 사용자도 항상 최신 데이터를 쿼리할 수 있습니다.

3.3 데이터 저장소 기술의 역사적 흐름

현재의 레이크하우스 개념은 다음과 같은 기술적 진화를 거쳐 등장했습니다.

1. **스프레드시트 (Spreadsheets):** 가장 원시적인 데이터 저장소 (예: CSV)
2. **데이터 웨어하우스 (Data Warehouse):**
   - **Bill Inmon (인몬):** ER 모델 기반, 정규화(3NF)된 중앙 DW를 구축. (데이터 일관성 중시)
   - **Ralph Kimball (킴볼):** 비즈니스 사용자에 초점, 비정규화된 차원 모델(Star/Snowflake Schema)을 제안. (BI 쿼리 속도 중시)
3. **MPP (대규모 병렬 처리):** Teradata, Greenplum 등. 데이터와 컴퓨팅을 여러 노드에 분산. 테라바이트(TB)급 처리가 가능해졌으나 매우 고가였습니다.
4. **NoSQL / BigTable:** Google이 시작. 페타바이트(PB)급 대용량 테이블 데이터를 처리.
5. **Hadoop / 데이터 레이크:** Doug Cutting. 저렴한 하드웨어(범용 서버)를 수평적으로 확장. 스토리지와 컴퓨팅을 분리하는 개념을 도입하며 '데이터 레이크' 시대를 열었습니다.
6. **데이터 메시 / 패브릭 (Data Mesh / Fabric):** 데이터를 '제품'으로 취급하는 분산형 아키텍처(Mesh)와 데이터 위치를 추상화하는 기술(Fabric)이 등장.
7. **레이크하우스 (Lakehouse):** 데이터 레이크 위에 DW 기능을 결합한 현재의 아키텍처.

4 핵심 개념 2: 비즈니스 인텔리전스 (BI) 란?

4.1 BI의 정의와 목적

비유: "오즈의 마법사"의 수정 구슬

BI는 기업의 경영진에게 마치 '수정 구슬'과 같습니다. 비즈니스는 데이터를 통해 현재 무슨 일이 일어나고 있는지 명확히 보고(Insight), 특히 미래에 어떤 일이 일어날지 예측(Foresight)하여 경쟁 우위(Competitive Advantage)를 점하고 싶어 합니다.

비즈니스 인텔리전스 (BI)란, 기업의 데이터를 수집, 통합, 분석, 시각화하여 **더 나은 비즈니스 의사결정을 지원**하는 모든 기술, 애플리케이션, 프로세스를 의미합니다.

BI의 핵심 철학

"데이터(Data)는 분석(Analytics)을 위해 필요한 것이다.
정보(Information)는 비즈니스(Business)를 위해 필요한 것이다."

BI는 원본 데이터(Data)를 가공하여 비즈니스에 유용한 '정보(Information)'로 만드는 과정입니다.

BI의 주요 구성 요소:

- **데이터 분석 (Data Analysis):** 데이터 탐색 및 쿼리
- **시각적 분석 (Visual Analytics):** 차트, 그래프, 대시보드
- **고급 분석 (Advanced Analytics):** 예측, 통계 (BA 영역과 겹침)
- **데이터 거버넌스 (Data Governance):** 데이터 품질, 보안, 접근 제어
- **전략 문서화 (Strategy Documentation):** 비즈니스 미션과 전략

4.2 BI vs. BA: 무엇이 다른가?

BI와 BA(Business Analytics)는 자주 혼용되지만, 초점과 질문이 다릅니다.

최근 BI의 트렌드는 단순한 '서술적(Descriptive)' 분석(과거 리포팅)에서 '처방적(Prescriptive)' 분석(미래에 무엇을 해야 하는지 제안)으로 나아가고 있습니다.

4.3 BI 프로세스 (5단계)

BI는 다음과 같은 5단계를 거쳐 비즈니스 가치를 창출합니다.

