KB 은행 PoC 시나리오 답변

[목차]

[**1.1. 데이터 수집 > 데이터 적재 > S3 → "대상 솔루션" 파일 기반 데이터 적재 및 확인 7**](#_togxz7425iu1)

[**1.2. 데이터 수집 > 데이터 적재 > 사용자 정의 구분자 (ex. \036, ||) 7**](#_9qpiik68ukvd)

[**1.3. 데이터 수집 > 데이터 적재 > euc-kr 파일을 utf-8 저장 8**](#_ky7s8xq8c3yy)

[**1.4. 데이터 수집 > 데이터 적재 > EBCDIC 등의 인코딩 지원 여부 8**](#_meb5014hsumd)

[**1.5. 데이터 수집 > 데이터 적재 > 고정길이 데이터 적재 8**](#_wghfl16tuyrp)

[**1.6. 데이터 수집 > 데이터 적재 > 압축파일 데이터 load 9**](#_dw6d76kve118)

[**1.7. JDBC/ODBC connection DB → "대상 솔루션" 데이터 적재 및 확인 10**](#_y1pb77wm6jzu)

[**1.8. JDBC/ODBC connection DB: euc-kr 인코딩 데이터 utf-8 저장 11**](#_bbfjurf3l015)

[**1.9. JDBC/ODBC connection DB: EBCDIC 등의 인코딩 지원 여부 11**](#_v91qjxtdxof1)

[**2.1. 계정 신청(추가) 및 승인 (개인계정, 공용계정) 12**](#_x19zq8ro1lwp)

[**2.2. 계정 신청(추가), 승인 등의 API 지원 여부 12**](#_mf2b5d9ta3of)

[**2.3. 별도의 계정 생성 지원 (IAM과 무관) 12**](#_jjptj7284y0s)

[**2.4. 내부 SSO 및 2차 인증 연동 (2.4~2.6) 12**](#_g5nxxtibz385)

[**2.7. 다단계 인증(MFA) 12**](#_cjh9jrf04iwe)

[**2.8. 비밀번호 변경 정책 설정 여부 13**](#_westjupaj5px)

[**3.1. 테이블 권한 신청 및 승인 14**](#_4mtcxihhp1ye)

[**3.2. 컬럼별 권한 신청 및 승인 14**](#_i32056z1ddin)

[**3.3. API 기능 제공 여부 16**](#_3pmnwck723sb)

[**3.4. External table 생성 시 S3 직접 접근 가능 여부 17**](#_4x1j737rbjal)

[**3.5. 가상환경 별 관리자 기능 여부 17**](#_etdzmccd9j7s)

[**4.1. 사용자 자원 통제 > 가상환경 별 최소~최대 사용량 설정 가능 여부 18**](#_29vwzj1fvn8o)

[**4.2. 사용자 자원 통제 > 가상환경이 suspend 된 이후 jdbc 호출 시 resume 여부 18**](#_uska2ofeqnyt)

[**4.3. 데이터 저장량 통제 > 사용자별, 스키마별 데이터 저장 공간 할당 및 제한 여부 18**](#_6r9t5tbhvihr)

[**4.4. 접근 및 사용이력 관리 > 로그인, 쿼리 CRUD 작업 이력 관리 여부 19**](#_1htkfjcw6dx9)

[**4.5. 접근 및 사용이력 관리 > 이력 저장 기간 관리 여부 19**](#_i7phqqp13uyz)

[**4.6. 접근 및 사용이력 관리 > 로그 조회 기능 19**](#_ohzsnw5exw0p)

[**4.7. 접근 및 사용이력 관리 > 샤크라 맥스, DB Safer 등 지원 여부 20**](#_7gq1ydcjgfsa)

[**5.1. 암호화 컬럼 관리 21**](#_y2xw1prerqag)

[**5.2. 제공 암호화 알고리즘 21**](#_t8ncbpzc3uz6)

[**5.3. 암호화 키 관리 21**](#_52uf9ruah0cn)

[**5.4. 쿼리 기반의 특정 컬럼 암호화, 복호화 21**](#_msi07x8n0ac0)

[**5.5. 사용자 복호화 권한 관리 22**](#_9c6djy1prqxa)

[**5.6. 마스킹 정책 생성 후 alter, create로 반영 되는지 확인 (신규) 22**](#_n9xsfjhyt18i)

[**5.7. 마스킹 정책 변경 후 기존 테이블 반영되는지 확인 (변경) 24**](#_u8inug1slvet)

[**5.8. 마스킹 정책이 적용된 테이블 확인 25**](#_llddopi2tfmh)

[**5.9. 외부 제공 시 적용 여부 확인 25**](#_lmea85opm9k9)

[**5.10. 계정별 정책 적용 여부 26**](#_fk05bq6lx584)

[**6.1. cold 데이터 자동 백업 및 사용 시 복구 27**](#_olwtbes12fyv)

[**6.2. 데이터 백업 및 복구 27**](#_m1te02iv3qfb)

[**6.3. 테이블 drop 후 rollback 가능 여부, 기간 27**](#_3wf9oiem1wo8)

[**6.4. 데이터 삭제 후 rollback 가능 여부, 기간 27**](#_gmxs5hjk2aiw)

[**6.5. 테이블별 삭제 정책 적용 및 기간 정의 (TTL, record) 28**](#_w4baortevisr)

[**6.6. 스키마, 테이블 폐기 이력 확인 28**](#_ifz4a8bav7f9)

[**7.1. 한글명 테이블 및 컬럼 생성, 변경 29**](#_3efaf4lp119k)

[**7.2. 전체 테이블 조회 29**](#_6dgl7y1sjpzy)

[**7.3. 전체 컬럼 조회 29**](#_34006ow6c0m6)

[**7.4. 특정 테이블 내 컬럼명 조회 29**](#_x90zq1o32lhi)

[**7.5. 특정 컬럼 포함 테이블 목록 조회 30**](#_pt5jr3rtvqwx)

[**7.6. 테이블 및 컬럼 변경이력 조회 30**](#_ls6eco4g18dn)

[**8.1. 쿼리 기반 수행 툴 여부 32**](#_82psqs62o404)

[**8.2. 분석환경 지원 여부 (ML,DL, R 등) 32**](#_7fngmbk6x8j)

[**8.3. 분석 협업 제공 기능 32**](#_x9tf6hohpwg4)

[**8.4. 개발환경 및 코드 타 사용자 공유 32**](#_uyi8nbfzpbyd)

[**8.5. 분석 데이터 타 사용자 공유 32**](#_jzr7y3vjdvix)

[**9.4. JDBC/ODBC 드라이버 지원 여부 34**](#_6y43kafg63r0)

[**9.5. 지원 프로그래밍 언어 34**](#_mdq69nat8pvi)

[**9.6. 태블로, 쿼리원 지원 여부 34**](#_m1jfzpza0aqc)

[**9.7. DataStage, TeraStream 등 지원 여부 35**](#_z2t3yz5oyvkm)

[**10.1. 배치 정기 및 비정기 Job 생성 - 쿼리기반 및 etl 툴(airflow, nifi) 호출 36**](#_suj85p7tf8f7)

[**10.2. JAS 연동을 위한 API 기능 제공 36**](#_j4ientq6k85w)

[**10.3. 배치 스케쥴 모니터링 기능 36**](#_t4n95vl20tyd)

[**10.4. 오류 시 알람 발송 (메일 or sms) 37**](#_s17lfvrn0m9x)

[**11.1. 플랫폼 운영자 배포 파이프라인 지원 (CI/CD) 38**](#_u8ywbhq727fn)

[**11.2. 분석가 배포 파이프라인 지원 (CI/CD) 38**](#_iml7vr5l37n5)

[**11.3. 로컬 개발환경 지원 여부 38**](#_1s17j9umfv6j)

[**11.4. 테스트 및 배포 39**](#_d90kego5amod)

[**12.1. 자원 및 리소스 사용 현황(H/W, Application) 40**](#_lcu90uxmg3jf)

[**12.2. 사용자별, 가상화 환경 별 비용 발생 현황 42**](#_y5wv2yb05p7f)

[**12.3. scale in, out 현황 모니터링 42**](#_siu3znms8bc)

[**12.4. 현재 사용 자원 상세 43**](#_qj17752em724)

[**12.5. Job 수행 및 처리(오류) 현황 43**](#_isguzy31s2dv)

[**13.1. SQL 및 Python UDF 지원 여부 44**](#_y6maxjwdmvzq)

[**13.2. UDF 권한 부여 44**](#_q900o5vlo7if)

[**14.1. 테이블 column 타입 변경 및 추가 45**](#_jv7e2ycw05gu)

[**14.2. 테이블 인코딩 변경 (요건 삭제 4/18) 45**](#_36hbeq5s6k70)

[**14.3. 테이블 pk 지정 및 not null 지정 45**](#_ajcur7kucylg)

[**14.4. DDL로 컬럼 변경 API지원 여부 45**](#_5smnn7v93afe)

[**14.5. 지원 데이터 타입 확인 (varchar, decimal …) 46**](#_ttil1dwfkiqv)

[**14.6. 지원되는 파일 형식 확인 (csv, txt, parquet,iceberg, delta …) 46**](#_1ivcvm84vxgg)

[**14.7. 테이블 포맷 지원 시 external table의 CRUD 여부 47**](#_ci2565qut4dk)

[**15.1. group by, having, count, case 등 가공 및 집계 성능 49**](#_nzu349ximioy)

[**15.2. 벌크 데이터 Replace 51**](#_6f0jhw2cafj4)

[**15.3. 벌크 데이터 Append 51**](#_vn1kilrwi2ge)

[**15.4. 벌크 데이터 Upsert 51**](#_c51ku592ulg0)

[**15.5. 데이터 삭제 → Glue(그룹 마이데이터) 51**](#_w15yyfphmg)

