한국기록관리학회지
21(3), 99-118, 2021.8
http://dx.doi.org/10.14404/JKSARM.2021.21.3.099
pISSN 1598-1487 eISSN 2671-7247

JKSARM
Journal of Korean Society of Archives and Records Management

# 행정정보 데이터세트 보존포맷으로서 SIARD 검증에 관한 연구

## A Study on SIARD Verification as a Preservation Format for Data Set Records

윤성호(Sung-Ho Yoon)[1], 이정은(Jung-eun Lee)[2], 양동민(Dongmin Yang)[3]

E-mail: tjdgh9410@naver.com, jungeun.lee@jbnu.ac.kr, dmyang@jbnu.ac.kr

[1] 제1저자 전북대학교 기록관리학과 석사
[2] 전북대학교 기록관리학과 4단계 BK21 교육연구단 박사후 연구원
[3] 교신저자 전북대학교 기록관리학과 부교수, 문화융복합아카이빙연구소 연구원

OPEN ACCESS

논문접수 2021-07-20
최초심사 2021-07-27
게재확정 2021-08-03

ORCID
Sung-Ho Yoon
https://orcid.org/0000-0001-6197-5336
Jung-eun Lee
https://orcid.org/0000-0003-2631-0245
Dongmin Yang
https://orcid.org/0000-0002-4029-9372



▪본 연구는 "2019년 행정안전부 국가기록원기록관리연구개발사업"의 연구비를 지원받아 수행되었음.
▪이 논문은 2019년 대한민국 교육부와 한국연구재단의 지원을 받아 수행된 연구임 (NRF-2019S1A5B8099507).

## 초 록

4차 산업혁명의 도래로 데이터의 중요성이 커지는 상황에 따라, 해외 각국은 데이터 장기보존 기술 연구를 추진하고 있다. 반면 우리나라는 행정정보 데이터세트가 기록관리 영역으로 법제화됐으나, 구체적인 장기보존 방안이 부재한 상황이다. 이에 본 연구는 여러 선행연구에서 행정정보 데이터세트 보존포맷으로 제안된 SIARD(Software Independent Archiving of Relational Database)에 대한 기초, 교차 검증 시험을 수행했다. 먼저 기초 검증 시험은 SIARD 포맷이 보존할 수 있는 데이터세트의 데이터, 구조, 기능 등을 도출하는데 방점을 두었다. 두 번째 교차 검증 시험은 DBMS 종류에 구애받지 않는 SIARD의 상호호환성 검증에 목적을 두었다. 2차례 검증 시험 결과, SIARD 포맷으로 JSON, UROWID 데이터 타입, FK(Foreign Key), 함수 계열 요소를 보존할 수 없으며, SIARD 2.0 표준에 명시된 기능과 실제 SIARD Suite이 제공하는 기능에 차이가 있음을 확인하였다. 본 연구는 실증적 검증 시험을 진행했으며, SIARD Suite의 기능을 보완하는 개발 방안과 SIARD Suite을 국내 환경에 맞춰 효율적으로 개발할 수 있는 방향성을 제시했다는 점에서 의의가 있다.

## ABSTRACT

As the importance of data grows because of the advent of the next industrial revolution, foreign countries are pushing for long-term data preservation technology research. On the other hand, in Korea, administrative information data sets have been legislated as records management areas without specific long-term preservation measures. As a response, this study conducted basic, cross-validation tests on the Software Independent Archiving of Relational Database (SIARD), which was proposed as an administrative information data set preservation format in several prior works. First, the underlying verification test focuses on deriving the data, structure, and functionality of the data set that SIARD can preserve. The second cross-validation test aimed at verifying the interoperability of SIARD independent of the DBMS class. In addition, two verification tests have confirmed the SIARD feature delivery range. Consequently, the differences between the feature types specified in the SIARD 2.0 standard and those provided by the actual SIARD Suite have been derived. Based on verification test results, we are proposing a development plan to broaden SIARD functionality and set a direction to efficiently enhance SIARD for local situations.

Keywords: 행정정보 데이터세트, 보존포맷, SIARD 포맷, 장기보존
Data Set Records, Preservation Format, SIARD Format, Long-Term Preservation

# 1. 서 론

## 1.1 연구 필요성

4차 산업혁명의 도래로 데이터의 중요성이 점차 커지는 작금이다. 이에 주요 선진국들은 국가의 주요 자산인 데이터를 장기적으로 활용하기 위해 데이터 장기보존 기술 연구 및 표준 제정 등을 추진하여 데이터 보존에 심혈을 기울여 왔다(한희정 외, 2020). 우리나라도 『공공데이터의 제공 및 이용활성화에 관한 법률』(이하 '공공데이터법')과 『데이터기반 행정 활성화에 관한 법률안』(이하 '데이터기반행정법') 등을 제정해 데이터세트 수집, 이용, 활용을 위한 제도적 기반을 마련하였다. 특히 국가기록원은 지난 2007년 『공공기록물 관리에 관한 법률 시행령』(이하 '공공기록물법 시행령')제2조11항에 '행정정보 데이터세트'의 정의를 명시하는 등 선제적으로 전자기록물의 범위를 데이터세트로 확장하는 법제화를 진행했다. 하지만 데이터세트 기록의 수집・관리・보존방안 등 구체적인 기록관리 프로세스의 부재로, 지난 10여년 동안 행정정보 데이터세트는 사실상 기록관리 영역 밖에 있었다.

이에 국가기록원은 지난 2020년, 행정정보 데이터세트 기록관리를 위한 표준인 『행정정보 데이터세트 기록관리 기준』을 제정하였다. 해당 표준에는 데이터세트 관리를 위한 기본적인 요건을 명시했으며, 행정정보 데이터세트 관리기준표 작성을 제도화했다. 또한 데이터세트 이관 및 폐기도구 등을 제안하는 등 데이터세트 기록관리를 위한 초석을 마련하였다. 하지만 행정정보 데이터세트를 장기간 보존하기 위한 전략(e.g., 마이그레이션, 에뮬레이션, DB Dump 등)이나 보존포맷[1] 등은 여전히 부재하다. 특히 데이터세트는 시스템과 DBMS에 종속되는 특성을 가지고 있으므로 시스템과 소프트웨어에 독립적인 행정정보 데이터세트 기록의 장기보존 전략 구축이 시급한 상황이다.

이러한 국내 실정과는 달리, EU, 호주, 스위스 등 주요 국가들은 SIARD(Software Independent Archiving of Relational Database)를 행정정보 데이터세트 보존포맷으로 채택해 활용하고 있다. SIARD는 관계형 데이터베이스(Relational Database, RDB) 보존을 위한 파일포맷으로, 대부분의 행정정보 데이터세트가 RDB 유형인 국내에서 활용하기 적합하다. 따라서 국가기록원을 포함한 여러 연구에서 데이터세트 보존포맷으로 SIARD 채택 및 도입을 제안한 바 있다(이규철, 2016; 소정의, 한희정, 양동민, 2018; 국가기록원, 2019a; 한희정, 윤성호, 오효정, 양동민, 2020; 김주연, 2020). 하지만 김주연(2020)의 연구를 제외하면 SIARD에 대한 실증적 검증이 진행된 연구는 부족하며, 특히 SIARD가 데이터세트를 구성하는 데이터, 구조, 기능 등 다양한 요소의 보존 여부를 실험 및 검증한 연구는 부재한 상황이다. 또한 DBMS 종류와 버전에 제약받지 않고 데이터 상호호환이 가능한 SIARD의 데이터 보존 기능을 확인한 연구마저 진행되지 않았다.

따라서 본 연구는 행정정보 데이터세트 보존포맷으로서 SIARD 적합 여부를 판단하고자 국내 공공기관에서 주로 활용되는 2종의 DBMS(Oracle, SQL Server)[2]와 1종의 오픈 프로젝트 DBMS(MySQL) 등 총 3종의 DBMS를 대상으로 SIARD Suite 기능을 검증한다. 이를 위해 기초 검증 시험과 교차 검증 시험을 수행한다. 먼저 전자의 경우, 각 DBMS에서 제공하는 모든 데이터 타입과 함수 계열 요소인 Function, Stored Procedures, Trigger와 데이터 관계를 표현하는 Primary Key(PK), Foreign Key(FK) 등 데이터베이스의 개별 요소가 생산 당시 모습을 SIARD가 보존할 수 있는지를 검증하는 시험이다. 후자는 SIARD를 통한 DBMS 간 상호호환성을 검증하는 시험이다. 특히 SIARD 파일을 다른 DBMS로 복원해 원본 데이터베이스와 복원된 데이터베이스 간 데이터 무결성 검증을 수행하는 교차 검증에 방점을 두고자 한다.[3] 상기 과정을 수행해 SIARD Suite이 제공하는 기능을 확인함과 동시

1) 국가기록원의 전자기록 유형별 포맷 정책(안)(국가기록원, 2019b), 전자기록물 장기보존 정책(안)(국가기록원, 2019c)에 따르면, 기존의 문서보존포맷이 아닌 보존포맷으로 용어를 변경해 활용하고 있으며, 보존포맷 선정기준과 관련한 표준화 작업을 진행 중이다. 이에 본 연구에서의 보존포맷은 문서보존포맷을 의미한다.
2) 2019년 12월 기준, 전체 DBMS 중 Oracle은 64.17%(1위), SQL Server는 16.57%(2위) 점유율을 보이고 있다(행정안전부, 2020).
3) 예를 들어 Oracle DBMS에서 생성한 데이터세트를 SIARD 파일로 변환한 뒤, MySQL DBMS로 복원해 두 데이터베이스(Oracle, MySQL) 간의 데이터 무결성을 검증한다.