1. **데이터 수집 (Collect):** 여러 소스 시스템(CRM, ERP 등)의 데이터를 데이터 웨어하우스(또는 레이크하우스)로 통합(ETL)합니다.
2. **데이터 조직 (Organize):** 수집된 데이터를 분석하기 좋은 모델(예: OLAP 큐브, Star Schema)로 구성합니다.
3. **데이터 분석 (Analyze):** BI 분석가나 사용자가 **SQL**을 사용해 데이터를 쿼리합니다.
4. **데이터 시각화 (Visualize):** 쿼리 결과를 차트, 대시보드, 리포트 등 이해하기 쉬운 형태로 만

Table 3: *Business Intelligence* (*BI*) *vs. Business Analytics* (*BA*)

비즈니스 인텔리전스 (BI)	**비즈니스 분석 (BA)**
과거와 현재의 데이터를 사용합니다.	**과거** 데이터를 사용합니다.
"무엇"이, "어떻게" 일어났는지에 집중합니다. (Descriptive)	**"왜"** 일어났는지 설명하고, **"무엇이 일어날지"** 예측합니다. (Explanatory, Predictive)
주요 질문 예시: • "지난 분기 매출은 얼마인가?" (What) • "가장 많이 팔린 제품은?" (Who) • "언제 가장 많이 팔렸나?" (When)	**주요 질문 예시:** • "왜 그 제품이 많이 팔렸나?" (Why) • "이 추세가 계속될까?" (Will it happen again?) • "가격을 10% 올리면 어떻게 될까?" (What if?)
주요 기술: 리포팅, 대시보드, OLAP, Ad-hoc 쿼리	**주요 기술:** 통계 분석, 데이터 마이닝, 예측 모델링, A/B 테스트

듭니다.

5. **의사 결정 (Decide):** 경영진과 실무자가 이 시각화된 '정보'를 보고 전략적 의사결정을 내립니다. (예: 어떤 신제품을 개발할지, 어떤 시장에 진출할지)

5 핵심 개념 3: BI 페르소나와 데이터 모델링

5.1 BI 분석가 (BI Analyst) 페르소나

BI 워크플로우에는 여러 역할(데이터 엔지니어, 데이터 과학자 등)이 있지만, BI의 핵심 소비자는 **BI 분석가**입니다.

- **주요 기술: SQL** BI 분석가의 주무기는 **SQL**입니다. 이들은 SQL을 사용해 데이터를 탐색하고, 비즈니스 질문에 답하며, 대시보드에 필요한 데이터를 추출합니다.
- **소비 데이터: 정제된 데이터 (Curated Data)** BI 분석가는 원본(Bronze) 데이터가 아닌, 데이터 엔지니어가 1차 정제(Silver)하고 비즈니스 용도에 맞게 집계/가공한(Gold) 데이터를 주로 사용합니다.
- **역할: 분석 엔지니어링 (Analytics Engineering)** 최근에는 BI 분석가가 SQL을 사용해 데이터를 모델링하고 골드 테이블을 직접 큐레이션하는 역할까지 맡기도 하며, 이를 '분석 엔지니어링'이라고 부릅니다.

5.2 BI를 위한 데이터 모델링 (Data Modeling for BI)

BI 쿼리는 매우 빨라야 하므로(Low Latency), 데이터를 BI에 최적화된 구조로 모델링해야 합니다. 이때 가장 널리 쓰이는 방식이 **차원 모델링(Dimensional Modeling)**, 특히 **스타 스키마(Star Schema)**입니다.

- **스타 스키마 (Star Schema):** 이름처럼 '별' 모양의 구조입니다.
  - **팩트 테이블 (Fact Table):** 중앙에 위치. 비즈니스 이벤트의 측정값(숫자 데이터)을 담습니다. (예: ‘sales$_a$*mount*‘, ‘*quantity*$_s$*old*‘)
  - **디멘션 테이블 (Dimension Tables):** 팩트 테이블을 둘러싼 '별'의 꼭짓점. 이벤트가 일어난 맥락(Context)을 설명합니다. (예: `dim_customer`, `dim_product`, `dim_time`)
- **스노우플레이크 스키마 (Snowflake Schema):** 스타 스키마의 변형. 디멘션 테이블이 추가로 정규화되어 또 다른 테이블에 연결된 구조. (예: ‘dim$_p$*roduct*‘ ‘*dim*$_c$*ategory*‘)

스타 스키마 예시: 온라인 상점 매출

- **Fact_Sales (팩트 테이블):** `{date_key, product_key, customer_key, sales_amount, quantity}`
- **Dim_Time (시간 차원):** `{date_key, date, month, year, quarter, day_of_week}`
- **Dim_Product (제품 차원):** `{product_key, product_name, category, brand}`
- **Dim_Customer (고객 차원):** `{customer_key, customer_name, city, country}`

"2025년 1분기, 서울에 거주하는 고객들의 카테고리별 매출액은?" 같은 BI 쿼리를 매우 빠르고 단순한 Join으로 처리할 수 있습니다.