| **#** | **대분류** | **중분류** | **소분류** |
| --- | --- | --- | --- |
| 1.1 | 데이터수집 | 데이터 적재 | S3 → "대상 솔루션" 파일 기반 데이터 적재 및 확인 |
| 1.2 |  |  | - 사용자 정의 구분자 (ex. \036, ||) |
| 1.3 |  |  | - euc-kr 파일을 utf-8 저장 |
| 1.4 |  |  | - EBCDIC 등의 인코딩 지원 여부 |
| 1.5 |  |  | - 고정길이 데이터 적재 |
| 1.6 |  |  | - 압축파일 데이터 load |
| 1.7 |  |  | JDBC/ODBC connection DB → "대상 솔루션" 데이터 적재 및 확인 |
| 1.8 |  |  | - euc-kr 인코딩 데이터 utf-8 저장 |
| 1.9 |  |  | - EBCDIC 등의 인코딩 지원 여부 |
| 2.1 | 계정관리 | 계정 신청 및 승인 지원 여부 | 계정 신청(추가) 및 승인 (개인계정, 공용계정) |
| 2.2 |  |  | 신청(추가), 승인 등의 API 지원 여부 |
| 2.3 |  |  | 별도의 계정 생성 지원 (IAM과 무관) |
| 2.4 |  | 내부 SSO 및 2차 인증 연동 여부 | 개인계정 및 공용계정 로그인 |
| 2.5 |  |  | - 내부 SSO로그인 |
| 2.6 |  |  | - ID & PW 로그인 |
| 2.7 |  |  | - 다단계 인증 (MFA) |
| 2.8 |  | 비밀번호 변경 정책 설정 여부 | 비밀번호 변경 정책 설정 (8자리 및 난이도, 사용기간) 지원 |
| 3.1 | 권한관리 | 테이블 및 컬럼 | 테이블 권한 신청 및 승인 |
| 3.2 |  |  | 컬럼별 권한 신청 및 승인 |
| 3.3 |  |  | API 기능 제공 여부 |
| 3.4 |  |  | External table 생성 시 S3 직접 접근 가능 여부 |
| 3.5 |  | 부서별 관리자 | 가상환경 별 관리자 기능 여부 |
| 4.1 | 통제 | 사용자 자원 사용량 통제 | 가상환경 별 최소~최대 사용량 설정 가능 여부 |
| 4.2 |  |  | 가상환경이 suspend 된 이후 jdbc 호출 시 resume 여부 |
| 4.3 |  | 데이터 저장량 통제 | 사용자별, 스키마별 데이터 저장 공간 할당 및 제한 여부 |
| 4.4 | 로깅 | 접근 및 사용이력 관리 | 로그인, 쿼리 CRUD 작업 이력 관리 여부 (사용자, 실행일시, IP, 성공여부 등) |
| 4.5 |  |  | 이력 저장 기간 관리 여부 |
| 4.6 |  |  | 로그 조회 기능 |
| 4.7 |  |  | 샤크라 맥스, DB Safer 등 지원 여부 |
| 5.1 | 개인신용정보저장 | 암/복호화 | 암호화 컬럼 관리 |
| 5.2 |  |  | 제공 암호화 알고리즘 |
| 5.3 |  |  | 암호화 키 관리 |
| 5.4 |  |  | 쿼리 기반의 특정컬럼 암호화, 복호화 |
| 5.5 |  |  | 사용자 복호화 권한 관리 |
| 5.6 |  | 비식별 (Hash, Mask) | 마스킹 정책 생성 후 alter, create로 반영 되는지 확인 (신규) |
| 5.7 |  |  | 마스킹 정책 변경 후 기존 테이블 반영되는지 확인 (변경) |
| 5.8 |  |  | 마스킹 정책이 적용된 테이블 확인 |
| 5.9 |  |  | 외부 제공 시 적용 여부 확인 |
| 5.1 |  |  | 계정별 정책 적용 여부 |
| 6.1 | 폐기 | 자동 백업 및 복구 | cold 데이터 자동 백업 및 사용 시 복구 |
| 6.2 |  |  | 데이터 백업 및 복구 |
| 6.3 |  | rollback 정책 | 테이블 drop 후 rollback 가능 여부, 기간 |
| 6.4 |  |  | 데이터 삭제 후 rollback 가능 여부, 기간 |
| 6.5 |  | 데이터 폐기 관리 | 테이블별 삭제 정책 적용 및 기간 정의 (TTL, record) |
| 6.6 |  | 이력 관리 | 스키마, 테이블 폐기 이력 확인 |
| 7.1 | 거버넌스 | 테이블 및 컬럼 메타 | 한글명 테이블 및 컬럼 생성, 변경 |
| 7.2 |  |  | 전체 테이블 조회 |
| 7.3 |  |  | 전체 컬럼 조회 |
| 7.4 |  |  | 특정 테이블 내 컬럼명 조회 |
| 7.5 |  |  | 특정 컬럼 포함 테이블 목록 조회 |
| 7.6 |  | 이력 관리 | 테이블 및 컬럼 변경이력 조회 |
| 8.1 | 분석 | 지원 | 쿼리 기반 수행 툴 여부 |
| 8.2 |  |  | 분석환경 지원 여부 (ML,DL, R 등) |
| 8.3 |  |  | 분석 협업 제공 기능 |
| 8.4 |  | 공유 | 개발환경 및 코드 타 사용자 공유 |
| 8.5 |  |  | 분석 데이터 타 사용자 공유 |
| 9.1 | 호환성 | 외부 연계 | JDBC/ODBC 드라이버 지원 여부 |
| 9.2 |  | 개발 언어 | 지원 프로그래밍 언어 |
| 9.3 |  | 상용 S/W 호환성 | 태블로, 쿼리원 지원 여부 |
| 9.4 |  |  | DataStage, TeraStream 등 지원 여부 |
| 10.1 | 스케쥴 | Job 관리 | 배치 정기 및 비정기 Job 생성  - 쿼리기반 및 etl 툴(airflow, nifi) 호출 |
| 10.2 |  |  | JAS 연동을 위한 API 기능 제공 |
| 10.3 |  | 모니터링 | 배치 스케쥴 모니터링 기능 |
| 10.4 |  | 알람 | 오류 시 알람 발송 (메일 or sms) |
| 11.1 | 배포관리 | 배포체계 | 플랫폼 운영자 배포 파이프라인 지원 (CI/CD) |
| 11.2 |  |  | 분석가 배포 파이프라인 지원 (CI/CD) |
| 11.3 |  |  | 로컬 개발환경 지원 여부 |
| 11.4 |  |  | 테스트 및 배포 |
| 12.1 | 모니터링 | 자원사용 | 자원 및 리소스 사용 현황(H/W, Application) |
| 12.2 |  |  | 사용자별, 가상화 환경 별 비용 발생 현황 |
| 12.3 |  |  | scale in, out 현황 모니터링 |
| 12.4 |  |  | 현재 사용 자원 상세 |
| 12.5 |  | ETL 및 파이프라인 | Job 수행 및 처리(오류) 현황 |
| 13.1 | 사용자함수 | UDF | SQL 및 Python UDF 지원 여부 |
| 13.2 |  |  | UDF 권한 부여 |
| 14.1 | 데이터운영 | DDL로 컬럼 및 인코딩 변경 | 테이블 column 타입 변경 및 추가 |
| 14.2 |  |  | 테이블 인코딩 변경 |
| 14.3 |  |  | 테이블 pk 지정 및 not null 지정 |
| 14.4 |  |  | API지원 여부 |
| 14.5 |  | 데이터 타입 및 파일 포맷 | 지원 데이터 타입 확인 (varchar, decimal …) |
| 14.6 |  |  | 지원되는 파일 형식 확인 (csv, txt, parquet,iceberg, delta …) |
| 14.7 |  |  | 테이블 포맷 지원 시 external table의 CRUD 여부 |
| 15.1 | 성능 | 쿼리 | group by, having, count, case 등 가공 및 집계 성능 |
| 15.2 |  |  | 벌크 데이터 Replace |
| 15.3 |  |  | 벌크 데이터 Append |
| 15.4 |  |  | 벌크 데이터 Upsert |
| 15.5 |  |  | 데이터 삭제 → Glue(그룹 마데) |

### 1.1. 데이터 수집 > 데이터 적재 > S3 → "대상 솔루션" 파일 기반 데이터 적재 및 확인

* 지원 여부: 지원
* 상세: Auto loader, Delta Live Tables, COPY INTO 등의 방식으로 파일 데이터 적재 가능
* 문서: <https://docs.databricks.com/aws/en/query/formats/csv>
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **%sql**  **CREATE TABLE diamonds AS**  **SELECT \* FROM read\_files(**  **"/databricks-datasets/Rdatasets/data-001/csv/ggplot2/diamonds.csv",**  **format => 'csv',**  **header => True,**  **sep => ',')**  **;**  **SELECT \* FROM diamonds;** |
| --- |

### 1.2. 데이터 수집 > 데이터 적재 > 사용자 정의 구분자 (ex. \036, ||)

* 지원 여부: 지원
* 상세: Autoloader의 CSV 옵션 중 sep 또는 delimiter 로 컬럼 구분자, lineSep 으로 라인 구분자 지정
* 문서: <https://docs.databricks.com/ingestion/auto-loader/options.html#csv-options>
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **%sql**  **CREATE TEMPORARY TABLE t12\_new**  **USING CSV**  **OPTIONS (path "dbfs:/FileStore/tables/KBPOC/data\_1\_2.csv", header "true", delimiter "\001")**  **;** |
| --- |

### 1.3. 데이터 수집 > 데이터 적재 > euc-kr 파일을 utf-8 저장

* 지원 여부: 지원
* 상세: Autoloader의 CSV 옵션 중 encoding or charset 을 지정
* 문서: <https://docs.databricks.com/ingestion/auto-loader/options.html#csv-options>
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **%sql**  **CREATE TEMPORARY TABLE t13\_new**  **USING CSV**  **OPTIONS (path "dbfs:/FileStore/tables/KBPOC/data\_cp949.csv", header "true", encoding "cp949")**  **;**  **SELECT \* FROM t13\_new;** |
| --- |

### 1.4. 데이터 수집 > 데이터 적재 > EBCDIC 등의 인코딩 지원 여부

* 지원 여부: 지원
* 상세: 파일을 python ebcdic 패키지를 사용해서 UDF 로 변환
* 문서: https://pypi.org/project/ebcdic/
* 예제:

### 1.5. 데이터 수집 > 데이터 적재 > 고정길이 데이터 적재

* 지원 여부: 지원
* 상세: 고정 길이 텍스트 포맷의 데이터에 대한 적재 방법 제시
* 문서: https://community.databricks.com/s/question/0D53f00001HKHCqCAP/how-to-read-a-fixed-length-file-in-spark-using-dataframe-api-and-scala
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **# borrowed from https://stackoverflow.com/questions/41944689/pyspark-parse-fixed-width-text-file**  **from pyspark.sql.functions import colff**  **schema = [**  **("id",1,5),**  **("ssn",6,10),**  **("name",16,4)**  **]**  **df = spark.read.text("dbfs:/FileStore/tables/KBPOC/data\_1\_5\_fixedlength.txt")**  **df.show()**  **for colinfo in schema:**  **df = df.withColumn(colinfo[0], df.value.substr(colinfo[1],colinfo[2]))**  **df2 = df.drop("value")**  **df2.show()** |
| --- |

### 1.6. 데이터 수집 > 데이터 적재 > 압축파일 데이터 load

* 지원 여부: 지원
* 상세: Spark 이 제공하는 압축 포맷을 그대로 로드 가능. gzip, bzip2, LZO, snappy, deflate
* 문서: <https://docs.databricks.com/external-data/index.html#data-formats-with-special-considerations>
  + LZO는 라이센스 이슈로 코덱 별도 설치 필요: https://docs.gcp.databricks.com/external-data/lzo.html
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **%sql**  **CREATE TEMPORARY TABLE t16**  **USING CSV**  **OPTIONS (path "dbfs:/FileStore/tables/KBPOC/data\_org\_csv.gz", header "true")**  **;**  **SELECT \* FROM t16;** |
| --- |

### 1.7. JDBC/ODBC connection DB → "대상 솔루션" 데이터 적재 및 확인

* 지원 여부: 지원
* 문서: https://docs.databricks.com/external-data/jdbc.html
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

| **employees\_table = (spark.read**  **.format("jdbc")**  **.option("url", "<jdbc\_url>")**  **.option("dbtable", "<table\_name>")**  **.option("user", "<username>")**  **.option("password", "<password>")**  **.load()**  **.write.mode("overwrite")**  **.saveAsTable("<target\_table>")**  **)** |
| --- |

[추가]

* Create table using JDBC
* 문서: <https://docs.databricks.com/external-data/jdbc.html#language-sql>

| **CREATE TEMPORARY VIEW employees\_table\_vw**  **USING JDBC**  **OPTIONS (**  **url "<jdbc\_url>",**  **dbtable "<table\_name>",**  **user '<username>',**  **password '<password>'**  **)** |
| --- |

### 1.8. JDBC/ODBC connection DB: euc-kr 인코딩 데이터 utf-8 저장

* 지원 여부: 지원
* 상세:
* 예제: [poc1 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018277/command/3641897765018278)

### 1.9. JDBC/ODBC connection DB: EBCDIC 등의 인코딩 지원 여부

* 지원 여부: 미지원
* 상세:
* 문서:
* 예제:

### 2.1. 계정 신청(추가) 및 승인 (개인계정, 공용계정)

* 지원 여부: 관리자가 사용자, 그룹, Service principal 추가 가능
* 상세: 계정 신청이나 승인 프로세스를 별도 포털에서 진행 후, 3.3과 같이 API 로 계정 추가 가능
* 문서: <https://docs.databricks.com/administration-guide/users-groups/users.html>
* 예제:
  + Account 레벨의 사용자 추가: Account 콘솔 > User management > Users > Add user
  + Workspace 레벨의 사용자 추가: Workspace > Admin Settings > Users > Add user
  + SCIM 이용하여 IDP로부터 계정 동기화 : <https://docs.databricks.com/administration-guide/users-groups/scim/index.html>

### 2.2. 계정 신청(추가), 승인 등의 API 지원 여부

* 지원 여부: 지원
* 상세: Account API 로 계정 추가 가능
* 문서: <https://docs.databricks.com/dev-tools/api/latest/scim/account-scim.html#operation/create-one-user>
* 예제: [poc2 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018247/command/3641897765018248)

### 2.3. 별도의 계정 생성 지원 (IAM과 무관)

* 지원 여부: 지원
* 상세: 데이터브릭스 account 에 별도 계정 생성 가능. 이 계정은 IAM 과 무관함.
* 문서: <https://docs.databricks.com/administration-guide/users-groups/users.html>

### 2.4. 내부 SSO 및 2차 인증 연동 (2.4~2.6)

* 지원 여부: 지원 (고객 상황에 따라 KB 내부 SSO와의 인증은 진행하지 않기로 함)
* 상세:
  + SSO for account console : OIDC 또는 SAML 2.0 이용.
  + SSO for Workspace: SAML 2.0 이용.
  + 2.5. 내부 SSO 로그인 : 사용중인 IDP 연결 정보로 테스트 진행
  + 2.6. ID & PW 로그인 : 로그인 가능
* 문서: <https://docs.databricks.com/security/auth-authz/index.html>

### 2.7. 다단계 인증(MFA)

* 지원 여부: SSO 연동시 IDP 에서 제공하는 MFA 이용

### 2.8. 비밀번호 변경 정책 설정 여부

* 지원 여부: 자체 비밀번호 정책 제공. 설정 기능은 현재 없음.
* 상세: 비밀번호 변경시, 길이 8자 이상이어야 하며, 영문 대문자, 영문 소문자, 숫자, 특수문자를 모두 하나 이상 포함해야 함. 비밀번호 길이, 난이도, 사용기한을 설정하는 기능은 현재 제공되지 않음.

### 

### 

### 3.1. 테이블 권한 신청 및 승인

* 지원 여부: 사용자, 그룹, Service Principal 에 대한 테이블 권한 부여 가능
* 상세: 권한 신청 및 승인 절차는 별도 포털에서 진행 후, 3.3과 같이 API 호출 방식으로 처리 가능
* 문서: <https://docs.databricks.com/sql/language-manual/sql-ref-privileges.html>

### 3.2. 컬럼별 권한 신청 및 승인

* 지원 여부: Column level mask 기능 사용 (Dynamic View 통해서도 구현 가능)
* 상세: 특정 컬럼에 대해 쿼리 실행 시점에 마스킹 함수를 적용하여 권한이 없는 사용자에게는 실제 값 대신 마스킹된 값을 보여줄 수 있음
* 문서:  
  https://docs.databricks.com/aws/en/tables/row-and-column-filters#apply-a-column-mask  
  <https://docs.databricks.com/data-governance/unity-catalog/create-views.html#dynamic-view>
* 예제: [poc3 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018227/command/3641897765018228)
* Note: Column level masks 기능 사용

-- Create a SQL user defined function.