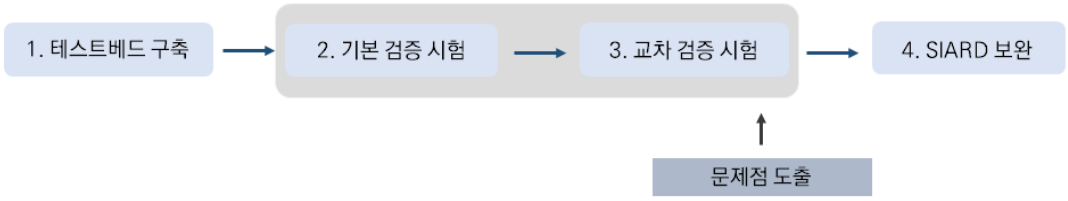

http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

에 이를 보완하는 개발을 수행해 SIARD의 활용성을 제고하고자 한다.

## 1.2 연구 방법 및 범위

본 연구는 행정정보 데이터세트 보존포맷으로서 SIARD Suite의 기본 기능을 확인함과 동시에 소프트웨어로부터 독립된 보존포맷이라는 이점인 DBMS 상호호환성을 검증하고자 한다. 이를 위해 본 연구는 크게 4단계로 구분할 수 있다. <그림 1>은 전체 연구 방법을 도식화한 모습이다. 첫 번째로 SIARD 검증을 위한 테스트베드를 구축한다. 이를 위해 공공기관 활용도가 높은 3종 DBMS(Oracel, MySQL, SQL Server)를 설치한 뒤 해당 DBMS에서 제공하는 모든 데이터 타입과 데이터 관계(PK, FK), 함수 계열 요소(Function, Stored Procedures, Trigger)를 포괄하는 각각의 원본 DB를 생성한다. 다음으로 SIARD 기초 검증 시험을 진행한다. 기초 검증 시험은 SIARD가 관계형 데이터베이스인 원본 DB의 필수보존속성(Significant Properties, SP)[4]인 기능과 구조, 데이터 등을 SIARD 포맷으로 변환 및 DBMS로 복원 등의 기능 지원 여부를 검증하고자 한다. 상기 과정을 수행해 SIARD가 보존할 수 있는 객체와 그렇지 않은 객체를 식별한다.

세 번째로 SIARD 교차 검증 시험을 진행한다. 해당 단계에서는 개별 DBMS에서 제공하는 유사한 데이터 타입들을 대상으로 공통 DB를 생성해 데이터 무결성을 검증한다. 데이터세트가 SIARD로 변환 및 DBMS로 복원되는 과정을 거치면서 SIARD 포맷을 통한 안정적인 데이터 보존 가능성을 확인한다. 이와 더불어 각각 DBMS에서 지원하는 유사하지만 상이한 데이터 타입에 저장하는 동일한 데이터가 SIARD 및 타 DBMS로 복원되더라도 데이터 무결성을 유지할 수 있는 지를 중점적으로 확인한다. 데이터 무결성 검증에는 Toad Data Point 5.1.0.142[5]를 활용한다. 마지막으로 선행 과정에서 도출된 SIARD의 문제점을 도출해 해당 문제를 보완하는 추가적인 개발 방안을 제시하고 더불어 국내 환경에 부합하는 SIARD 개발 방향성을 제안한다. 본 연구의 검증 시험에서 활용한 3종의 DBMS에서 제공하는 모든 데이터 타입의 보존을 가능케 해 소프트웨어 독립성에 기초한 SIARD의 상호호환성을 제고하고자 한다.




〈그림 1〉 전체 연구 방법 도식화

## 1.3 선행연구

그동안 기록관리 영역에서 벗어나 있던 행정정보 데이터세트를 제도권 안으로 들여오기 위한 선행연구가 주로 수행되었다. 왕호성과 설문원(2017)은 현행 전자기록관리정책으론 데이터세트 기록을 관리하기가 매우 어려운 실정임을 언급하며, 이를 해소하기 위한 정책방향을 제안하였다. 특히 데이터세트 기록의 진본성은 기록의 내용,

4) 필수보존속성(SP)은 전자기록물이 접근 가능하고 진본성이 보장된 상태를 유지할 수 있도록 보존되어야 하는 디지털 객체의 요소로, Appearane(외관), Behavior(기능), Content(내용), Context(맥락), Structure(구조)로 구분할 수 있다(The National Archives, 2008). 본 연구에서 관계형 데이터베이스의 SP를 기능(함수 계열 요소), 구조(PK・FK), 내용(데이터)으로 보았다.

5) Toad Data Point는 퀘스트소프트웨어에서 개발한 소프트웨어로, Oracle, MySQL, SQL Server, PostgreSQL 등의 환경에서 데이터 조회, 분석 및 DBMS 관리를 지원한다. 본 연구에서는 Toad Data Point의 기능 중 DBMS 접속 및 데이터 비교 기능을 사용했다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

구조, 맥락뿐만 아니라 외형과 기능을 함께 보존해야 확보될 수 있음을 언급하며, 데이터세트를 전자객체가 아닌 기록으로 관리하기 위해선 재현성이 전제되어야 함을 강조하였다. 이러한 맥락으로 데이터세트 기록의 재현 및 장기보존 전략으로 에뮬레이션을 제안하였다. 오세라, 박승훈, 임진희(2018)는 IT 기술의 발전으로 인해 행정정보시스템의 신규 구축과 재개발이 증가하고 있음에도 불구하고 실제 공공기관에서 운영 중인 행정정보시스템에서 생산된 데이터세트가 관리되지 못함을 문제점으로 언급하였다. 이에 대한 구체적인 원인으로 현실 적용이 가능한 데이터세트 관리 방안의 부재를 지적하였다. 따라서 구현 가능한 행정정보 데이터세트 관리 방안은 데이터세트 관리 환경을 기초로 해야 한다는 판단 하에 실제 공공기관에서 운영 중인 6종의 행정정보시스템 생산 및 관리 환경을 조사하였다. 해당 연구의 결과는 데이터세트 관리 방안 개발의 기초 자료로 활용될 수 있으며, 유사 연구에서 활용될 수 있는 조사방법론을 제안했다는 점에서 의의가 있다. 오세라와 이해영(2019)은 행정정보 데이터세트 기록의 관리 필요성과 시급성에 비해 실제 현장에서는 데이터세트가 관리되지 않으며, 현행 종이기록 중심의 표준 기록관리 지침과 절차를 데이터세트에 적용하지 못함을 문제점으로 제기하였다. 이를 해소하고자 실제 공공기관에서 운영 중인 6종 행정정보시스템의 데이터세트 현황 조사 및 분석을 진행하였다. 분석 결과를 기반으로 데이터세트 관리방안과 데이터세트 관리기준표를 제안하였다. 대부분의 선행연구는 행정정보 데이터세트 관리를 위한 전반적인 프로세스 구축에 방점을 두고 있으며, 구체적인 장기보존 방안과 보존포맷에 대한 연구는 수행되지 못한 상황이었다.

행정정보 데이터세트 보존포맷에 관한 연구로는 한희정 외(2020)의 연구가 있다. 한희정 외(2020)는 데이터와 데이터세트의 중요성이 부각되면서 해외 각국은 데이터 장기보존 기술 연구 및 표준 제정을 추진하는 등 데이터 관리 및 보존에 노력을 기울이고 있지만, 국내는 구체적인 관리방안이 부재한 실정을 언급하였다. 특히 행정정보시스템에서 생산되는 엄청난 규모의 데이터세트를 관리・보존하기 위한 방안이 시급하지만 데이터세트에 대한 관리지침과 보존포맷 선정기준이 선제적으로 필요함을 강조하였다. 이를 위해 행정정보 데이터세트 보존포맷 선정기준에 대한 평가체계를 도출하고, 평가 결과에 따라 적합성을 가지는 파일포맷에 대한 실증적 검증을 수행하였다. 해당 연구는 연구 결과로 도출된 평가체계와 보존포맷으로서 SIARD에 대한 검증을 수행하였다는 의의가 있다. 김주연(2020)은 현장에서 행정정보 데이터세트의 실질적인 관리가 이뤄지지 못하고 있으며, 데이터베이스가 시스템과 DBMS 제조사에 종속되어 있어 이관 및 장기보존 시 기술적인 지원이 필요하다는 문제점을 제기하였다. 특히 행정정보시스템과 데이터세트의 유형이 방대하므로 소프트웨어에 독립적인 SIARD를 기반으로 행정정보 데이터세트 기록관리 및 장기보존 방안을 제안하였다.

한희정 외(2020), 김주연(2020) 등으로 대표되는 SIARD 관련 연구는 주로 SIARD의 기본적인 기능인 다운로드(DBMS에 탑재된 DB를 SIARD 파일로 추출), 업로드(SIARD 파일의 DB를 DBMS로 탑재)를 확인하는 수준의 검증만 이뤄졌다. SIARD 표준에 명시된 기본적인 성능을 실제로 확인하였다는 점에서 의의가 있으나, SIARD의 활용성을 제고하기 위한 여러 DBMS 간 상호호환성 검증과 개발이 필요하다고 사료된다. 따라서 본 연구는 행정정보 데이터세트 보존포맷으로서 SIARD의 적합성을 확인하는 교차검증과 더불어 DBMS에서 제공하는 다양한 기능을 보존할 수 있도록 추가적인 개발을 진행하고자 한다.

