Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.5, Oct. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.5.823

개인정보 비식별 환경에서의 개선된 응용프로그램 테스트 데이터 범위 선정 방법

백 송 이,[1†] 이 경 호[2‡]
[1,2]고려대학교(대학원생, 교수)

Improved Application Test Data Range Selection Method in a Non-Personal Information Identification Environment

Song-yi Baek,[1†] Kyung-ho Lee[2‡]
[1,2]Korea University(Graduate student, Professor)

요 약

과거 카드3사 개인정보유출 사건을 계기로 전산 프로그램 개발 시에도 운영환경과 동일 수준의 엄격한 전자금융감독규정을 준수하고 있다. 하지만, 전산 프로그램 개발시 해당 응용 프로그램과 연관된 테스트 데이터 변환 대상범위 식별이 불명확하여 테스트 데이터의 무결성이 훼손된 상태로 응용 프로그램을 검증하고 있어, 이 단계에서 발견되지 못한 결함이 서비스 장애로 이어지는 IT운영리스크가 증가되고 있다. 따라서, 본 논문에서는 특정 응용 프로그램과 연관된 테스트 데이터 변환대상 범위 선정을 위한 프로세스와 알고리즘을 제시하여 실증하였다.

ABSTRACT

In the past, when the personal information leakage incident of the three card companies, the computer program development was followed by the same strict electronic financial supervision regulations as the operating environment. However, when developing a computerized program, the application data is being verified with the integrity of the test data being compromised because the identification of the scope of conversion of the test data associated with the application is unclear. Therefore, in this paper, we proved by presenting a process and algorithm for selecting a range of sufficient test data conversion targets associated with a specific application.

Keywords: Test Data Conversion Range, Data Integrity, System Catalog, Table Relationship, 6sigma, DPMO

I. 서 론

과거 발생한 카드3사의 개인정보 유출 사건은 부정사용방지시스템(FDS) 전산프로그램 테스트를 위해 내부직원이 실제 개인정보를 변환 없이 테스트시스템에 제공되어 외부직원이 개인정보를 무작위로 USB로 내려 받아 유통한 사건으로 전산 개발 시 전자금융감독규정 등의 법규에 대해 실질적인 감독 및 관리를 하지 못한 것이 원인이었다. 이 사건으로 테스트 데이터 이행시 개인식별정보 익명화, 테스트데이터 변환 범위 사전 허가 절차, 응용 프로그램 테스트 종료 후 데이터 삭제, 데이터 복사 및 삭제 이력 저장 등[1] 테스트시스템의 보안 통제 프로세스가 강화되었다.

하지만, 테스트시스템에 대한 데이터 보안 통제가 강화됨에 따라 응용 프로그램 변경시마다 필요한 테스트 데이터를 변환하고 삭제하는 작업을 매번 수행하므로 애플리케이션 개발 생산성 저하되어 Table

Received(05. 04. 2020), Modified(1st: 06. 29. 2020, 2nd: 08. 20. 2020), Accepted(08. 21. 2020)
† 주저자, jangmi4rang@naver.com
‡ 교신저자, kevinlee@korea.ac.kr(Corresponding author)

Table 1. A Financial Company Disability Status(2015~2019)

Disorder Types	Number of disorder	Failure time (Minute)	Number Ratio(%)	Time Ratio(%)
poor policy	17	1,087	6,91	0.18
Insufficient infrastructure	60	4,216	24,39	0.69
Unclear business requirements	2	95,430	0.81	15.54
Business analysis and design errors	1	29	0.41	0.01
Program logic error	22	268,796	8.94	43.77
Poor program performance	15	940	6.10	0.15
Poor test verification	51	237,275	20.73	38.64
Operator operation mistakes	78	6,331	31.71	1.03

1. A금융회사 장애현황과 같이 테스트 검증 미흡으로 장애 발생 건수, 장애 발생 시간(분)이 20.73%, 38.54%를 차지하고 있어 유무형의 손실로 이어지는 IT 운영리스크가 증가하고 있다.

이 중 응용 프로그램 테스트 검증 미흡에 따른 장애 발생 빈도는 Table 2. 와 같이 변경된 응용 프로그램과 연관된 데이터 영향도 파악 미흡이 전체 장애 발생 시간의 97.83% 를 차지하고 있고, 불량 데이터에 의한 장애 발생 건수가 전체의 11.76%를 차지하고 있어 변경된 응용 프로그램과 연관된 데이터 범위 선정이 중요하다.

따라서, 본 논문에서는 변경된 응용 프로그램과 연관된 테스트 데이터 변환 범위를 객관적으로 판단할 수 있는 방법을 제시하고, 운영환경과 동일한 수준의 데이터 무결성이 보장되는지 입증해 본다.

Table 2. A Financial Company Test validation error detail type

Disorder Types	Number of disorder	Failure time (Minute)	Number Ratio(%)	Time Ratio(%)
Poor test peer review	1	59	1.96	0.02
Insufficient test environment	3	43	5.88	0.02
Insufficient grasp of related programs and data	11	232.449	21.57	97.83
Defects due to bad data	6	623	11.76	0.26
Test case missing	24	2,136	47.06	0.90
other	2	2,286	3.92	0.96

II. 선행연구

2.1 테스트시스템 보안통제 연구

테스트시스템 보안에 대한 전자금융감독규정 제13조(전산자료 보호대책) 10항, 제29조(프로그램 통제) 4항[1] 및 개인정보보호 관리체계(PIMS) 개인정보 보호대책 요구사항 8.6.3(개발과 운영환경 분리), 8.6.4(시험 데이터 및 소스프로그램 보안) 등을 준수하도록 규정하고 있다[2].

이 규정에 따라 공공기관 및 대기업은 시험데이터 변환 및 사용에 따른 기준·절차수립·이행, 운영 데이터 사용 승인 절차 마련, 시험용 운영 데이터 사용 기한 만료 후 폐기절차 마련 및 이행, 중요 데이터 사용에 대한 시험환경에서의 접근통제 대책 적용, 운영데이터 복제·사용에 대한 모니터링 및 정기검토 수행, 소스 프로그램 변경 절차 수립, 이전 시스템 환경에 활용한 소프트웨어 보관, 소스프로그램은 운영환경이 아닌 별도의 환경에 저장·관리 등 테스트 데이터 보호조치를 행하고 있다[3].