CREATE FUNCTION ssn\_mask(ssn STRING)

RETURN IF(IS\_MEMBER('admin'), ssn, "\*\*\*\*");

-- Apply the new function to a table as a column mask.

-- Future users' queries from the 'ssn' column then return

-- altered values.

CREATE TABLE users (region STRING, table\_ssn STRING) USING delta;

ALTER TABLE users ALTER COLUMN table\_ssn SET MASK ssn\_mask;

-- Disable the column mask. Future users' queries from the 'ssn'

-- column then return original values.

ALTER TABLE users ALTER COLUMN ssn DROP MASK;

-- Create another table with the new function applied as a column

-- mask in one step, as part of the create table statement.

-- Future users' queries from the 'ssn' column then return

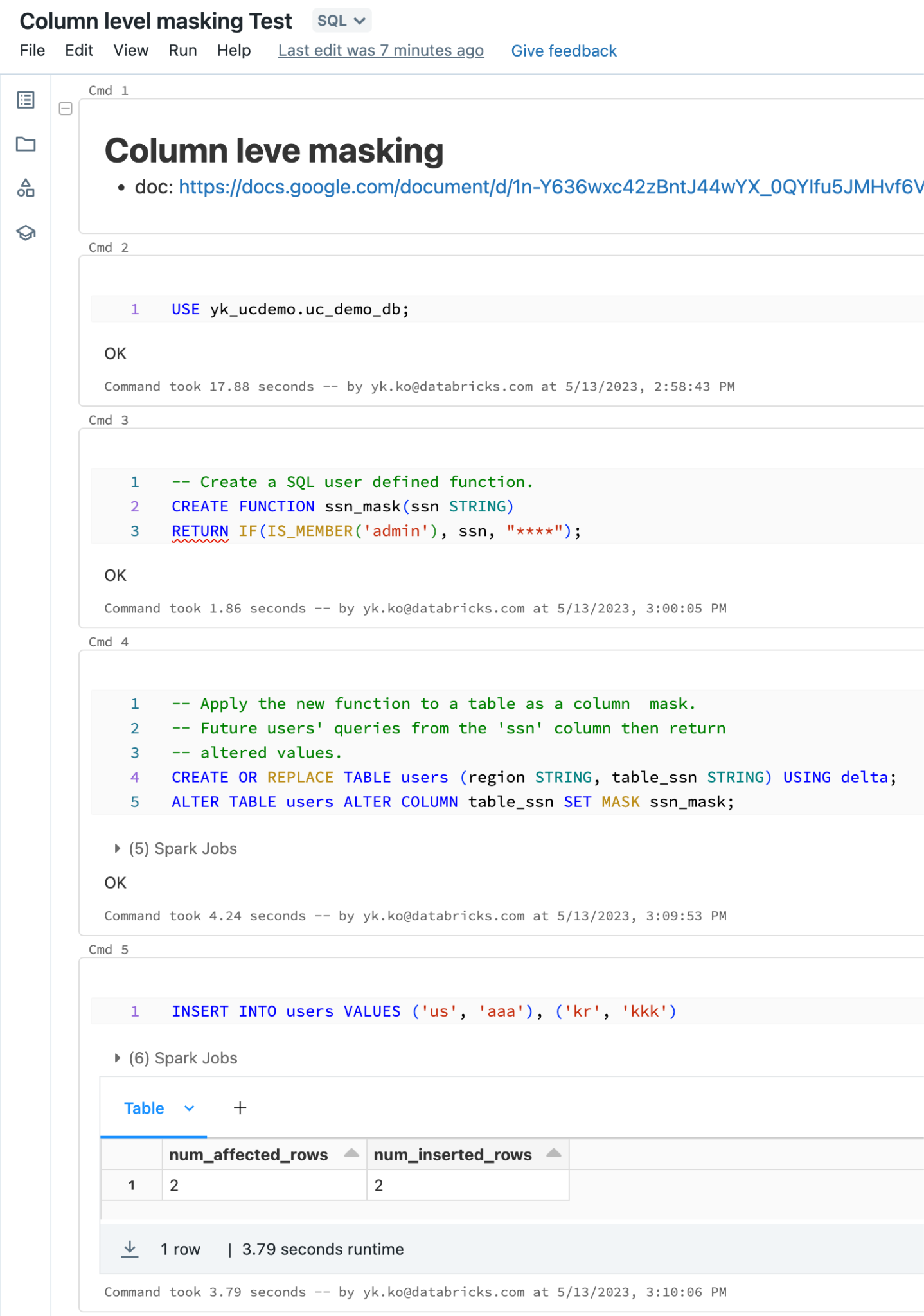
-- altered values.

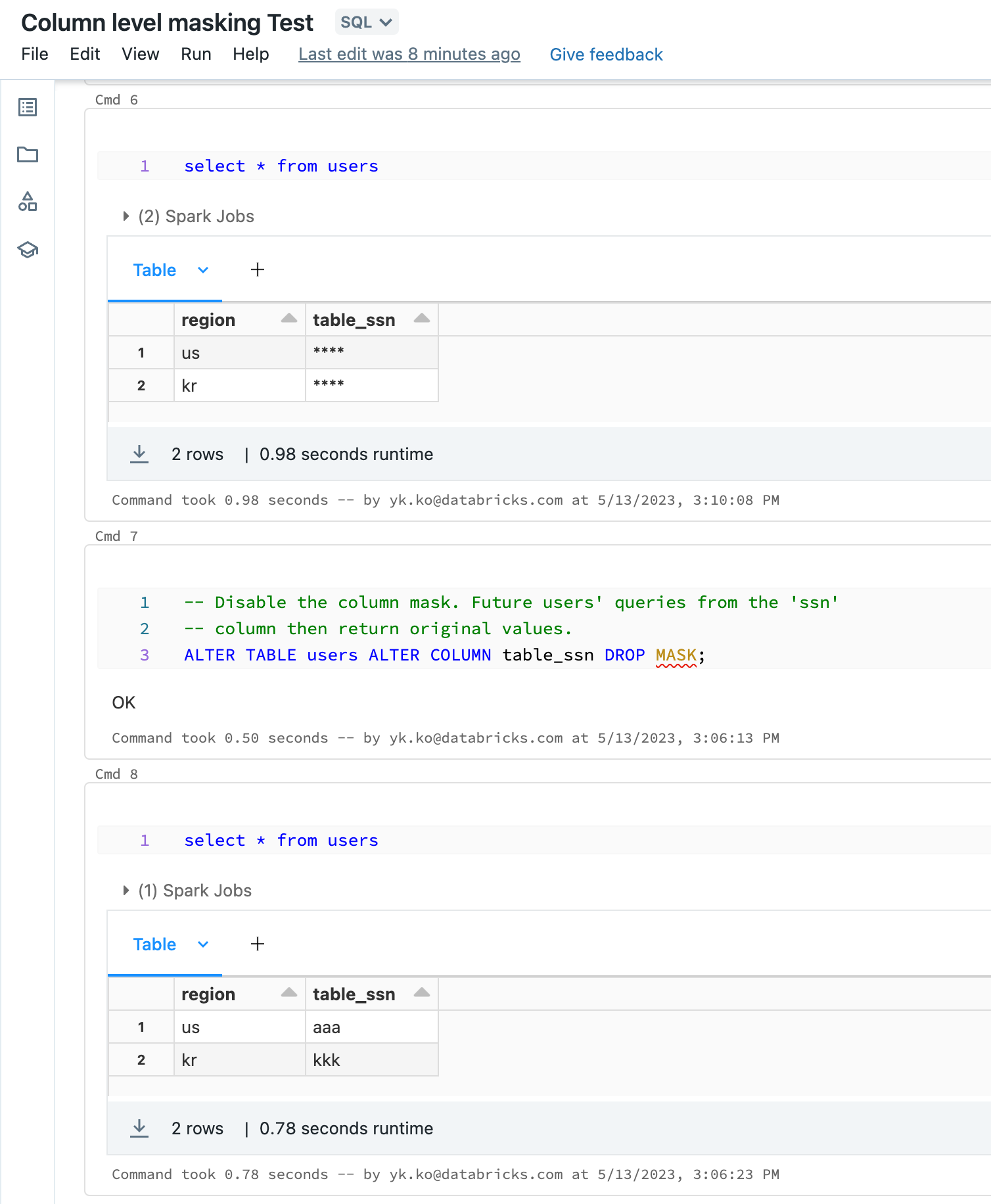
CREATE TABLE region\_to\_ssn (

region STRING,

ssn STRING MASK ssn\_mask)

USING delta;





### 3.3. API 기능 제공 여부

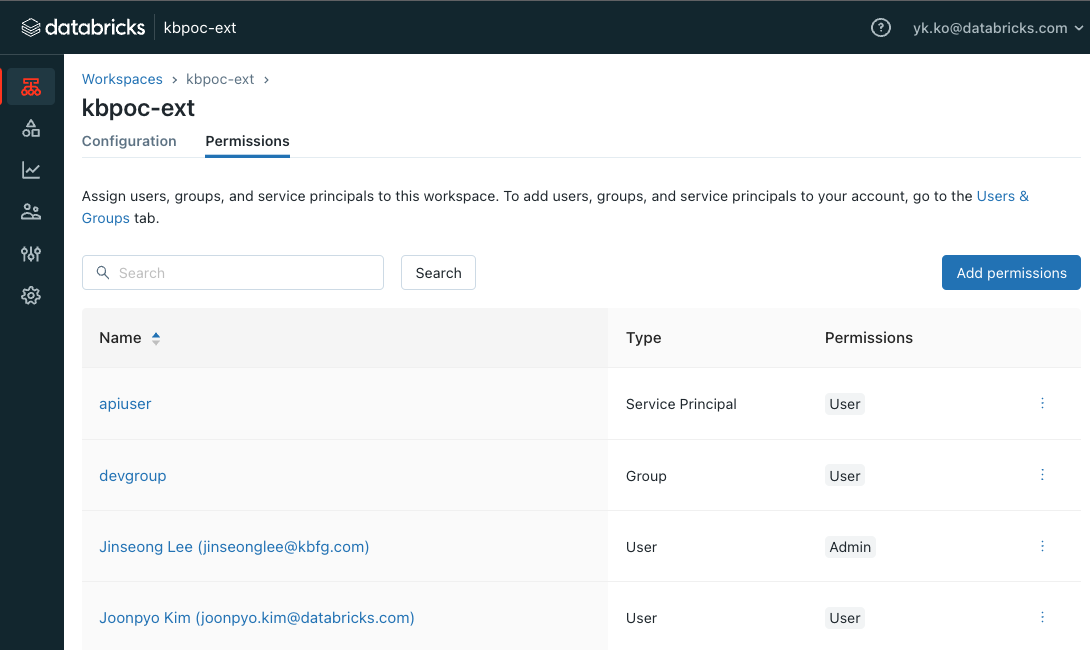
* 지원 여부: 제공
* 상세: Unity Catalog API 의 Update permissions 이용
* 문서: <https://docs.databricks.com/api-explorer/workspace/grants/update>
* 예제: [poc3 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018227/command/3641897765018228)

### 3.4. External table 생성 시 S3 직접 접근 가능 여부

* 지원 여부: 요구 기능 지원 (S3 File access 가 직접 되면 안됨)
* 상세: 테이블 생성시 storage credential 을 지정하거나, External Location (storage credential 정보 포함)을 지정해야 함.
* 문서: <https://docs.databricks.com/data-governance/unity-catalog/manage-external-locations-and-credentials.html>
* 예제: [poc3 노트북](https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/3641897765018227/command/3641897765018228)

### 3.5. 가상환경 별 관리자 기능 여부

* 지원 여부: 지원
* 상세: workspace 별로 관리자를 지정할 수 있음.
* 문서: <https://docs.databricks.com/administration-guide/index.html>



### 4.1. 사용자 자원 통제 > 가상환경 별 최소~최대 사용량 설정 가능 여부

* 지원 여부: 지원
* 상세: 사용자나 그룹별로 cluster policy를 적용하여 클러스터 자원의 크기, 인스턴스 타입, 비용(시간당 DBU) 등을 제한할 수 있음. 특정 워크스프페이스 및 태그 기반으로 월별 예산 설정 및 모니터링 가능(현재는 사용량을 제한하지는 않음)
* 문서:
  + <https://docs.databricks.com/administration-guide/clusters/policies.html>
  + <https://docs.databricks.com/aws/en/admin/account-settings/budgets>

### 4.2. 사용자 자원 통제 > 가상환경이 suspend 된 이후 jdbc 호출 시 resume 여부

* 지원 여부: 지원
* 상세: 클러스터가 종료 상태일 때 Can Restart 권한이 있는 유저가 JDBC로 연결 시도하면 클러스터가 자동으로 재실행됨.
* 문서: https://docs.databricks.com/integrations/jdbc-odbc-bi.html

### 4.3. 데이터 저장량 통제 > 사용자별, 스키마별 데이터 저장 공간 할당 및 제한 여부

* 지원 여부: 지원
* 상세: 스키마별 저장공간을 external location 과 storage credential 을 이용하여 설정하고, 저장 공간 사용에 대한 권한과 제한은 GRANT 으로 관리할 수 있음.

