

개인정보보호 관련 패키지 교육

2016년 9월 21일

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form or by any means - electronic, mechanical, photocopying, recording, or otherwise - without permission. This document provides an outline of a presentation and is incomplete without the accompanying oral commentary and discussion.

Contents

- 관련법령
- 개인정보 암호화
- 개인정보 접속기록
- 개인정보 분리보관

개인정보 보호법

행정자치부(개인정보보호정책과)02-2100-4105

제2조(정의) 이 법에서 사용하는 용어의 뜻은 다음과 같다.<개정 2014.3.24>

1. “개인정보”란 살아 있는 개인에 관한 정보로서 성명, 주민등록번호 및 영상 등을 통하여 개인을 알아볼 수 있는 정보(해당 정보만으로는 특정 개인을 알아볼 수 없더라도 다른 정보와 쉽게 결합하여 알아볼 수 있는 것을 포함한다)를 말한다.

.....

5. “개인정보처리자란” 업무를 목적으로 개인정보파일을 운영하기 위하여 스스로 또는 다른 사람을 통하여 개인정보를 처리하는 공공기관, 법인, 단체 및 개인 등을 말한다.

제15조(개인정보의 수집·이용)

① 개인정보처리자는 다음 각 호의 어느 하나에 해당하는 경우에는 개인정보를 수집할 수 있으며 그 수집 목적의 범위에서 이용할 수 있다.

1. 정보주체의 동의를 받는 경우

.....

② 개인정보처리자는 제1항제1호에 따른 동의를 받을 때에는 다음 각 호의 사항을 정보주체에게 알려야 한다. 다음 각 호의 어느 하나의 사항을 변경하는 경우에도 이를 알리고 동의를 받아야 한다.

1. 개인정보의 수집·이용 목적

2. 수집하려는 개인정보의 항목

3. 개인정보의 보유 및 이용 기간

4. 동의를 거부할 권리가 있다는 사실 및 동의 거부에 따른 불이익이 있는 경우에는 그 불이익의 내용

개인정보 보호법

행정자치부(개인정보보호정책과)02-2100-4105

제21조(개인정보의 파기)

- ① 개인정보처리자는 보유기간의 경과, 개인정보의 처리 목적 달성 등 그 개인정보가 불필요하게 되었을 때에는 지체 없이 그 개인정보를 파기하여야 한다. 다만, 다른 법령에 따라 보존하여야 하는 경우에는 그러하지 아니하다.
- ② 개인정보처리자가 제1항에 따라 개인정보를 파기할 때에는 복구 또는 재생되지 아니하도록 조치하여야 한다.
- ③ 개인정보처리자가 제1항 단서에 따라 개인정보를 파기하지 아니하고 보존하여야 하는 경우에는 해당 개인정보 또는 개인정보 파일을 다른 개인정보와 분리하여서 저장·관리하여야 한다.(개인정보 분리보관)
- ④ 개인정보의 파기방법 및 절차 등에 필요한 사항은 대통령령으로 정한다.

제24조(고유식별정보의 처리제한)

- ① 개인정보처리자는 다음 각 호의 경우를 제외하고는 법령에 따라 개인을 고유하게 구별하기 위하여 부여된 식별정보로서 대통령령으로 정하는 정보(이하 "고유식별정보"라 한다)를 처리할 수 없다.
 1. 정보주체에게 제15조제2항 각 호 또는 제17조제2항 각 호의 사항을 알리고 다른 개인정보의 처리에 대한 동의와 별도로 동의를 받는 경우
 2. 법령에서 구체적으로 고유식별정보의 처리를 요구하거나 허용하는 경우
- ② 삭제 <2013.8.6>
- ③ 개인정보처리자가 제1항 각 호에 따라 고유식별정보를 처리하는 경우에는 그 고유식별정보가 분실·도난·유출·위조·변조 또는 훼손되지 아니하도록 대통령령으로 정하는 바에 따라 암호화 등 안전성 확보에 필요한 조치를 하여야 한다.<개정 2015.7.24.>(개인정보 암호화)

.....

개인정보 보호법 시행령

행정자치부(개인정보보호정책과)02-2100-4105

제16조(개인정보의 파기방법)

① 개인정보처리자는 법 제21조에 따라 개인정보를 파기할 때에는 다음 각 호의 구분에 따른 방법으로 하여야 한다.<개정 2014.8.6.>

1. 전자적 파일 형태인 경우: 복원이 불가능한 방법으로 영구 삭제
2. 제1호 외의 기록물, 인쇄물, 서면, 그 밖의 기록매체인 경우: 파쇄 또는 소각

② 제1항에 따른 개인정보의 안전한 파기에 관한 세부 사항은 행정자치부장관이 정하여 고시한다.<신설 2014.8.6.,2014.11.19.>

제19조(고유식별정보의 범위) 법 제24조제1항 각 호 외의 부분에서 “대통령령으로 정하는 정보”란 다음 각 호의 어느 하나에 해당하는 정보(이하 “고유식별정보”라 한다)를 말한다. 다만, 공공기관이 법 제18조제2항제5호부터 제9호까지의 규정에 따라 다음 각 호의 어느 하나에 해당하는 정보를 처리하는 경우의 해당 정보는 제외한다. **(개인정보 암호화)**

1. 「주민등록법」 제7조제3항에 따른 **주민등록번호**
2. 「여권법」 제7조제1항제1호에 따른 **여권번호**
3. 「도로교통법」 제80조에 따른 운전면허의 **면허번호**
4. 「출입국관리법」 제31조제4항에 따른 **외국인등록번호**

정보통신망 이용촉진 및 정보보호 등에 관한 법률 시행령

제15조(개인정보의 보호조치)

.....

③ 법 제28조제1항제3호에 따라 정보통신서비스 제공자등은 접속기록의 위조·변조 방지를 위하여 다음 각 호의 조치를 하여야 한다.

1. 개인정보취급자가 개인정보처리시스템에 접속하여 개인정보를 처리한 경우 접속일시, 처리내역 등의 저장 및 이의 확인·감독
2. 개인정보처리시스템에 대한 접속기록을 별도 저장장치에 백업 보관(개인정보 접속기록)

④ 법 제28조제1항제4호에 따라 정보통신서비스 제공자등은 개인정보가 안전하게 저장·전송될 수 있도록 다음 각 호의 보안조치를 하여야 한다.
<개정 2014.11.28.>

1. 비밀번호의 일방향 암호화 저장 (개인정보 암호화)
2. 주민등록번호, 계좌정보 및 바이오정보(지문, 홍채, 음성, 필적 등 개인을 식별할 수 있는 신체적 또는 행동적 특징에 관한 정보를 말한다) 등 방송통신위원회가 정하여 고시하는 정보의 암호화 저장 (개인정보 암호화)

.....

제16조(개인정보의 파기 등)

② 정보통신서비스 제공자등은 이용자가 정보통신서비스를 법 제29조제2항의 기간 동안 이용하지 아니하는 경우에는 이용자의 개인정보를 해당 기간 경과 후 즉시 파기하거나 다른 이용자의 개인정보와 분리하여 별도로 저장·관리하여야 한다.<개정 2016.5.31> (개인정보 분리보관)

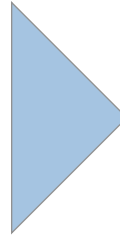
.....

개인정보 암호화



BEFORE

- 암호화, 복호화는 쿼리 또는 Java에 직접구현
- 암호화 컬럼이 추가 되면 해당 컬럼을 사용하는 프로그램을 모두 수정해야 함
- 암호화를 사용하지 않거나 다른 암호화를 사용하는 경우 프로그램을 모두 수정해야 함



AFTER

- 암호화 되는 컬럼을 지정하면 자동으로 암복호화
- 추가된 컬럼을 기준에 추가 등록
- huneCryptoUtil에서 암호화 메소드 재정의

암복호화
코딩에서
해방!

- 설정에 따라 예전방식 그대로 사용 가능 (system.properties의 PSN_INFO_AUTO_ENC_COLUMN_YN=N 으로 설정)
- 일부의 프로그램(로그인시 패스워드 비교, 인사기본사항의 탭 권한 때문에 사용한 암호화)은 자동 암복호화에서 제외
- 연말정산은 기존사이트와의 호환을 위해 자동암복호화를 적용하지 않음

암호화 적용 시점

VarStatement.setParameter

CUDSQLManager.addKey
.addField
.addEtcParameter

- Varstatement , CUDSQLManager 를 사용하지 않는 경우
기존과 동일하게 코딩

(ex. 프로시저, 평션... 등 ..)

복호화 적용 시점

xsheetResultXML.jsp
ibsheetResultJSON.jsp
export.jsp

Sys_common.**getResultTxtRow**
.getResultTxt

- 위에 나열된 소스를 거치지 않고 조회하는 경우
기존과 동일하게 코딩

(ex. 프로시저, 평션... 등 ..)

암호화 컬럼 관리

화이트 리스트

- 암호화 할 컬럼명을 지정한다.(정확히 말하자면 ALIAS임)
- PER_NO, ACC_NO, 주민등록번호.....등등

예외 지정

- 화이트 리스트에 포함되었으나 암,복호화 하면 안되는 컬럼인 경우
 - CLASS, METHOD, COLUMN 으로 예외지정 가능
- ex) biz.pas.pa_bas.Pa_bas_120_ul01, save01, PER_NO

암복호화예제

기존

- `vstmt.setParameter("PER_NO", huneiCrypto.getEncrypt(form.getValue("S_PER_NO")));`
- **PKG_CRYPTOF.GET_DEC_DATA**(T1.PER_NO) AS PER_NO

변경

- `vstmt.setParameter("PER_NO", form.getValue("S_PER_NO"));`
- `T1.PER_NO`

암복호화를 코어에서 처리함

추가된 내용 및 설정

1. system.properties

- PSN_INFO_AUTO_ENC_COLUMN_YN=Y 추가
- PSN_INFO_AUTO_ENC_COLUMN_YN 변수가 없거나 값이 N이면 기존 소스에서 암호화 하는 방식으로 그대로 사용가능.

2. 암호화 컬럼관리(ServerStartUtil)

- src/common/util 경로에 ServerStartUtil.java 생성.
- was기동시에 코어에서 ServerStartUtil의 init() 메소드를 호출하기 때문에 해당경로에 파일이 존재 해야 함.
- ServerStartUtil의 setEncColMap() 메소드에 암호화 할 컬럼들을 등록

```
/**
 * 암호화할 컬럼들을 static map에 담는다.
 */
private void setEncColMap() {
    log.debug("===== setEncColMap() was start!!!");
    HashMap<String,String> encColMap = new HashMap();
    encColMap.put("PSPT_NO", "PSPT_NO");
    encColMap.put("TAX_DEFER_ACC_NO", "TAX_DEFER_ACC_NO");
    encColMap.put("PER_NO", "PER_NO");
    encColMap.put("주민등록번호", "주민등록번호");
    encColMap.put("ACC_NO", "ACC_NO");
    encColMap.put("GRNT_PER_NO", "GRNT_PER_NO");
    encColMap.put("DEC_PER_NO", "DEC_PER_NO");
    encColMap.put("M_PER_NO", "M_PER_NO");
    encColMap.put("RST_PER_NO", "RST_PER_NO");
    encColMap.put("M_주민번호", "M_주민번호");
    encColMap.put("RET_PNS_ACC_NO", "RET_PNS_ACC_NO");

    hunelServlet.setStaticMap("ENC_COL_MAP", encColMap);
    for(String key : encColMap.keySet()) {
        log.debug("ENC_COLUMN : "+key);
    }
    log.debug("===== setEncColMap() was succeeded...");
}
```

3. 암복호화 예외컬럼 관리 프로그램 추가

- 암복호화 하지 않을 컬럼을 메소드의 컬럼별로 등록해서 관리함
- WebRoot/sy_enc/sy_enc_010_ul01.jsp 파일추가
- src/biz/sys/sy_enc/Sy_enc_010_ul01.java, Sy_enc_010_ul01.xml 파일추가
- SY9040(암복호화 예외컬럼 관리) 테이블 생성