이에 기업은 변경된 응용 프로그램 테스트시마다 운영시스템과 유사한 정도의 데이터와 보안수준을 유지하기 위해 테스트시스템의 데이터 유입은 테스트 데이터 변환 프로세스로 통제하고 있다. 이는 메타데이터를 기초하여 개인식별정보를 관리하고 임의의 데이터로 변환하여 테스트시스템에 미변환된 개인식별정보가 적재되지 않도록 미연에 방지할 수 있다[4].

또한, 테스트 데이터 변환 프로세스를 통하지 않고 테스트시스템에서 온라인 거래, 배치작업 등으로 개인식별정보 미변환 사례가 발생하는 바, DBA 관점에서 주기적으로 로우 변동 상황을 기록하여 변환되지 않은 데이터를 확인하고 재변환 하는 등의 프로세스를 적용하여 테스트시스템의 개인식별정보 노출을 낮출 수 있다고 제안했다[5].

하지만, 해당 프로세스는 응용 프로그램과 연관된 대상 데이터 간의 일관성을 고려하지 않고 DBA 관점에서 일괄 데이터를 갱신하므로 개인식별정보 노출은 보장되나, 데이터 간의 일관성이 훼손되어 변경된 응용 프로그램이 잘못된 데이터를 참조하게 되어 그 처리결과가 달라져 장애 발생의 원인이 된다.

1) 제13조(전산자료 보호대책) 10항: 이용자 정보의 조회,출력에 대한 통제를 하고 테스트 시 이용자 정보사용 금지
제29조(프로그램 통제) 4항: 변경 필요시 해당 프로그램을 개발 또는 테스트 시스템으로 복사 후 수정할 것

2.2 테스트 데이터 관리 연구

테스트 데이터 관리는 테스트 대상 시스템의 데이터베이스에 있는 데이터가 테스트케이스의 사전조건(preconditions)을 충족하기 위한 모든 개념과 도구를 다루는 것으로 데이터 프라이버시를 보장하고, 적절하고 반복 가능한 테스트 환경을 지원하고, 효율적인 테스트 데이터 공급을 보장하는 것이 중요하다[6].

응용 프로그램을 테스트 하는 것보다 요구되는 테스트 데이터 범위를 분석하고, 필요한 만큼의 테스트 데이터를 변환 적재하고, 새로운 테스트 조건이 발생 시마다 유효하지 않은 테스트 데이터를 재사용이 어려우므로 반복적인 테스트 데이터 재적재 작업이 발생하여 개발 생산성은 떨어진다[7]. 또한, 유효하지 않은 테스트 데이터로 응용 프로그램 결함의 원인이 된다.

또한, 데이터가 비즈니스 서비스에 중심 역할을 하므로 기능 처리가 제대로 동작하는지 테스트하기 위해 실제 운영 데이터와 최대한 가까운 특징을 가진 테스트 데이터가 보장되어야 하며, 테스트 데이터가 사용될 응용 프로그램에 특정한 요구사항, 데이터 요소의 잠재적인 값과 비즈니스 관련성을 이해해야 한다고 기술했다[8].

2.3 데이터 무결성 연구

데이터 무결성(Data Integrity)는 데이터의 정확성과 일관성을 확보하는 것으로 개체 무결성, 참조 무결성, 도메인 무결성, 사용자정의 무결성이 있다. 이 중 참조 무결성(Referential Integrity)는 관계 데이터베이스 모델에서 테이블 간 데이터 값의 일관성을 의미하는 것으로 기본 키[2] 또는 키가 아닌 후보 키와 외래 키[3] 조합으로 정의되며, 데이터 정의어(DDL)[4]에 의한 참조 무결성 제약조건(Constraint)[5]에 의해 정의된다.

일반적으로 논리 데이터모델 설계과정에서 ERD(Entity Relationship Diagram)[6] 도구로 테이블간의 연관성을 확인할 수 있으나, 물리 데이터모델 구현과의 차이가 발생할 수 있으므로 기존 연구에서는 논리 데이터모델 설계에 대한 역공학(Reverse Engineering)[7] 방식으로 데이터베이스 내 시스템 카탈로그[8]를 분석하여 테이블 연관관계 획득하고, 이를 객체화하여 표현하였다[9][10]. 이 연구에서는 데이터 구조 추출을 데이터베이스에 생성된 테이블의 기본 키와 외래 키를 구성하는 필드를 획득 한 후 테이블 간의 상하 관계를 확인하고, 기본 키와 외래 키 사이의 카디널리티[9]를 획득하여 관계를 가진 두 테이블간의 관계 유효성을 확인하고, 디지털포렌식 조사를 위한 데이터베이스 구조분석에 관한 연구에서는 데이터베이스 설계 및 관리상 키를 정의한 컬럼명을 활용하여 부모테이블(Parent Table)과 자식테이블(Child Table)를 검색한다[11].

또한, 한국데이터산업진흥원에서 제시한 테이블 간의 컬럼 값이 참조관계라면 제시된 두 개 이상의 테이블 및 컬럼 간에 상호 연관 관계가 존재하고, 동일한 값으로 유지해야 일관성이 확보된다고 제시했다. 이에 기업은 품질기준을 마련하고, 지속적으로 측정하여 데이터 품질 수준을 평가하고 있다[12]. 따라서, 테이블 관계 목록을 취합하여 자식테이블의 데이터 중 부모테이블에 존재하지 않는 데이터 오류 건수, 오류률 등을 정기적으로 측정하고 있고, 이 수치가 적을수록 데이터 품질수준이 높다고 할 수 있다.

2) 기본 키(Primary Key): 관계형 데이터베이스에서 레코드를 식별하는 후보키(속성 또는 속성의 집합) 가운데, 설계자가 일반적으로 이용되어야한다고 정한 후보키

3) 외래 키(Foreign Key): 한 테이블의 필드(attribute) 중 다른 테이블의 행(row)을 식별할 수 있는 키

4) 데이터 정의어(DDL, Data Definition Language): 새로운 데이터베이스를 구축하기 위해 스키마를 정의하거나, 기존 스키마의 정의를 삭제 또는 수정하기 위해 사용하는 데이터 언어

5) 참조 무결성 제약조건(Referential Integrity Constraint): 외래키는 참조할 수 없는 값을 가질 수 없다는 규칙으로 다른 릴레이션의 기본키를 참조하는 속성이고 릴레이션 간의 관계를 표현하는 역할

6) ERD(Entity Relationship Diagram): 데이터베이스의 논리적 구조를 도식화한 것

7) 역공학(Reverse Engineering): 완성된 제품을 상세하게 분석하여 그 기본적인 설계 내용을 추적하는 것으로 제품이 어떻게 작동하는지 분석하여 프로그램이나 보안 매커니즘을 어떻게 작동시키는지를 알아내기 위해 사용하는 기법