Data Vault 모델

또 다른 모델로 **Data Vault**가 있습니다. 이는 허브(Hubs: 핵심 비즈니스 키), 링크(Links: 관계), 새틀라이트(Satellites: 설명 속성)로 구성되며, 변화에 유연하게 대응할 수 있습니다.

하지만 Data Vault가 Silver 레이어에서 데이터를 유연하게 통합하는 데 쓰이더라도, 최종적으로

BI 사용자가 쿼리하는 **Gold 레이어는 여전히 Star Schema 같은 차원 모델로 변환**되는 경우가 많습니다.

6 Databricks를 활용한 BI 및 데이터 웨어하우징

Databricks는 '레이크하우스' 아키텍처를 기반으로 BI 및 데이터 웨어하우징 기능을 제공합니다.

6.1 Databricks 데이터 인텔리전스 플랫폼

Databricks 플랫폼은 데이터 흐름에 따라 여러 구성요소로 나뉩니다.

Source (소스): 모든 종류의 데이터 (정형, 비정형, 스트리밍)

Ingest (수집): 데이터를 레이크하우스로 가져옵니다.

- **ETL:** 전통적인 방식. 데이터를 복사하여 레이크하우스가 '소유'합니다.
- **Data Federation (데이터 연합):** 데이터를 복사하지 않고, 외부 시스템(예: Oracle, Redshift)에 **읽기 전용(read-only) 쿼리**를 날려 가상으로 데이터를 가져옵니다. **소유권(Ownership)이 없는 것**이 ETL과의 핵심 차이입니다.

Transform (변환): Medallion Architecture (Bronze → Silver → Gold)에 따라 데이터를 정제하고 가공합니다.

Query and Process (쿼리/처리):

- **Databricks SQL (DBSQL):** BI 및 데이터 웨어하우징 워크로드를 위한 SQL 엔진입니다. **ANSI SQL** 표준을 따릅니다.
- **Data Science & ML:** 데이터 과학 및 머신러닝 워크로드.

Governance (거버넌스): Unity Catalog가 모든 데이터 자산(테이블, 파일, 모델)의 접근 제어, 데이터 계보(Lineage), 감사를 중앙에서 관리합니다.

Core Engine (엔진): Photon (Spark을 C++로 재작성한 차세대 벡터화 엔진)이 쿼리 성능을 높여줍니다.

Serve/Analysis (제공/분석): Lakeview Dashboards (내장 대시보드), **BI Tools** (Tableau, PowerBI 연동), Lakehouse Apps 등을 통해 최종 사용자에게 데이터를 제공합니다.

ETL vs. Federation 선택 기준

Federation은 데이터를 이동할 필요가 없어 편리해 보이지만 만능이 아닙니다.

- **Federation이 적합할 때:** 외부 시스템의 데이터 용량이 작거나(Modest data), 단순 참조/조회용(Lookup)일 때 사용합니다.
- **ETL이 적합할 때:** 대용량 데이터를 처리해야 하거나, 낮은 지연 시간(Low Latency)의 빠른 쿼리 성능이 반드시 필요할 때는 ETL을 통해 데이터를 레이크하우스로 물리적으로 가져와야 합니다.

6.2 Databricks SQL (DBSQL)의 주요 BI 기능

DBSQL은 BI 분석가를 위해 다음과 같은 강력한 기능들을 제공합니다.

- **서버리스 컴퓨팅 (Serverless Compute):** 쿼리 요청 시 즉시 컴퓨트 자원을 할당받습니다. 기존 방식처럼 VM(클러스터)이 켜지는 데 3 6분씩 기다릴 필요가 없습니다.
- **스트리밍 테이블 (Streaming Tables):** 복잡한 코드 없이, SQL만으로 스트리밍 데이터 소스

(예: Kafka, 클라우드 파일)를 읽어 자동으로 업데이트되는 테이블을 정의할 수 있습니다.