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions	St
Column Name	ID	PK	Index Pos	Null?	Comments	Data Type				
DSCLASS	1	1	1	N	DSCLASS	VARCHAR2 (100 Byte)				
DSMETHOD	2	2	2	N	DSMETHOD	VARCHAR2 (100 Byte)				
ENC_EXEC_COL	3	3	3	N	암호화에 외컬럼	VARCHAR2 (100 Byte)				
NOTE	4			Y	비고	VARCHAR2 (500 Byte)				
INS_USER_ID	5			Y	입력자ID	VARCHAR2 (20 Byte)				
INS_YMDHMS	6			Y	입력일시	TIMESTAMP(6)				
MOD_USER_ID	7			Y	작업자ID	VARCHAR2 (20 Byte)				
MOD_YMDHMS	8			Y	수정일시	TIMESTAMP(6)				
INS_GYMDHMS	9			Y	입력일시_ 현지	TIMESTAMP(6) WITH TIME ZONE				
MOD_GYMDHMS	10			Y	수정일시_ 현지	TIMESTAMP(6) WITH TIME ZONE				
<										
<input type="checkbox"/> Editable Table Comments										
(SY9040)암복호화 예외컬럼 관리										

개인정보 암호화

- 시스템 모듈에 메뉴 추가

> 시스템관리

- 환경변수/평가유형

- 배치작업관리

- 배치로그조회

- 테이블명세

- 프로시저명세

- 메일발송

- SMS발송

- 도움말관리

- 사변삭제

- 웹서비스관리

> 개인정보보호

- **암복호화 예외컬럼 관리**

- 개인정보분리보관테이블관리

> Main

- 공지사항등록

암복호화 예외컬럼 관리

* DSCLASS * DSMETHOD * COLUMN Q 조회

> 암복호화 컬럼 리스트 DOWN

> 암복호화 예외컬럼 관리 + 추가 복사 저장 DOWN

No	*COLUMN
1	M_주민번호
2	DEC_PER_NO
3	GRNT_PER_NO
4	ACC_NO
5	PER_NO
6	주민등록번호
7	RET_PNS_ACC_NO
8	TAX_DEFER_ACC_NO
9	RST_PER_NO
10	PSPT_NO
11	M_PER_NO

No	*DSCLASS	*DSMETHOD	*COLUMN	비고	<input type="checkbox"/> 삭제	상태
1	biz.pas.pa_bas.Pa_bas_120_ul01	save01	PER_NO		<input type="checkbox"/>	

암복호화를 하지 않을 서비스를 등록해준다.

ex) biz.pas.pa_bas.Pa_bas_120_ul01 클래스의 save01 메소드에서는 PER_NO를
암복호화 하지 않겠다.

HCG 14
Hanson Consulting Group

4. hunelCryptoUtil.java

- hunelCrypto를 상속받는 hunelCryptoUtil을 src/common/util 경로에 생성
- 프로젝트에서 hunel의 암호화를 사용하지 않을 경우 암호화로직을 건너내거나 재 정의하여 사용할 수 있게 함.

```
package common.util;
import java.sql.Connection;
import java.sql.SQLException;
import javax.servlet.http.HttpServletRequest;
import hcg.hunel.core.action.hunelCrypto;
public class hunelCryptoUtil extends hunelCrypto{
    /**
     * hunel 암호화 외에 다른 암호화를 사용한다면
     * 해당 메소드들의 주석을 삭제 후 재정의하여 사용.
     */
    //public static String getSessionKey(HttpServletRequest request){ return null; }
    //public static void delDecData(Connection conn, HttpServletRequest request) throws SQLException{}
    /**
     * 암호화된 문자열을 받아 복호화 처리한 평문을 반환한다.(주민등록번호 등 복호화)
     * @param codedID 암호화된 문자열
     * @return 복호화한 평문
     */
    //public static String getDecrypt(String codedID) { return codedID; }
    /**
     * 세션ID를 key로 사용하여 암호화된 문자열을 받아 복호화 처리한 평문을 반환한다.
     * @param codedID 암호화된 문자열
     * @return 복호화한 평문
     */
    //public static String getDecrypt(String codedID, HttpServletRequest request) { return codedID; }
    /**
     * 문자열을 받아 암호화된 문자열을 반환한다.(주민등록번호 등 암호화)
     * @param message 암호화할 문자열
     * @return 암호화 처리된 문자열
     * @throws Exception
     */
    //public static String getEncrypt(String message){ return message;}
    /**
     * 세션ID를 key로 사용하여 문자열을 받아 암호화된 문자열을 반환한다.
     * @param message 암호화할 문자열
     * @return 암호화 처리된 문자열
     * @throws Exception
     */
    //public static String getEncrypt(String message, HttpServletRequest request){ return message; }
    /**
     * 문자열을 받아 암호화된 문자열을 반환한다.(SHA-2) (비밀번호암호화)
     * @param message 암호화할 문자열
     * @return 암호화 처리된 문자열
     * @throws Exception
     */
}
```

hunelCrypto를 상속받은
hunelCryptoUtil을 생성함.
재 정의하려면 주석을 풀어 메소드들의
내용을 코딩해주면 된다.

소스 수정

1. 소스에서 암호화 걷어냄

- java에서 hunelCrypto.getEncrypt, hunelCrypto.getDecrypt를 직접 사용한부분 제거
- xml에서 PKG_CRYPTOF_GET_DEC_DATA 를 사용하여 복호화하는 부분 제거

```

/**
 * 생성자
 * @param conn Connection
 * @param request HttpServletRequest
 */
public Pa_bas_216_pl01(Connection conn, HttpServletRequest request, HttpServletResponse response)
{
    super(conn, request, response);
}

/**
 * 이전화사 발령내역을 조회한다.
 * @param form hunelBaseForm
 * @throws SQLException
 */
public void search01(hunelBaseForm form) throws SQLException
{
    String S_SMR_PRT_YN = form.getValue("S_SMR_PRT_YN");
    StringBuffer sb = new StringBuffer();
    if (S_SMR_PRT_YN.trim().equals("sum")) sb.append("\n AND T2.SMR_PRT_YN = 'Y'");
    String block = sb.toString();

    String query = xmlQuery.getElement(this, "search01",
        vstmt.setParameter("PER_NO", form.getValue("S_PER_NO"));
        vstmt.setParameter("ENTER_YMD", form.getValue("S_ENTER_YMD"));
        vstmt.setParameter("LANG_TYPE", ehrbean.get("LANG_TYPE"));
        vstmt.setParameter("PER_NO", hunelCryptoUtil.getEncrypt(form.getValue("S_PER_NO")));
        vstmt.setParameter("ENTER_YMD", form.getValue("S_ENTER_YMD"));
        vstmt.setParameter("LANG_TYPE", ehrbean.get("LANG_TYPE"));
        // vstmt.setParameter("YMD", form.getValue("S_YMD"));

    log.debug(vstmt.getQueryString());
    setResultSet(SQLUtil.getResultSetWithClose(vstmt));
}

```

hunelCryptoUtil.getEncrypt 제거

```

Pa_bas_200_m01.xml 838
Pa_bas_200_m01.xml
VarStatement.java

1 <?xml version='1.0' encoding='utf-8'?>
2 <SQLResource version='1'>
3
4 <query name='search01'>![CDATA[
5 SELECT T1.C_CD,
6 T1.EMP_ID,
7 T1.EMP_NM,
8 PKG_CRYPTOF_GET_DEC_DATA(T1.PER_NO) PER_NO,

```

PKG_CRYPTOF_GET_DEC_DATA 제거

3. header_nologin.jsp 수정

- hunelCryptoUtil을 사용하는 jsp가 있기 때문에 hunelCryptoUtil을 import 시킴.

```
1 <%@ page import="java.util.*,  
2 hcq.hunel.core.util.*,  
3 hcq.hunel.core.action.*,  
4 common.util.hunelCryptoUtil,  
5 biz.user.*,  
6 org.apache.log4j.Logger,  
7 common.util.XSSFilterList  
8 ">  
9 %>  
10 <%@taglib uri="/WEB-INF/common.tld" prefix="com"%>  
11 <%>  
12 Logger log = Logger.getRootLogger();  
13 CommonProperties commProp = CommonProperties.getInstance();  
14 request.setCharacterEncoding("utf-8");  
15 response.setCharacterEncoding("utf-8");  
16 String httpVersion = request.getProtocol();  
17 if (httpVersion.equals("HTTP/1.1"))  
18 {
```

header_nologin.jsp에
common.util.hunelCryptoUtil을 import 시킴

4. 데이터를 파싱해주는 jsp 수정

- xsheetResultXML.jsp, ibsheetResultJSON.jsp, export.jsp

```

14 String _DSCCLASS = StringUtil.nvl(request.getParameter("S_DSCCLASS"), "");
15 String _DSMETHOD = StringUtil.nvl(request.getParameter("S_DSMETHOD"), "");
16 String S_PSN_INFO_AUTO_ENC_COLUMN_YN = StringUtil.nvl(commProp.getProperty("PSN_INFO_AUTO_ENC_COLUMN_YN"), "N");
17 try
18 {
19 > //log.debug("xsheetResultXML.jsp start");
20 if ("Y".equals(request.getParameter("_DOIT")))
21 {
22 // 초기화
23 request.setAttribute("O_ETC", null);
24 request.setAttribute("O_RS", null);
25 request.setAttribute("O_RSMAP", null);
26 setResult(request, response);
27 }
28
29 log.debug("xsheetResultXML.jsp === "+request.getAttribute("O_ETC"));
30 etcMap = (HashMap)request.getAttribute("O_ETC");
31
32 msg = StringUtil.nvl((String)request.getAttribute("O_MSG"));
33 urs = (MemResultSet)request.getAttribute("O_RS");
34 ursmap = (HashMap)request.getAttribute("O_RSMAP");
35 if (ursmap == null) ursmap = new HashMap();
36 if (urs != null) ursmap.put("default", urs);
37 }
38 catch (Exception e)
39 {
40 msg = e.toString();
41 }
42 if (etcMap == null) etcMap = new HashMap();
43 <?xml version="1.0" encoding="utf-8" ?>
44 <SHEET>
45 <ETC-DATA>
46 <
47 Iterator etcIter = etcMap.keySet().iterator();
48 String etcKey, etcVal;
49 while ( etcIter.hasNext() )
50 {
51 etcKey = (String)etcIter.next();
52 etcVal = StringUtil.nvl((String)etcMap.get(etcKey));
53 if ("Y".equals(S_PSN_INFO_AUTO_ENC_COLUMN_YN))
54 {
55 ServerStartUtil serverStartUtil = new ServerStartUtil();
56 boolean existEncColumn = serverStartUtil.getExistEncColMap(StringUtil.nvl(etcKey, ""));
57 boolean existExceEncColumn = serverStartUtil.getExistExceEncColMap(_DSCCLASS+"."+_DSMETHOD+"."+StringUtil.nvl(etcKey, ""));
58 if (existEncColumn && !existExceEncColumn && !" ".equals(etcVal))
59 {
60 etcVal = hunelCryptoUtil.getDecrypt(etcVal);
61 }
62 }
63 }

```

PSN_INFO_AUTO_ENC_COLUMN_YN이 Y이고
호출한 서비스의 컬럼이 암복호화 컬럼이면
복호화를 해서 data 파싱

5. RD의 데이터를 만들어주는 부분 수정

- sys_common.java의 getResultTxt, getResultTxtRow 메소드를 오버로딩하여 생성

```
public static String getResultTxt(MemResultSet mrs, String splitStr, String subYn, String subParam, HttpServletRequest request) throws SQLException {
    > String[] arrSubParam = subParam.split("\\|"); //기타 파라미터 구분자로 넘김
    > StringBuffer sb = new StringBuffer();
    > int COL_CNT = mrs.getColumnCount(); //컬럼 갯수
    > CommonProperties commProp = CommonProperties.getInstance();
    > String S_DSCLASS = StringUtil.nvl(request.getParameter("S_DSCLASS"), "");
    > String S_DSMETHOD = StringUtil.nvl(request.getParameter("S_DSMETHOD"), "");
    > String S_PSN_INFO_AUTO_ENC_COLUMN_YN = StringUtil.nvl(commProp.getProperty("PSN_INFO_AUTO_ENC_COLUMN_YN"), "N");

    while(mrs.next()) {
        > //쿼리 결과 순서대로 append
        > for(int i=1; i <= COL_CNT; i++) {
            > if(!"WRAP_ORDER".equals(mrs.getColumn(i))) {
                > String value = StringUtil.nvl(mrs.getString(i));
                > if("Y".equals(S_PSN_INFO_AUTO_ENC_COLUMN_YN)) {
                    > ServerStartUtil serverStartUtil = new ServerStartUtil();
                    > boolean existEncColumn = serverStartUtil.getExistEncColMap(mrs.getColumn(i));
                    > boolean existExceEncColumn = serverStartUtil.getExistExceEncColMap(S_DSCLASS+"."+S_DSMETHOD+"."+mrs.getColumn(i));
                    > if(existEncColumn && !existExceEncColumn && !"Y".equals(value)) {
                        > sb.append(hunelCryptoUtil.getDecrypt(value));
                    }
                    > else {
                        > sb.append(value);
                    }
                }
                > else {
                    > sb.append(value);
                }
            }
            > sb.append(splitStr);
        }
    }
}
```

request를 추가로 받음

PSN_INFO_AUTO_ENC_COLUMN_YN이 Y이고
호출한 서비스의 컬럼이 암호화 컬럼이면
복호화를 해서 data 를 만들어줌.