8) 시스템 카탈로그(System Catalog): 시스템이 필요로 하는 데이터베이스, 테이블, 뷰, 인덱스, 접근 권한 등에 관한 정보를 메타 데이터 형태로 포함하는 시스템 데이터베이스

9) 카디널리티(Cardinality): 하나의 릴레이션에서 튜플의 전체 개수

2.4 선행 연구와의 차이점

이전 연구가 테스트시스템의 개인식별정보 노출 빈도를 최소화하여 기밀성을 보장하는 방법을 제시했다면, 본 연구는 테스트시스템의 기밀성 뿐 만 아니라, 운영환경과 동일한 수준의 응용 프로그램 테스트 결과를 보장하기 위한 테스트 데이터의 무결성을 확보하기 위한 방법을 제시하였다. 특히, 데이터 관점에서의 연관성 분석이 아닌 특정 응용 프로그램을 중심으로 참조하는 데이터 범위를 객관적으로 대상 식별할 수 있도록 구현하였다.

현실적으로 다양하고 복잡한 비즈니스 서비스를 처리하고 있는 업무에서는 데이터가 다수의 데이터베이스에 흩어져 있고, 데이터베이스 서비스 부하 등을 고려하여 데이터베이스 내 참조 무결성 제약조건 설정을 지양하고 있다. 또한, 개발자는 운영 시스템에 대하여 제한된 접근만 허용되므로 응용 프로그램과 연관된 데이터를 분석하는데 한계가 있다.

Table 3. 시중은행의 시스템 운영 환경과 같이 거래 및 데이터에 대한 조건의 정합성 처리는 데이터베이스 내 참조 무결성 제약조건 정의 보다는 응용 프로그램으로 처리하고 있다. 따라서, 선행 연구에서 제시한 데이터베이스 내 관리키 정보로는 응용 프로그램과 데이터의 연관성을 파악하는데 한계가 있고, ERD 정보를 통한 테이블의 관계 분석은 Table 3. 와 같이 ERD 작성은 일부 업무만 적용되어 있다.

Table 3. A Financial Company System Operation Status

구분		ERD & Metasystem	Metasystem
Business Information	Unit Business System	13	144
	subject area	1,148	468
	volume (TB)	70.5	547
Database Information	Table	22,030	614,310
	Index	40,427	929,318
	Primary Key Constraints	20,427	606,225
	Foregin Key Constraints	0	2,566

III. A금융회사 전산 프로그램 개발단계의 테스트 데이터 변환 운영현황

3.1 A금융회사 전산 프로그램 개발 및 테스트 데이터 변환 프로세스 운영현황

A금융회사는 대고객 금융서비스를 제공하기 위해 Table 3. 과 같이 계정계 이외의 약 150 여개의 단위 업무 시스템을 운영하고 있고, 약 63만 테이블에 고객정보를 포함한 데이터를 저장하고 있다. 또한, 연관된 데이터의 일관성 보장을 위한 무결성 제약조건에 대한 외래키 제약조건 생성은 전체 테이블의 약 0.03% 정도만 관리되고 있는 바, 대부분 데이터 정합성은 응용 프로그램으로 처리하고 있다. 이에 개발자는 비즈니스 요구사항에 맞게 응용 프로그램을 변경하고 있으며, 변경된 응용 프로그램을 운영시스템에 적용 전에 비즈니스 요구사항이 정확하게 구현되었는지, 데이터에 따라 조건 처리가 누락 없이 구현되었는지 테스트시스템에서 검증하는 과정을 거친다.

이에 개발자는 응용 프로그램을 변경할 때마다 Fig. 1. 의 테스트 데이터 변환 프로세스를 준수하여 변경된 응용 프로그램과 연관된 변환 대상 테이블 목록을 선정하고, 사전에 정의된 메타데이터 기초하여 변환 대상 테이블 내 개인식별정보가 포함되었는

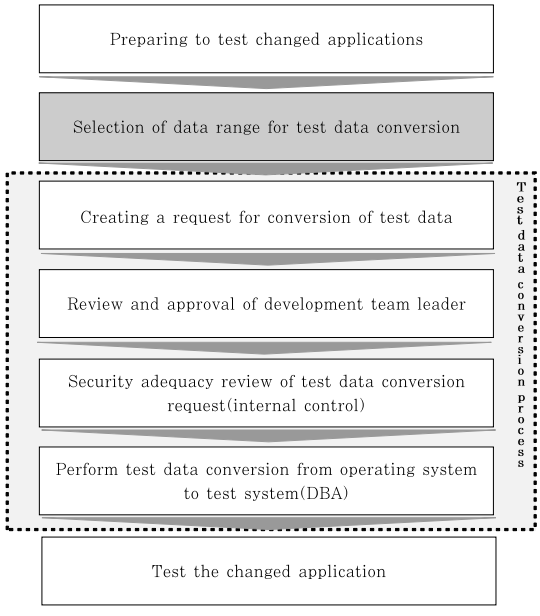


Fig. 1. A Financial company Test Data Conversion process

지 자동 판단하여 비식별화 변환 룰을 적용하여 테스트 데이터 변환 신청하게 된다. 이후 내부 통제자는 신청내역 중 개인식별정보의 변환 정책이 적용되었는지 검토되면 DBA(DataBase Administrator)에 의해 신청한 변환 대상 테이블의 데이터만 운영시스템에서 테스트시스템으로 변환하고 적재하는 작업을 수시로 진행한다.

3.2 응용 프로그램과 연관된 테스트 데이터 누락 발생 사례 분석

A금융회사는 Fig 1. 통제 프로세스 기반으로 변경된 응용 프로그램과 연관된 데이터를 개인식별정보 노출 없이 테스트시스템에 변환된다. 하지만, 개발자 판단에 의한 응용 프로그램 테스트를 위해 필요한 데이터 범위 식별 시 데이터 간 RI(Referential Integrity)[10] 고려되지 않아 데이터 변환 대상 목록에서 일부 테이블이 누락되거나, 연관된 데이터 값이 일치하지 않아 응용 프로그램의 처리 결과가 테스트환경과 운영환경이 상이하게 처리될 수 있고, 운영환경에서 예기치 못한 결함이 발생한다.

3.3 응용 프로그램의 테스트 결함 유형 분석

운영환경과 테스트환경에서의 응용 프로그램 테스트 결과가 상이하게 발생하는 유형은 다음과 같다.