USE CATALOG <catalog>;

CREATE { DATABASE | SCHEMA } [ IF NOT EXISTS ] <schema\_name>

[ MANAGED LOCATION '<location\_path>' ]

[ COMMENT <comment> ]

[ WITH DBPROPERTIES ( <property\_key = property\_value [ , ... ]> ) ];

GRANT CREATE EXTERNAL TABLE ON STORAGE CREDENTIAL <storage\_credential\_name> TO <principal>;

GRANT READ FILES ON STORAGE CREDENTIAL <storage\_credential\_name> TO <principal>;

* 문서:
  + <https://docs.databricks.com/data-governance/unity-catalog/create-schemas.html>
  + <https://docs.databricks.com/data-governance/unity-catalog/manage-external-locations-and-credentials.html#manage-permissions-for-a-storage-credential>

### 4.4. 접근 및 사용이력 관리 > 로그인, 쿼리 CRUD 작업 이력 관리 여부

* 지원 여부: 지원
* 상세: audit log 설정을 통해 login, create table 등의 이벤트에 대한 로그를 확인 가능
  + Audit log Event 종류: <https://docs.databricks.com/administration-guide/account-settings/audit-logs.html#auditable-events-account>
* 문서: <https://docs.databricks.com/data-governance/unity-catalog/audit.html>
* 예제: 문서내 노트북 제공

### 4.5. 접근 및 사용이력 관리 > 이력 저장 기간 관리 여부

* 지원 여부: audit log는 기간제한 없이 저장되며, 보존기간 적용 필요시 삭제 로직 적용 가능
* 상세:
  + audit log는 지정된 S3 경로에 저장됨. <bucket-name>/<delivery-path-prefix>/workspaceId=<workspaceId>/date=<yyyy-mm-dd>/auditlogs\_<internal-id>.json
  + Audit log 보존기간을 적용해야 할 경우, batch 삭제 또는 s3 lifecycle 정책 등으로 보관 기한을 조정 가능.
* 문서: <https://docs.databricks.com/administration-guide/account-settings/audit-logs.html>

### 4.6. 접근 및 사용이력 관리 > 로그 조회 기능

* 지원 여부: 지원
* 상세: audit log 조회 가능
* 문서: <https://docs.databricks.com/administration-guide/account-settings/audit-logs.html#analyze>
* 예제: 문서 참조

| **%scala**  **val df = spark.read.format("json").load("s3a://bucketName/path/to/auditLogs")**  **df.createOrReplaceTempView("audit\_logs")**  **%sql**  **SELECT DISTINCT userIdentity.email, sourceIPAddress**  **FROM audit\_logs**  **WHERE serviceName = "accounts" AND actionName LIKE "%login%"** |
| --- |

### 4.7. 접근 및 사용이력 관리 > 샤크라 맥스, DB Safer 등 지원 여부

* 지원 여부: 미지원
* 상세: 해당 도구들이 제공하는 기능은 다른 방식으로 제공할 수 있음.

### 

### 5.1. 암호화 컬럼 관리

* 지원 여부: 요구 기능 확인 필요 (암호화 적용된 컬럼 목록을 메타스토어에서 관리하는 기능?)
* 상세:
* 문서: <https://www.databricks.com/blog/2020/11/20/enforcing-column-level-encryption-and-avoiding-data-duplication-with-pii.html>
* 예제: <https://www.databricks.com/notebooks/enforcing-column-level-encryption.html>

### 5.2. 제공 암호화 알고리즘

* 지원 여부: 지원
* 상세: aes\_encrypt, aes\_decrypt 빌트인 함수에서 key 길이에 따라 AES-128, AES-192, AES-256 알고리즘을 제공
* 문서: https://docs.databricks.com/sql/language-manual/functions/aes\_encrypt.html

### 5.3. 암호화 키 관리

* 지원 여부: 지원
* 상세: Secrets API 를 이용하여 암호화 키를 관리할 수 있음. 암호화의 용도에 따라 AWS 의 KMS를 이용 가능.
* 문서: <https://docs.databricks.com/security/secrets/index.html>
* 예제: poc5 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/1846427390996847/command/1846427390996848>

### 5.4. 쿼리 기반의 특정 컬럼 암호화, 복호화

* 지원 여부: 지원
* 상세: Dynamic View 기능으로 구현 가능하며, Column level masking 기능으로 별도의 view 생성 없이도 컬럼 MASK 추가 가능
* 문서: <https://docs.databricks.com/data-governance/unity-catalog/create-views.html#dynamic-view>
* 예제: poc5 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/1846427390996847/command/1846427390996848>

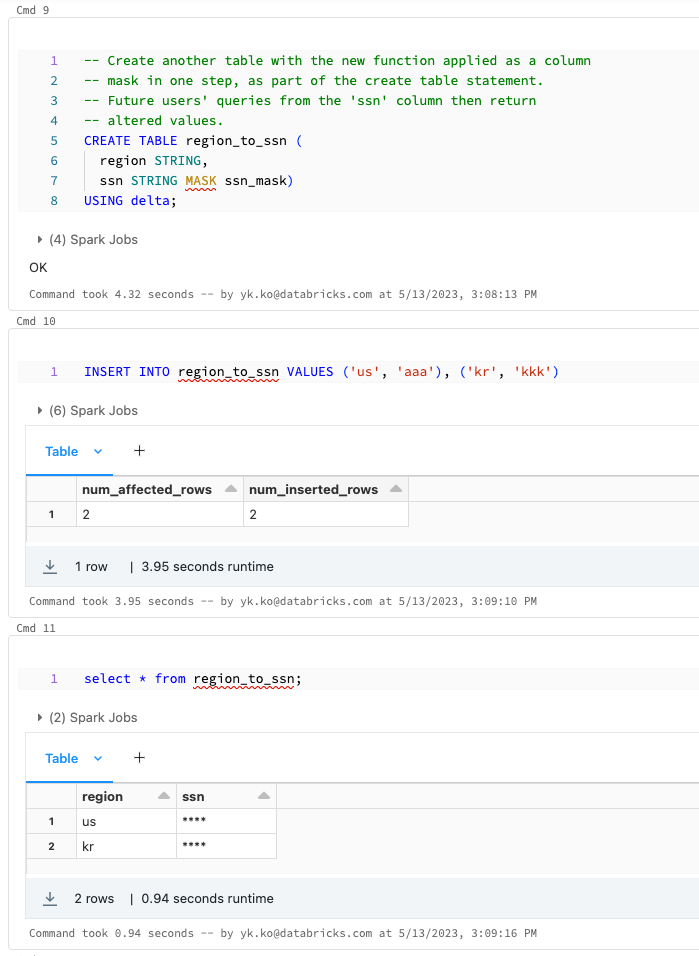
| -- 마스킹 함수 생성  CREATE FUNCTION mask\_ssn(ssn STRING)  RETURN CASE  WHEN is\_account\_group\_member('HumanResourceDept') THEN ssn  ELSE '\*\*\*-\*\*-\*\*\*\*'  END;  -- 테이블 생성 시 컬럼에 마스킹 적용  CREATE TABLE persons (  name STRING,  ssn STRING MASK mask\_ssn  );  -- 데이터 삽입  INSERT INTO persons VALUES ('James', '123-45-6789');   |
| --- |

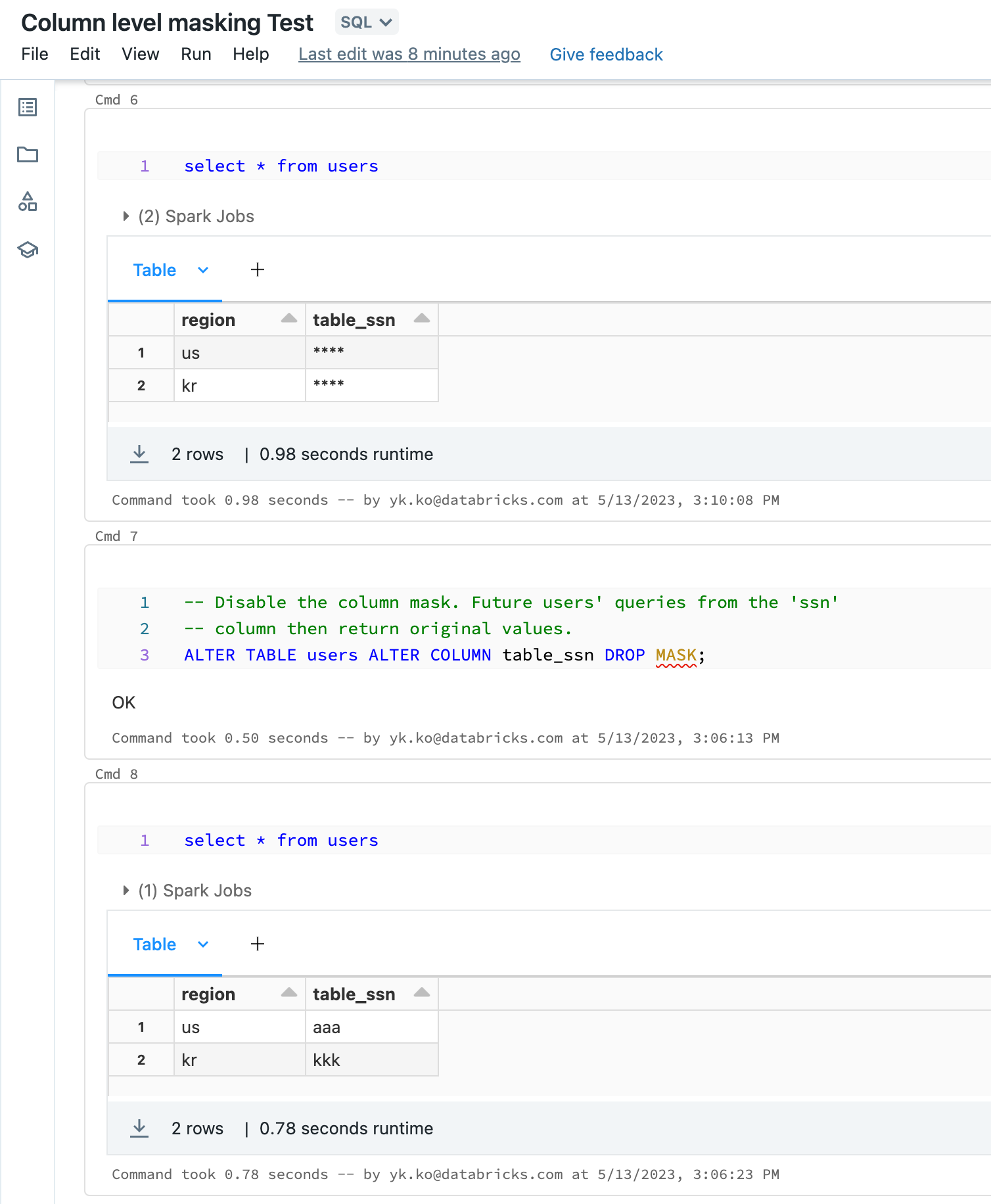
### 5.5. 사용자 복호화 권한 관리

* 지원 여부: 지원
* 상세: Dynamic View 나 Column level masking 이용시 사용자의 그룹 멤버십에 따라 권한 관리됨. 만약 암/복호화를 UDF로 구현한다면 Function 사용 privilege 로 권한 관리 가능.

### 5.6. 마스킹 정책 생성 후 alter, create로 반영 되는지 확인 (신규)

* 지원 여부: 지원
* 상세:
  + CREATE TABLE 시 컬럼에 MASK 지정 가능.
  + 또한 ALTER COLUMN DROP MASK로 mask 삭제 가능.