- getResultTxt, getResultTxtRow 메소드를 사용하는 소스 수정(메소드 호출시 request를 추가로 넘겨줌)

```
//주쿼리의 경우 Row별로 append
sb.append(biz.sys.Sys_common.getResultTxtRow(mrs, "|", "N", PHOTO_PATH+"|"+IP_ADDRESS+"|"+LOGIN_EMP_NM, row, request));

//서브쿼리2 start
String query2 = xmlQuery.getElement(this, "getSub2", null);
VarStatement vstmt2 = new VarStatement(conn, query2, request);

vstmt2.setParameter("C_CD", C_CD);
vstmt2.setParameter("EMP_ID", M_EMP_ID);
vstmt2.setParameter("LANG_TYPE", LANG_TYPE);

log.debug(vstmt2.getQueryString());
MemResultSet mrs2 = SQLUtil.getResultSetWithClose(vstmt2);

sb.append(biz.sys.Sys_common.getResultTxt(mrs2, "|", "Y", "", request));
```

코어 수정 내용

1. hunelServlet.java

- HashMap<String, HashMap> 변수 STATIC_MAP추가
- ServerStartUtil.init() 메소드를 init시에 호출함

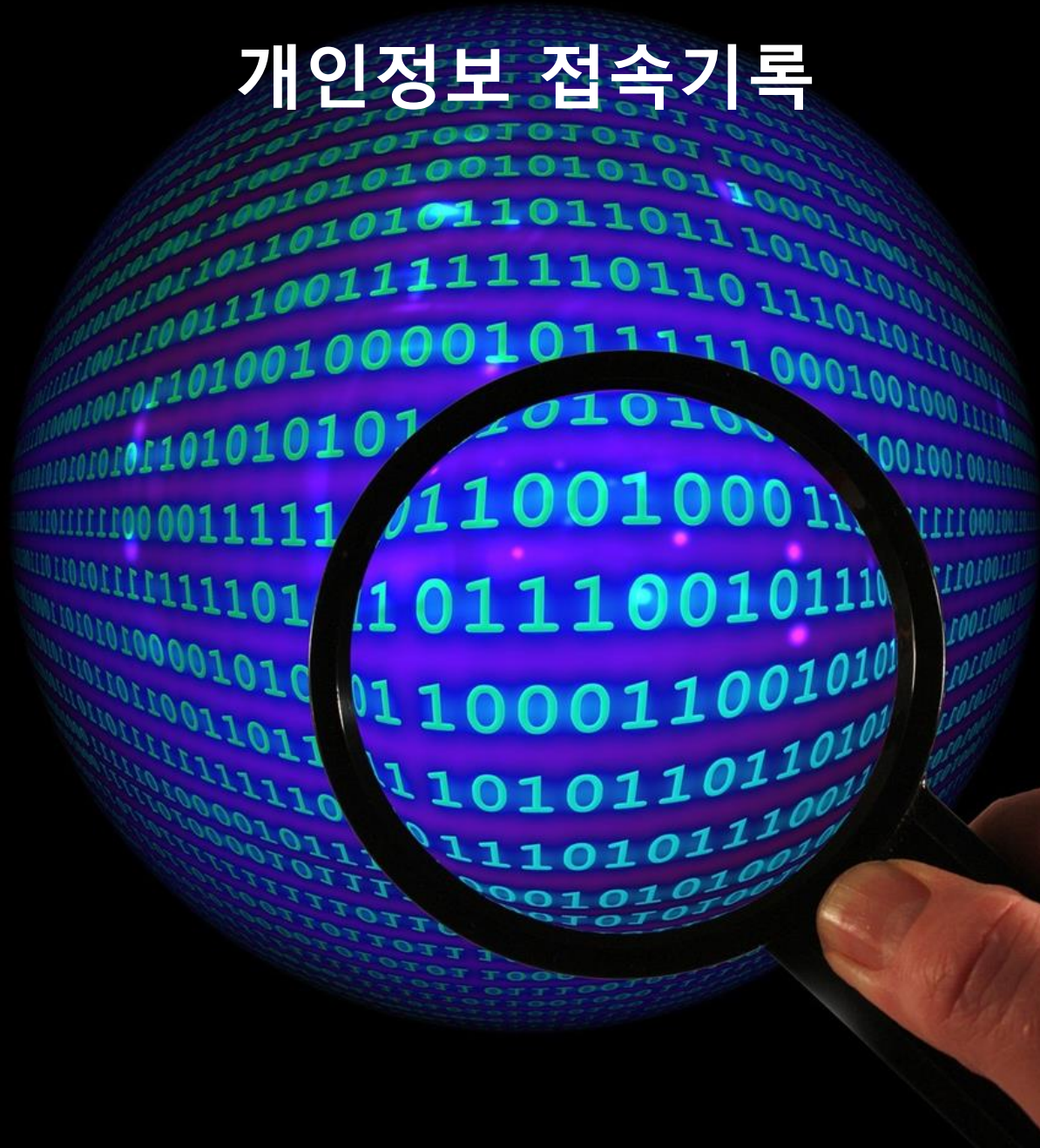
2. VarStatement.java

- setParameter 메소드에서 컬럼체크하여 암호화하는 로직 추가

3. CUDSQLManager.java

- addEtcParameter, addKey, addField 메소드에서 컬럼체크하여 암호화하는 로직 추가

개인정보 접속기록





setting을 이용한 개인정보 접속 기록 적용 방법

조회, CUD

테이블명

기준컬럼

개인정보컬럼



SY9041

해당정보를 SY9041에 등록

프로시저

뒤에 설명

coding을 이용한
개인정보 접속 기록
적용 방법

조회

VarStatement.setPsnIdenStdCol

CUD

CUDSQLManager.setPsnIdenStdCol

프로시저

뒤에 설명

로그 기록 시점

조회

SQLUtil.**getResultSetWithClose**
.getResultSet

CUD

CUDSQLManager.**insert**
.update
.delete

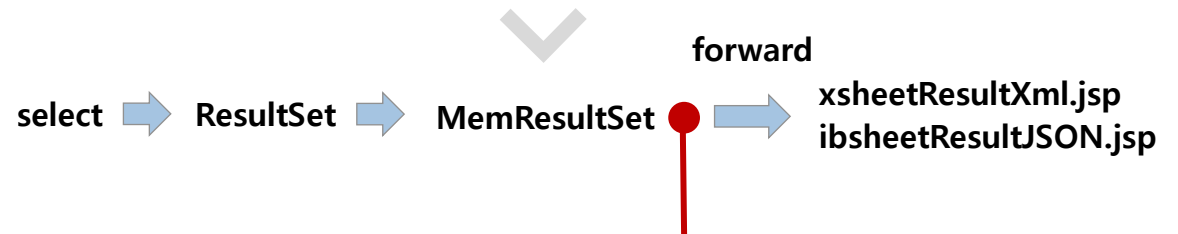
프로시저

VarStatement.**executeUpdate**

조회 LOG 기록 방법 (setting)

조회

SQLUtil.**getResultSetWithClose(vstmt)**



(SY9041) 개인정보 분리보관 접근로그 테이블 관리

테이블명(PK)	개인식별기준컬럼	개인정보컬럼
PA1010	EMP_ID	PER_NO,EMP_NM,.....
PA2030	PER_NO	PER_NO,FAM_NM,.....

- 조회쿼리에서 SY9041에 등록된 테이블명 존재
- MemResultSet에서 개인정보컬럼이 존재
- MemResultSet에서 개인식별기준컬럼이 존재

Y

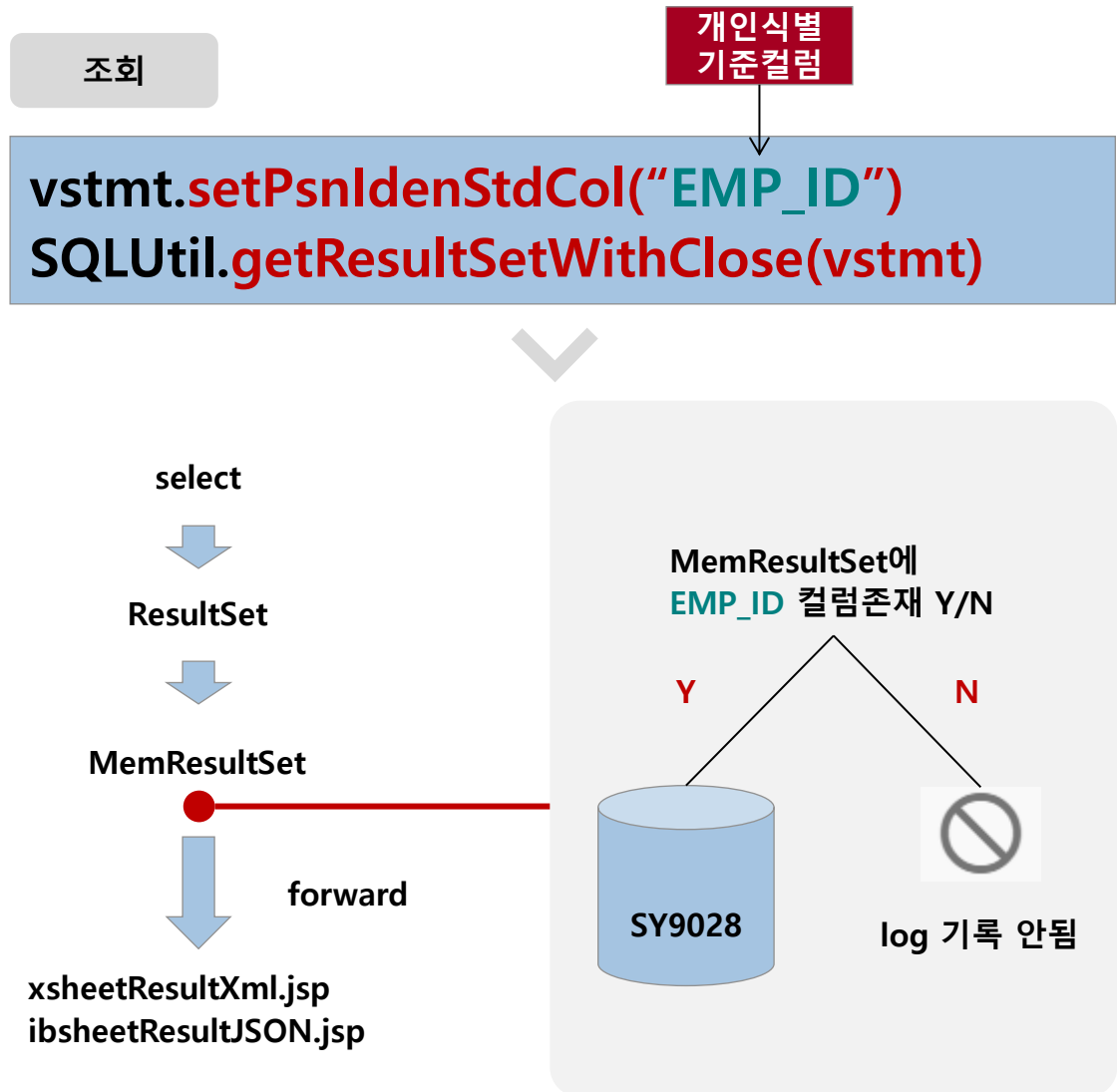


N



log 기록 안됨

조회 LOG 기록 방법 (coding)