□ 응용 프로그램과 연관된 일부 테이블 누락

Fig 2. 과 같이 변경된 응용 프로그램의 관련된 테이블의 일부가 식별되지 않아 해당 테이블의 데이터가 테스트시스템으로 적재되지 않는 경우에 발생한다.

□ 연관되는 데이터 값의 비일치

Fig. 3. 과 같이 테스트시스템의 데이터를 개발자 또는 DBA에 의해 임의로 특정 데이터로 일괄 데이터 값이 변경되거나, 임의의 프로그램에 의해 일부 데이터 값이 변경되어 응용 프로그램에서 테이블의 참조 조건절에 일치되는 데이터 값이 없는 경우에 발생한다.

10) RI(Referential Integrity): 관계 데이터베이스 모델에서 2개의 관련있는 관계 테이블간의 일관성 유지 특성[11]

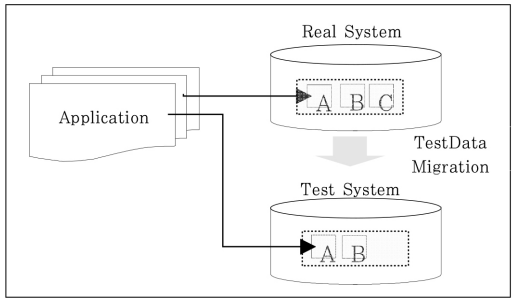


Fig. 2. Some table missing types associated with the application

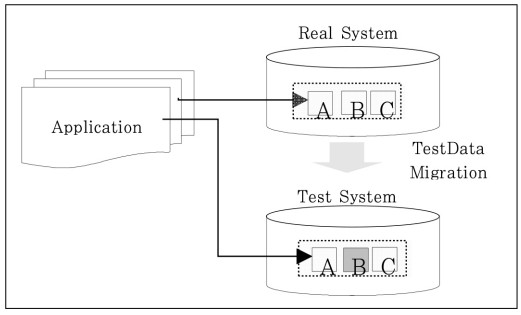


Fig. 3. Inconsistency in associated data values

3.4 응용 프로그램과 연관된 데이터 무결성 보장방안 제시

3.3 과 같은 결함을 예방하기 위해 개발자에 의한 주관적인 테스트 데이터 변환 대상 선정을 배제하고, 변경된 응용 프로그램을 기준으로 참조하는 테이블 목록을 선정하는 알고리즘을 전산화하여 Fig. 4. 과 같이 테스트 데이터 변환 프로세스 사전 단계에 처리하면, 응용 프로그램 테스트와 관련된 테스트 데이터 변환 대상 범위를 객관적으로 판단되므로 개발자의 분석 오류와 테스트 데이터 준비 시간이 단축된다.

IV. 테스트 데이터 변환대상 범위 선정 모델

4.1 테스트 데이터 변환대상 선정 프로세스

응용 프로그램과 데이터의 연관정보를 개별적인 응용 프로그램별, 데이터별 단독으로 식별할 수 없으므로 대상이 되는 응용 프로그램을 기준으로 관련된 테스트 데이터 변환대상 선정하는 프로세스는 다음과 같다.

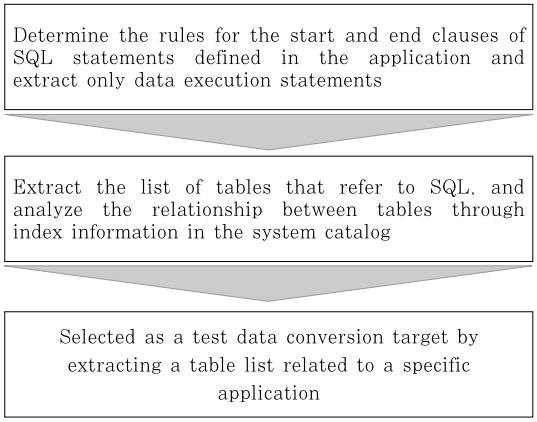


Fig. 4. Application-based test data conversion target selection process

- 응용 프로그램의 정의된 SQL 통한 테이블 식별
- 시스템카탈로그 기준으로 테이블 연관성 분석
- 응용 프로그램과 데이터간 일치된 테이블목록 추출

4.2 단계별 전산 프로세스 구현

4.2.1 응용 프로그램의 정의된 SQL 통한 테이블 파악

Fig. 4. 는 대상이 되는 응용 프로그램을 기준으로 연관된 테스트 데이터 변환 대상 범위를 선정하는 프로세스를 도식화 한 것이다. 제4장 4.2 에서는 개인정보 비식별 환경에서의 개선된 응용 프로그램 테스트 범위 선정 방법을 단계별 전산 프로세스 알고리즘 제시하고, 제4장 4.3에서는 C언어와 ORACLE DBMS 환경 기반으로 본 프로세스를 구현된 전산 프로세스를 테스트하고, 제4장 4.4에서는 데이터 구조 무결성 관점에서 구현된 프로세스의 실효성을 검증해 본다.

4.2.1.1 요구사항 정의

비즈니스 서비스는 관련된 응용 프로그램에 의해 데이터 변화가 발생한다. 이에 비즈니스 서비스가 요구하는 기능과 데이터에 따른 조건처리가 정확하게 구현되었는지 테스트시스템 환경에서 충분히 확인하게 된다. 이에 운영시스템과 동일 수준의 테스트 데이터가 필요하며, 개발자는 해당된 응용 프로그램과 연관된 데이터를 식별할 수 있는 객관적인 방법으로 테스트 데이터 변환 대상 범위 식별이 필요하다.

따라서, 응용 프로그램과 연관된 데이터 범위 선정을 위해 변경된 응용 프로그램에 정의된 SQL을 추출하고, SQL 구문 분석으로 변환이 필요한 테이블 일부를 파악할 수 있다.

4.2.1.2 SQL 구문 분석을 통한 테이블 식별 방법

4.2.1.1의 요구사항에 따라 응용 프로그램에 정의된 SQL를 추출하고, 추출된 SQL 구문을 분석하는 방법은 정규 표현식(Regular Expression)을 이용하여 응용 프로그램에서 SQL 문장의 시작절과 종료절의 규칙으로데이터 실행 문장만을 추출한다. 또한, 테이블 식별은 Table 4. 과 같이 SQL 유형별로 데이터 변화가 발생하는 테이블이 명시적으로 정의되므로 SQL 유형별 테이블 구성절을 정규 표현식 방법을 이용하여 테이블명만 추출한다.