## 2. 이론적 배경

### 2.1 데이터세트와 행정정보 데이터세트

기록학 용어 사전에 명시된 데이터세트의 정의는 '컴퓨터가 처리하거나 분석할 수 있는 형태로 존재하는 관련 정보의 집합체'이며, 영국 국가기록원(The National Archives, TNA)은 '특별한 목적을 위해 생산된 구조화된 데이


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

터 집합으로, 다양한 포맷과 기술과 저장, 관리, 생산이 가능'하다고 하였다. 이처럼 데이터세트는 컴퓨터에 의해 처리되는 것을 전제로 함과 동시에 텍스트, 숫자, 이미지, 공간 정보 등 다양한 형태의 데이터로 이뤄짐을 알 수 있다. 이는 데이터세트가 다른 전자기록물들과 구분되는 가장 큰 이유로, 컴퓨터가 내용을 확인할 수 있으면 되므로 외관은 전혀 고려되지 않는다. 즉, 문자, 표, 이미지 등의 크기・폰트・색상・음영 등은 중요하지 않고 표현하고자 하는 내용이 중요한 것이다(한희정 외, 2020).

데이터세트는 파일에 저장하는 방식과 데이터베이스에 저장하는 방식 등 크게 2가지로 구분할 수 있다. 파일에 저장되는 방식도 텍스트 파일(Text File)로 저장되는 방식과 문자열 외에 다른 여러 형태의 데이터를 포함하는 이진 파일(Binary File)로 나눌 수 있다. CSV, JSON, TXT, XML 등이 대표적인 텍스트 파일 방식이고, XLS, CELL 등이 대표적인 이진 파일이며, 스프레드시트(Spreadsheet)로 불리운다. 또한 데이터베이스는 Oralce, MySQL, SQL Server로 대표되는 관계형 데이터베이스 유형과 MongoDB, DynamoDBN, DataStax 등 NoSQL 유형으로 구분된다(노종원, 소정의, 2020). 이중 행정정보 데이터세트는 데이터베이스 유형에 해당한다. 국가기록원 행정정보 데이터세트 기록관리기준-관리기준표의 작성 및 이관규격에 따르면 대부분의 공공기관에서 관계형 데이터베이스를 사용하고 있으나, 최근 빅데이터 처리를 위해 비관계형 데이터베이스(NoSQL 등) 사용이 증가하는 추세이다.

## 2.2 데이터세트의 장기보존 전략

데이터세트의 장기보존 전략으로 크게 3가지를 제안할 수 있다. DBMS에서 제공하는 기능인 DB 덤프(Dump), 데이터 생산 당시에 사용한 소프트웨어와 데이터를 재현하는 에뮬레이션(Emulation), 데이터의 무결성을 유지한 채 데이터의 파일포맷, 어플리케이션 등을 변환하는 프로세스인 마이그레이션(Migration) 등이 있다. 먼저 DB 덤프는 Oracle, MySQL, SQL Server, Cubrid 등과 같은 DBMS에서 지원하는 기능으로 데이터 손실 시 또는 이관 시 데이터베이스의 사본을 제작해 해당 내용을 복원하거나 백업하는데 사용된다. 덤프는 개별 DBMS에서 수행하는 기능이므로 가장 간단한 보존전략이지만, 덤프 데이터를 타 DBMS로 복원 및 백업이 불가하다. 최소 30년 이상 보존해야하는 장기보존 대상을 특정 DBMS에서만 데이터를 확인할 수 있는 상황은 장기보존 측면에서 큰 애로사항이다. 이는 데이터베이스가 DBMS 제조사에 종속되기 때문에 장기보존 측면에서는 부적합하다.

에뮬레이션은 '디지털 원본에 적용된 기술적인 조건들에 변경이 있어도 인코딩되어 있는 콘텐츠를 재생할 수 있는 환경을 프로그램으로 만들어내어 디지털정보의 접근성을 보장하는 방식'(한국도서관협회, 2010)이다. 에뮬레이션은 생산 당시의 외관과 기능을 그대로 재현할 수 있어서 데이터세트를 재활용하기에 가장 적합한 방식이다. 또한 DBMS의 데이터세트는 독자적으로 활용되지 않고 WAS/WEB 서버들과 연계되어 운용되는 경우가 많기 때문에 데이터세트만 보존하기보단 시스템 전체를 보존하기 위한 에뮬레이션 방식이 적합한 경우가 많다(국가기록원, 2019a). 하지만 에뮬레이션을 공공기관에서 데이터세트 장기보존 전략으로 채택해 활용하기엔 고도의 IT 기술이 요구되고 비용이 많이 소요된다는 단점이 존재한다.

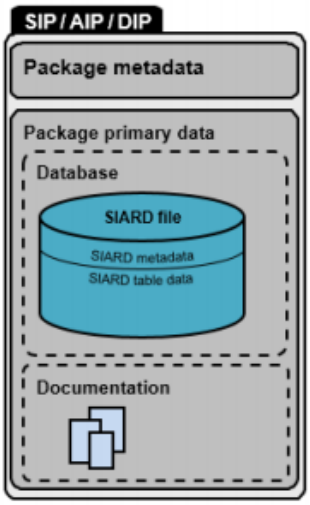
마지막으로 마이그레이션은 데이터의 형식과 파일포맷, 어플리케이션 등을 변환하는 프로세스를 의미한다. 예컨대 종이기록물을 전자화하여 전자기록물로 관리하는 것도 일종의 마이그레이션이다. 데이터세트의 경우, 타 DBMS로 변환하거나 파일로 변환해 보존하는 방안을 고려할 수 있다. 마이그레이션은 특별한 IT 기술과 큰 금액이 필요하지 않은 장점이 있다. 또한 데이터세트를 보존하기 위한 국제 표준격인 SIARD가 이미 여러 국가에서 활용 중이므로 마이그레이션을 위한 파일포맷으로 활용이 가능하다. 하지만 마이그레이션 전략을 수행함에 있어 중요한 사항은 마이그레이션을 지원하는 도구의 성능과 원본 DBMS와 마이그레이션 후 데이터를 보존할 파일포맷 및 DBMS 간 상호호환성을 확보하는 것이다. 이를 위해 후행 절에서 SIARD와 지원 도구인 SIARD Suite에 대해 살펴보고 여러 DBMS 간 호환성을 가지는지 검증 실험을 수행하고자 한다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

## 2.3 SIRAD와 SIARD Suite

SIARD는 스위스 연방 기록원(Swiss Federal Archives, SFA)에서 개발한 관계형 데이터베이스 보존포맷으로, 소프트웨어로부터 독립해 데이터를 보존할 수 있는 파일 포맷이다. 스위스 연방 기록원의 ARELDA 프로젝트의 장기보존 전략을 기반으로 개발되었다. 2008년 SIARD 1.0 버전은 European Open PLANETS 프로젝트에서 관계형 데이터베이스를 보존하기 위한 공식 파일포맷으로 채택되었으며, SIARD 2.0 버전은 2015년 SFA와 E-ARK(Europeana Archival Records and Knowledge Preservation) 프로젝트에 의해 개발되었다(김주연, 2020). 특히 SIARD는 SQL:2008, XML, Unicode, ZIP 표준을 기반으로 제작되어 원본 데이터베이스를 지원하는 DBMS를 활용할 수 없더라도 이들 표준에 근거해 데이터베이스를 접근 및 변환할 수 있으므로 보존포맷으로서 가치를 지닌다.

한편, SIARD는 OAIS 패키지 모델 구조와 독립적으로 설계되어 OAIS 패키지 메타데이터와 관계없이 자체적으로 메타데이터를 가지고 있으며, 다른 문서들(외부 Large Object 파일, 외부 파일 이름에 대한 변환 맵, DB 문서, DB 구조와 관련 문서 등)과 함께 보존되는 것으로 가정한다(<그림 2> 참조). SIARD 아카이브 구조를 보면 메타데이터와 테이블데이터가 결합된 구조로 하나의 관계형 DB는 단일의 SIARD 파일로 저장되며, 모든 DB 콘텐츠는 XML 스키마 1.0의 스키마 정의에 따라 XML 1.0 포맷의 파일 집합으로 보관된다. 스키마 정의와 SQL 코드는 앞서 언급한 바와 같이 SQL:2008을 따른다(한희정 외, 2020).




〈그림 2〉 SIARD 구조
출처: eCH-0165

SFA는 SIARD 표준과 함께 SIARD를 지원하는 도구로 SIARD Suite 오픈 프로젝트를 개발하고 있다. SIARD Suite은 국제 표준을 준수하며, 전 세계 50개국에서 사용 중에 있다(Swiss Federal Archives, 2021). 또한 JAVA를 기반으로 제작되었으며, 운영체제에 독립적으로 작동한다. SIARD Suite은 오픈소스 라이선스인 CDDL v1.0이므로 SFA의 Gitgub를 통해 무료로 활용할 수 있다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

# 3. SIARD 검증 시험

## 3.1 테스트베드 구축

본 장에서는 행정정보 데이터세트 보존포맷으로서 SIARD의 변환 및 복원 기능을 검증하는 기초 검증 시험과 DBMS 간 데이터 무결성을 검증하는 교차 검증 시험을 진행하고자 한다. 먼저 SIARD 기초 검증 시험에서는 데이터베이스의 필수보존속성인 모든 데이터 타입과 함수 계열 요소, 데이터 관계(PK, FK 등)를 SIARD 포맷으로 변환 및 DBMS로 복원 과정을 수행해 SIARD Suite이 제공하는 변환, 복원 기능을 검증하고자 한다. 이어서 진행하는 교차 검증 시험에서는 원본 DB가 변환 및 복원 과정을 거친 뒤 DBMS에 업로드 된 DB와 데이터 무결성을 검증한다. 무결성 검증 시 Toad Data Point 5.1.0.142를 활용해 원본 DB와 업로드 DB의 데이터를 비교 검증한다.