CUD LOG 기록 방법 (setting)

CUD

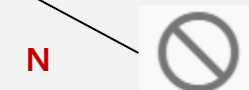
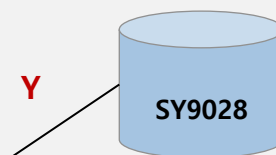
`cud.insert(), update(), delete()`

makeQuery → Varstatement.executeUpdate() → 결과 반환

(SY9041) 개인정보 분리보관 접근로그 테이블 관리

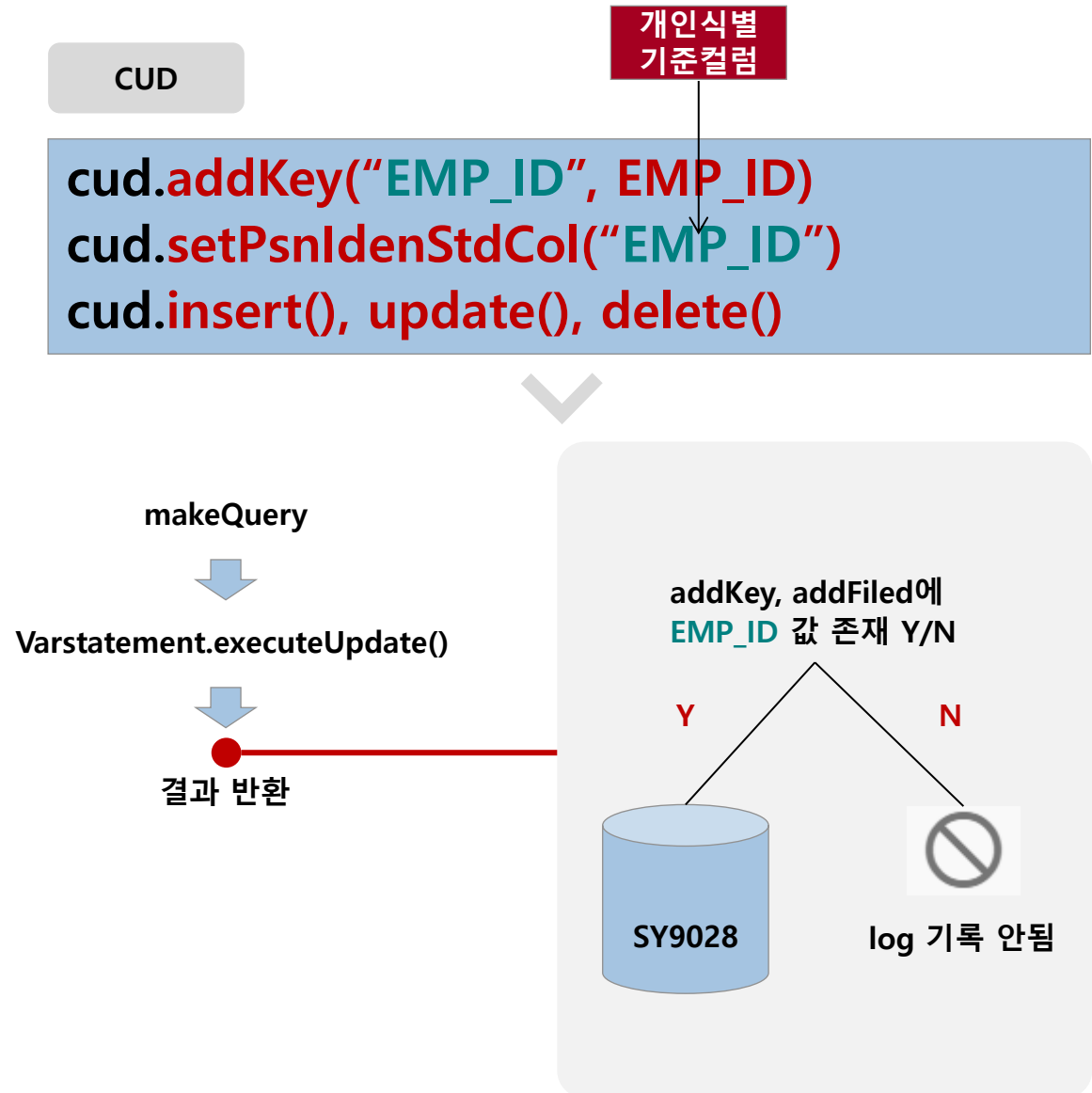
테이블명(PK)	개인식별기준컬럼	개인정보컬럼
PA1010	EMP_ID	PER_NO,EMP_NM,.....
PA2030	PER_NO	PER_NO,FAM_NM,.....

- `cud.setTable()`의 테이블이 SY9041에 등록된 테이블명
- `addKey, addField` 에서 개인정보컬럼이 존재
- `addKey, addField` 에서 개인식별기준컬럼이 존재

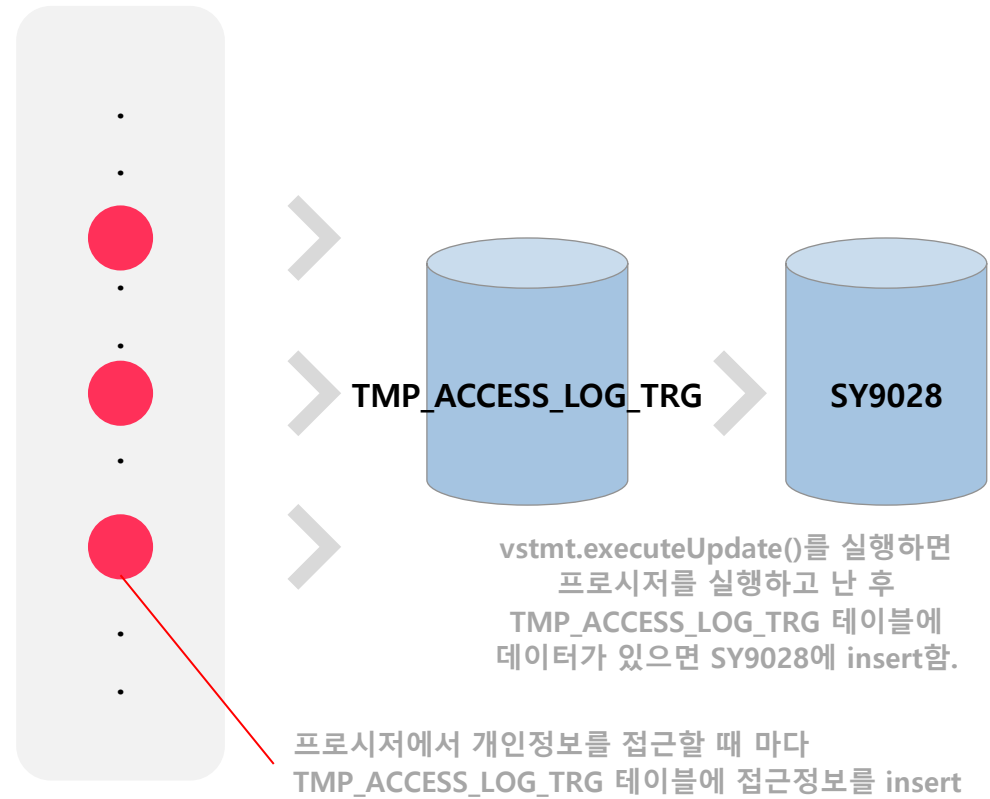


log 기록 안됨

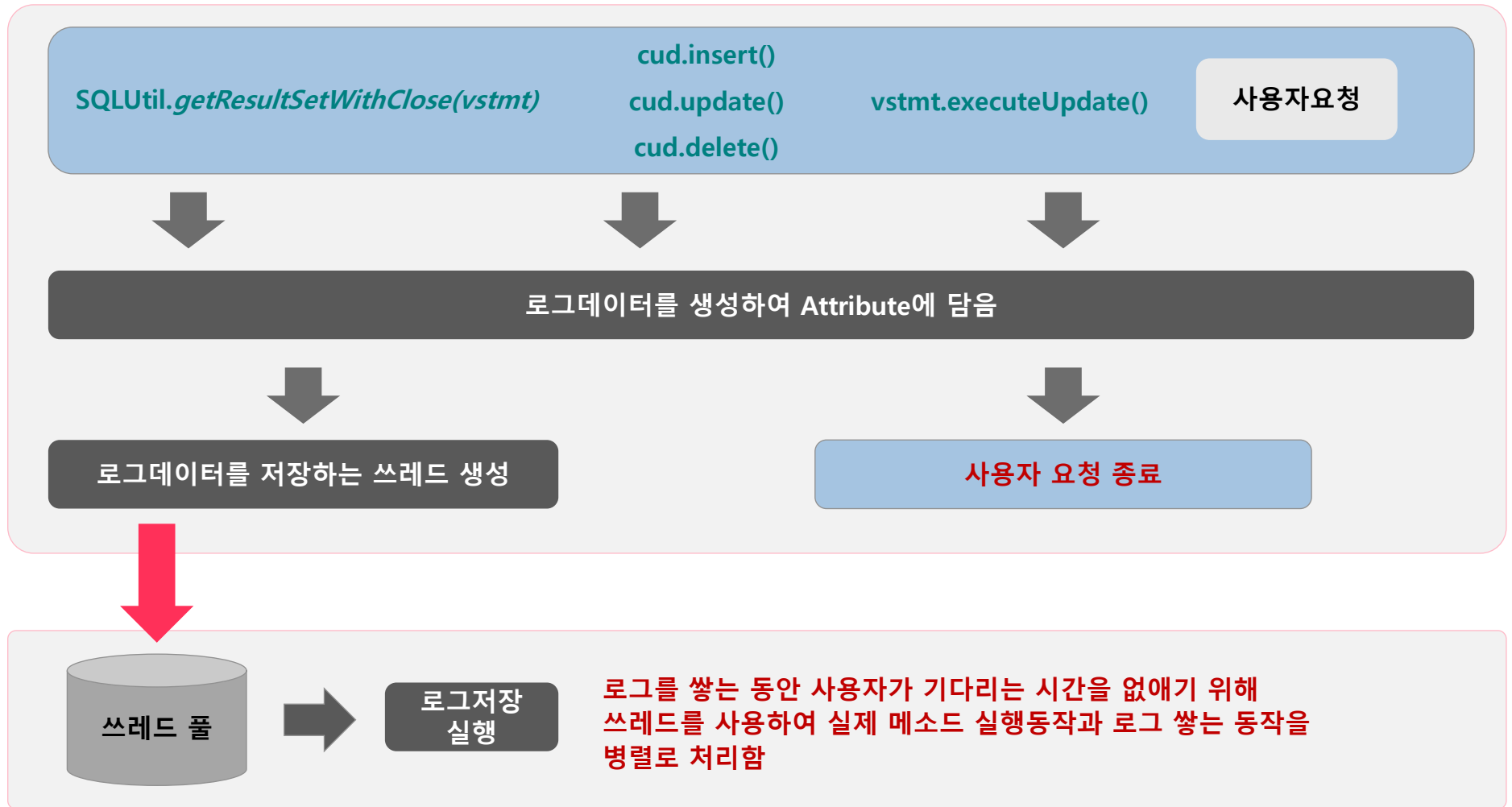
CUD LOG 기록 방법 (setting)