Table 4. Table construction clause by SQL type

SQL Type	SQL Syntax	Table clause
SELECT	SELECT A.Col1, B.Col2 FROM Tab A, Tab B WHERE A.Col1 = B.Col2	FROM
INSERT	INSERT INTO Tab(Col1,Col2) VALUES (Val1, Val2) ;	INTO
UPDATE	UPDATE Tab1 SET Col1=Val1 WHERE [Condition Clause] ;	UPDATE
DELETE	DELETE FROM Tab1 WHERE [Condition Clause] ;	FROM

4.2.1.3 응용 프로그램의 정의된 SQL 통한 테이블 식별 알고리즘

4.2.1.2 에 의하여 테이블을 식별하는 과정은 크게 3단계로 구성된다. 1단계는 응용 프로그램의 정의된 SQL 문장 추출하는 과정, 2단계는 추출된 SQL 문장에 정의된 테이블 식별하는 과정, 마지막 3단계는 2단계에서 식별된 테이블을 중복 배제하는 과정이다.

1단계는 변경된 응용 프로그램을 중심으로 SELECT, INSERT, UPDATE, DELETE 으로 시작하는 부분부터 종료를 의미하는 세미콜론(;) 까지 SQL 문장을 추출한다.

2단계는 1단계에 추출된 SQL 문장을 기준으로 정규 표현식으로 테이블 구성절 다음에 정의된 구성절을 테이블로 식별하는 과정이다.

마지막 3단계는 2단계에서 식별된 동일 테이블 중복을 배제하여 고유한 기준 테이블을 식별한다.

4.2.2 시스템카탈로그 기준으로 테이블 연관성 분석

4.2.2.1 요구사항 정의

4.2.1에서 식별된 테이블을 기준으로 연관된 테이블 목록을 추출하는 과정은 기존 연구인 테이블 기반의 레거시 데이터베이스에서 엔티티간 관계 추출에 관한 연구에서 데이터베이스 내 시스템카탈로그의 제약조건 정보로 테이블의 기본키와 외래키로 테이블 간 관계 기수성(Cardinality)으로 쉽게 분석할 수 있으나,대부분 복잡한 비즈니스 프로세스를 처리하고 있는 업무에서는 서비스 부하 등을 고려하여 데이터베이스 내 참조 무결성 제약조건 정의하지 않고, 응용 프로그램으로 서비스 정합성을 보장하고 있다.

하지만 응용 프로그램을 통한 데이터 처리를 빠르게 하기 위해 검색 조건이 되는 컬럼을 인덱스로 구성하므로 시스템카탈로그 내 인덱스 메타정보 이용하여 테이블의 기본키, 외래키 역할을 하는 인덱스를 찾아 테이블 간 연관 관계를 분석 할 수 있다.

4.2.2.2 시스템카탈로그의 기본키와 외래키 분석방법

데이터베이스 시스템카탈로그의 인덱스 관련 뷰에 기본키와 외래키의 메타 정보가 Table 5. 과 같이 저장된다[13]. 이 메타 정보에서 UNIQUE와 NOT NULL로 정의된 인덱스를 기본키 이고, 외래키는 인덱스 구성 컬럼을 포함하고 있는 다른 테이블의 기본키 구성 컬럼을 참조한다.

Table 5. Primary key and foreign key constraints in the DB system catalog

Type	Constraint
primary key	Automatically creates a unique index, and the columns that make up the primary key are NOT NULL.
Foreign key	Only columns that are primary keys of other tables or composed of Unique Index can be defined, and the data type must match.

4.2.2.3 시스템카탈로그 기준의 테이블 연관성 분석 알고리즘

4.2.2.2 에서 분석된 테이블의 기본키와 외래키 역할을 수행하는 인덱스를 통해 테이블 연관관계를 분석할 수 있다. 4.2.1.3 에서 식별된 테이블을 기준테이블로 정의하고, 비교를 하는 테이블을 대상테이블이라고 정의한다. 그리고, 기준 테이블을 중심으로 연관 테이블간의 관계를 식별하고, 식별한 테이블 중 대상테이블이 자식테이블(Child Table) 역할을 수행한다면, 그와 연관된 테이블을 반복하여 분석함으로써 연관된 모든 테이블을 도출할 수 있다.

먼저, 기준 테이블의 기본키(Primary Key) 역할을 수행하는 인덱스 구성 컬럼을 획득한 후, 해당 컬럼을 포함한 다른 테이블(대상 테이블)의 인덱스 정보를 확인한다. 이 때 기준 테이블과 대상 테이블의 구성 컬럼이 동일하면 2개의 테이블은 일대일 관계[11]라고 할 수 있고, 포함된다면 일대다 관계[12]로 볼 수 있고 포함하는 테이블을 자식테이블(Child Table), 그 반대는 부모테이블(Parent Table)로 볼 수 있다. 또한, 기준 테이블의 일반 인덱스 구성 컬럼 중 다른 테이블(대상 테이블)의 기본키로 포함된다면, 해당 인덱스는 외래키 역할을 수행하고, 2개의 테이블은 일대다 관계로 해당 테이블은 자식테이블로 볼 수 있다. 이와 같이 분석한 결과에서 대상 테이블이 자식테이블(Child Table) 이라면 대상 테이블을 기준 테이블로 변경하여 동일한 분석 과정을 수행하여 연관된 모든 테이블을 도출한다.

이를 통해 도출된 모든 테이블에 대한 관계 기수성 확인할 수 있고, 응용 프로그램과 연관된 테이블로 테스트 데이터 변환 대상 범위가 된다.

4.2.3 응용 프로그램과 연관된 테이블 목록 추출

4.2.2.3 분석으로 응용 프로그램과 연관된 모든 테이블 목록을 추출하고, 테이블 간 관계 기수성도 확인할 수 있다. 따라서, 해당된 테이블 목록만 응용 프로그램 테스트를 위한 필요한 테스트 데이터 변환 대상범위로 정의할 수 있다.

11) 일대일(1:1) 관계: 하나의 레코드가 오직 하나의 레코드와 연결될 수 있는 데이터베이스 관계

12) 일대다(1:n) 관계: 하나의 레코드가 많은 레코드와 연결될 수 있는 데이터베이스 관계로서, 테이블의 한 레코드가 다른 테이블의 여러 레코드에 관련되는 것을 말함

4.2.4 응용 프로그램과 연관된 데이터 범위 선정 통합 모델

4.2에서 제시한 단계별 프로세스를 통합한 모델은 Fig. 5. 과 같다.