해당 시험을 통해 SIARD가 한희정 외(2020)의 연구에서 제안한 행정정보 데이터세트 보존포맷 선정을 위한 고유기준인 '일반화', '수용성', '활용성'[6] 등 부합 여부를 확인할 수 있다. 특히 '일반화' 기준과 같이 다양한 DBMS의 호환성 검증에서 한 단계 더 나아가, 서로 상이한 DBMS 간 SIARD의 호환성 및 데이터 무결성 검증이 가능하다.

검증 시험을 위해 본 연구진이 구축한 테스트베드는 <표 1>, <표 2>와 같다. 동일한 환경에서 시험을 진행해야 일관성 있는 결과를 도출할 수 있으므로 <표 2>에 제시된 환경과 SIARD Suite으로 검증 시험을 수행했다. 검증 시험 대상으로는 공공기관 활용도가 높은 DBMS 2종(Oracle, SQL Server)과 오픈 프로젝트 DBMS 1종(MySQL)등 총 3종의 DBMS를 선정했다. 개별 DBMS에서 제공하는 데이터 타입을 파악하고자 제조사 매뉴얼 및 홈페이지를 참조해 데이터 타입을 종류별로 분류했다(<표 3> 참조). 데이터 타입 중 숫자, 문자 및 이진 계열, Large Object 계열, 날짜 및 시간 계열 데이터 타입은 일반 데이터 타입으로 분류하였으며, 이에 해당하지 않는 데이터 타입은 특수 데이터 타입으로 정의하였다. 개별 DBMS에 생성한 원본 DB는 5개 테이블로 구성되어 있으며, 테이블별 100건의 레코드, 추가적으로 함수 계열 요소(Function, Stored Procedures, Trigger)와 테이블 간 PK(Primary Key), FK(Foreign Key) 관계를 부여해 해당 기능들의 보존 여부를 검증하고자 한다.

〈표 1〉 SIARD 검증 시험 테스트 환경 스펙 및 SIRAD Suite 정보

| 종류 | 상세내용 |
|---|---|
| 1. HW 정보 | CPU: i7-8750H 2.2GHz, RAM: 32GB, SSD: 1TB |
| 2. OS 정보 | Windows 10 |
| 3. SIARD Suite 정보 | SIARD Suite 2.1.105 |

6) 일반화(Normalization)란 행정정보 데이터세트 보존포맷이 상용화된 여러 DBMS와 호환이 되어야하다는 기준.
수용성(Acceptability)이란 DBMS에서 제공하는 다양한 기능(데이터 타입, 함수, 데이터 구조 및 관계 등)을 수용할 수 있어야한다는 기준.
활용성(Usability)이란 데이터세트를 보존포맷으로 변환 후 다시 활용 가능해야 한다는 기준.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

〈표 2〉 3종 DBMS 원본 DB 정보

| 종류 | | Oracle 11g | MySQL 8.0 | SQL Server 2017 |
|---|---|---|---|---|
| 일반 데이터 타입 | 숫자 | NUMBER, FLOAT, BINARY_FLOAT, BINARY_DOUBLE | BIT, INT, TINYINT, SMALLINT, MEDIUMINT, BIGINT, NUMERIC, DECIMAL, DOUBLE, REAL, FLOAT, BOOLEAN | BIT, INT, TINYINT, SMALLINT, BIGINT, MONEY, SMALLMONEY, NUMERIC, DECIMAL, FLOAT, REAL |
| | 문자/이진 | CHAR, VARCHAR2, NCHAR, NVARCHAR2 | CHAR, VARCHAR, BINARY, VARBINARY | CHAR, NCHAR, VARCHAR, NVARCHAR, BINARY, VARBINARY |
| | Large Object | LONG, RAW, LONG RAW, BLOB, BFILE, CLOB, NCLOB | BLOB, TINYBLOB, MEDIUMBLOB, LONGBLOB, TEXT, TINYTEXT, MEDIUMTEXT, LONGTEXT | TEXT, NTEXT, IMAGE |
| | 날짜/시간 | DATE, TIMESTAMP, TIMESTAMP WITH TIME ZONE, TIMESTAMP WITH, LOCAL TIME ZONE, INTERVAL YEAR TO MONTH, INTERVAL DAY TO SECOND | DATE, TIME, DATETIME, TIMESTAMP, YEAR | DATE, TIME, DATETIME, DATETIME2, DATETIMEOFFSET, SMALLDATETIME |
| 특수 데이터 타입 | 기타 | ROWID, UROWID | JSON, GEOMETRY, POINT, MULTIPOINT, LINESTRING, MULTILINESTRING, POLYGON, MULTIPOLYGON, GEOMETRY, COLLECTION | GEOGRAPHY, GEOMETRY |
| 함수 계열 요소 | | Function, Stored Procedures, Trigger | | |
| 데이터 관계 | | Primary Key(PK), Foreign Key(FK) | | |

## 3.2 SIARD 기초 검증 시험

선행 절에서 언급한 테스트베드에서 수행한 SIARD 기초 검증 결과는 <표 3>과 같다. 각 DBMS별 검증 결과는 후행 절에서 자세히 설명하고자 한다. DBMS에서 생성한 원본 DB를 SIARD로 변환 및 DBMS로 복원하는 일련의 시험 과정을 수행한다.

〈표 3〉 3종 DBMS↔SIARD 기초 검증 결과 요약표

| 항목 | 일반 데이터 타입 (숫자, 문자, 날짜 등) | 특수 데이터 타입 | 데이터 관계 (PK, FK) | 함수 계열 요소 |
|---|---|---|---|---|
| Oracle → SIARD (다운로드) | ◎ | ○ | ◎ | X |
| SIARD → Oracle (업로드) | ◎ | ○ | ○ | X |
| MySQL → SIARD (다운로드) | ◎ | ○ | ◎ | X |
| SIARD → MySQL (업로드) | ◎ | ○ | ◎ | X |
| SQL Server → SIARD (다운로드) | ◎ | ◎ | ◎ | X |
| SIARD → SQL Server (업로드) | ◎ | ◎ | ◎ | X |

(◎: 모두 변환 가능, ○: 부분 변환 가능, X: 변환 불가능)


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

### 3.2.1 Oracle ↔ SIARD 기초 검증 결과

Oracle과 SIARD 간 검증 결과에 따르면 데이터 관계 중 FK와 함수 계열 요소를 제외한 대부분은 SIARD로 정상 변환이 가능함을 확인하였다. 먼저 데이터 타입의 경우, <표 4>와 같이 Oracle의 데이터 타입이 다른 데이터 타입으로 변환돼 SIARD로 저장되며, 데이터도 동일하게 저장되는 것을 알 수 있다. 이 중 SIARD로 변환되지 않는 데이터 타입은 'UROWID' 타입으로 해당 데이터 타입을 포함하는 DB를 SIARD 파일로 변환 시 SIARD Suite이 종료되는 현상이 반복적으로 발생하였다. 이 후 'UROWID' 타입을 제외한 뒤 변환을 시도한 경우, 정상적으로 변환이 되는 것을 확인하였다.

다음으로 데이터 관계를 표현하는 PK와 FK 중 FK는 SIARD 변환이 불가하다. PK와 FK 모두 SIARD 파일로 변환은 가능하지만, DBMS로 업로드 시 FK는 누락되는 것을 확인하였다. 해당 문제의 경우, 원본DB에서 생성한 PK의 제약조건 이름(SYS_C0012339)이 SIARD 변환을 거치면서 업로드 시 임의로 제약조건 이름(SYS_C0012347)이 변경된 것에서 기인한다고 판단된다(<그림 3>, <그림 4> 참조). 구조적으로 FK는 기존에 정의된 PK와 관계를 맺는다. 하지만 앞서 언급한 바와 같이 임의로 PK의 제약조건 이름이 변경된다면 FK는 링크를 맺고 있던 PK을 인식할 수 없으므로 누락되는 것으로 사료된다.

마지막으로 함수 계열 요소의 경우, 모든 유형을 SIARD로 변환할 수 없다. <그림 5>와 같이 함수 계열 요소를 생성해 SIARD 파일로 변환할 경우, 개별 객체의 명칭은 SIARD로 변환이 가능하다(<그림 7> 참조). 하지만 DBMS로 업로드 시 객체 명칭이 누락되는 것을 확인하였다(<그림 6> 참조).

〈표 4〉 Oracle ↔ SIARD 기초 검증 결과 요약표

| 항목 | Oracle 11g | SIARD(SQL:2008) |
|---|---|---|
| 일반 데이터 타입 | NUMBER | DECIMAL |
| | FLOAT | FLOAT |
| | BINARY_FLOAT | REAL |
| | BINARY_DOUBLE | DOUBLE PRECISION |
| | CHAR | CHAR |
| | VARCHAR2 | VARCHAR |
| | NCHAR | NCHAR |
| | NVARCHAR2 | NCHAR VARYING |
| | LONG | CLOB |
| | RAW | VARBINARY |
| | LONG RAW | BLOB |
| | BLOB | |
| | BFILE | |
| | CLOB | CLOB |
| | NCLOB | NCLOB |
| | DATE | DATE |
| | TIMESTAMP | TIMESTAMP |
| | TIMESTAMP WITH TIME ZONE | |
| | TIMESTAMP WITH LOCAL TIME ZONE | |
| | INTERVAL YEAR TO MONTH | INTERVAL YEAR TO MONTH |
| | INTERVAL DAY TO SECOND | INTERVAL DAY TO SECOND |
| 특수 데이터 타입 | ROWID | BIGINT |
| | **UROWID** | **변환불가** |
| 데이터 관계 | PK | 변환가능 |
| | **FK** | **변환불가** |
| 함수 계열 요소 | **Function, Stored Procedures, Trigger** | **변환불가** |