Procedure Log 기록 방법



Log 기록 프로세스



추가된 내용 및 설정

1. 테이블 생성












1.1 SY9028

- 로그를 저장할 테이블을 생성. (SY9028 로그_개인정보접근로그)

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions	S
<div><div></div><div> </div></div>										
Column Name	ID	PK	Index Pos	Null?	Comments	Data Type				
C_CD	1			N	회사코드	VARCHAR2 (20 Byte)				
USER_ID	2			N	사용자ID	VARCHAR2 (20 Byte)				
TABLE_NM	3			Y	테이블명칭	VARCHAR2 (100 Byte)				
CRUD	4			N	CRUD	VARCHAR2 (1 Byte)				
PSN_IDEN_STD_COL	5			N	개인식별 기준컬럼	VARCHAR2 (30 Byte)				
ACCESS_TRG	6		1	Y	접근대상	CLOB				
ACCESS_QUERY	7		1	Y	접근쿼리	CLOB				
PGM_ID	8			N	프로그램 ID	VARCHAR2 (150 Byte)				
DSCLASS	9			N	DSCLASS	VARCHAR2 (100 Byte)				
DSMETHOD	10			N	DSMETHO D	VARCHAR2 (100 Byte)				
IP	11			N	IP	VARCHAR2 (150 Byte)				
INS_YMDHMS	12			N	입력일시	TIMESTAMP(6)				
INS_GYMDHMS	13			N	입력일시_ 현지	TIMESTAMP(6) WITH TIME ZONE				
<div><input type="checkbox"/> Editable Table Comments</div>										
(SY9028)로그_개인정보접근로그										













1.2 SY9041

- SY9041개인정보 분리보관 접근로그 테이블 관리 테이블 생성
- 데이터 생성(SY9041.xls 참고)

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions	Stats/Size
										
Column Name	ID	PK	Index Pos	Null?	Comments	Data Type	Default			
TABLE_NM	1	1	1	N	테이블명	VARCHAR2 (100 Byte)				
TABLE_TXT	2			Y	테이블내용	VARCHAR2 (500 Byte)				
SEPA_CLASS	3			Y	분리구분	VARCHAR2 (20 Byte)				
PSN_IDEN_STD_COL	4			Y	개인식별기준컬럼	VARCHAR2 (30 Byte)				
PSN_INFO_COL	5			Y	개인정보컬럼	VARCHAR2 (1000 Byte)				
LOG_USE_YN	6			N	로그사용여부	VARCHAR2 (1 Byte)	'Y'			
APPL_TABLE_YN	7			N	신청서테이블여부	VARCHAR2 (1 Byte)	'N'			
NOTE	8			Y	비고	VARCHAR2 (500 Byte)				
INS_USER_ID	9			Y	입력자ID	VARCHAR2 (20 Byte)				
INS_YMDHMS	10			Y	입력일시	TIMESTAMP(6)				
MOD_USER_ID	11			Y	작업자ID	VARCHAR2 (20 Byte)				
MOD_YMDHMS	12			Y	수정일시	TIMESTAMP(6)				
INS_GYMDHMS	13			Y	입력일시_현지	TIMESTAMP(6) WITH TIME ZONE				
MOD_GYMDHMS	14			Y	수정일시_현지	TIMESTAMP(6) WITH TIME ZONE				
<										
<input type="checkbox"/> Editable Table Comments										
(SY9041)개인정보 분리보관 접근로그 테이블 관리										

1.3 TMP_ACCESS_LOG_TRG

- 프로시저안에서 데이터를 접근하였을경우 로그를 쌓기 위해 생성. (접근로그저장임시테이블)

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions
									
	Column Name	ID	PK	Index Pos	Null?	Comments	Data Type		
	TABLE_NM	1			Y	테이블명 칭	VARCHAR2 (100 Byte)		
	CRUD	2			Y	CRUD	VARCHAR2 (1 Byte)		
	PSN_IDEN_STD_COL	3			Y	개인식별 기준컬럼	VARCHAR2 (30 Byte)		
	ACCESS_TRG	4			Y	접근대상	VARCHAR2 (100 Byte)		
	ACCESS_TXT	5			Y	접근내용	VARCHAR2 (4000 Byte)		
<input type="checkbox"/> Editable Table Comments									
접근로그저장임시테이블									

2. system.properties

- PSN_INFO_ACCESS_LOG_YN=Y 추가 (접근로그 사용여부)
- PSN_INFO_ACCESS_LOG_DEBUG_YN=Y 추가 (접근로그를 insert 하는 query의 debug 여부)
- PSN_INFO_ACCESS_LOG_THREADPOOL_CNT=20 추가(개인정보 접근로그 스레드 풀 생성갯수)

3. 프로그램 추가

3.1 개인정보분리,로그테이블관리 프로그램 추가

- WebRoot/sys/sy_bas 경로에 sy_bas_250_ul01.jsp 파일추가
- src/biz/sys/sy_bas 경로에 Sy_bas_250_ul01.java, Sy_bas_250_ul01.xml 파일추가
- 시스템 모듈에 메뉴 추가

시스템관리

환경변수/평가유형

배치작업관리

배치로그조회

테이블명세

프로시저명세

메일발송

SMS발송

도움말관리

사변삭제

웹서비스관리

개인정보보호

암복호화 예외컬럼 관리

개인정보분리,로그테이블관리

개인정보접근로그 조회

개인정보분리보관

개인정보분리,로그테이블관리

테이블명

Q 조회

개인정보분리,로그테이블관리

추가

저장

선택테이블생성

DOWN

No	<input type="checkbox"/> 선택	*테이블명	테이블내용	분리보관			접속로그			비고	<input type="checkbox"/> 삭제	상태
				분리구분	신청서테이블 여부	생성 여부	개인식별 기술표준	개인정보컬럼	로그사용 여부			
9	<input type="checkbox"/>	PA1010	인사기본_기본사항	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM,BIRTH_YMD,FAM_MASTI	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
10	<input type="checkbox"/>	PA1050	발령_사번관리	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
11	<input type="checkbox"/>	PA1055	발령_전입자사번관리	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
12	<input type="checkbox"/>	PA2010	인사기본_주소	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	TEL_NO,ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
13	<input type="checkbox"/>	PA2030	인사기본_가족	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PER_NO	PER_NO,FAM_NM,BIRTH_YMD,CONTACT_T	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
14	<input type="checkbox"/>	PA2040	인사기본_자격	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	CERT_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
15	<input type="checkbox"/>	PA2070	인사기본_병역	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	MILSVC_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
16	<input type="checkbox"/>	PA2100	인사기본_보훈	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PATRIOT_NO,PATRIOT_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
17	<input type="checkbox"/>	PA2110	인사기본_신원보증	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	GRNT_PER_NO,GRNT_NO,ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
18	<input type="checkbox"/>	PA2130	인사기본_장애	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	DISABLE_REG_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
19	<input type="checkbox"/>	PA2180	인사기본_여권	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PSPT_NO,ENG_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
20	<input type="checkbox"/>	PA2190	인사기본_비자	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	VISA_NO,ENG_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
21	<input type="checkbox"/>	PA2240	인사기본_차량	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	CAR_NO,CAR_OWN_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
22	<input type="checkbox"/>	PA7020	재증명_발급대장	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
23	<input type="checkbox"/>	PA8030	인사기본_자인정보	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	MOBILE_NO,ACQ_NM,OFFICE_TEL_ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
24	<input type="checkbox"/>	PA8050	인사기본_해외주재원관리	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	LOCAL_PSN_TEL_NO,LOCAL_MOBILE_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
25	<input type="checkbox"/>	PY2060	급여_개인별계좌	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
26	<input type="checkbox"/>	PY2370	급여_은행이체파일	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
27	<input type="checkbox"/>	PY7010	급여_계좌변경신청	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
28	<input type="checkbox"/>	TE1200	교육_강사	정규직외기타	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	LECT_ID		<input checked="" type="checkbox"/>	LECT_NM	<input type="checkbox"/>	
29	<input type="checkbox"/>	TM5300	근태_근태신청	퇴직직후	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	EGC_TEL_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
30	<input type="checkbox"/>	TX0030	세무_일용소득자관리	정규직외기타	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	PER_NO		<input type="checkbox"/>	INCMN_NM	<input type="checkbox"/>	

[30 / 30]

3.2 개인정보접근로그 조회 프로그램 추가

- WebRoot/sy_log/sy_log_110_sl01.jsp, sy_log_110_p01.jsp 파일추가
- src/biz/sys/sy_log/Sy_log_110_sl01.java, Sy_log_110_sl01.xml, Sy_log_110_p01.java 파일추가
- 시스템 모듈에 메뉴 추가



개인정보접근로그 조회

회사

유만권설팅그룹

기간

2016.08.26 ~ 2016.08.26

기간선택

프로그램

사용자 ID/명

접근대상

CRUD

전체

조회

개인정보접근로그 조회

접근자

회사	사용자ID	사번	성명	조직	대상컬럼	접근대상	프로그램ID	프로그램명	테이블명	CRUD	DSCLASS	DSMETHOD	IP	일자
유만권설팅그룹	SUPER	02096	백대리	테스트조직	PER_NO	pa_bas_280_ul01				R	biz.pas.pa_bas.Pa_bas_280_ul01	search01	0-0-0-0-0-0-1	2016/08/26
유만권설팅그룹	SUPER	02096	백대리	테스트조직	PER_NO	pa_bas_280_ul01		VR_PA280		D	biz.pas.pa_bas.Pa_bas_280_ul01	save01	0-0-0-0-0-0-1	2016/08/26
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												
유만권설팅그룹	SUPER	02096												

접근대상

접근대상

대상건수 : 6

개인식별기준컬럼 : PER_NO

4701291023456

4902252023456

7410251555946

7803071023456

9701011892832

9901012839283

사용된 쿼리

SELECT T1.* FROM (

SELECT T1.*, ROWNUM AS WRAP_ORDER FROM (

SELECT T1.C_CD,

T1.EMP_ID,

TA.EMP_NM,

F_GET_GLOBAL_OBJNM(TB.C_CD, 'O',

TB.ORG_ID, '20160920', 'ko') ORG_NM,

T1.FAM_NM,

T1.PER_NO,

T1.FAM_CD,

T1.BIRTH_YMD,

T1.SCHO_CD,

T1.SCHL_NM,

T1.JOB_NM,

T1.COM_NM,

T1.TOGETHER_YN,

T1.FAM_ALW_YN,

T1.FAM_ALW_STA_YMD,

T1.FAM_ALW_END_YMD

4. Util 추가

- src/common/util 경로에 **InsertLog.java**, **InsertLogVO.java**, **Sy9041VO.java** 추가

4.1 InsertLog.java

- Runnable 인터페이스를 구현한 객체
- InsertLogVO 객체를 받아 SY9028(개인정보접근로그) 에 저장하는 동작을 함

4.2 InsertLogVO.java

- SY9028(개인정보접근로그)의 데이터를 담아놓는 VO

4.3 Sy9041VO.java

- SY9041(개인정보 분리보관 접근로그 테이블 관리) 의 데이터를 담아놓는 VO
- hunelServlet의 STATIC_MAP에 SY9041의 정보를 담아놓기 위해 생성