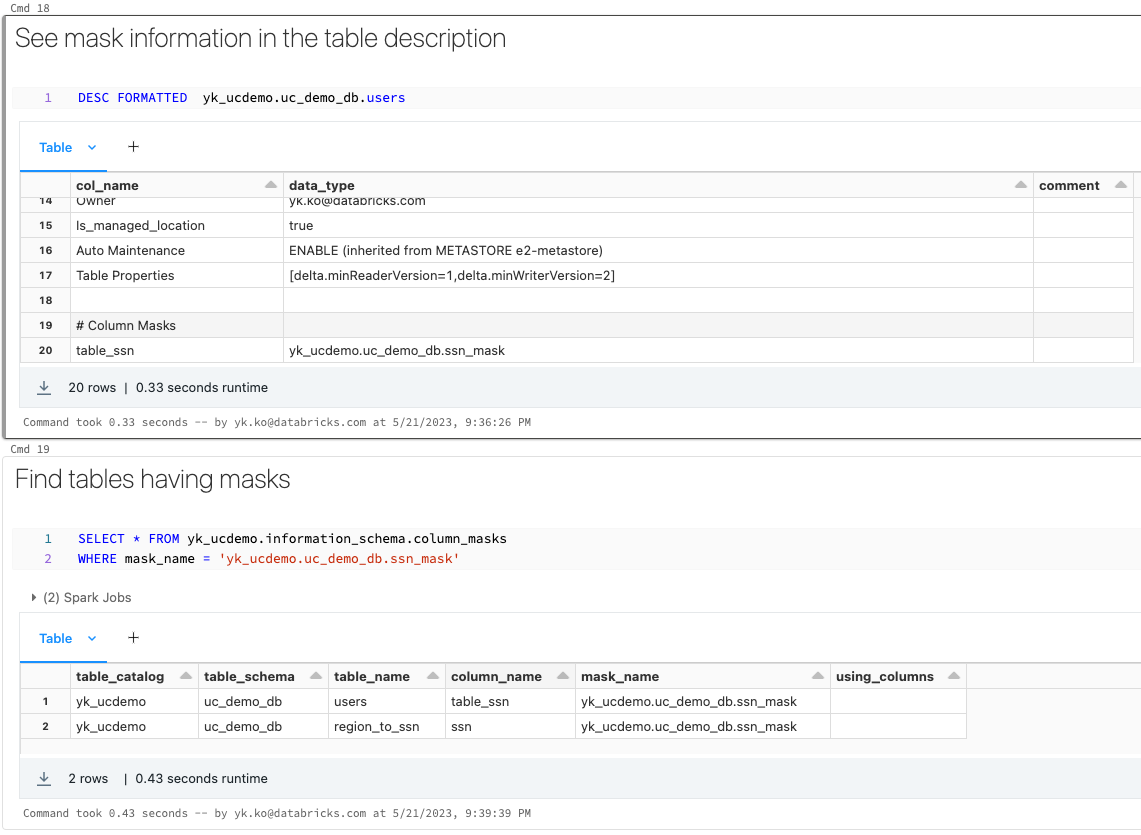


### 5.7. 마스킹 정책 변경 후 기존 테이블 반영되는지 확인 (변경)

* 지원 여부: 지원
* 상세: 3.2. 컬럼 레벨 마스킹 스크린샷 참조. DROP MASK 실행 후, 마스킹 없이 원래 값 조회됨.
* 문서:
* 예제:

### 5.8. 마스킹 정책이 적용된 테이블 확인

* 지원 여부: 지원
* 상세: information\_schema.column\_masks 에서 조회 가능. 또한, DESC FORMATTED TABLE 로 표시되는 테이블 상세정보에서도 #Column masks 섹션에서 확인 가능.



### 5.9. 외부 제공 시 적용 여부 확인

* 지원 여부: 부분 지원
* 상세: 마스킹 적용된 데이터를 export 또는 materialize 하여 제공할 수 있음. 그러나 이 경우 사용자별 마스킹 처리는 불가. (Delta Sharing 또한 Row/Column Masking 미지원)

### 5.10. 계정별 정책 적용 여부

* 지원 여부: 지원
* 상세: current\_user(), is\_member() 등의 멤버십 function 을 이용하여, 특정 사용자나 특정 그룹에게만 마스킹 정책을 적용 가능

### 6.1. cold 데이터 자동 백업 및 사용 시 복구

* 지원 여부: 지원
* 상세:
  + S3 Lifecycle Policy을 이용하여 s3 storage tier 이동 정책을 적용하거나, S3 intelligent-tiering 을 사용하여 cold 데이터에 대한 아카이빙을 적용할 수 있음.
  + Cold 데이터 저장을 위한 S3 storage tier 에서 아카이브된 객체 복원 시간이 소요되는 클래스들이 있음. 비용과 복원 소요시간을 고려하여 스토리지 클래스를 선택할 수 있음.
  + (권고) 현재 Intelligent Tiering 대신 정적인 Storage Lifecycle Policy 사용을 권고함: SELECT \* LIMIT 수행시 old file들이 샘플링 될 수 있어, intelligent tiering에서 colder tier 에 있는 파일들이 warm tier로 이동하여 비용이 증가할 수 있음 (관련 기능 개발중).
  + (권고) Delta lake 에서는 data expiration rule 을 사용하지 않기를 권고. Transaction Log와 mismatch 가 발생하여 data corruption 가능
* 문서: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/object-lifecycle-mgmt.html>

### 6.2. 데이터 백업 및 복구

* 지원 여부: 지원
* 상세:
  + Delta lake의 time travel 기능을 이용하여, 과거 버전 또는 특정 시점으로 복구할 수 있음.
  + <https://docs.databricks.com/delta/history.html>
  + 또는, Delta lake 의 Deep clone 기능을 이용하여 백업 또는 아카이빙 용도로 이용 가능.
  + <https://docs.databricks.com/delta/clone.html>
* 예제: (노트북) poc6 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/2503819615984510/command/2503819615984511>

### 6.3. 테이블 drop 후 rollback 가능 여부, 기간

* 지원 여부: 테이블 유형에 따라 다름
* 상세: External table 의 경우 테이블을 drop해도 데이터는 유지되므로 테이블 재생성 가능. Managed table 의 경우 테이블 drop 시 데이터도 삭제되므로 롤백 불가.
* 문서: <https://docs.databricks.com/sql/language-manual/sql-ref-syntax-ddl-drop-table.html>

### 6.4. 데이터 삭제 후 rollback 가능 여부, 기간

* 지원 여부: 지원
* 상세: Delta 테이블의 경우 Vacuum 기간 내에는 time travel 통한 롤백 가능
* 문서: <https://docs.databricks.com/delta/history.html>

### 6.5. 테이블별 삭제 정책 적용 및 기간 정의 (TTL, record)

* 지원 여부: 지원
* 상세:
  + Delta 테이블의 경우 파티션 컬럼 기준의 delete 는 drop partition 처럼 효율적으로 동작함. 그러므로, 주기적으로 delete 수행하여 삭제 정책과 기간 적용 가능.
  + 또는 주기적으로 vacuum 수행하여 데이터 보존 기간 관리
* 문서: <https://docs.databricks.com/delta/vacuum.html>

### 6.6. 스키마, 테이블 폐기 이력 확인

* 지원 여부: 지원
* 상세:
* 문서: <https://docs.databricks.com/data-governance/unity-catalog/audit.html>
  + Audit 대상 이벤트: <https://docs.databricks.com/administration-guide/account-settings/audit-logs.html#auditable-events-account>

### 

### 7.1. 한글명 테이블 및 컬럼 생성, 변경

* 지원 여부: 지원
* 예제 노트북 (7.1 ~ 7.6 동일): <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/2503819615984525/command/2503819615984535>

CREATE TABLE pocdemo.uc\_demo\_db.`테스트1` (

`name` STRING ,

`번호` INT

)

;

INSERT INTO pocdemo.uc\_demo\_db.`테스트1` (`name`, `번호`) VALUES ('aaa', 1), ('홍길도',1), ('여의도', 2);

### 7.2. 전체 테이블 조회

* 지원 여부: 지원

SELECT \* FROM pocdemo.information\_schema.tables

### 7.3. 전체 컬럼 조회

* 지원 여부: 지원

SELECT \* FROM pocdemo.information\_schema.columns

WHERE table\_catalog='pocdemo'

AND table\_schema = 'uc\_demo\_db'

ORDER BY table\_name, ordinal\_position

### 7.4. 특정 테이블 내 컬럼명 조회

* 지원 여부: 지원

SELECT \* FROM pocdemo.information\_schema.columns

WHERE table\_catalog='pocdemo'

AND table\_schema = 'uc\_demo\_db'

AND table\_name = '테스트1';

### 

### 7.5. 특정 컬럼 포함 테이블 목록 조회

* 지원 여부: 지원
  + Workspace UI 내 통합 검색 혹은 쿼리 통한 특정 컬럼 조회 가능
* 특정 컬럼 포함 테이블 쿼리 조회

SELECT \* FROM pocdemo.information\_schema.columns

WHERE table\_catalog='pocdemo'

AND table\_schema = 'uc\_demo\_db'

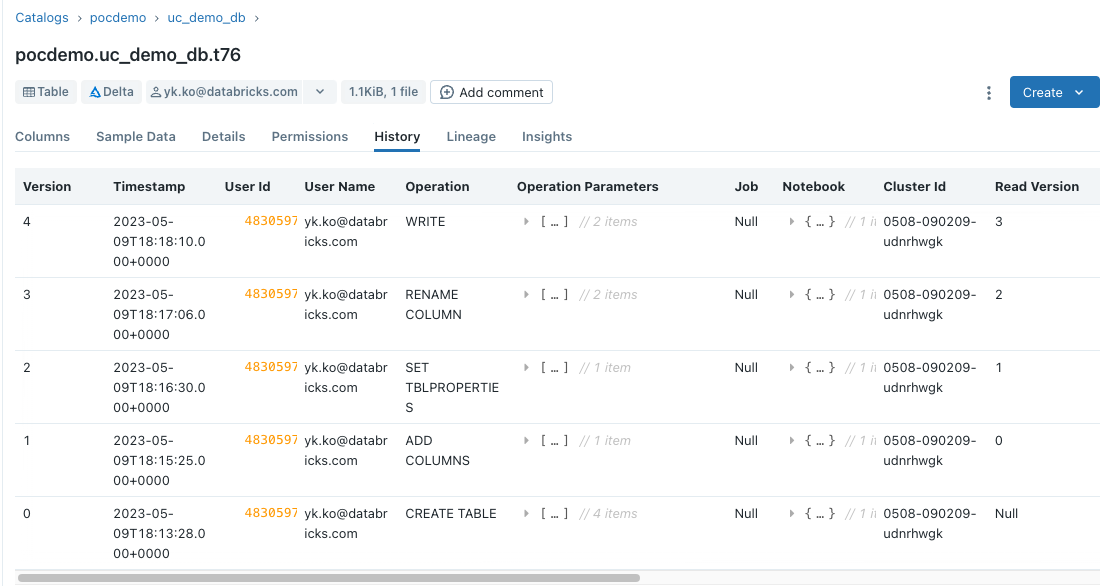
AND column\_name = 'num'

* Workspace UI 컬럼 검색



### 7.6. 테이블 및 컬럼 변경이력 조회

* 지원 여부: 지원
* DESC HISTORY tablename
* 또는 Workspace UI : Data explorer > table > History
* 예제: <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/2503819615984525/command/1846427390996862>



### 8.1. 쿼리 기반 수행 툴 여부

* 지원 여부: 지원
* SQL Editor, 대시보드 등 웹 UI 기반 쿼리 수행 환경 제공
* Tableau, PowerBI, Looker 등의 BI 툴들과 connector를 제공하며 ODBC/JDBC로 연동 가능
* 외부 쿼리 툴 (eg. DBeaver)과 ODBC/JDBC로 연동 가능

### 8.2. 분석환경 지원 여부 (ML,DL, R 등)

* 지원 여부: 지원
* 데이터 과학자를 위한 머신러닝 통합 분석 환경 제공 (AutoML, feature store, mlflow 등)
* ML 용 Databricks Runtime 으로 ML용 클러스터를 쉽게 생성
  + TensorFlow, PyTorch, keras 및 XGBoost, Horovod 등의 ML 인기 라이브러리 탑재
  + <https://docs.databricks.com/runtime/mlruntime.html>
* Python, Scala, R, SQL을 모두 사용할 수 있는 노트북 환경 제공
* R-studio 데스크탑에서 databricks 클러스터에 연결하거나, R-studio server를 databricks 클러스터에 셋업하여 사용 가능
  + <https://docs.databricks.com/sparkr/rstudio.html>

### 8.3. 분석 협업 제공 기능

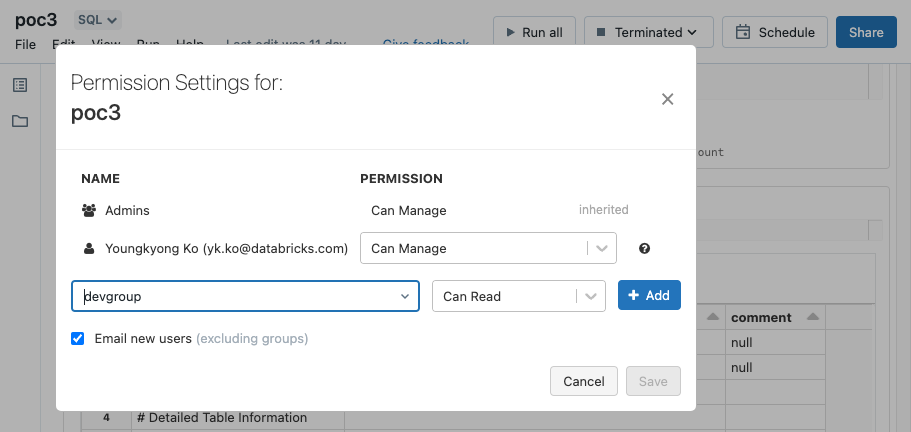
* 지원 여부: 지원
* 테이블, 노트북, 쿼리, 대시보드 등 데이터와 분석 asset 들을 간편하게 공유하고 권한 제어
* 노트북 환경에서 코멘트 달기, 동시 편집, 버전 관리(Revision history, restore version) 가능

### 8.4. 개발환경 및 코드 타 사용자 공유

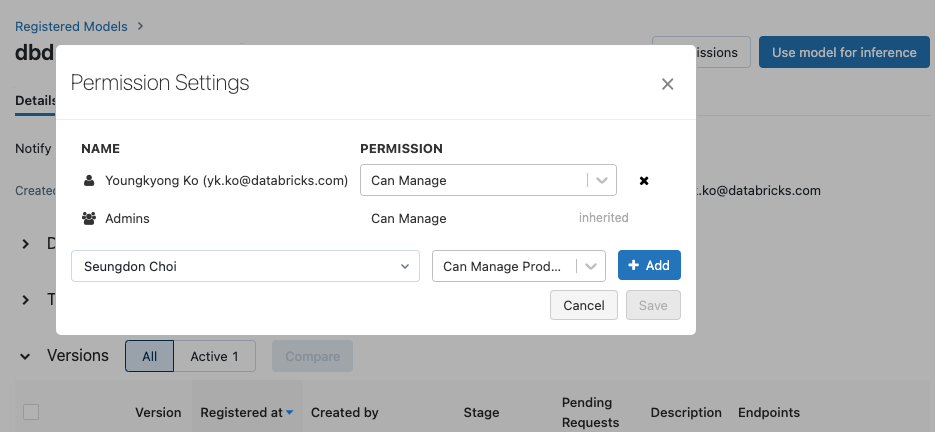
* 지원 여부: 지원
* 클러스터와 노트북 공유를 통해 여러 사용자가 동일한 개발 환경에서 협업 가능
* Git repository와 바로 연결하는 Repos 기능 제공
  + 다양한 git 서비스와 설치형 서버에 연결 (Github, Gitlab, Bitbucket, AWS CodeCommit, Azure DevOps 등)
* Repos를 이용하여 CI/CD 워크플로우와 연동
  + <https://docs.databricks.com/repos/ci-cd-techniques-with-repos.html>
* 노트북 및 라이브러리 코드를 dbc archive 등으로 export/import 하여 전달 가능.