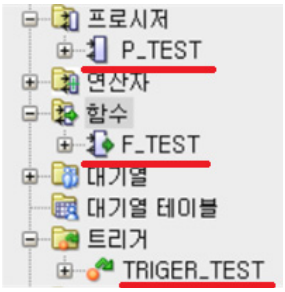

http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

| | CONSTRAINT_NAME | CONSTRAINT_TYPE |
|---|---|---|
| 1 | FK_PUB_ID_PUBLISHED_PUB_ID | Foreign_Key |
| 2 | FK_WRI_ID_WRITERS_WRI_ID | Foreign_Key |
| 3 | SYS_C0012338 | Check |
| 4 | SYS_C0012339 | Primary_Key |

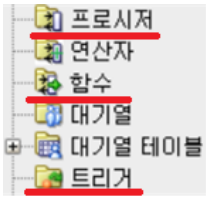
〈그림 3〉 원본DB의 "BOOK" 테이블 PK, FK

| | CONSTRAINT_NAME | CONSTRAINT_TYPE |
|---|---|---|
| 1 | SYS_C0012346 | Check |
| 2 | SYS_C0012347 | Primary_Key |

〈그림 4〉 업로드 DB의 "BOOK" 테이블 PK




〈그림 5〉 Oracle의 함수 계열 요소 변환 전 화면




〈그림 6〉 Oracle의 함수 계열 요소 업로드 화면

```
<routines>
  <routine>
    <specificName>F_TEST</specificName>
    <name>F_TEST</name>
    <returnType>VARCHAR</returnType>
  </routine>
  <routine>
    <specificName>P_TEST</specificName>
    <name>P_TEST</name>
  </routine>
  <routine>
    <specificName>TRIGER_TEST</specificName>
    <name>TRIGER_TEST</name>
  </routine>
</routines>
```

〈그림 7〉 Oracle의 함수 계열 요소 SIARD 변환 모습

### 3.2.2 MySQL ↔ SIARD 기초 검증 결과

MySQL과 SIARD 간 검증 결과는 <표 5>와 같다. 먼저 데이터 타입의 경우, 일반 데이터 타입은 모두 정상적으로 변환이 되며, 데이터도 안정적으로 보존되는 것을 확인하였다. 특수 데이터 타입의 경우, 'JSON'을 제외한 다른 항목들은 'CLOB'으로 정상 변환이 가능하다. 'JSON' 타입은 자바스크립트를 토대로 개발된 언어로, 텍스트로 기술되어 사람도 쉽게 읽고 작성이 가능하다. 또한 속성과 값을 한 쌍으로 묶어 구성되어, "속성:값" 형식으로 데이터 객체를 표현한다(한국정보통신기술협회, 2018). 이후 ISO/IEC 21778:2017 표준에 의해 제정이 되었으나, SIARD가 참조한 표준인 SQL:2008에 정의되지 않은 데이터 타입이므로, SIARD가 지원하지 못하는 것으로 판단된다.

다음으로 PK와 FK는 모두 정상적으로 변환되는 것을 확인하였다. 반면 함수 계열 요소의 객체들은 Oracle과 마찬가지로 객체명은 SIARD로 변환하지만 DBMS로 업로드 시 모두 누락된다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

〈표 5〉 MySQL ↔ SIARD 기초 검증 결과 요약표

| 항목 | MySQL | SIARD(SQL:2008) |
|---|---|---|
| 일반 데이터 타입 | BIT | BOOLEAN |
| | INT | INTEGER |
| | TINYINT | SMALLINT |
| | SMALLINT | |
| | MEDIUMINT | INTEGER |
| | BIGINT | BIGINT |
| | NUMERIC | DECIMAL |
| | DECIMAL | |
| | DOUBLE | DOUBLE PRECISION |
| | REAL | |
| | FLOAT | FLOAT |
| | BOOLEAN<br>(SIARD에서 TINYINT로 인식) | SMALLINT |
| | CHAR | CHARACTER |
| | VARCHAR | VARCHAR |
| | BINARY | BINARY |
| | VARBINARY | VARBINARY |
| | TINYBLOB | |
| | BLOB | BLOB |
| | MEDIUMBLOB | |
| | LONGBOLB | |
| | TINYTEXT | VARCHAR |
| | TEXT | CLOB |
| | MEDIUMTEXT | |
| | LONGTEXT | |
| | ENUM | VARCHAR |
| | SET | |
| | DATE | DATE |
| | TIME | TIME |
| | DATETIME | TIMESTAMP |
| | TIMESTAMP | |
| | YEAR | SMALLINT |
| 특수 데이터 타입 | **JSON** | **변환불가** |
| | GEOMETRY | CLOB |
| | POINT | |
| | MULTIPOINT | |
| | LINESTRING | |
| | MULTILINESTRING | |
| | POLYGON | |
| | MULTIPOLYGON | |
| | GEOMETRYCOLLECTION | |
| 데이터 관계 | PK | 변환가능 |
| | FK | |
| 함수 계열 요소 | **Function, Stored Procedures, Trigger** | **변환불가** |


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

### 3.2.3 SQL Server ↔ SIARD 기초 검증 결과

SQL Server와 SIARD 간 검증 결과는 <표 6>과 같다. 모든 데이터 타입이 SIARD로 변환되어 아래의 표와 같이 모두 매핑되는 것을 알 수 있다. 또한 PK와 FK 모두 정상적으로 변환되는 것을 확인하였으며, 함수 계열 요소의 경우 객체명은 SIARD로 변환이 가능하지만 업로드 시 누락되는 것을 확인하였다.

〈표 6〉 SQL Server ↔ SIARD 기초 검증 결과 요약표

| 항목 | SQL Server 2014 | SIARD(SQL:2008) |
|---|---|---|
| 일반 데이터 타입 | BIT | BOOLEAN |
| | INT | INTEGER |
| | TINYINT | SMALLINT |
| | SMALLINT | |
| | BIGINT | BIGINT |
| | MONEY | DECIMAL |
| | SMALLMONEY | |
| | NUMERIC | NUMERIC |
| | DECIMAL | DECIMAL |
| | FLOAT | DOUBLE PRECISION |
| | REAL | REAL |
| | CHAR | CHARACTER |
| | NCHAR | NCHAR |
| | VARCHAR | VARCHAR |
| | NVARCHAR | NCHAR VARYING |
| | BINARY | BINARY |
| | VARBINARY(MAX) | VARBINARY |
| | TEXT | CLOB |
| | NTEXT | NCLOB |
| | IMAGE | BLOB |
| | DATE | DATE |
| | TIME | TIME |
| | DATETIME | TIMESTAMP |
| | DATETIME2 | |
| | DATETIMEOFFSET | VARCHAR |
| | SMALLDATETIMEOFFSET | TIMESTAMP |
| 특수 데이터 타입 | GEOGRAPHY | VARCHAR |
| | GEOMETRY | VARCHAR |
| 데이터 관계 | PK | 변환가능 |
| | FK | |
| 함수 계열 요소 | **Function, Stored Procedures, Trigger** | **변환불가** |

### 3.2.4 소결

기초 검증 시험은 데이터세트를 구성하는 여러 요소에 대한 SIARD의 보존 기능을 확인하는데 목적이 있다. 시험 결과 DBMS에서 제공하는 대부분의 데이터 타입은 SIARD로 변환 및 보존이 가능했지만, Oracle과 MySQL


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

의 특수 데이터 타입(UROWID, JSON)과 함수 계열 요소 및 데이터 관계를 표현하는 PK, FK는 Oracle을 제외한 MySQL과 SQL Server에서 SIARD로 변환이 가능한 것으로 확인되었다. 이는 SIARD가 Oracle, MySQL, SQL Server 등 여러 DBMS에 적용가능하며, 함수 계열 요소의 변환이 가능하다고 명시된 SIARD 표준과는 상반된 검증 결과이다. 또한 본 검증 시험을 통해 실제 SIARD 표준의 기능과 실제 SIARD Suite을 통해 제공되는 기능에 차이가 있음을 알 수 있다.

이에 교차 검증 시험은 SIARD로 보존이 가능한 데이터 타입과 데이터를 3종 DBMS에 공통 DB로 생성한 뒤 SIARD를 통한 데이터 무결성 및 DBMS 간 상호호환성 검증을 진행하고자 한다.

## 3.3 SIARD 교차 검증 시험

본 절에서는 특정 DBMS에서 생성한 원본 DB를 SIARD 변환 후, DBMS로 업로드한 뒤 원본 DB의 데이터와 업로드 DB의 데이터가 서로 무결성을 보장하는지 검증하는 교차 검증 시험에 대해 설명하고자 한다. 본 검증 시험에서는 모든 데이터 타입을 활용한 앞선 시험과는 달리 SIARD(SQL:2008)와 정상적으로 매핑되는 데이터 타입 중 3종의 DBMS가 공통적으로 지원하는 유사 데이터 타입을 선정해 이용했다. 선정된 데이터 타입은 <표 7>과 같으며, 이를 활용해 개별 DBMS에 공통 DB(5개 테이블, 테이블별 100건 레코드)를 생성하였다. 공통 DB 생성 후 데이터 무결성을 검증하는 과정은 <표 8>과 같다. 각각의 DBMS에서 생성한 공통 DB는 SIARD 파일로 변환한 뒤 3종의 DBMS에 업로드한다. 이 과정을 거쳐 원본 DB와 업로드 DB의 데이터는 Toad Data Point 5.1.0.142를 통해 비교 검증된다.