소스수정

1. common.js

- \$(document).ready(function () { 평선에 S_PAGE_PGM_ID 추가
- function makeParameters(dsClass, dsMethod, otherParams) 평선에 S_PAGE_PGM_ID 추가 (로그샷을때 PGM_ID가 필요)

```

55 $(document).ready(function () {
56   //chkAuthMenu 권한체크를 위해 추가함
57   if(window.Page != null)
58   {
59     form = ($("#f1").length > 0)? $("#f1") : $("form").first();
60     form.append('<input type="hidden" id="S_PAGE_PROFILE_ID" name="S_PAGE_PROFILE_ID" value="'+Page.PROFILE_ID+'">');
61     form.append('<input type="hidden" id="S_PAGE_MODULE_ID" name="S_PAGE_MODULE_ID" value="'+Page.MODULE_ID+'">');
62     form.append('<input type="hidden" id="S_PAGE_MENU_ID" name="S_PAGE_MENU_ID" value="'+Page.MENU_ID+'">');
63     form.append('<input type="hidden" id="S_PAGE_PGM_URL" name="S_PAGE_PGM_URL" value="'+Page.PGM_URL+'">');
64     form.append('<input type="hidden" id="S_PAGE_ENC_VAL" name="S_PAGE_ENC_VAL" value="'+Page.ENC_VAL+'">');
65     form.append('<input type="hidden" id="S_PAGE_PGM_ID" name="S_PAGE_PGM_ID" value="'+Page.PGM_ID+'">');
66     form.append('<input type="hidden" id="__viewState" name="__viewState" value="">');
67     if(typeof(Page) != "undefined" && ! Page.MENU_ID) displayElement($(".btMsg"), false);
68   }
69   3087 function makeParameters(dsClass, dsMethod, otherParams)
70   {
71     3088   otherParams = otherParams||{};
72     3089   //otherParams = $(otherParams.toString()).replace(null,"")||{};
73     3090   if ( typeof otherParams == 'string') otherParams = $(otherParams).toQueryParams();
74     3091   var forceParams = {S_DSCLASS: dsClass, S_DSMETHOD: dsMethod, S_FORWARD: otherParams.S_FORWARD || "xsheetResultXML"};
75     3092   if ( ehrTopFrame._LOGIN_INFO ) forceParams.X_LOGIN_INFO = ehrTopFrame._LOGIN_INFO;
76     3093   //메뉴별 권한체크를위해 추가함 2014-02-24 w.y.c
77     3094   if(window.Page != null)
78     {
79       3095     forceParams.S_PAGE_PROFILE_ID = Page.PROFILE_ID;
80       3096     forceParams.S_PAGE_MODULE_ID = Page.MODULE_ID;
81       3097     forceParams.S_PAGE_MENU_ID = Page.MENU_ID;
82       3098     forceParams.S_PAGE_PGM_URL = Page.PGM_URL;
83       3099     forceParams.S_PAGE_PGM_ID = Page.PGM_ID;
84       3100     forceParams.S_PAGE_ENC_VAL = Page.ENC_VAL;
85     }
86     3101   var parameters = $.extend(otherParams, forceParams);
87     3102   parameters = $.extend(parameters, {__viewState:toJsonString(parameters)});
88     3103   parameters = nullToUndefinedForJSON(parameters);
89     3104   return parameters;
90   }
91 }

```

2. export.jsp

- S_DSCLASS, S_DSMETHOD, S_PAGE_PGM_ID를 request.setAttribute로 넘겨줌
- excel download시 export.jsp를 호출하여 호출한 원래 화면의 S_DSCLASS, S_DSMETHOD를 가지고 데이터를 직접 조회 하여 excel파일을 만들게 되어있다. 조회시 로그가 쌓일려면 S_DSCLASS, S_DSMETHOD, S_PAGE_PGM_ID의 정보가 필요한데, export.jsp에서 **request**를 **MultipartRequest** 로 받아서 **request**를 조회 메소드에 넘겨주게 되어있다. **MultipartRequest** 생성에 쓰인 request는 request.getParameter(); 로 값을 가져올수 없어서 Attribute에 해당정보를 셋팅하여 로그를 쌓는부분에서 **request.getAttribute("S_DSCLASS");** 와 같이 불러올 수 있게 request.setAttribute로 넘겨줌.

```
try {
    > int maxPostSize = 10 * 1024 * 1024; // 10MB
    > conn = SQLUtil.getConnection(request, null);
    > multi = new MultipartRequest(request, application.getRealPath(""), maxPostSize, "utf-8");
    > hunelBaseForm baseform = new hunelBaseForm(request, response);
    > baseform.setMultipartRequest(multi);
    > String S_DSCLASS = baseform.getValue("S_DSCLASS");
    > String S_DSMETHOD = baseform.getValue("S_DSMETHOD");
    > String Data = baseform.getValue("Data");
    > String S_PAGE_PGM_ID = baseform.getValue("S_PAGE_PGM_ID");
    > request.setAttribute("DATA", Data);
    > request.setAttribute("S_DSCLASS", S_DSCLASS);
    > request.setAttribute("S_DSMETHOD", S_DSMETHOD);
    > request.setAttribute("S_PAGE_PGM_ID", S_PAGE_PGM_ID);

    Object objDs;
    Class dsClass;
    try {
        dsClass = Class.forName(S_DSCLASS);
        Class[] parameterTypes = new Class[] { Connection.class, HttpServletRequest.class, HttpServletResponse.class };
        Constructor constructor = dsClass.getConstructor(parameterTypes);
        objDs = constructor.newInstance(new Object[] { conn, request, response });
    } catch (Throwable e) {
```

개인정보 접속기록

3. 데이터 조회로그 관련 java소스 수정(선택사항)

- '(SY9041) 개인정보 분리보관 접근로그 테이블관리' 에서 관리하지 않지만 로그를 쌓고 싶을경우, 혹은 테이블관리로 해결되지 않는부분이 있다면

search메소드에서 **SQLUtil.getResultSetWithClose(vstmt)** 를 하기전에 **vstmt.setPsnIdenStdCol("EMP_ID");** 메소드를 호출한다.

- **vstmt.setPsnIdenStdCol("EMP_ID");** 에서 EMP_ID는 SY9041 테이블에서 관리되는 PSN_IDEN_STD_COL(개인식별기준컬럼)을 대체 한다.
ex) SY9041의 PSN_IDEN_STD_COL이 "PER_NO" 여도 "EMP_ID" 를 개인식별기준으로 본다.
- 호출되는 쿼리의 select절에 EMP_ID 컬럼(ALIAS)이 존재하여야 한다.
- SY9041에 등록된 정보가 있어서 로그가 쌓이는데, **로그를 쌓고싶지 않을경우 vstmt.setPsnInfoAccessLogUse(false);** 메소드를 호출한다.

```
public void search01(hunelBaseForm form) throws SQLException {
    String S_C_CD = ehrbean.getCCD();
    StringBuffer sb = new StringBuffer(100);
    sb.append("AND NVL(T1.WEEK_FINANC_YN, 'N') = 'N' \n");
    String block = sb.toString();
    String query = xmlQuery.getElement(this, "search01", block, null);

    VarStatement vstmt = new VarStatement(conn, query, request, null);
    vstmt.setParameter("C_CD", S_C_CD);
    vstmt.setParameter("STA_YMD", form.getValue("S_STA_YMD"));
    vstmt.setParameter("END_YMD", form.getValue("S_END_YMD"));
    log.debug(vstmt.getQueryString());

    vstmt.setPsnIdenStdCol("EMP_ID");

    setResultSet(SQLUtil.getResultSetWithClose(vstmt));
}
```

```
<query name='search01'><![CDATA[
SELECT T1.C_CD,
       T1.EMP_ID,
       EMP_NM,
       PER_NO,
       STD_YY,
       EMP_TYPE,
       RE_TYPE,
       T1.MOD_YMDHMS,
       EXPT_ENTER_YMD,
```

```
if (!"".equals(S_EMP_ID.trim())) vstmt.setParameter("EMP_ID", S_EMP_ID);
if (!"".equals(S_ORG_ID.trim())) vstmt.setParameter("ORG_ID", S_ORG_ID);

log.debug(vstmt.getQueryString());

vstmt.setPsnInfoAccessLogUse(false);
setResultSet(SQLUtil.getResultSetWithClose(vstmt));
```

4. 데이터 수정로그 관련 java소스 수정(선택사항)

- '(SY9041) 개인정보 분리보관 접근로그 테이블관리' 에서 관리하지 않지만 로그를 쌓고 싶을경우, 혹은 테이블관리로 해결되지 않는부분이 있다면

save메소드에서 **cud.insert(), cud.update(), cud.delete()** 를 하기전에 **cud.setPsnIdenStdCol("EMP_ID");** 메소드를 호출한다.

- **cud.setPsnIdenStdCol("EMP_ID");** EMP_ID는 SY9041 테이블에서 관리되는 PSN_IDEN_STD_COL(개인식별기준컬럼)을 대체 한다.
ex) SY9041의 PSN_IDEN_STD_COL이 "PER_NO" 여도 "EMP_ID" 를 개인식별기준으로 본다.
- CUDSQLManager에 key나 filed로 EMP_ID 컬럼이 정의되 있어야 한다.
- SY9041에 등록된 정보가 있어서 로그가 쌓이는데, **로그를 쌓고싶지 않을경우 cud.setPsnInfoAccessLogUse(false);** 메소드를 호출한다.

```
public void save01(hunelBaseForm form) throws SQLException {
    > HashMap<String, String> authMap = new HashMap<String, String>();
    authMap.put("EMP_ID", "EMP_ID");
    xmlQuery.chkAuthTrans(conn, request, ehrbean, form, authMap);
    CUDSQLManager cud = new CUDSQLManager( conn, request );
    int sz = form.getValues( "CSTATUS" ).length;
    for ( int n = 0 ; n < sz ; n++ ) {
        cud.setTable( "PA1010" );
        cud.addKey( "C_CD", ehrbean.getCCD() );
        cud.addKey( "EMP_ID", form.getValues( "EMP_ID" )[ n ] );
        cud.addField( "CEN_RNW_YMD", form.getValues( "CEN_RNW_YMD" )[ n ] );
        cud.addField( "MOD_USER_ID", ehrbean.get( "USER_ID" ) );
        cud.addFieldRaw( "MOD_YMDHMS", "SYSDATE" );
        cud.setPsnIdenStdCol( "EMP_ID" );
        cud.update();
    }
}
```

```
cud.setTable( "PA1010" );
cud.addKey( "C_CD", ehrbean.getCCD() );
cud.addKey( "EMP_ID", form.getValues( "EMP_ID" )[ n ] );
cud.addField( "CEN_RNW_YMD", form.getValues( "CEN_RNW_YMD" )[ n ] );
cud.addField( "MOD_USER_ID", ehrbean.get( "USER_ID" ) );
cud.addFieldRaw( "MOD_YMDHMS", "SYSDATE" );
cud.setPsnInfoAccessLogUse( false );
cud.update();
```

5. 프로시저 로그 관련 프로시저 수정

- 프로시저내에서 접근한 데이터의 정보를 TMP_ACCESS_LOG_TRG 테이블에 insert 한다.

```
INSERT INTO TMP_ACCESS_LOG_TRG
(TABLE_NM, CRUD, PSN_IDEN_STD_COI, ACCESS_TRG, ACCESS_TXT)
SELECT 'PM5210'
, 'I'
, 'EMP_ID'
, COL2
, 'INSERT INTO PM5210
(
    C_CD
, EVU_STD_ID
, EMP_EVU_NO
, EMP_ID
, ORG_ID
, TRG_YN
, STA_YMD
, END_YMD
, STD_YMD
, WT
, EVU_T_GRP
, INS_USER_ID
, INS_YMDHMS
, MOD_USER_ID
, MOD_YMDHMS
, INS_GYMDHMS
, MOD_GYMDHMS
)
SELECT T1.COL1 C_CD
, I_EVU_STD_ID EVU_STD_ID
, ROWNUM + V_EMP_EVU_NO EMP_EVU_NO
, T1.COL2 EMP_ID
, T1.COL3 ORG_ID
, 'Y' TRG_YN
, R_PM5010.STA_YMD STA_YMD
, R_PM5010.END_YMD END_YMD
, R_PM5010.STD_YMD STD_YMD
, 100 WT
, T1.COL4 EMP_GRP_CD
, I_MOD_USER_ID INS_USER_ID
, SYSTIMESTAMP INS_YMDHMS
, I_MOD_USER_ID MOD_USER_ID
, SYSTIMESTAMP MOD_YMDHMS
, CURRENT_TIMESTAMP INS_GYMDHMS
, CURRENT_TIMESTAMP MOD_GYMDHMS
FROM TMP_TAB10 T1'
FROM TMP_TAB10;
```