- **구체화된 뷰 (Materialized Views - MV):** 복잡하고 오래 걸리는 쿼리 결과를 미리 계산하여 물리적 테이블로 저장합니다. BI 대시보드가 이 MV를 조회하면 매우 빠른 응답을 얻을 수 있습니다. 데이터 원본이 변경되면 MV는 증분(incrementally) 업데이트됩니다.
- **지오스페이셜 지원 (Geospatial Support):** 위치/지리 정보(예: H3)를 처리하는 함수를 SQL에서 바로 지원합니다.

6.3 SQL의 AI 기능 (AI Functions in SQL)

Databricks SQL은 LLM(거대 언어 모델)을 SQL 쿼리 내에서 직접 호출하는 혁신적인 기능을 제공합니다.

6.3.1 1. ai_query(): 외부 LLM 호출

`ai_query()` 함수를 사용하면 SQL 문 내에서 OpenAI의 GPT나 Anthropic의 Claude 같은 외부 모델을 직접 호출하여 데이터를 보강(enrich)할 수 있습니다.

```sql
-- 'my-openai-chat'이라는' 모델엔드포인트를호출
SELECT
  sku_id,
  product_name,
  ai_query(
    "my-openai-chat",  -- 미리등록한모델엔드포인트
    -- 프롬프트: 제품이름을포함하여단어  30 홍보문구생성
    "You are a marketing expert. Generate a promotional text
     in 30 words for product: " || product_name
  ) AS promotional_text
FROM
  retail_products;
```

Listing 1: ai_query()를 사용한 제품 홍보 문구 생성 예시

6.3.2 2. 내장 AI 함수 (Built-in AI Functions)

자주 사용되는 AI 작업을 위해 미리 구축된 모델을 제공합니다. 외부 모델을 설정할 필요 없이 즉시 사용 가능합니다.

```sql
-- 감성분석  (Sentiment Analysis)
-- 'positive', 'negative', 'neutral' 반환
SELECT ai_analyze_sentiment('I am happy');
-- 결과: positive

-- 텍스트분류  (Classification)
-- 주어진레이블중하나로분류
SELECT ai_classify('My password is leaked.', ARRAY('urgent', 'not urgent
    '));
-- 결과: urgent
```

```
10
11 -- 정보추출 (Extraction)
12 -- 텍스트에서원하는정보이름 (, 이메일등 ) 추출
13 SELECT ai_extract('John Doe lives in New York', ARRAY('person', '
       location'));
14 -- 결과: {"person": "John Doe", "location": "New York"}
15
16 -- 문법교정 (Grammar Fix)
17 SELECT ai_fix_grammar('This sentence have some mistake');
18 -- 결과: 'This sentence has some mistakes'
19
20 -- 민감정보마스킹 (Masking)
21 SELECT ai_mask('My email is john.doe@example.com', ARRAY('email'));
22 -- 결과: 'My email is [MASKED]'
```

Listing 2: 내장 AI 함수 사용 예시

7 실습: Lakeview 대시보드와 Genie

Databricks는 내장 대시보드 도구인 **Lakeview**와 대화형 AI 분석 도구인 **Genie**를 제공합니다.

7.1 Lakeview 대시보드 (Lakeview Dashboards)

- **정의:** Databricks 플랫폼 내에서 직접 데이터를 시각화하고 대시보드를 구축하는 도구입니다.
- **구성:** '데이터' 탭에서 대시보드에 사용할 데이터를 정의하고, 캔버스에 '시각화 위젯', '텍스트 상자', '필터' 등을 추가하여 대시보드를 만듭니다.
- **자연어 생성:** 차트를 만들 때, 내장된 **Assistant**에게 자연어로 요청할 수 있습니다. (예: "show me total revenue by zip code")

Lakeview vs. PowerBI/Tableau: 언제 무엇을 쓸까?

- **Lakeview (Databricks 내장):** 추가 라이선스 비용이 없습니다. 데이터 엔지니어, 분석가가 빠르게 데이터를 확인하고 공유하는 **'운영용' 대시보드**에 적합합니다.
- **PowerBI / Tableau (외부 BI 도구):** 경영진(CEO 등)에게 보고하기 위한 매우 정교하고 복잡한(예: 3D 효과, 특정 브랜드 색상) **'전략적' 대시보드**에 적합합니다.