〈표 7〉 DBMS별 공통 DB 데이터 타입

| | SIARD(SQL:2008) | Oracle 11g | MySQL | SQL Server 2014 |
|---|---|---|---|---|
| 1 | INT | NUMBER | INT | INT |
| 2 | SMALLINT | NUMBER | SMALLINT | SMALLINT |
| 3 | BIGINT | NUMBER | BIGINT | BIGINT |
| 4 | FLOAT | FLOAT | FLOAT | FLOAT |
| 5 | DECIMAL | NUMBER | DECIMAL | DECIMAL |
| 6 | DOUBLE | BINARY DOUBLE | DOUBLE | DOUBLE |
| 7 | DECIMAL | NUMBER | DECIMAL | NUMERIC |
| 8 | DOUBLE | BINARY DOUBLE | DOUBLE | REAL |
| 9 | SMALLINT | NUMBER | TINYINT | SMALLINT |
| 10 | CHAR | CHAR | CHAR | CHAR |
| 11 | VARCHAR | VARCHAR2 | VARCHAR | VARCHAR |
| 12 | NCHAR | NCHAR | CHAR | NCHAR |
| 13 | VARCHAR | VARCHAR2 | VARCHAR | VARCHAR |
| 14 | VARCHAR | VARCHAR2 | VARCHAR | VARCHAR |
| 15 | BLOB | BLOB | BLOB | VARBINARY |
| 16 | CLOB | CLOB | TEXT | TEXT |
| 17 | DATE | DATE | DATE | DATE |
| 18 | TIME | VARCHAR2 | TIME | TIME |
| 19 | TIMESTAMP | TIMESTAMP | TIMESTAMP | DATETIME2 |


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

〈표 8〉 SIARD 교차 검증 시험 순서

| 순서 | 상세 내용 | |
|---|---|---|
| 1. 원본 DB 생성 | 〈원본 DB〉<br>MySQL | 공통된 쿼리문을 이용해 원본 DB 생성 |
| 2. SIARD 파일 생성 | 생성한 DB를 SIARD 파일로 변환 | |
| 3. DBMS로 Upload | 〈업로드 DB〉<br>Oracle, SQL Server, MySQL | 원본DB(MySQL)의 SIARD 파일을 비교 검증하기 위해<br>업로드DB(SQL Server와 Oracle)로 Upload 진행 |
| 4. 데이터 무결성 검증 | Toad Data Point 5.1.0.142를 활용해 원본 DB와 업로드 DB의 데이터 무결성 검증 | |

SIARD 교차 검증의 결과는 <표 9>와 같다. 9가지 검증 결과는 각각의 사례별로 자세히 살펴보고자 한다.

〈표 9〉 SIARD 교차 검증 결과 요약표

| 원본 DB / 업로드 DB | Oracle | MySQL | SQL Server |
|---|---|---|---|
| Oracle | 1) ◎ | 4) △ | 7) △ |
| MySQL | 2) ○ | 5) ◎ | 8) ○ |
| SQL Server | 3) ○ | 6) ○ | 9) ◎ |

(◎: 데이터 일치, ○: Toad Data Point에선 데이터가 일치하지 않지만, 실제 데이터는 일치하는 경우, △: 시험 결과 데이터가 일치하지 않으며 실제로 일치하지 않는 경우)

1) Oracle → Oracle 교차 검증 결과

Oracle과 Oracle 교차 검증 결과 모든 데이터가 일치하는 것으로 나타났다.

2) Oracle → MySQL 교차 검증 결과

Oracle과 MySQL 교차 검증 결과 Toad Data Point 상으론 데이터가 일치하지 않은 것으로 나타났으나, SELECT 문을 활용해 실제 데이터를 확인해보니 데이터가 모두 일치하는 것을 확인하였다.

3) Oracle → SQL Server 교차 검증 결과

Oracle과 SQL Server 교차 검증 결과 데이터가 일치하지 않은 것으로 나타났다. 이에 SELECT문을 활용해 확인해보니 원본 DB와 업로드 DB의 "book_weight" 컬럼의 데이터 타입이 서로 상이해, 실제 데이터 값의 자릿수에 차이가 있었다(<그림 8>, <그림 9> 참조). 자릿수는 다르지만 실제 데이터 값은 동일한 점을 확인하였다.

| book_weight |
|---|
| 1.280000000000000E002 |
| 2.280000000000000E002 |
| 3.280000000000000E002 |

〈그림 8〉 Oracle의 "BOOK" 테이블

| book_weight |
|---|
| 1.280000E002 |
| 2.280000E002 |
| 3.280000E002 |

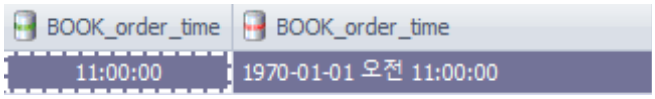
〈그림 9〉 SQL Server의 "BOOK" 테이블

4) MySQL → Oracle 교차 검증 결과

MySQL과 Oracle 교차 검증 결과 데이터가 일치하지 않은 것으로 나타났다. 이에 SELECT문을 활용해 데이터


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

를 확인해보니, MySQL의 'TIME(시-분-초 저장)' 데이터 타입이 SIARD 변환 후 Oracle 업로드 시 'TIMESTAMP (년-월-일 및 시-분-초 저장)' 데이터 타입으로 변환되는 것을 확인하였다. 이로 인해 <그림 10>과 같이, 데이터 무결성이 손상된 점을 확인하였다.



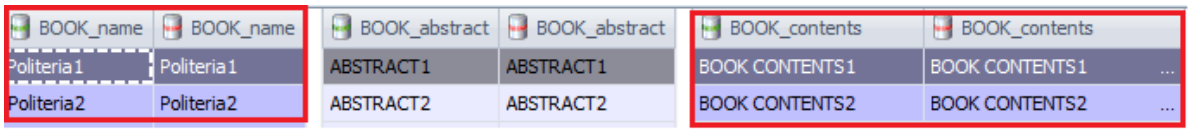

〈그림 10〉 MySQL과 Oracle의 "BOOK" 테이블

5) MySQL → MySQL 교차 검증 결과

MySQL과 MySQL 교차 검증 결과 모든 데이터가 일치하는 것으로 나타났다.

6) MySQL → SQL Server 교차 검증 결과

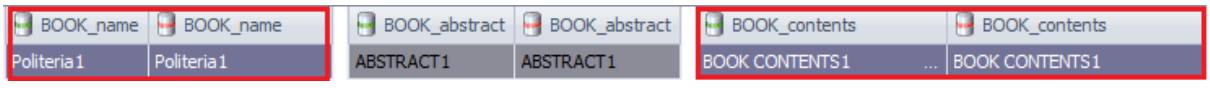
MySQL과 SQL Server 교차 검증 결과 데이터가 일치하지 않은 것으로 나타났으나 실제 데이터를 검토하였다. <그림 11>과 같이 "BOOK_name", "BOOK_contents" 컬럼의 데이터가 일치하지 않는 것으로 나타나지만 실제로는 <그림 11>에서처럼 동일한 데이터로 확인되었다("BOOK_abstract" 컬럼은 동일한 데이터).



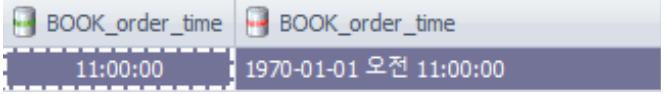

〈그림 11〉 MySQL과 SQL Server의 "BOOK" 테이블

7) SQL Server → Oracle 교차 검증 결과

Toad Data Point 상으론 SQL Server와 MySQL 교차 검증 결과 데이터가 일치하지 않은 것으로 나타나 실제 데이터를 확인하였다. <그림 12>와 같이, "BOOK_name", "BOOK_contents" 컬럼의 데이터가 일치하지 않는 것으로 나타나지만 실제로는 동일한 것으로 확인되었다. 또한 원본 DB가 SQL Server에서 SIARD 변환과 Oracle로 업로드 과정을 거치면서 기존에 SQL Server의 "TIME" 데이터 타입(시-분-초 저장)이 Oracle의 "TIMESTAMP" 데이터 타입(년-월-일 및 시-분-초 저장)으로 변경되면서 "TIME" 컬럼에 해당하는 데이터가 변경되는 것을 확인하였다 (<그림 13> 참조).




〈그림 12〉 SQL Server와 Oracle "BOOK" 테이블 1

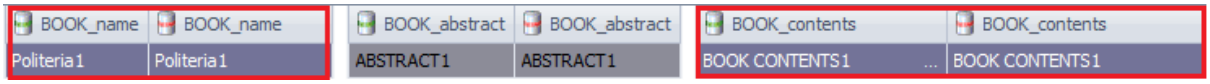



〈그림 13〉 SQL Server와 Oracle "BOOK" 테이블 2


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

8) SQL Server → MySQL 교차 검증 결과

Toad Data Point 상으론 SQL Server와 MySQL 데이터가 일치하지 않은 것으로 나타났다. 이에 실제 데이터를 확인해보니, <그림 14>와 같이 “BOOK_name”, “BOOK_contents” 컬럼의 데이터가 일치하지 않는 것으로 나타나지만 실제로는 동일한 데이터임이 확인되었다.




〈그림 14〉 SQL Server와 MySQL “BOOK” 테이블

9) SQL Server → SQL Server 교차 검증 시험 결과

SQL Server와 SQL Server 교차 검증 결과 모든 데이터가 일치하는 것으로 나타났다.