해당 프로시저는 P_PM2_CRE_EVU_T(피평가자생성) 프로시저이다.
 TMP_TAB10 테이블은 평가대상자를 담고 있는 테이블이다.
 TMP_TAB10 테이블의 데이터를 읽어와 TMP_ACCESS_LOG_TRG 테이블에 담아둔다.

core 수정 내용

1. 조회로그관련 core 수정

1.1 VarStatement.java 수정

- **String** psnIdenStdCol 전역변수 선언(개인식별기준컬럼)
- **boolean** psnInfoAccessLogUse 전역변수 선언(개인정보접근로그사용여부)
- setPsnIdenStdCol(String psnIdenStdCol), getPsnIdenStdCol(), setPsnInfoAccessLogUse(boolean bool), getPsnInfoAccessLogUse(), insertAccessLog 메소드 추가

1.2 SQLUtil.java 수정

- insertAccessLog 메소드 추가
- getResultSetWithClose, getResultSet 메소드에서 insertAccessLog 메소드 호출

2. 수정로그관련 core 수정

2.1 CUDSQLManager.java 수정

- **String** psnIdenStdCol 전역변수 선언(개인식별기준컬럼)
- **String** accessTrgVal 전역변수 선언(개인식별기준컬럼의 값을 담을 변수)
- **boolean** psnInfoAccessLogUse 전역변수 선언(개인정보접근로그사용여부)
- setPsnIdenStdCol(String psnIdenStdCol), setPsnInfoAccessLogUse(boolean bool), insertAccessLog(String CUD) 메소드 추가
- insert(), update(), delete() 에서 insertAccessLog메소드 호출
- setTable에서 psnIdenStdCol null로 초기화

3. 프로시저로그관련 core 수정

3.1 VarStatement.java 수정

- insertAccessLog() 메소드 추가

4. 로그저장처리관련 core 수정

4.1 hunelServlet.java 수정

- was 기동시 ServerStartUtil.init()을 실행함.
- was 기동시 system.properties의 PSN_INFO_ACCESS_LOG_YN 값이 'Y' 이면 쓰레드 풀을 생성함.

4.2 hunelActionServlet.java 수정

- doAction요청이 끝난 후 **request.getAttribute("INSERT_LOG_LIST")** 의 값을 읽어�.
읽어온 값을 매개변수로 Runnable 인터페이스를 구현한 InsertLog객체를 생성하여 쓰레드 풀에 던져줌.

개인정보 분리보관



원래테이블명 (예: PA1010)

사번(PK)	시퀀스(PK)	컬럼1	민감정보	분리보관
2016001	1	일반정보	민감정보	
2016002	2	일반정보	민감정보	Y

NULL로
UPDATE

P_PSN_INFO_SEPA_STOR

- 퇴직후 테이블의 정보를 PK를 남기고 나머지 모든 데이터를 NULL처리
분리보관 표시 후 분리보관테이블로 데이터를 옮긴다

분리보관테이블명 (예: PA1010_RET)

사번(PK)	시퀀스(PK)	컬럼1	민감정보
2016002	2	일반정보	민감정보

조회 때는 원테이블과
분리보관테이블의
데이터를 Union 하여
View로 조회



1. 같은 USER
2. 다른 USER
3. DBLINK

추가된 내용 및 설정

1. 테이블생성 및 변경

1.1 테이블생성

- SY9041(개인정보 분리보관 접근로그 테이블 관리) 테이블 생성(접근로그에서 이미 생성함)

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions	Stats/Size
Column Name	ID	PK	Index Pos	Null?	Comments	Data Type	Default			
TABLE_NM	1	1	1	N	테이블명	VARCHAR2 (100 Byte)				
TABLE_TXT	2			Y	테이블내용	VARCHAR2 (500 Byte)				
SEPA_CLASS	3			Y	분리구분	VARCHAR2 (20 Byte)				
PSN_IDEN_STD_COL	4			Y	개인식별기준컬럼	VARCHAR2 (30 Byte)				
PSN_INFO_COL	5			Y	개인정보컬럼	VARCHAR2 (1000 Byte)				
LOG_USE_YN	6			N	로그사용여부	VARCHAR2 (1 Byte)	'Y'			
APPL_TABLE_YN	7			N	신청서테이블여부	VARCHAR2 (1 Byte)	'N'			
NOTE	8			Y	비고	VARCHAR2 (500 Byte)				
INS_USER_ID	9			Y	입력자ID	VARCHAR2 (20 Byte)				
INS_YMDHMS	10			Y	입력일시	TIMESTAMP(6)				
MOD_USER_ID	11			Y	작업자ID	VARCHAR2 (20 Byte)				
MOD_YMDHMS	12			Y	수정일시	TIMESTAMP(6)				
INS_GYMDHMS	13			Y	입력일시_현지	TIMESTAMP(6) WITH TIME ZONE				
MOD_GYMDHMS	14			Y	수정일시_현지	TIMESTAMP(6) WITH TIME ZONE				

< Editable Table Comments
(SY9041)개인정보 분리보관 접근로그 테이블 관리

개인정보 분리보관

- SY9042(개인정보분리보관) 테이블 생성

Columns	Indexes	Constraints	Triggers	Data	Script	Grants	Synonyms	Partitions	Subpartitions	Stats/Size
Column Name	ID	PK	Index Pos	Null?	Comments	Data Type	Default			
C_CD	1	1	1	N	회사코드	VARCHAR2 (20 Byte)				
EMP_ID	2	2	2	N	사번	VARCHAR2 (100 Byte)				
SEPA_CLASS	3	3	3	N	분리구분	VARCHAR2 (20 Byte)				
SEPA_STOR_YN	4			N	분리보관 여부	VARCHAR2 (1 Byte)	'N'			
DESTOR_YMD	5			Y	파기일자	VARCHAR2 (8 Byte)				
INS_USER_ID	6			Y	입력자ID	VARCHAR2 (20 Byte)				
INS_YMDHMS	7			Y	입력일시	TIMESTAMP(6)				
MOD_USER_ID	8			Y	작업자ID	VARCHAR2 (20 Byte)				
MOD_YMDHMS	9			Y	수정일시	TIMESTAMP(6)				
INS_GYMDHMS	10			Y	입력일시_현지	TIMESTAMP(6) WITH TIME ZONE				
MOD_GYMDHMS	11			Y	수정일시_현지	TIMESTAMP(6) WITH TIME ZONE				
<input type="checkbox"/> Editable Table Comments										
(SY9042)개인정보분리보관										

1.2 테이블 변경

- PA1010테이블에 PSN_INFO_DATA_STOR_Y_CNT 컬럼 추가(개인정보데이터보관년수)

- 테이블변경 스크립트

```
: ALTER TABLE PA1010 ADD(PSN_INFO_DATA_STOR_Y_CNT NUMBER DEFAULT 5);
```

```
: COMMENT ON COLUMN PA1010.PSN_INFO_DATA_STOR_Y_CNT IS '개인정보데이터보관년수';
```

2. 환경설정

2.1 system.properties 수정

2.1.1 같은유저에서 분리보관 할 경우

- RET_JNDINAME=HUNELWEBSTAND 추가

2.1.2 다른유저 또는 물리적보관장소가 다른곳에서 분리보관 할 경우

- RET_JNDINAME=HUNELWEBSTAND_RET 추가

2.2 server.xml 수정(tomcat기준)

2.1.2 다른유저 또는 물리적보관장소가 다른곳에서 분리보관 할 경우

- resource에 HUNELWEBSTAND_RET등록(system.properties의 RET_JNDINAME과 같아야 함)

```
<Service name="HUNELWEBSTAND">
  <Connector URIEncoding="utf-8" connectionTimeout="20000" port="8090" protocol="HTTP/1.1" redirectPort="8443" maxPostSize="0"/>
  <Connector URIEncoding="utf-8" port="8010" protocol="AJP/1.3" redirectPort="8443"/>
  <Engine defaultHost="localhost" name="HUNELWEBSTAND">
    <Realm className="org.apache.catalina.realm.UserDatabaseRealm" resourceName="UserDatabase"/>
    <Host appBase="webapps" autoDeploy="true" name="localhost" unpackWARs="true" xmlNamespaceAware="false" xmlValidation="false">
      <Valve className="org.apache.catalina.valves.AccessLogValve" directory="logs" pattern="common" prefix="mh_ehr_access_log." resolveHosts="false" suffix=".txt"/>
      <Context docBase="C:/EHR_PROJECT/HUNEL_EHRS_WEB_STAND/WebRoot" path="" privileged="true" reloadable="true">
        <Resource auth="Container" connectionProperties="SetBigStringTryClob=true" driverClassName="oracle.jdbc.driver.OracleDriver" maxActive="20" maxIdle="10" name="HUNELWEBSTAND" password=
        <Resource auth="Container" connectionProperties="SetBigStringTryClob=true" driverClassName="oracle.jdbc.driver.OracleDriver" maxActive="20" maxIdle="10" name="HUNELWEBSTAND_RET" passw
        <Resource auth="Container" connectionProperties="SetBigStringTryClob=true" driverClassName="oracle.jdbc.driver.OracleDriver" maxActive="20" maxIdle="10" name="REEHR" username="REEHR"
      </Context>
    </Host>
  </Engine>
</Service>
```

개인정보 분리보관 테이블 관리 프로그램에서 분리보관할 테이블을 등록하고 분리보관할 테이블 및 뷰를 생성해준다. 분리보관할 테이블을 다른유저 또는 물리적공간이 다른 DB에서 관리할 경우가 있기때문에 별도의 jndi가 필요함.

2.3 환경변수 추가

- 시스템 -> 시스템관리 -> 환경변수/평가유형 프로그램의 환경변수를 등록한다.

No	*환경변수명	*변수값	모듈	설명	<input type="checkbox"/> 삭제
63	BASE_WORK_FORM_CD	10	근태	<input checked="" type="checkbox"/> 기본 근무형태 코드	<input type="checkbox"/>
64	BASE_HOLI_DPS_CLASS	002	근태	<input checked="" type="checkbox"/> 기본 근무조 휴일처리 구분	<input type="checkbox"/>
65	BASE_SHIFT_CD	Z	근태	<input checked="" type="checkbox"/> 기본 근무조 코드	<input type="checkbox"/>
66	BASE_STD_WORK_YMD	20141201	근태	<input checked="" type="checkbox"/> 기본 근무 생성 기준일	<input type="checkbox"/>
67	QUOTA_CD	0040	근태	<input checked="" type="checkbox"/> 근태코드(연차) - 메인 연차현황관리사용	<input type="checkbox"/>
68	NIGHT_ADD_RATE	0.5	근태	<input checked="" type="checkbox"/> 야간 가급률	<input type="checkbox"/>
69	OVER_ADD_RATE	0.5	근태	<input checked="" type="checkbox"/> 연장 가급률	<input type="checkbox"/>
70	WH_CD	20	근태	<input checked="" type="checkbox"/> 주휴코드	<input type="checkbox"/>
71	WORK_STD_LAW_YN	N	근태	<input checked="" type="checkbox"/> 연차 1년미만자 회계년도 도래 이후 연차 부여시 월 1개가 아니고 근무	<input type="checkbox"/>
72	OVERTIME_PLAN_YN	Y	근태	<input checked="" type="checkbox"/> 휴일특근계획을 받을지 말지 여부(Y/N)	<input type="checkbox"/>
73	REALTIME_MODIFY_YN	N	근태	<input checked="" type="checkbox"/> 휴일특근신청시 실근무시간 수정 여부(Y/N)	<input type="checkbox"/>
74	HOLI_OT_WORK_TIME_CD	2	근태	<input checked="" type="checkbox"/> 휴일특근신청시 시작근무시간 코드(TM102)	<input type="checkbox"/>
75	BASE_WORK_AGGR_TYPE	DEFAULT_TIME	근태	<input checked="" type="checkbox"/> 일집계 시 기본 집계시간(DEFAULT_TIME : 기본시간, TIME_CARD	<input type="checkbox"/>
76	PSN_INFO_SEPA_STOR_CLASS	2	시스템	<input checked="" type="checkbox"/> 개인정보 분리보관 구분(1 = 같은유저, 2 = 다른유저, 3 = DBLink)	<input type="checkbox"/>
77	PSN_INFO_SEPA_STOR_DB_ALIAS	HUNELHTMLEHR_RET	시스템	<input checked="" type="checkbox"/> 개인정보 분리보관 DB ALIAS	<input type="checkbox"/>

개인정보 분리보관 테이블 관리 프로그램 에서 뷰를 생성시 위의 두 환경변수에 따라 다르게 생성된다.