7.2 게시 (Publish) 및 공유 (Share)

- **공유 (Share):** 조직 내부의 다른 Databricks 사용자에게 대시보드 접근 권한(보기/편집)을 부여합니다.
- **게시 (Publish):** Databricks 계정이 **없는** 외부 사용자에게 대시보드를 공유하는 기능입니다.
  - **'자격 증명 포함(Embed credentials)'** 옵션으로 게시합니다.
  - **비용 주의:** 게시된 대시보드의 쿼리가 캐시(Cache)되지 않은 상태에서 외부 사용자가 대시보드를 조회하면, **대시보드를 게시한 사람(Publisher)의 계정에서 컴퓨트 비용이 발생**합니다.

7.3 Databricks Genie: 대화형 AI 분석

Genie는 게시된 대시보드에서 사용할 수 있는 대화형 AI 비서입니다.

- **사용 시나리오:** BI 분석가가 만든 대시보드를 현업 사용자(예: 마케팅 매니저)가 보다가 추가적인 궁금증이 생겼습니다. (예: "이 데이터에서 20대 남성 고객만 보면 어떨까?")
  과거에는 이 요청을 BI팀에 보내고 답변까지 몇 주를 기다려야 했지만, 이제는 Genie에게 **자연어로 직접 질문**하여 즉시 답을 얻을 수 있습니다.
- **작동 방식 (예시):**
  1. **사용자 질문:** "평균 여행 시간은 얼마야?"
  2. **Genie 답변:** "평균 13.7분입니다."
  3. **투명성 (Transparency):** Genie는 이 답을 얻기 위해 실행한 **SQL 쿼리를 함께 보여줍니다.** (예: `SELECT avg(dropoff_time - pickup_time) ...`)

- **환각(Hallucination)이 없는 이유:** Genie는 ChatGPT 같은 범용 챗봇이 아닙니다. Genie의 답변 범위는 대시보드를 생성할 때 사용된 **특정 테이블의 메타데이터(컬럼명, 타입 등)로 엄격하게 제한**됩니다.
  만약 사용자가 "날씨가 어때?"처럼 데이터와 무관한 질문을 하면, Genie는 "죄송합니다. 저는 해당 데이터에 대해서만 답변할 수 있습니다"라고 답하며 환각을 일으키지 않습니다.

8 학습 점검 체크리스트

이 강의의 핵심 내용을 잘 이해했는지 다음 항목들로 점검해 보세요.

데이터 웨어하우스(DW)와 데이터 레이크(DL)의 4가지 주요 차이점(데이터, 스키마, 비용, 용도)을 설명할 수 있는가?

"레이크하우스" 아키텍처가 왜 등장했으며, 기존 2-tier (Lake + DW) 아키텍처의 어떤 문제를 해결하는가? (힌트: 데이터 중복, 복잡성)

BI와 BA의 차이점을 "질문"과 "목적" 관점에서 비교 설명할 수 있는가?

BI 분석가 페르소나의 핵심 기술은 무엇인가? (정답: SQL)

데이터 모델링에서 킴볼(Kimball)의 "스타 스키마"가 무엇인지 팩트/디멘션 테이블로 설명할 수 있는가?

"데이터 연합(Data Federation)"과 "ETL"의 가장 큰 차이점은 무엇인가? (정답: 데이터 소유권/이동 여부)

"뷰(View)"와 "구체화된 뷰(Materialized View)"의 차이점(저장 공간, 성능)을 설명할 수 있는가?

Databricks SQL의 `ai_query()`와 `ai_analyze_sentiment()`의 차이점과 용도를 아는가?

Lakeview 대시보드를 PowerBI 대신 사용하는 이유는 무엇인가? (운영용, 라이선스 비용 없음)

Databricks Genie가 챗GPT와 달리 "환각(Hallucination)"을 일으키지 않는 이유는 무엇인가? (정답: 지정된 테이블 메타데이터만 참조)

외부 사용자와 대시보드를 "게시(Publish)" 기능으로 공유할 때 발생할 수 있는 비용 문제는 무엇인가?

9 FAQ (자주 묻는 질문)

Q: 데이터 레이크가 있는데 왜 굳이 데이터 웨어하우스가 필요한가요?