### 8.5. 분석 데이터 타 사용자 공유

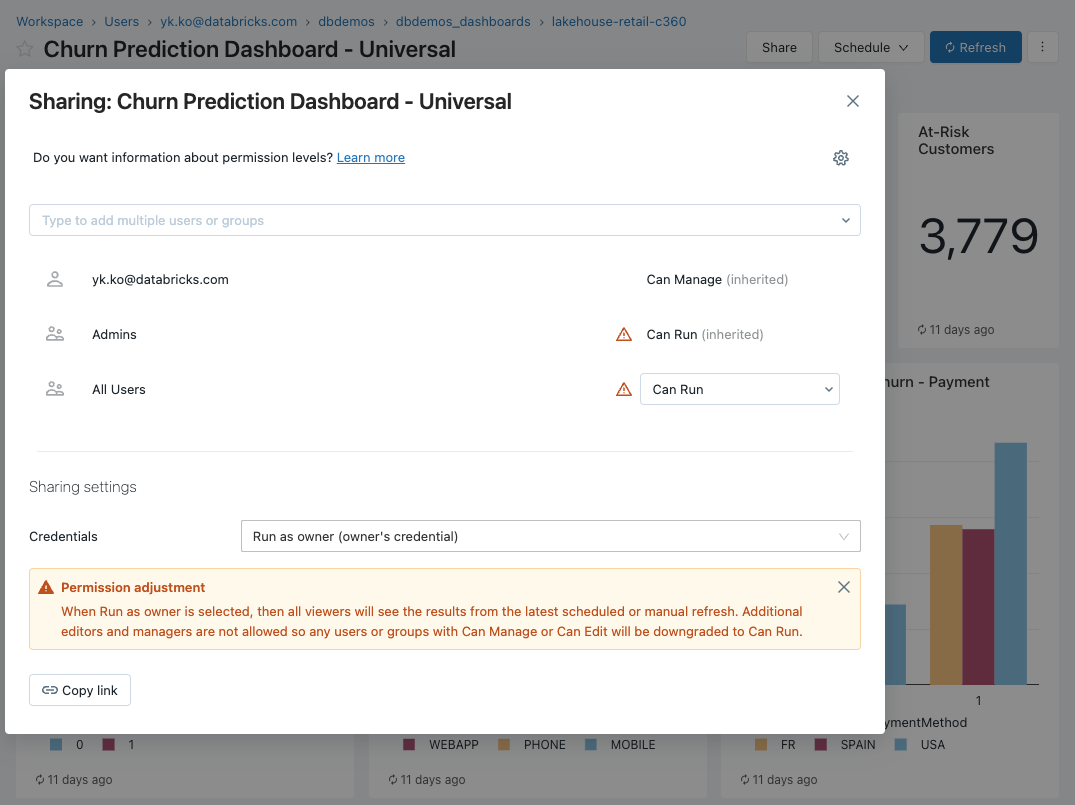
* 지원 여부: 지원
* 테이블을 사용자/그룹과 공유하고 read, modify 로 구분하여 권한 관리
* 노트북, 쿼리, 대시보드, ML 실험, ML model 등의 분석 결과물을 공유 & 권한 관리
* Ex. 노트북 공유



* ex) ML 모델 공유



* ex) 대시보드 공유



### 9.4. JDBC/ODBC 드라이버 지원 여부

* 지원 여부: 지원
* 향상된 ODBC/JDBC 드라이버 지원
* 대량의 결과셋을 클라우드 스토리지를 이용하여 병렬 전송하는 Cloud Fetch 기능 제공
* 문서 : ​​<https://docs.databricks.com/integrations/jdbc-odbc-bi.html>
  + ODBC driver download: <https://www.databricks.com/spark/odbc-drivers-download>
  + JDBC driver download: <https://www.databricks.com/spark/jdbc-drivers-download>

### 9.5. 지원 프로그래밍 언어

* 지원 여부: 지원
* 노트북 환경에서 Python, Scala, SQL, R 언어로 분석 가능
* 하나의 노트북에서 Cell 별로 다른 언어를 사용하여 작업 가능
* SQL Editor, Dashboard 등에서는 SQL 분석 환경 제공

### 9.6. 태블로, 쿼리원 지원 여부

* 지원 여부: 태블로 지원, 쿼리원 미지원
* Tableau Connector 를 제공 (Warehouse cluster 선택 > Connection details 탭
  + Tableau desktop, Databricks ODBC driver 필요
  + 인증방법: Personal Access Token(권장), username/password, OAuth
  + 문서: <https://docs.databricks.com/partners/bi/tableau.html>
* 쿼리원 툴에서 databricks 클러스터에 연결 가능한지는 해당 벤더에 확인
  + .NET 기반의 ODBC 사용 가능

### 9.7. DataStage, TeraStream 등 지원 여부

* 지원 여부: 지원
* DataStage : 데이터브릭스의 JDBC 드라이버 Jar를 등록 후 사용
  + 문서 : https://www.ibm.com/docs/en/iis/11.7.0?topic=SSZJPZ\_11.7.0/com.ibm.swg.im.iis.conn.jdbc.usage.doc/topics/jdbc\_config\_driver.html
* TeraStream : 데이터브릭스의 ODBC 드라이버를 TeraStream 에 설치 후 사용

### 10.1. 배치 정기 및 비정기 Job 생성 - 쿼리기반 및 etl 툴(airflow, nifi) 호출

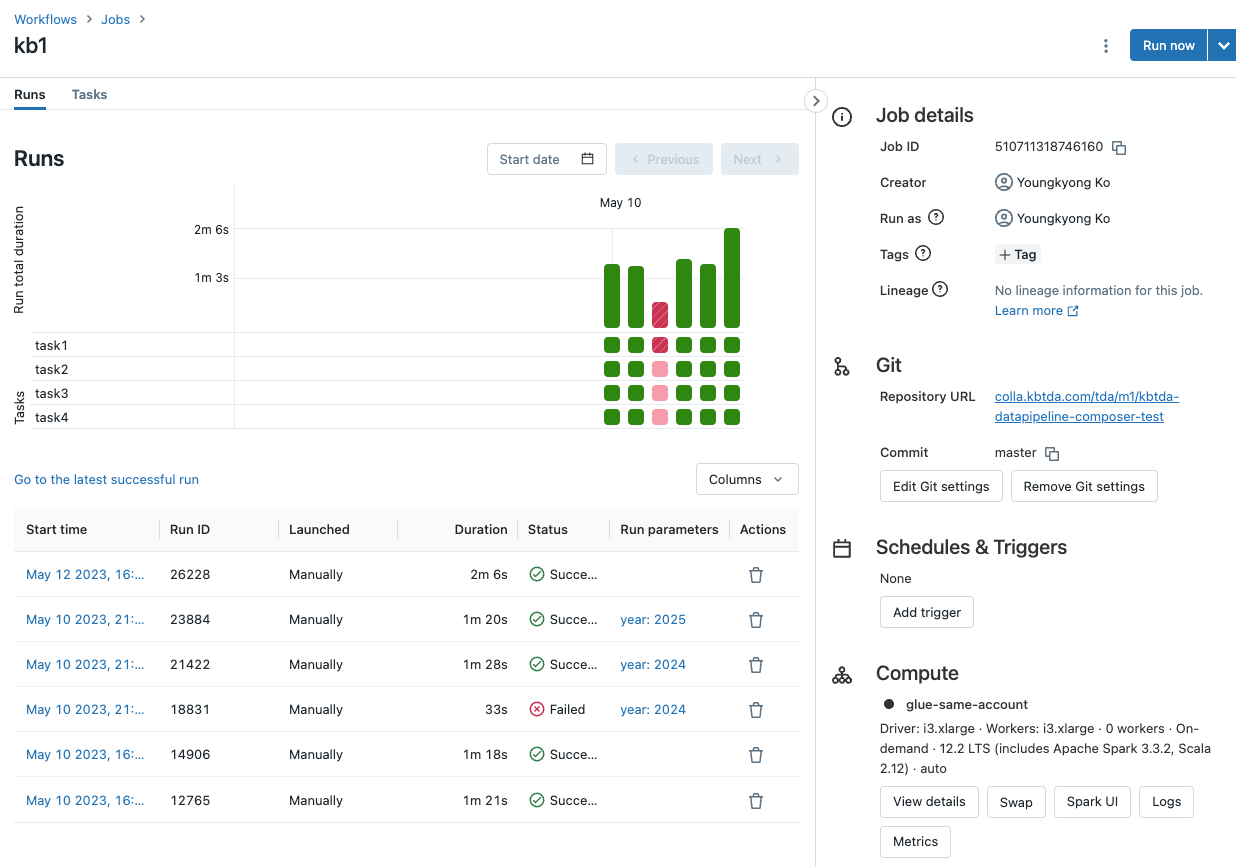
* 지원 여부: 지원
* 완전 관리형 오케스트레이션 서비스인 Workflows 를 탑재하여, 오케스트레이션을 위한 별도의 시스템을 운영하지 않고도 작업 스케줄과 운영, 관리 가능
* 배치 스케줄, 신규 파일 유입, 상시 유지 의 3가지 트리거 방식 제공
* Airflow 에서 operators for databricks 를 통해 job 호출 가능
  + 문서: <https://docs.databricks.com/workflows/jobs/how-to/use-airflow-with-jobs.html>
  + Airflow Databricks Operators: <https://airflow.apache.org/docs/apache-airflow-providers-databricks/stable/operators/index.html>

### 10.2. JAS 연동을 위한 API 기능 제공

* 지원 여부: 지원
* Databricks Jobs API 를 이용하여, job 생성과 관리, 실행 등의 작업을 처리할 수 있음. (job 생성, 삭제, 설정, 실행, 실행취소, 실행 목록, 실행결과, repair 등)
* Jobs API 문서: <https://docs.databricks.com/api-explorer/workspace/jobs>

### 10.3. 배치 스케쥴 모니터링 기능

* 지원 여부: 지원
* Workflows > jobs 에서 스케줄된 Jobs 리스트와 Last run 상태를 조회
* Workflows > Job runs 에서 최근 실행된 job 목록과 상태, 소요시간, parameter 등을 조회
* 개별 job 에서는 각 run 별 상세 실행기록을 조회

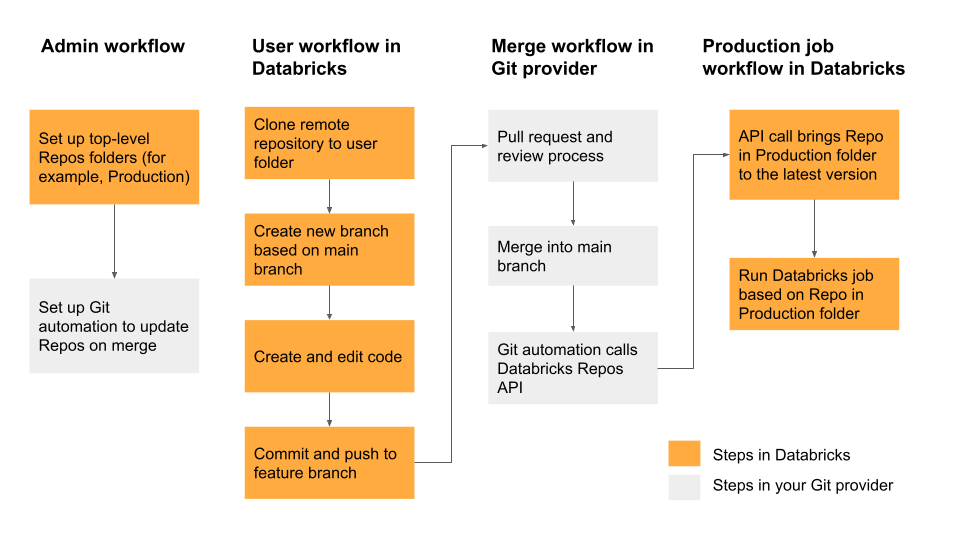


### 10.4. 오류 시 알람 발송 (메일 or sms)

* 지원 여부: 지원
* Job 설정에서 작업 시작시, 성공시, 실패시 등 트리거 조건을 지정하여 Notification 설정 가능
* 알림 대상으로 Email, Slack, webhook 등을 지정할 수 있음
* Admin settings > Notification destinations 에서 Email, Slack, Webhook, PagerDuty, MS Teams 설정 가능
* 이메일 발송은 Databricks Control Plane에서 이루어지며, 수신 대상 시스템에서는 Databricks Control Plane IP 허용 필요
* 문서:
  + <https://docs.databricks.com/sql/admin/notification-destinations.html>
  + https://docs.databricks.com/gcp/en/admin/workspace-settings/notification-destinations