요약하자면, 응용 프로그램 소스에 정의된 SQL 문장을 분석하여 기준 테이블을 추출하고, 그 기준 테이블을 중심으로 DB 시스템카탈로그 메타정보로 기준 테이블의 자식과 부모 테이블을 식별·추출하여 애플리케이션과 연관된 데이터 범위를 결정하게 된다.

4.3 프로세스 테스트

4.3.1 응용 프로그램의 정의된 SQL 통한 테이블 파악

본 논문의 테스트는 C언어로 작성된 애플리케이션과 오라클 DBMS 기반 환경에서 검증하였다. A 금융회사에서 변경 가능성이 있는 응용 프로그램과 데이터베이스 현황은 애플리케이션 약 36,178,087개이고, 데이터베이스는 Fig. 4와 같이 총 용량 19.3 TB(테라바이트)로 테이블 7,105개, 인덱스 11,911개, 기본키 제약조건은 7,010개 정의되었고, 외래키 제약조건은 정의되지 않았다.

또한, C언어 기반 응용 프로그램 소스 Fig. 7 와 같이 작성되어 있고, 해당 응용 프로그램 소스 내 데이터 처리하는 SQL 문장이 정의되어 있다. 응용 프

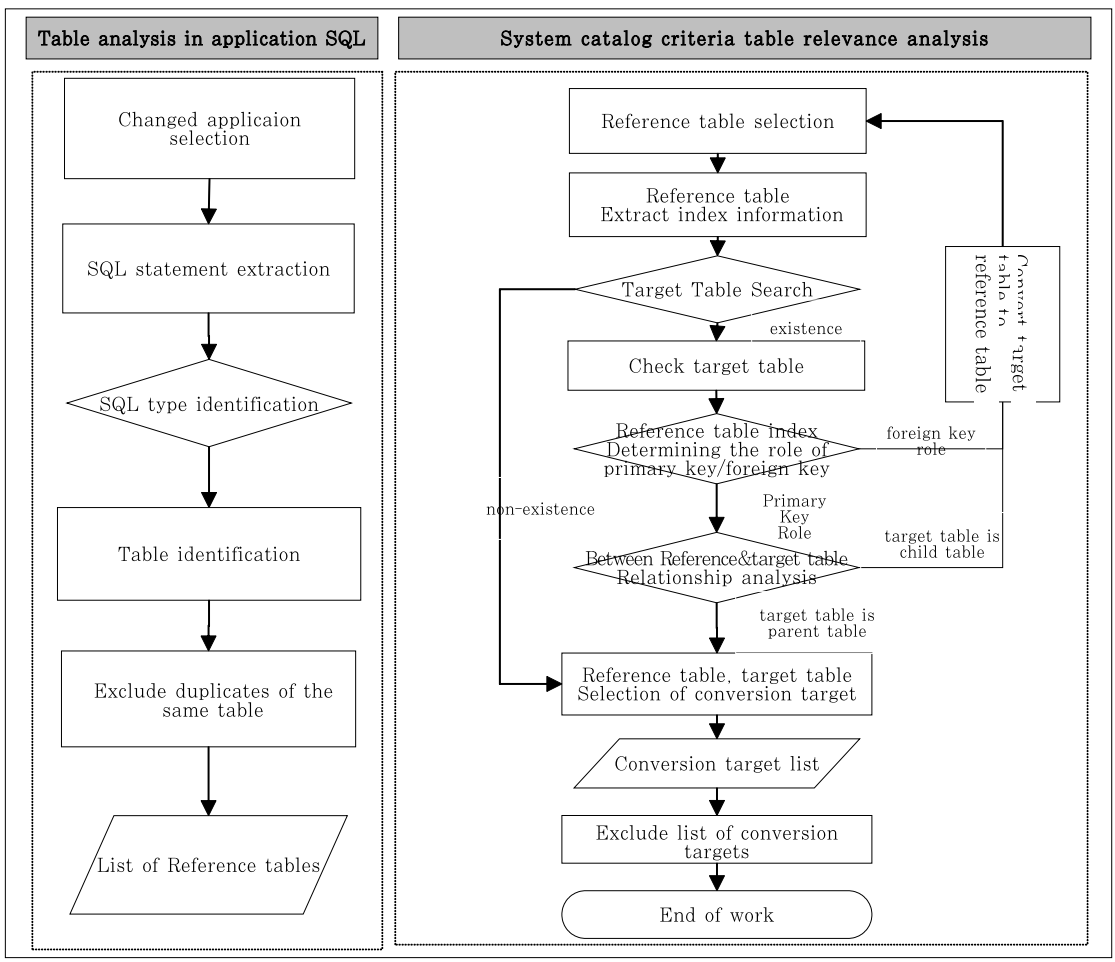
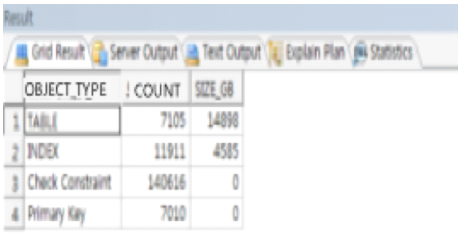


Fig. 5. Data range selection model for test data conversion



Result

Grid Result | Server Output | Text Output | Explain Plan | Statistics

	OBJECT_TYPE	COUNT	SIZE_GB
1	TABLE	7105	14898
2	INDEX	11911	4585
3	Check Constraint	140616	0
4	Primary Key	7010	0

Fig. 6. Database Usage Status

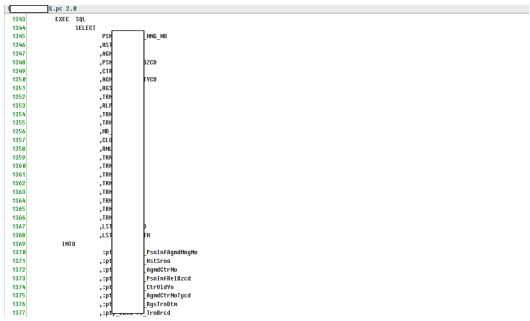
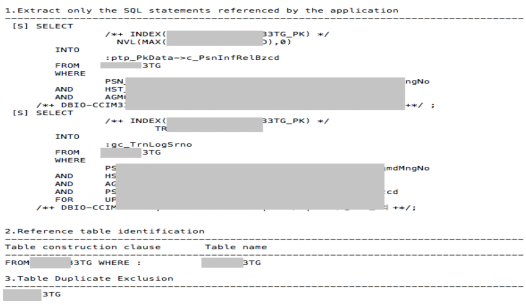