A: 데이터 레이크는 모든 데이터를 *저장*하는 데는 뛰어나지만, 정제되지 않아 *분석*하기에는 느리고 복잡합니다. 데이터 웨어하우스는 BI 리포팅 및 대시보드처럼 *매우 빠른 응답 속도*(Low Latency)와 *높은 동시성*(High Concurrency)이 필요한 BI 워크로드에 특화되어 있습니다. 레이크하우스는 이 두 장점을 결합하려는 시도입니다.

Q: 레이크하우스는 그냥 마케팅 용어 아닌가요? 데이터 레이크와 뭐가 다른가요?

A: 레이크하우스는 데이터 레이크의 저렴한 저장소(S3, ADLS 등) 위에 ACID 트랜잭션, 데이터 버전 관리, 인덱싱 등 웨어하우스의 핵심 기능을 제공하는 *기술적 아키텍처*(예: Delta Lake)입니다. 덕분에 별도의 DW 없이 레이크에서 직접 BI와 ML을 모두 수행할 수 있습니다.

Q: '데이터 연합(Federation)'은 항상 ETL보다 좋은 것 아닌가요?

A: 아닙니다. 연합은 데이터를 복제/이동하지 않아 편리하지만, 실시간으로 원격 시스템에 쿼리를 날립니다. 이는 *소량의 데이터*나 *참조용 데이터*(lookup)에는 좋지만, 대용량 데이터를 조인하거나 빠른 성능이 필요할 때는 매우 비효율적입니다. 이 경우 ETL을 통해 데이터를 레이

크하우스로 가져오는 것이 성능상 유리합니다.

Q: Genie가 SQL을 생성해준다면, 이제 SQL을 배울 필요가 없나요?

A: 아닙니다. Genie는 *보조* 도구입니다. Genie가 생성한 SQL이 100% 정확하지 않을 수 있으며, 복잡한 비즈니스 로직을 구현하려면 여전히 SQL 지식이 필수입니다. Genie가 생성한 SQL을 *검증하고 수정*할 수 있어야 합니다.

10 빠르게 훑어보기 (1-Page Summary)

저장소 비교: 웨어하우스 vs. 레이크

- **웨어하우스(DW):** 깐깐한 도서관 (정형 데이터, Schema-on-Write, BI/SQL 최적화, 고비용, 고성능)
- **레이크(Lake):** 막 쌓는 차고 (모든 데이터, Schema-on-Read, ML/DS 최적화, 저비용, 저성능)

레이크하우스 (Lakehouse): 두 세계의 통합

레이크의 저렴한 스토리지 + 웨어하우스의 성능/안정성/거버넌스

- 단일 시스템에서 BI와 ML/DS 워크로드를 모두 지원.
- 데이터 중복 및 ETL 파이프라인 복잡성 해소.

질문의 차이: BI vs. BA

- **BI (Business Intelligence):** "무엇이 일어났나?" (과거/현재 리포팅, 대시보드)
- **BA (Business Analytics):** "왜 일어났나? 무엇이 일어날까?" (통계, 예측 모델링)

BI 분석가와 데이터 모델링

- **BI 페르소나:** 핵심 기술은 **SQL**.
- **데이터 모델링: Star Schema** (중앙의 팩트 테이블 + 주변의 디멘션 테이블)가 BI 쿼리 성능에 가장 효율적임. (킴볼 방식)

Databricks SQL 핵심 기능

- **서버리스 컴퓨트:** 클러스터 부팅 대기 시간 없음 (Instant Compute).
- **Materialized Views (MV):** 복잡한 쿼리 결과를 미리 계산/저장하여 대시보드 속도 향상.
- **Streaming Tables:** SQL만으로 스트리밍 데이터 처리.

SQL + AI: 지능형 쿼리

- **ai_query():** SQL 내에서 외부 LLM (GPT 등) 호출.
- **ai_analyze_sentiment(), ai_classify()...:** 내장 AI 함수로 감성분석, 분류 등을 SQL로 즉시 수행.

대시보드와 AI 비서: Lakeview Genie

- **Lakeview:** Databricks 내장 대시보드 (운영용, 라이선스 무료).
- **Genie:** 게시된 대시보드에서 사용하는 **대화형 AI**. 자연어 질문을 SQL로 변환.
- **Genie의 특징:** 환각(Hallucination) 없음. 지정된 테이블의 메타데이터만 참조.