Ex) 같은유저일경우 : 사용안함

다른유저일경우 : FROM HUNELHTMLEHR_RET.PA1010 (유저명)

DBLINK일경우 : FROM PA1010@HUNELHTMLEHR_RET (DBLINK명)

같은유저일경우라도 PSN_INFO_SEPA_STOR_DB_ALIAS의 환경변수명은 무조건 등록되어 있어야 함.

2.4 다국어 프로퍼티 추가

- MSG_ERROR_SY_0001 //시스템 환경변수 PSN_INFO_SEPA_STOR_CLASS 에 구분값을 입력하세요.
- MSG_ERROR_SY_0002 //시스템 환경변수 PSN_INFO_SEPA_STOR_DB_ALIAS 에 DB ALIAS값을 입력하세요.
- MSG_ERROR_DBMS_SYS_0001 //시스템 환경변수 PSN_INFO_SEPA_STOR_CLASS, PSN_INFO_SEPA_STOR_DB_ALIAS 가 존재하는지 확인하세요.
- MSG_ERROR_DBMS_SYS_0002 //개인정보관리 테이블을 확인 하세요.

3. 공통코드 추가

- /SY104 코드 추가(SY104코드.xls 파일참조)

4. 프로그램 추가

4.1 개인정보 분리보관 테이블 관리 프로그램 추가(접근로그에서 이미 추가됨)

- WebRoot/sys/sy_bas 경로에 sy_bas_250_ul01.jsp 파일추가
- src/biz/sys/sy_bas 경로에 Sy_bas_250_ul01.java, Sy_bas_250_ul01.xml 파일추가
- 시스템 모듈에 메뉴 추가

> 시스템관리

- 환경변수/평가유형
- 배치작업관리
- 배치로그조회
- 테이블명세
- 프로시저명세
- 메일발송
- SMS발송
- 도움말관리
- 사번삭제
- 웹서비스관리

> 개인정보보호

- 암호호화 예외컬럼 관리
- 개인정보분리.로그테이블관리
- 개인정보접근로그 조회
- 개인정보분리보관

개인정보분리.로그테이블관리

테이블명

Q 조회

> 개인정보분리.로그테이블관리

+ 추가

저장

선택테이블생성

DOWN

No	선택	*테이블명	테이블내용	분리보관			접속로그			비고	삭제	상태
				분리구분	신청서테이블 여부	생성 여부	개인식별 기준컬럼	개인정보컬럼	로그사용 여부			
9	<input type="checkbox"/>	PA1010	인사기본_기본사항	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM,BIRTH_YMD,FAM_MASTI	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
10	<input type="checkbox"/>	PA1050	발령_사번관리	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
11	<input type="checkbox"/>	PA1055	발령_전입자사번관리	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PER_NO,EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
12	<input type="checkbox"/>	PA2010	인사기본_주소	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	TEL_NO,ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
13	<input type="checkbox"/>	PA2030	인사기본_가족	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	PER_NO	PER_NO,FAM_NM,BIRTH_YMD,CONTACT_T	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
14	<input type="checkbox"/>	PA2040	인사기본_자격	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	CERT_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
15	<input type="checkbox"/>	PA2070	인사기본_병역	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	MILSVC_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
16	<input type="checkbox"/>	PA2100	인사기본_보훈	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PATRIOT_NO,PATRIOT_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
17	<input type="checkbox"/>	PA2110	인사기본_신원보증	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	GRNT_PER_NO,GRNT_NO,ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
18	<input type="checkbox"/>	PA2130	인사기본_장애	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	DISABLE_REG_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
19	<input type="checkbox"/>	PA2180	인사기본_여권	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	PSPT_NO,ENG_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
20	<input type="checkbox"/>	PA2190	인사기본_비자	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	VISA_NO,ENG_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
21	<input type="checkbox"/>	PA2240	인사기본_차량	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	CAR_NO,CAR_OWN_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
22	<input type="checkbox"/>	PA7020	재종영_발급대장	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	EMP_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
23	<input type="checkbox"/>	PA8030	인사기본_자민정보	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	MOBILE_NO,ACQ_NM,OFFICE_TEL_ADDR	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
24	<input type="checkbox"/>	PA8050	인사기본_해외주재원관리	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	LOCAL_PSN_TEL_NO,LOCAL_MOBILE_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
25	<input type="checkbox"/>	PY2060	급여_개인별계좌	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
26	<input type="checkbox"/>	PY2370	급여_은행이체파일	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
27	<input type="checkbox"/>	PY7010	급여_계좌변경신청	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	ACC_NO,DEPO_NM	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
28	<input type="checkbox"/>	TE1200	교육_강사	정규직외기타	<input type="checkbox"/>	<input checked="" type="checkbox"/>	LECT_ID		<input type="checkbox"/>	LECT_NM	<input type="checkbox"/>	
29	<input type="checkbox"/>	TM5300	근태_근태신청	퇴직직후	<input type="checkbox"/>	<input checked="" type="checkbox"/>	EMP_ID	EGC_TEL_NO	<input checked="" type="checkbox"/>		<input type="checkbox"/>	
30	<input type="checkbox"/>	TX0030	세무_일용소득자관리	정규직외기타	<input type="checkbox"/>	<input checked="" type="checkbox"/>	PER_NO		<input type="checkbox"/>	INCMN_NM	<input type="checkbox"/>	

[30 / 30]

개인정보 분리보관

4.2 개인정보분리보관 프로그램 추가

- WebRoot/sys/sy_bas 경로에 sy_bas_260_m01.jsp 파일추가
- src/biz/sys/sy_bas 경로에 Sy_bas_260_m01.java, Sy_bas_260_m01.xml 파일추가
- 시스템 모듈에 메뉴 추가




> 시스템관리

- 환경변수/평가유형
- 배치작업관리
- 배치로그조회
- 테이블명세
- 프로시저명세
- 메일발송
- SMS발송
- 도움말관리
- 사변삭제
- 웹서비스관리

> 개인정보보호

- 암호호화 예외컬럼 관리
- 개인정보분리,로그테이블관리
- 개인정보접근로그 조회
- 개인정보분리보관

개인정보분리보관

* 사변/성명  * 기간 2016.08.22  ~ 2016.09.21  기간선택 * 파기여부 -전체- ☐ ☒ 파기포함여부

> 개인정보분리보관

분리구분 -선택-

No	<input type="checkbox"/> 선택	사변	성명	분리구분				파기일	보관 년수	파기에정일	파기일	상태
				퇴직직후	퇴직자소득세	기타	파기					
1	<input type="checkbox"/>	01037		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/21	
2	<input type="checkbox"/>	20120601		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/20	
3	<input type="checkbox"/>	96027		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/20	
4	<input type="checkbox"/>	07004		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/12	
5	<input type="checkbox"/>	08013		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/09	
6	<input type="checkbox"/>	96075		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
7	<input type="checkbox"/>	03096		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
8	<input type="checkbox"/>	07045		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
9	<input type="checkbox"/>	05030		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
10	<input type="checkbox"/>	08030		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
11	<input type="checkbox"/>	05002		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
12	<input type="checkbox"/>	06071		<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>		5		2016/09/08	
13	<input checked="" type="checkbox"/>	06020	오X훈	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	2016/09/08	5	2021/09/08		

개인정보 분리보관

4.3 개인정보분리보관(그외) 프로그램 추가

- WebRoot/sys/sy_bas 경로에 sy_bas_270_m01.jsp 파일추가
- src/biz/sys/sy_bas 경로에 Sy_bas_270_m01.java, Sy_bas_270_m01.xml 파일추가
- 시스템 모듈에 메뉴 추가

> 시스템관리

- 환경변수/평가유형
- 배치작업관리
- 배치로그조회
- 테이블명세
- 프로시저명세
- 메일발송
- SMS발송
- 도움말관리
- 사번삭제
- 웹서비스관리

> 개인정보보호

- 암호호화 예외컬럼 관리
- 개인정보분리,로그테이블관리
- 개인정보접근로그 조회
- 개인정보분리보관
- 개인정보분리보관(그외)

개인정보분리보관(그외)

* 테이블내용 -전체-
* 파기여부 -전체-
Q 조회

> 개인정보분리보관(그외)
선택분리보관
선택복구
선택파기
DOWN

No	<input type="checkbox"/> 선택	테이블내용	기준		대상자	분리구분 여부	파기여부	파기일	상태
			COLUMN	ID					
1	<input type="checkbox"/>	교육_강사	LECT_ID	10000196		<input type="checkbox"/>	<input checked="" type="checkbox"/>	2016/09/21	
2	<input type="checkbox"/>	교육_강사	LECT_ID	10000197	마크 저커버그	<input type="checkbox"/>	<input type="checkbox"/>		
3	<input type="checkbox"/>	교육_강사	LECT_ID	10000198		<input type="checkbox"/>	<input checked="" type="checkbox"/>	2016/09/21	
4	<input type="checkbox"/>	교육_강사	LECT_ID	10000200	김건희	<input type="checkbox"/>	<input type="checkbox"/>		
5	<input type="checkbox"/>	교육_강사	LECT_ID	10000201		<input type="checkbox"/>	<input checked="" type="checkbox"/>	2016/09/21	
6	<input type="checkbox"/>	교육_강사	LECT_ID	10000202	강사2	<input type="checkbox"/>	<input type="checkbox"/>		
7	<input type="checkbox"/>	교육_강사	LECT_ID	10000203	강사1	<input type="checkbox"/>	<input type="checkbox"/>		
8	<input type="checkbox"/>	교육_강사	LECT_ID	10000204	안X일	<input type="checkbox"/>	<input type="checkbox"/>		
9	<input type="checkbox"/>	세무_일용소득자관리	PER_NO	870806-2853772	김동개	<input type="checkbox"/>	<input type="checkbox"/>		
10	<input type="checkbox"/>	세무_일용소득자관리	PER_NO	651024-2330950	테스트99	<input type="checkbox"/>	<input type="checkbox"/>		
11	<input type="checkbox"/>	세무_일용소득자관리	PER_NO	590316-1553087		<input type="checkbox"/>	<input checked="" type="checkbox"/>	2016/09/21	

5. 프로시저 생성 및 수정

5.1 P_PSN_INFO_SEPA_STOR 프로시저 생성

- SY9041(개인정보 분리보관 테이블 관리) 테이블에서 분리보관사용여부가 Y 인 테이블 리스트를 읽어와 해당 테이블의 데이터를 분리보관 테이블(_RET)에 복사, 원래테이블의 PSN_INFO_SEPA_STOR_YN 컬럼의 값을 'Y'로 업데이트 및 key를 제외한 값들을 null로 업데이트 함.

5.2 P_PSN_INFO_SEPA_STOR_ETC 프로시저 생성

- 정규직 외에 EMP_ID가 없는 데이터들을 분리보관 하는 프로시저(P_PSN_INFO_SEPA_STOR 프로시저와 유사)

6. 소스수정(java, xml, procedures)

- 분리보관목록에 있는 테이블들을 사용하여 조회하는 프로그램 또는 프로시저에서 테이블을 뷰로 변경.
ex) SELECT EMP_ID FROM PA1010 --> SELECT EMP_ID FROM VR_PA1010