### 11.1. 플랫폼 운영자 배포 파이프라인 지원 (CI/CD)

* 지원 여부: 지원
* Git 레포지토리와 연동하여 CI/CD 파이프라인과 연계할 수 있음
* 운영자는Repos API 와 Github Actions를 이용하여 [배포 자동화 설정](https://docs.databricks.com/dev-tools/ci-cd/ci-cd-github.html) 가능.
* [Jenkins 를 이용한 CI/CD 파이프라인](https://docs.databricks.com/dev-tools/ci-cd/ci-cd-jenkins.html)을 구성할 수도 있음
* 문서: <https://docs.databricks.com/repos/ci-cd-techniques-with-repos.html>



### 11.2. 분석가 배포 파이프라인 지원 (CI/CD)

* 지원 여부: 지원
* 11.1 의 CI/CD 기능을 사용
* SQL/BI 분석가들은 Workspace, Queries, Dashboard 기능을 이용하여 소스코드를 보관하고 공유 가능

### 11.3. 로컬 개발환경 지원 여부

* 지원 여부: 지원
* Databricks Connect V2 를 설치하여 VS Code, PyCharm, Eclipse 등의 IDE 에서 원격 Spark cluster (DBR 13.0+) 에 연결하여 Spark Job 을 실행하고 step-through 디버깅 가능
* 문서: <https://docs.databricks.com/dev-tools/databricks-connect.html>
* SQL/BI 사용자들은 ODBC/JDBC 드라이버를 이용하여 접속하거나, 각 BI 툴 전용 connector 를 이용하여 SQL warehouse 에 접속할 수 있음.
* SQL을 사용하는 개발자들은 VS Code Extension 인 [Databricks Driver for SQLTools](https://marketplace.visualstudio.com/items?itemName=databricks.sqltools-databricks-driver) 를 이용하여 로컬에서 SQL 코드를 개발하고 원격 클러스터에 실행 가능 (Public Preview)
  + https://docs.databricks.com/aws/en/dev-tools/sqltools-driver

### 11.4. 테스트 및 배포

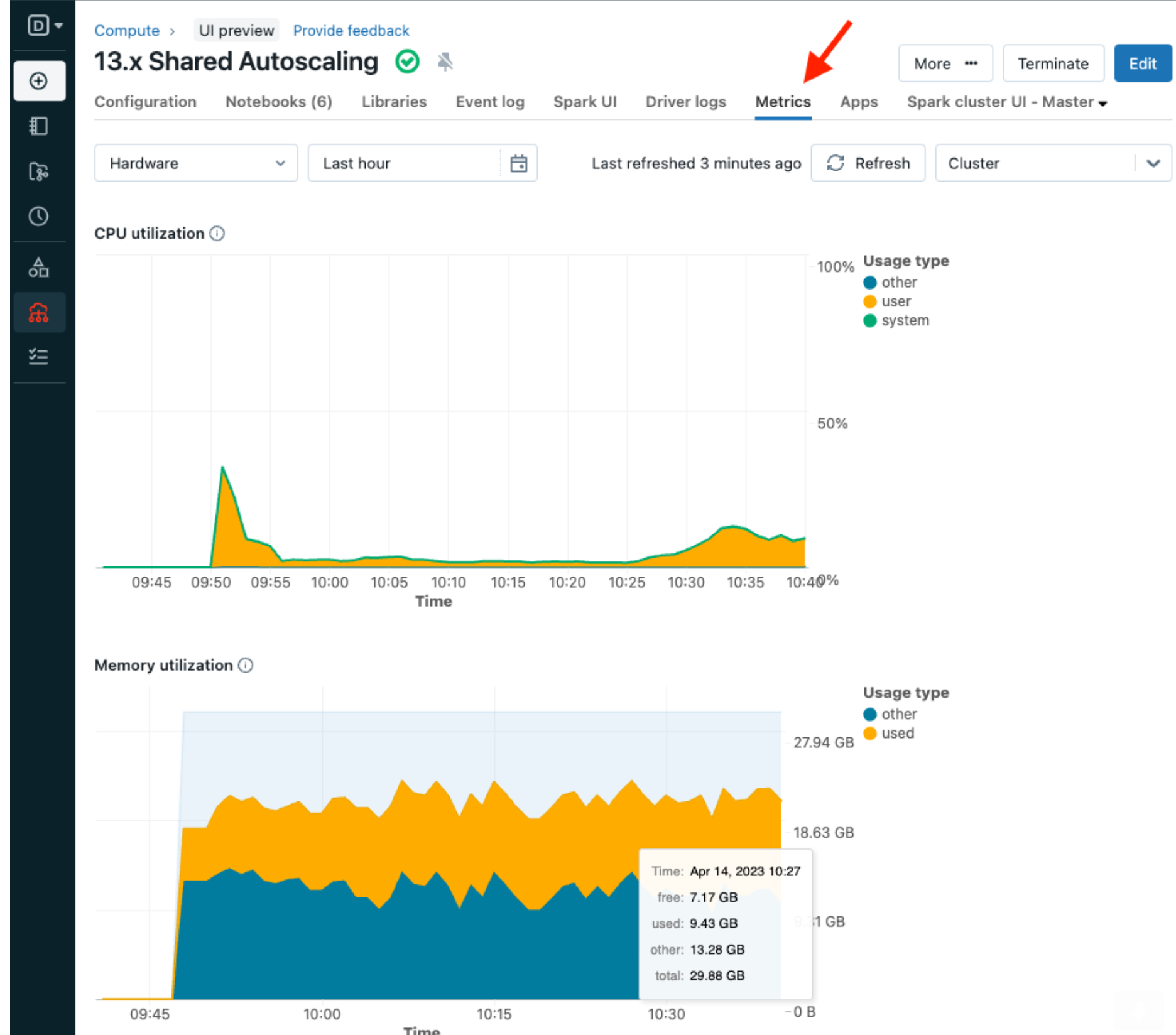
* 지원 여부: 지원
* 노트북 코드에 대한 unit test 를 %run 이나 Repos 를 이용하여 수행할 수 있음.
  + 문서: <https://docs.databricks.com/notebooks/test-notebooks.html>
    - Notebook Unit testing : <https://docs.databricks.com/notebooks/testing.html>
  + 예제 노트북: poc11 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353902/command/402993171353906>
* Delta Live Tables 에서는 Unit Test용 companion 노트북을 추가하여, expectations 로 데이터에 대한 unit test 를 수행할 수 있음.
  + DLT Unit testing 예제 노트북: <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353854/command/402993171353859>
  + DLT unit testing 예제 파이프라인: <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#joblist/pipelines/1c11ce64-1b76-4598-8534-e5cc86235706/updates/684473ab-ecfd-4a3b-b3c8-969083b5c2a6>

### 12.1. 자원 및 리소스 사용 현황(H/W, Application)

* 지원 여부: 지원
* 각 클러스터별로 Spark UI 및 메트릭을 제공 (Cluster 선택 > Metrics 탭)
* DBR 버전 ~12.2 : ganglia 로 Live metric 과 과거 데이터 스냅샷 제공
* DBR 13 이후 자체 metric UI(prometheus endpoint도 제공해서 외부 모니터링 툴과 연동 가능)
* 클러스터/잡 수행 데이터를 기반으로 리소스 사용량이나 최적화 분석을 위한 overwatch tool도 적용해서 사용할 수 있습니다. <https://databrickslabs.github.io/overwatch/>
* 추후 금년도 하반기내에 별도의 system table 형태로 상기 클러스터들의 metric 을 테이블에서 확인할 수 있는 기능도 예정되어 있습니다.
* Ganglia (DBR ~12.2)



* 자체 Metric UI (DBR 13)

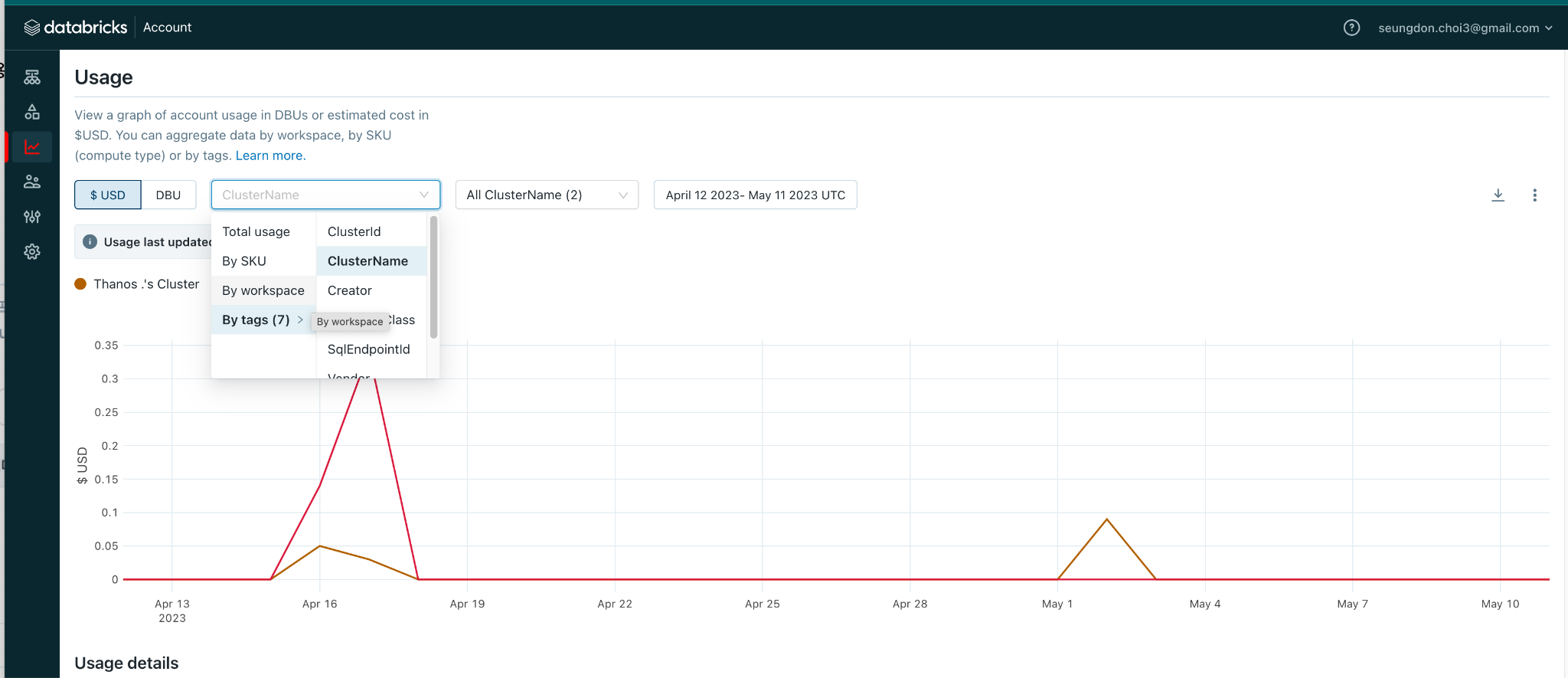


* Overwatch



### 12.2. 사용자별, 가상화 환경 별 비용 발생 현황

* 지원 여부: 지원
* Account console > Usage 에서 SKU, Workspace, Tag기반으로 cluster나 job별 비용 split 산정이 가능합니다. 기본적으로 vendor나 cluster명, owner등의 tag가 적용되고 별도로 custom tag를 넣어서 보통 부서별/작업별 비용산정에 이용 됩니다.