## 4. SIARD Suite 기능 보완

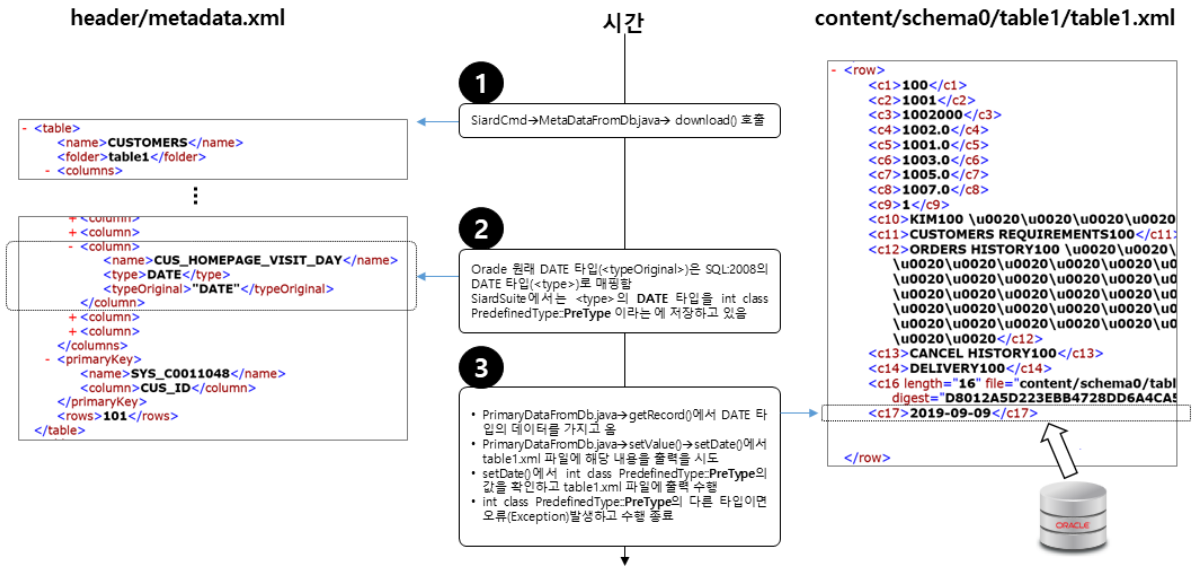
본 장에서는 선행 장에서 수행한 SIARD 검증 시험을 통해 도출된 SIARD Suite의 기능을 보완하는 개발을 진행하고자 한다. 검증 시험 결과, 몇몇 데이터 타입과 함수 계열 요소는 SIARD로 변환되지 않으며, 데이터 중 일부가 누락되는 것을 확인하였다. 데이터가 누락되는 경우는 DBMS와 SIARD 간 대응되는 데이터 타입이 저장하는 데이터 범위가 일치하지 않기 때문에 발생하는 문제로, 이를 해결할 수 있는 방안을 제안하고자 한다. 반면 함수 계열 요소의 경우, DBMS 종류에 따라 Function, Stored Procedures, Trigger를 정의하는 언어 및 구조가 상이하다. 그러므로 개별 DBMS 구조에 따른 맞춤형 개발이 필요하며, 이를 위해선 DBMS에 대한 높은 수준의 지식이 요구되므로 함수 계열 요소 변환 기능을 개발하는데 많은 어려움이 따른다. 따라서 SIARD 2.1 표준은 함수 계열 요소의 변환 기능을 명시하고 있으나, 실제 SIARD Suite에서는 해당 기능이 구현되지 않은 것으로 판단된다. 이에 본 연구에서는 SIARD 개발 범위는 데이터 타입에 한정지어 개발 방안을 제안하고자 한다.

SIARD 변환 시 데이터 누락 문제가 발생하는 데이터 타입인 Oracle의 “DATE”를 지원하는 개발 방안을 제안하고자 한다. SIARD Suite은 JDBC를 활용해 DBMS와 연결해 DB 변환 및 복원을 지원한다. 특히 SIARD Suite에서는 각각의 상용 DBMS가 지원하는 데이터 타입과 매핑되는 SQL:2008의 데이터 타입을 사전에 정의해 놓았다. 정의된 데이터 타입을 변환하기 위한 개별 JDBC API 함수도 정의되어 있다. 이 중 Oracle의 ‘DATE(년-월-일 및 시-분-초 저장)’은 SQL:2008의 “DATE”와 대응되어 있으므로 JDBC API 중 java.sql.getDate(Date getDate(int columnIndex) throws SQLException)를 활용해 Oracle로부터 해당 데이터를 가지고 오도록 구현되어 있다. 하지만 SQL:2008의 ‘DATE(java.sql.Date)’는 년-월-일만 저장할 수 있으므로 데이터의 시-분-초는 누락된 상태로 저장된다(<그림 15> 참조).

이러한 문제를 해결하고자 <표 10>과 같이 3개의 해결방안을 제시한다.

첫 번째 방안은 SIARD Suite 내 ‘SiardCmd’의 소스코드를 수정하는 방안이다. 앞서 언급한 바와 같이, Oracle의 ‘DATE’는 SIARD 변환 과정에서 SQL:2008의 ‘DATE’과 대응되어 데이터 누락이 발생한다. 따라서 SQL:2008의 데이터 타입 중 년-월-일 및 시-분-초를 저장하는 데이터 타입인 ‘TIMESTAMP’를 Oracle의 ‘DATE’와 대응하도록 소스코드를 수정한다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099




〈그림 15〉 SIARD Suite 소스코드 내 Oracle의 'DATE' 타입 처리 과정

〈표 10〉 SIARD의 Oracle 'DATE' 타입 개발 방안

| 연번 | 구분 | 구현 방안 | |
|---|---|---|---|
| 방안1 | SIARD Suite의 SiardCmd 소스코드 수정 | 수정코드 | SiardCmd의 PrimaryDataFromDb.java 수정 |
| | | 수정 전 | ...<br>case Types.DATE:<br>oValue = rs.getDate(iPosition);<br>break;<br>... |
| | | 수정 후 | ...<br>case Types.DATE:<br>if (__dbms.equals("Oracle")) {<br>mc.setPreType(Types.TIMESTAMP, 0, mc.getScale());<br>oValue = rs.getTimestamp(iPosition);<br>}<br>else {<br>oValue = rs.getDate(iPosition);<br>}<br>break;<br>... |
| 방안2 | Oracle JDBC 소스코드 수정 | • SIARD Suite의 JdbcOracle 프로젝트는 Oracle JDBC 드라이버의 Wrapper 함수로 JDBC API를 SIARD Suite의 다른 프로젝트에서 일관된 방식으로 호출할 수 있도록 설계되어 있음<br>• Oracle JDBC API중 java.sql.ResultSet.getDate(...)함수가 년/월/일과 시/분/초 모두 가져올 수 있도록 하기 위해서는 Oracle JDBC 드라이버 소스코드 자체를 수정해야 함<br>• (주의사항) JDBC API의 동작과정을 수정하는 것은 JDBC라는 모든 DBMS에 대해 일반화(Normalization)된 구조에 위배되는 것으로 추후 Oracle JDBC에 대해 호환성에 문제가 발생할 수 있음 | |
| 방안3 | SIARD Suite의 SqlParser 프로젝트 소스코드 수정 | • SIARD Suite에서는 각 DBMS에서 제공하고 있는 타입들에 대해 SQL:2008의 타입들도 대응시키는 규칙을 가지고 있음<br>• 여기에서 DATE 타입이 TIMESTAMP 타입과 호환이 되도록 코드를 수정 방안임<br>• (주의사항) SqlParser 프로젝트는 SIARD Suite 중에서 가장 복잡하면서 정교하게 작업된 프로젝트로 소스코드 분석에 많은 시간과 노력이 예상됨 | |

두 번째 방안은 Oracle JDBC 소스코드를 수정하는 방안이다. SIARD Suite의 JdbcOracle 프로젝트는 Oracle JDBC 드라이버의 Wrapper 함수로 JDBC API를 SIARD Suite의 다른 프로젝트에서 일관된 방식으로 호출할 수 있도록


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

설계되어 있다. Oracle JDBC API 중 java.sql.ResultSet.getDate(...)함수가 년-월-일과 시-분-초 모두 가져오기 위해서는 Oracle JDBC 드라이버 소스코드 자체를 수정해야 한다. 하지만 JDBC API의 동작과정을 수정하는 것은 JDBC라는 모든 DBMS에 대해 일반화(Normalization)된 구조에 위배되는 것으로, 추후 Oracle JDBC에 대해 호환성에 문제가 발생할 수 있다는 제약사항이 있는 방안이다.

마지막 방안은 SIARD Suite내의 SqlParser 프로젝트 소스코드를 수정하는 방안이다. 앞서 언급한 바와 같이, SIARD Suite은 각 DBMS에서 제공하고 있는 데이터 타입을 SQL:2008의 데이터 타입으로 대응시키는 규칙을 가지고 있다. 여기에서 DATE 타입이 TIMESTAMP 타입과 호환이 가능하도록 소스코드를 수정하는 방안이다. 이 방안은 SIARD Suite 중 가장 복잡하고 정교하게 제작된 소스코드인 SqlParser 프로젝트를 수정해야 하므로 선제적으로 해당 소스코드를 완벽하게 분석해야 한다.

본 연구팀에서는 첫 번째 방안으로 DATE 타입의 시-분-초 데이터가 누락되는 문제를 해결하였다. 이 3개의 해결 방안은 SIARD Suite으로 변환 또는 복원에 문제가 발생하는 데이터 타입들에 대해서도 모두 적용이 가능하다.

## 5. 결 론

본 연구는 행정정보 데이터세트 보존포맷으로서 SIARD Suite의 기능을 검증하고자 Oracle, MySQL, SQL Server 총 3종의 DBMS를 대상으로 두 차례 검증 시험을 수행했다. 먼저 SIARD 기초 검증 시험은 SIARD가 데이터세트를 구성하는 기능, 구조, 데이터 등을 안정적으로 보존할 수 있는지 검증하고자 했다. 이를 위해 개별 DBMS에서 제공하는 모든 데이터 타입, 함수 계열 요소(Function, Stored Procedures, Trigger), 데이터 관계를 의미하는 PK, FK를 SIARD 포맷이 안정적으로 변환 및 보존하는지 여부를 확인하였다. 시험 결과, 데이터 타입 중 Oracle DBMS의 'UROWID' 타입, MySQL DBMS의 'JSON'을 제외한 모든 데이터 타입은 SIARD로 변환 및 보존이 가능하며, 다시 DBMS로 복원도 가능함을 확인하였다. 반면 PK는 모든 DBMS에서, FK는 Oracle을 제외한 MySQL과 SQL Server에서 생성한 경우 SIARD 포맷으로 변환이 가능했다. 함수 계열 요소는 DBMS 종류에 관계없이 SIARD로 변환이 불가했다. 상기 검증 시험을 통해 SIARD 포맷으로 보존이 가능한 데이터세트의 구성요소를 도출했다.