Fig. 7. C language based application program source



1.Extract only the SQL statements referenced by the application

[S] SELECT
/++ INDEX(33TG_PK) */
NVL(MAX( ),0)
INTO :ptp_PkData->c_PsnInfRelBzcd
FROM 3TG
WHERE
PSN ngNo
AND HST
AND AGM
/++ DBIO-CCIM3 ++/ ;
[S] SELECT
/++ INDEX(33TG_PK) */
TR
INTO :gc_TrnLogSrno
FROM 3TG
WHERE
PS mdMngNo
AND HS
AND AC
AND PS cd
FOR UP
/++ DBIO-CCIM ++/;

2.Reference table identification

Table construction clause Table name

FROM 3TG WHERE : 3TG

3.Table Duplicate Exclusion

3TG

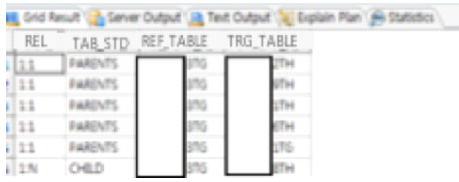
Fig. 8. Program source defined SQL statement and table identification result

로그램 소스명을 조건으로 검색하면, 해당 소스에 정의된 모든 SQL 문장을 추출하고, 해당 SQL 문장에 정의된 테이블 목록만 아래 Fig. 8 와 같이 식별할 수 있다.

해당 응용 프로그램은 데이터 변경이 발생하는 UPDATE SQL 문장으로만 작성되어 있고, 한 개의 테이블에 대하여 데이터 변경을 하고 있는 것을 확인할 수 있다.

4.3.2 시스템카탈로그 기준으로 테이블 연관성 분석

4.3.1에서 확인된 테이블 XXX3TG를 기준 테이블로 하고 그에 따른 DB 시스템카탈로그 인덱스 메타정보를 참조하여 구현한 테이블 연관성 분석 알고리즘 실행하면 Fig. 9 과 같이 6개의 연관되는 대상테이블로 추출되었고, 기준 테이블과 일대다(1:N) 관계로 1개 테이블XXX8TH는 XXX3TG의 부모테이블임을 확인할 수 있다.



Grid Result | Server Output | Text Output | Explain Plan | Statistics

REL	TAB_STD	REF_TABLE	TRG_TABLE
1:1	PARENTS	3TG	8TH
1:1	PARENTS	3TG	8TH
1:1	PARENTS	3TG	8TH
1:1	PARENTS	3TG	8TH
1:1	PARENTS	3TG	8TG
1:N	CHILD	3TG	8TH

Fig. 9. Extract target table associated with base table

4.3.3 응용 프로그램과 연관된 테이블 목록 추출

기준 테이블이 대상 테이블에 대하여 모두 부모테이블이라면 추출된 기준 테이블과 대상 테이블을 중 배제하여 프로그램이 종료되고, 위의 Fig.7 과 같이 특정 응용 프로그램과 연관된 6개의 테이블 목록을 최종 확인할 수 있다.

4.4 구현된 프로세스의 데이터 무결성 검증

4.4.1 데이터 무결성 검증 방법

개인정보 비식별 환경에서의 개선된 응용 프로그램의 테스트 데이터 변환 대상 범위 선정 알고리즘을 탑재한 프로그램으로 기준 테이블과 대상 테이블의 인덱스 구성 컬럼이 추출된다. 한국데이터산업진흥원에서 제시한 기준 테이블과 대상 테이블 간 Table 5. 데이터 구조 무결성 오류 검증 SQL 스크립트를 통하여 전체 진단 건수와 오류건수를 확인하여 두 테이블의 연관 정도를 확인한다[12].

데이터 품질관리 수준평가를 위해 대부분 기업은 6시그마 방법론[13] 참조하여 품질진단, 오류원인분석, 품질개선, 통제 활동을 수행하고 있고[14], A금융회사도 6시그마 수준으로 품질을 측정·평가하고 있다.

13) 6시그마 방법론: 기업 또는 조직 내의 다양한 문제를 구체적으로 정의하고 현재 수준으로 계량화하고 평가한 다음 개선하고 이를 유지 관리하는 경영 기법으로 기존의 프로세스를 향상시키거나, 새로운 제품을 만들거나 예측 가능하고 결함이 없는 성능을 내기 위한 관리방법론

Table 6. Data integrity error verification SQL script

```
SELECT COUNT(*)
FROM CHILD_TAB TAB1 LEFT OUTER
JOIN PARENTS_TAB TAB2 ON TAB1.Col1
= TAB2.Col2
WHERE TAB1.Col1 IS NULL;
```

4.4.2 6시그마 수준의 데이터 품질 지수 산출

Table 6 의 SQL 스크립트로 테이블 간 오류건수를 확인하고, 오류 정도를 나타내는 DPMO(Defects Per Million Oppertunities)를 산출하고, 데이터 품질 지수를 시그마 수준으로 평가한다.

DPMO 는 100만 개당 오류 발생률을 의미하며, 다음과 같이 산출한다[15].

DMPO 기준으로 산출된 시그마 수준이 높을수록 데이터 품질지주가 높다는 것을 의미하며, 이는 기준 테이블과 대상테이블 간 연관성이 있다는 것을 의미한다.

$$DPMO = \left(\frac{\text{total number of defects found in a sample}}{\text{total number of defect opportunities in the sample}}\right) \times 1{,}000{,}000$$

$$= \left(\frac{\text{total number of defects found in a sample}}{\text{Sample size x number of defect opportunities per unit in the sample}}\right) \times 1{,}000{,}000$$

Fig. 10. DPMO Calculation formula

4.4.3 테이블 간 데이터 연관성 확인 결과

5.3에서 추출된 기준 테이블과 대상 테이블에 대하여 Table 6.의 SQL 스크립트로 테이블 간 데이터 구조 무결성 품질 측정을 위한 관계 목록을 취합하여, 전체 데이터 건수, 오류 데이터 건수, DPMO, 시그마 수준을 Table 7.과 같이 확인할 수 있다. A금융회사는 최소 시그마 수준은 5.29S, 기대 시그마 수준은 5.42S 으로 관리하고 있으며, 6개 중 5개는 A 금융회사가 관리하는 시그마 수준을 만족하고 평균 5.39S임을 확인할 수 있다. 따라서, XXX9TH를 제외한 테이블은 XXX3TG 테이블과 데이터는 무결성이 유지된다고 볼 수 있으므로 테이블 간 연관관계가 있다고 볼 수 있다.