### 12.3. scale in, out 현황 모니터링

* 지원 여부: 지원
* DBR Cluster 에서는 Ganglia (DBR 12.2 이하) 또는 Cluster metrics (DBR 13)에서 확인
  + Cluster metrics 에서는 “number of active nodes” 그래프에서 확인
* DBSQL 에서는 SQL Warehouse 선택 > Monitoring 탭에서 Running Cluster 수의 추이를 확인

### 

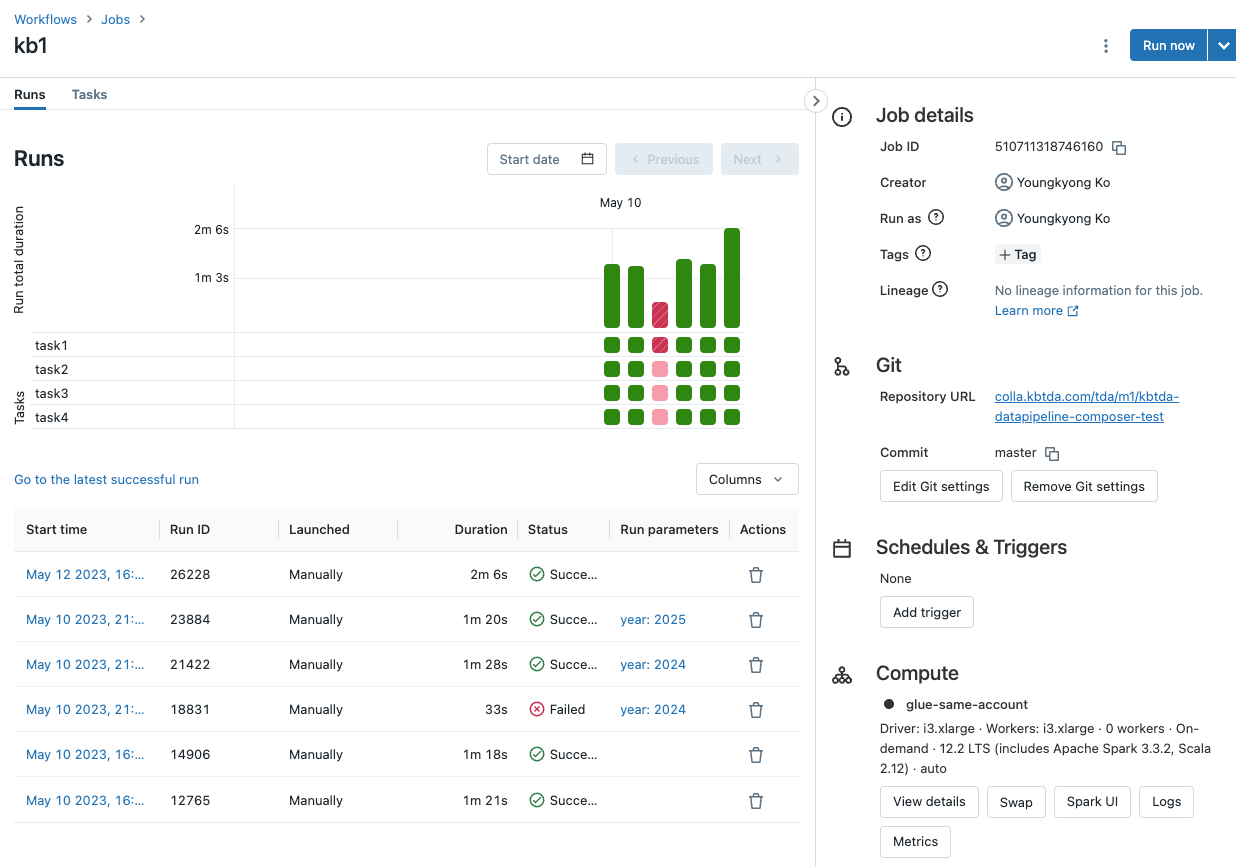
### 

### 12.4. 현재 사용 자원 상세

* 지원 여부: 지원
* Ganglia (DBR 12.2 이하) 또는 Cluster metrics (DBR 13)에서 Load average, CPU, Memory, Swap, Network, Disk 등에 대한 메트릭과 트렌드를 제공

### 12.5. Job 수행 및 처리(오류) 현황

* 지원 여부: 지원
* Workflows > jobs 에서 스케줄된 Jobs 리스트와 Last run 상태를 조회
* Workflows > Job runs 에서 최근 실행된 job 목록과 상태, 소요시간, parameter 등을 조회
* 개별 job 에서는 각 run 별 상세 실행기록을 조회



### 13.1. SQL 및 Python UDF 지원 여부

* 지원 여부: 지원
* SQL, Python, Scala UDF 제공
* 문서 : <https://docs.databricks.com/udf/index.html>
* 예제 노트북: poc13 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353730/command/402993171353748>

### 13.2. UDF 권한 부여

* 지원 여부: 지원
* FUNCTION 에 대해, legacy HMS 에서는 SELECT, Unity Catalog 에서는 EXECUTE 권한을 부여할 수 있음.
* Hive metastore 의 Privilege type: <https://docs.databricks.com/sql/language-manual/sql-ref-privileges-hms.html>
* Unity catalog 의 Privilege type : <https://docs.databricks.com/data-governance/unity-catalog/manage-privileges/privileges.html>
* 예제 노트북: poc13 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353730/command/402993171353755>

### 14.1. 테이블 column 타입 변경 및 추가

* 지원 여부: 지원
* Delta 테이블에서는 column mapping을 name으로 설정하면 컬럼명을 rename 할 수 있음.
  + 참고: <https://docs.databricks.com/delta/delta-column-mapping.html>
  + 예제 노트북 : poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353938>
* 컬럼 타입을 변경하려면 아래와 같이 overwriteSchema 옵션으로 rewrite 해야 합니다.
  + 참고: <https://docs.databricks.com/delta/update-schema.html#explicitly-update-schema-to-change-column-type-or-name>
  + 예제 노트북: poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353938>

### 14.2. 테이블 인코딩 변경 (요건 삭제 4/18)

### 14.3. 테이블 pk 지정 및 not null 지정

* 지원 여부: 지원
* NOT NULL, CHECK constraint 를 테이블에 추가할 수 있음. Delta 테이블에 적용.
* PK, FK 를 테이블의 Constraint 로 추가할 수 있음. Delta 테이블, Unity Catalog 에 적용.
* PK, FK 는 쿼리 실행 계획 최적화 등에 사용되는 정보성 기능이며, 강제성은 없음
* 문서:
  + <https://docs.databricks.com/aws/en/tables/constraints>
* 예제 노트북: poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353938>

### 14.4. DDL로 컬럼 변경 API지원 여부

* 지원 여부: 지원
* Execute SQL Statement API 를 이용하여 DDL문을 실행 가능
* 문서: <https://docs.databricks.com/api-explorer/workspace/statementexecution/executestatement>
* 예제 노트북: poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353958>

curl -X POST -k --header "Authorization: Bearer $DATABRICKS\_TOKEN" \

https://dbpoc-kbpoc-ext.cloud.databricks.com/api/2.0/sql/statements/ \

-d '{

"warehouse\_id": "2a28a21a4d8e39bc",

"statement": "ALTER TABLE pocdemo.uc\_demo\_db.ext34 RENAME COLUMN newname TO newname\_by\_api;",

"wait\_timeout": "0s"

}'

### 14.5. 지원 데이터 타입 확인 (varchar, decimal …)

* 지원 여부: 지원
* Spark SQL 데이터 타입 문서: <https://spark.apache.org/docs/latest/sql-ref-datatypes.html>
  + Varchar(n), Char(n), decimal(p,s) 타입 지원
* 예제 노트북: poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353958>

### 14.6. 지원되는 파일 형식 확인 (csv, txt, parquet,iceberg, delta …)

* 지원 여부: 지원
* 문서 : <https://docs.databricks.com/external-data/index.html>
* 지원 데이터 포맷
  + [Delta Lake](https://docs.databricks.com/delta/index.html)
  + [Delta Sharing](https://docs.databricks.com/external-data/deltasharing.html)
  + [Parquet](https://docs.databricks.com/external-data/parquet.html)
  + [ORC](https://docs.databricks.com/external-data/orc.html)
  + [JSON](https://docs.databricks.com/external-data/json.html)
  + [CSV](https://docs.databricks.com/external-data/csv.html)
  + [Avro](https://docs.databricks.com/external-data/avro.html)
  + [Text](https://docs.databricks.com/external-data/text.html)
  + [Binary](https://docs.databricks.com/external-data/binary.html)

### 14.7. 테이블 포맷 지원 시 external table의 CRUD 여부

* 지원 여부: 지원
* External Table이 Delta 포맷으로 생성된 경우, 다음과 같은 DML(데이터 조작 언어) 작업 가능
  + UPDATE,DELETE,, INSERT INTO / INSERT OVERWRITE:
  + Csv,parquet 등의 포맷은 INSERT INTO, INSERT OVERWRITE 만 사용 가능
* 문서:
  + <https://www.databricks.com/blog/2020/09/29/diving-into-delta-lake-dml-internals-update-delete-merge.html>
  + https://docs.delta.io/latest/delta-update.html
* 예제 노트북: poc14 <https://dbpoc-kbpoc-ext.cloud.databricks.com/?o=2187597122825188#notebook/402993171353935/command/402993171353958>

### 

| -- Delta 포맷의 External Table 생성  CREATE TABLE catalog.schema.table\_name  USING DELTA  LOCATION 's3://your-bucket/path/to/table';  -- 데이터 삽입  INSERT INTO catalog.schema.table\_name VALUES (...);  -- 데이터 업데이트  UPDATE catalog.schema.table\_name SET column = value WHERE condition; [oai\_citation\_attribution:4‡Delta Lake Documentation](https://docs.delta.io/latest/delta-update.html?utm\_source=chatgpt.com)  -- 데이터 삭제  DELETE FROM catalog.schema.table\_name WHERE condition; [oai\_citation\_attribution:5‡Databricks 문서](https://docs.databricks.com/gcp/en/tables/external?utm\_source=chatgpt.com)  -- 데이터 병합(업서트)  MERGE INTO catalog.schema.table\_name AS target  USING source\_table AS source  ON target.id = source.id  WHEN MATCHED THEN UPDATE SET target.column = source.column  WHEN NOT MATCHED THEN INSERT (columns) VALUES (values); |
| --- |

### 

### 

### 15.1. group by, having, count, case 등 가공 및 집계 성능

* 테스트 쿼리

--1. group by 성능

SET use\_cached\_result = false;

select `etl기준년월일`, count (\*)

from poc\_data

group by `etl기준년월일`;

--2. case 성능

SET use\_cached\_result = false;

select

case

when movieid < 100 then '1'

when movieid < 1000 then '2'

when movieid < 1000 then '3'

else '4'

end as movieid

from poc\_data;

--3.having 성능

SET use\_cached\_result = false;

select dates, count (\*)

from poc\_data

group by dates

having dates >= '1999-11-11';

--4. sum 성능

SET use\_cached\_result = false;

select sum (rating)

from poc\_data;

--5. count 성능

SET use\_cached\_result = false;

select count (\*)

from poc\_data;

--6. subquery

SET use\_cached\_result = false;

select movieid

from poc\_data

where movieid in (

select movieid

from poc\_data

where `etl기준년월일` = '20230401'

);

-- 7 1억건 left outer join 2500만건

SET use\_cached\_result = false;

select a.movieid, b.rating

from poc\_temp as a

left outer join append as b on a.movieid = b.movieid;

--8. 단일 컬럼 select

SET use\_cached\_result = false;

select movieid

from poc\_data;

--9. min/max

SET use\_cached\_result = false;

select min (dates)

from poc\_data;

[의견과 제안]

* 200억건, 100 GB 의 데이터셋은 성능 테스트의 변별력이 크게 드러나지 않을 수 있습니다.
  + ⇒ 현실적인 크기의 데이터셋으로 테스트하면 차이가 더 잘 드러날 수 있습니다.
* 테스트용 Warehouse 클러스터는 다른 작업에 영향받지 않도록, 전용 warehouse 클러스터를 별도로 구성하여 사용하는 것이 좋겠습니다.
* 성능 테스트 비용을 별도 측정하기 위해, Warehouse 생성시 별도의 tag 를 붙이는 것이 좋겠습니다.
* Result cache 를 타지 않도록, 각 쿼리 수행 전에 SET use\_cached\_result = false; 를 수행하고 측정하는 것이 좋겠습니다.
* 전체적으로 scan 의 비중이 큰 심플한 유형의 쿼리들이 많아서, 네트워크 성능과 Disk cache 에 따른 차이가 클 것으로 보입니다.
* 디스크 캐시는 SQL warehouse 에서 default ON 이고, 클러스터 종료 후 재시작하는 방식으로 캐시를 purge 할 수 있으나, 테스트가 번거롭고, 현실적인 상황에도 맞지 않습니다. 특히 캐시된 데이터가 없는 상태에서 처음 실행하는 쿼리와 그 이후 쿼리들의 편차를 줄이려면, 테스트 시작 전 SELECT \* FROM table LIMIT 1; 같은 간단한 쿼리로 상황을 맞추면 좋겠습니다.
* Q1, Q3, Q4, Q5 는 full scan + aggr 유형으로, 읽어들이는 컬럼에 따른 차이를 제외하고는 비슷한 성능을 보일 것으로 보입니다.
* Q2, Q8 은 full scan + projection 유형이어서, LIMIT 유무에 따른 차이가 클 것으로 생각됩니다.

LIMIT 이 없으면 전체 데이터를 처리하며, worker 에서 driver 로 결과 데이터를 전송하는 시간이 많은 비중을 차지하게 될 것입니다.

* ⇒ 전체 결과가 필요하다면 CTAS 또는 INSERT 로 결과를 저장하는 방식이 더 일반적인 사용사례로 보입니다.
* ⇒ 전체 결과를 추출하려면 SQL Editor UI 에서 LIMIT 1000 옵션을 해제해야 합니다.
* selectivity 가 높은 필터 조건이 사용된 쿼리가 없어서, zorder 인덱싱의 효과를 보기가 어려운 조건으로 보입니다.
  + ⇒ 현실적인 데이터셋과 쿼리로 테스트하면 더 실무에 가까운 테스트 결과를 얻을 수 있겠습니다.

### 15.2. 벌크 데이터 Replace

### 15.3. 벌크 데이터 Append

### 15.4. 벌크 데이터 Upsert

### 15.5. 데이터 삭제 → Glue(그룹 마이데이터)