기초 검증 시험 결과를 토대로 교차 검증 시험을 수행했다. 기초 검증 시험 결과로 도출된 SIARD로 변환 가능한 데이터 타입을 선정해 공통 DB를 3종 DBMS에 생성했다. 생산된 공통 DB는 SIARD 파일로 변환 및 3종 DBMS로 복원됐으며, 최초의 공통 DB와 DBMS로 업로드된 DB의 데이터를 비교했다. Toad Data Point 5.1.0.142를 활용해 두 DB 간 데이터를 비교했으며, 데이터 무결성을 유지하는 경우와 무결성이 손상된 경우를 도출했다. 대부분의 경우에서 데이터 무결성이 보장됐지만, "MySQL → Oracle", "SQL Server → Oracle" 경우에는 특정 데이터 타입의 데이터가 누락되는 것을 확인하였다(<표 9> 참조).

두 차례 검증 시험을 통해 SIARD가 특정 데이터 타입과 함수 계열 요소 보존에 한계점을 가지고 있음을 확인하였다. 이에 SIARD Suite이 제공하는 기능을 보완하는 추가 개발을 수행했다. 데이터 누락이 발생하는 데이터 타입의 문제를 해소하는 3가지 개발 방안을 제시하였으며(<표 10> 참조), 개발 방안은 모든 유형의 데이터 타입에 적용가능한 방법론이다. 반면 함수 계열 요소의 경우는 DBMS에 따라 함수 객체를 구성하는 요소와 구조가 상이하기 때문에 모든 DBMS에 맞춰 개발하는 것은 비효율적이다. 따라서 본 연구진은 국내 환경에 맞춰 국내 대표 오픈소스 DBMS인 큐브리드(CUBRID)와 티베로(Tibero)의 구조에 부합하는 개발 방향을 제안한다. 큐브리드의 경우 G-클라우드 공공표준 DBMS이며, 티베로는 국산 DBMS 중 가장 높은 점유율[7]을 보이고 있어(대한민국.

7) 2019년 12월 기준, 티베로는 점유율 2.71%로 국내 DBMS 중 가장 높은 점유율을 보이고 있다.


http://dx.doi.org/10.14404/JKSARM.2021.21.3.099

행정안전부, 2020) 향후 공공영역에서 해당 DBMS의 활용도는 점차 늘어날 것으로 예측된다. 따라서 큐브리드와 티베로를 기반으로, SIARD Suite의 기능 범위를 확장하는 개발 방향을 고려해 볼 수 있다.

본 연구는 SIARD가 제공하지 않는 기능인 함수 계열 요소 보존을 위한 기능 개발을 진행하지 못했다는 한계점이 있지만, SIARD Suite 기능을 확인하는 실증적 검증 시험을 수행했다. 이를 통해 SIARD 2.1 표준과 SIARD Suite이 제공하는 기능 간 차이점을 확인했다는 점에서 의의가 있다. 또한 SIARD Suite의 기능을 보완하는 개발 방안과 방향성을 제안했다는 점에서 SIARD 개발에 대한 추가 연구를 기대할 수 있다. 특히, SIARD가 지원하지 못하는 비관계형 데이터베이스 활용도가 점차 커지고 있으므로, 해당 유형의 데이터베이스를 위한 보존포맷의 연구가 후속 과제로 진행되기를 희망한다.

## 참고문헌


국가기록원 (2019a). 데이터세트 유형 전자기록의 장기보존기술 연구.

국가기록원 (2019b). 전자기록 유형별 포맷 정책(안).

국가기록원 (2019c). 전자기록물 장기보존 정책(안).

김주연 (2020). SIARD를 활용한 행정정보데이터세트 장기 보존 방안 연구. 석사학위논문, 명지대학교 기록정보과학전문대학원 기록관리전공.

노종원, 소정의 (2020). 데이터세트의 장기적인 보존 및 활용을 위한 관리 방안에 관한 연구. 디지털문화아카이브지, 3(1), 51-64.

대한민국. 행정안전부 (2020). 2020년 범정부EA기반 공공부문 정보자원 현황 통계보고서.

소정의, 한희정, 양동민 (2018). 국외 전자기록물의 장기보존 정책 비교 분석: 미국, 캐나다, 영국, 호주, 스위스를 중심으로. 한국기록관리학회지, 18(4), 125-148. https://doi.org/10.14404/JKSARM.2018.18.4.125

오세라, 박승훈, 임진희 (2018). 행정정보 데이터세트 사례 조사 연구. 한국기록관리학회지, 18(2), 109-133. https://doi.org/10.14404/JKSARM.2018.18.2.109

오세라, 이해영 (2019). 행정정보 데이터세트의 기록관리 방안. 한국기록관리학회지, 19(2), 51-76. https://doi.org/10.14404/JKSARM.2019.19.2.051

왕호성, 설문원 (2017). 행정정보 데이터세트 기록의 관리방안. 한국기록관리학회지, 17(3), 23-47. https://doi.org/10.14404/JKSARM.2017.17.3.023

이규철 (2016). 행정정보시스템 데이터세트의 이해와 기록관리 고려사항. 기록관리 표준 거버넌스 포럼 자료집, 72-78.

한국도서관협회 (2010). 문헌정보학용어사전. 출처: http://www.kla.kr/jsp/fileboard/termdic.do

한국정보통신기술협회 (2018). IT용어사전. 출처: http://terms.tta.or.kr/main.do

한희정, 윤성호, 오효정, 양동민 (2020). 데이터세트 보존포맷 검증방안에 관한 연구: 재난안전정보 데이터세트의 SIARD 적용을 통해. 한국정보관리학회지, 37(2), 251-284. http://dx.doi.org/10.3743/KOSIM.2020.37.2.251

행정정보 데이터세트 기록관리기준-관리기준표의 작성 및 이관규격. NAK 35:2020(v1.0).

Digital Preservation Guidance Note 1 - Selecting File Formats for Long-Term Preservation. DPGN-01.

Swiss Federal Archives (2021.07.01.). SIARD Suite. Swiss Federal Archives. Available: https://www.bar.admin.ch/bar/en/home/archiving/tools/siard-suite.html


• **국문 참고자료의 영어 표기**
(English translation / romanization of references originally written in Korean)


Han, Hui-Jeong, Yoon, Sung-Ho, Oh, Hyo-Jung, & Yang, Dong-Min (2020). Empirical Verification of Conversion and Restoration of Preservation Format for Dataset: Application of Dataset with Disaster Safety Information to SIARD. Journal of Korean

Society for Information Management, 37(2), 251-287. http://dx.doi.org/10.3743/KOSIM.2020.37.2.251

Kim, Joo-Yeon (2020). A Study on the Long-term Preservation of Administrative Information Datasets Using SIARD. Master's thesis, Graduate School of Records, Archives & Information Science, Myongji University.

Korea. Ministry of the Interior and Safety (2020). 2020 National Government EA based Public Sector Information Resources Statistical Report.

Korean Library Association (2010). Dictionary of Libraries and Information Sciences. Available: http://www.kla.kr/jsp/fileboard/termdic.do

Lee, Kyu-Chul (2016). Understanding for administration information system dataset and considerations for recordkeeping. Records Management Standard Forum Resources, 72-78.

National Archives of Korea (2019a). Study on long-term preservation technology of dataset-type electronic records.

National Archives of Korea (2019b). Policy of Format by Electronic Records Type.

National Archives of Korea (2019c). Policy of Long-Term Preservation of Electronic Records.

Oh, Seh-Ra & Rieh, Hae-young (2019). Managing Data Set in Administrative Information Systems as Records. Journal of Korean Society of Archives and Records Management, 19(2), 51-76. https://doi.org/10.14404/JKSARM.2019.19.2.051

Oh, Seh-Ra, Park, Seung-Hoon, & Yim, Jin-Hee (2018). A Case Study of Dataset Records in Information Management System. Journal of Korean Society of Archives and Records Management, 18(2), 109-133. https://doi.org/10.14404/JKSARM.2018.18.2.109

Record Keeping Criteria for Dataset: Composition of Dataset Management Reference Table & Exchange of Dataset. NAK 35:2020(v1.0).

Roh, Jong-Won & So, Jeong-Eui (2020). A Study on the Management Plan for Preservation and Long-Term Use of Datasets. Journal of D-Culture Archives, 3(1), 51-64.

So, Jeong-Eui, Han, Hui-Jeong, & Yang, Dong-Min (2018). A Comparative Analysis of Long-Term Preservation Policies in Foreign Electronic Records: NARA, LAC, TNA, NAA, and SFA. Journal of Korean Society of Archives and Records Management, 18(4), 125-148. https://doi.org/10.14404/JKSARM.2018.18.4.125

Telecommunications Technology Association (2018). Dictionary of Information Technology. Available: http://terms.tta.or.kr/main.do

Wang, Ho-Sung & Seol, Moon-won (2017). A Study on Managing Dataset Records in Government Information Systems. Journal of Korean Society of Archives and Records Management, 17(3), 23-47. https://doi.org/10.14404/JKSARM.2017.17.3.023