V. 결론 및 향후 연구방향

다양한 비즈니스 요구사항을 IT서비스에 능동적으로 반영하기 위해서는 응용 프로그램 분석부터 테스트 전 과정이 효율적으로 구성되어야 한다. 이 중 테스트 단계는 비즈니스 요구사항이 응용 프로그램에 올바르게 반영되었는지 고객에게 제공 전에 마지막으로 검증하는 과정이므로 운영환경과 유사한 수준의 데이터 확보가 된 상태에서 다양한 경우를 충분히 검증해야 한다.

이에 운영 데이터와 유사한 수준의 테스트 데이터량 확보를 위한 테스트 데이터 변환 프로세스는 제시되었지만, 변경 처리된 특정 응용 프로그램과 연관된 테스트 데이터 범위를 선정하는 것은 복잡한 관계를 가진 데이터를 분석하는 것으로 변경된 데이터에 의해서 영향받는 모든 영역의 데이터를 파악하는 것은 어려운 과정이다. 일부 데이터 누락 등으로 검증되지 않는 테스트가 예상치 못한 애플리케이션 결함 및 장애 발생로 이어져 기업의 유형적 손실과 고객신뢰도 훼손 등 IT운영리스크가 증가하고 있다.

본 논문에서는 응용 프로그램과 연관된 테스트 데이터 변환 대상을 추출할 수 있는 알고리즘을 반영한 의사결정 모델을 제시하고, 전산 프로세스로 구현하였다. 또한 변환된 테스트 데이터 간 일관성이 운영환경과 유사한 수준으로 보장되는지 확인하였다. 다

Table 7. Data quality index between reference data and target table

No.	Reference Table	Target Table	Relationship	Total number of data	Number of error data	DPMO	Sigma_VAL	Degree of association
1	XXX3TG	XXX2TH	1:1	948,010	10	10.59	5.75	Y
2	XXX3TG	XXX9TH	1:1	1,494,015	2,405	1,609.75	4.45	N
3	XXX3TG	XXX1TH	1:1	62,776,516	3,970	63.24	5.33	Y
4	XXX3TG	XXX6TH	1:1	62,776,516	594	9.46	5.77	Y
5	XXX3TG	XXX1TG	1:1	62,776,516	2,346	37.37	5.46	Y
6	XXX3TG	XXX8TH	N:1	23,540,621	625	26.54	5.54	Y

만, 일부의 경우에는 테이블 간 인덱스 구성 컬럼이 동일하더라도 테이블 간 데이터 연관이 없는 경우도 발생한다.

그러므로 향후 연구방향은 동일한 메타 정보를 사용하더라도 다른 의미를 갖는 테이블을 식별하여 테스트 데이터 변환 대상 목록에서 삭제할 수 있는 프로세스가 추가적으로 마련이 필요하다. 또한, 실무적으로 동일한 의미를 갖는 데이터는 메타시스템에 정의된 같은 표준용어를 사용하여 테이블 컬럼으로 구성하지만, 일부의 경우 동일한 의미의 데이터를 갖더라도 다른 표준용어로 테이블 컬럼을 정의하는 경우도 발생하므로 이를 관리할 수 있는 프로세스 마련도 추가적으로 필요하다.

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 대학ICT연구센터육성지원사업의 연구결과로 수행되었음

References

[1] Security world, "Strengthen development security, introduce test data conversion system," https://www.boannews.com/media/view.asp?idx=44642, 2014

[2] Korea Ministry of Government Legislation, "Regulations for Electronic Banking Supervision of the Financial Supervisory Service," http://www.law.go.kr/administrative_rules/electronic_Financial_supervision_regulations, 2019

[3] Korea Internet & Security Agency, "PIMS Certification System Guide," pp. 181-185, Apr. 2017.

[4] Yang-Ho Kim, In-Hyun Cho and Kyung-Ho Lee, "A Decision-Making Model for Handling Personal Information Using Metadata," Journal of the Korea Institute of Information Security & Cryptology, 26(1), pp. 259-270, Feb. 2016

[5] Yeong-jin Choi, Jeong-hwan Kim and Kyung-ho Lee, "A Study on Data Security Control Model of the Test System in Financail Institutions," Journal of the Korea Institute of Information Security & Cryptology, 24(6), pp. 1293-1307, Dec. 2014

[6] Klaus Haller, "Test Data Management in Practice," Journal paper of conference, pp. 1-11, 2013

[7] Purnima Khurana and P Bindal, "Test Data Management," International Journal of Computer Trends, pp. 1-6, 2014

[8] Stephanie Chace, "Test Data Management Best Practice," Technical Document, pp. 1-14, 2011

[9] Dowming Yeh, Yuwen Li and William Chu, "Extracting entity-relationship diagram from a table-based legacy database," The Journal of Systems and Software, pp. 764-771, 26 July 2007.

[10] Jean-Luc Hainaut, "Introduction to Database Reverse Engineering, LIBD-Laboratory of Database Application Engineering Institut d"Informatique - University of Namur," 24 Sep 2002

[11] DongChan Lee and Sangjin Lee, "Reserach of organized data extraction method for digital investigation in relational database system," Graduate School of Information Management and Security Korea University, pp. 566-571, May 2012.

[12] Korea Data Agency, "Data Quality Assessment Procedure Manual, 4th Ed., Prentice Hall," pp. 30-154, Nov. 2005.

[13] Korea Data Agency, "SQL basics and usage," http://www.dbguide.net/db.db?cmd=view&boardUid=148190&boardConfigUid=9&categoryUid=216&boardIdx=134&boardStep=1, 2013

[14] National Information Society Agency, "Data Quality Management Solution

Data Collection," pp. 13, Nov. 2012.
[15] Six sigma daily, "Six Sigma Tools: DPU, DPMO, PPM and RTY," https://www.sixsigmadaily.com/dpu-dpmo-ppm-and-rty/, 2020

〈저자소개〉

백 송 이 (Song-yi Baek) 정회원
2001년 2월: 성균관대학교 전기전자컴퓨터공학부 졸업
2015년 3월~현재: 고려대학교 정보보호대학원 석사과정
〈관심분야〉 정보보호 정책, 개인정보보호 정책, 데이터베이스

이 경 호 (Kyung-ho Lee) 종신회원
1989년 8월: 서강대학교 수학과 학사
1997년 8월: 서강대학교 정보통신대학원 석사 졸업
2009년 8월: 고려대학교 정보보호대학원 박사 졸업
2011년~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 정보보호 정책, 개인정보보호 정책, 위험관리, 정보보호컨설팅