LG전자

70%미만(3개월예상) / 70%미만(3개월예상) 발송

프로시져 설계서

**STEP**

**문서번호**

**작성자**

**작성일**

**TASK**

**STAGE**

이상훈C

2016.05.03

프로시져 설계서

ED\_진화적 전달

문서 이력

검토

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 번호 | 검토 일자 | 개정 | 검토자 | 서명 |
| 1 |  |  |  |  |
| 2 |  |  |  |  |
| 3 |  |  |  |  |
| 4 |  |  |  |  |
| 5 |  |  |  |  |
| 6 |  |  |  |  |
| 7 |  |  |  |  |
| 8 |  |  |  |  |
| 9 |  |  |  |  |
| 10 |  |  |  |  |

개정 기록

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| 번호 | 변경 일자 | 버전 | 변경 내용 | 작성자 | 승인자 |
| 1 | 2016-05-03 | V.1.0 | Initial Release | 이상훈C |  |
| 2 |  |  |  |  |  |
| 3 |  |  |  |  |  |
| 4 |  |  |  |  |  |
| 5 |  |  |  |  |  |
| 6 |  |  |  |  |  |
| 7 |  |  |  |  |  |
| 8 |  |  |  |  |  |
| 9 |  |  |  |  |  |
| 10 |  |  |  |  |  |
| 11 |  |  |  |  |  |
| 12 |  |  |  |  |  |
| 13 |  |  |  |  |  |
| 14 |  |  |  |  |  |
| 15 |  |  |  |  |  |
| 16 |  |  |  |  |  |
| 17 |  |  |  |  |  |
| 18 |  |  |  |  |  |
| 19 |  |  |  |  |  |
| 20 |  |  |  |  |  |
| 21 |  |  |  |  |  |
| 22 |  |  |  |  |  |

목 차

[1. 개요 1](#_Toc442348355)

[1.1 Definitions 1](#_Toc442348356)

[1.2 기본 흐름 1](#_Toc442348357)

[2. 70%미만(3개월예상) 1](#_Toc442348358)

[2.1 Procedure개요 1](#_Toc442348359)

[2.1.1 m\_opsmr\_sp\_rate\_70\_less 1](#_Toc442348360)

[2.2 Return Value 24](#_Toc442348361)

[2.3 Table and View Usage 25](#_Toc442348362)

# 개요

## Definitions

사용자가 원하는 데이터를 추출하기 위하여 지정된 조건을 임의로 입력하여 조회하고 그 결과를 보고서 엑셀파일로 생성하는 작업에 대해 정의한다.

1. 70%미만(3개월예상)/ 70%미만(3개월예상)-발송
2. 70%미만(3개월예상) 조회.

## 기본 흐름

1) 가동율 COPY 시트 / 운영CAPA시트

DB2

EXCEL

**8000자 제한으로 인해서 엑셀에서 DB2를 호출 후 엑셀에서 MS-SQL로 적재한다..**

**현재월 이전까지의 데이터는 엑셀로 UPLOAD 한다.**

EXCEL

MS-SQL

**M\_OPSMR\_TB\_OP\_RATE**

**에 적재한다.**

# 70%미만(3개월예상)/ 70%미만(3개월예상)-발송

## Procedure개요

## 2.1.1 m\_opsmr\_sp\_rate\_70\_less

|  |  |
| --- | --- |
| **프로시져 설명** | 10% 이상변동(차월) 등을 조회함. |
| **관련 Application** |  |
| **사전 조건** | M\_OPSMR\_TB\_OP\_RATE 적재 |
| **상세 로직** | 1.과거데이타는 엑셀에서 기준월부터는 쿼리로 적재 후 그 결과를 프로시저로 호출하도록 함.  2. m\_opsmr\_sp\_rate\_70\_less 을 실행 후 보고서 생성함.  3. 호출 파라메터    (1) opsmr\_type : 기준/운영 구분  (2) base\_yyyymm : 조회기준일  (3) base\_rate : 기준율    4. 사용 쿼리  시작년월이 의미가 없어, 제외하고 추후 기준율과 변동율이 바뀌었을 때를 대비하여 기준율과 변동율을 입력받기로 함.  **m\_opsmr\_sp\_rate\_70\_less**  ALTER PROCEDURE [dbo].[m\_opsmr\_sp\_rate\_70\_less]  (  @opsmr\_type VARCHAR(5)  ,@base\_yyyymm VARCHAR(6)  ,@base\_rate FLOAT  )  AS  /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  1. 프 로 젝 트 : M\_OPSMR  2. 프로그램 ID : m\_opsmr\_sp\_rate\_70\_less  3. 기 능 : DB2 기준가동률 및 운영가동률을 m\_opsmr\_sp\_rate\_70\_less  -- EXEC m\_opsmr\_sp\_rate\_70\_less 'STD', @base\_yyyymm, 70  4. 관 련 화 면 :  버전 작 성 자 일 자 내 용  ---- --------- ---------- -----------------------------------------------  1.0 shlee 2016.04.11 최초작성  1.1 shlee 2016.04.21 날짜와 subsidiary/ Product Master적용  1.2 shlee 2016.04.22 \* 100 제거  \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/  DECLARE @vc\_pre\_13 AS VARCHAR(6);  DECLARE @vc\_py\_yyyy AS VARCHAR(6);  DECLARE @vc\_start\_yyyymm AS VARCHAR(6);  DECLARE @vc\_py\_act\_start\_yyyymm AS VARCHAR(6);  DECLARE @vc\_py\_act\_end\_yyyymm AS VARCHAR(6);  DECLARE @vc\_py\_plan\_start\_yyyymm AS VARCHAR(6);  DECLARE @vc\_py\_plan\_end\_yyyymm AS VARCHAR(6);  DECLARE @vc\_post\_1 AS VARCHAR(6);  DECLARE @vc\_post\_2 AS VARCHAR(6);  DECLARE @vc\_post\_3 AS VARCHAR(6);  SET NOCOUNT ON  SET @vc\_pre\_13 = CONVERT(VARCHAR(6), DATEADD(m,-12, CONVERT(DATETIME,@base\_yyyymm + '01')), 112); -- 전13월  SET @vc\_py\_yyyy = CONVERT(VARCHAR(4), DATEADD(yy, -1, CONVERT(DATETIME,@base\_yyyymm + '01')), 112); -- 전년  SET @vc\_start\_yyyymm = SUBSTRING(@base\_yyyymm,1,4)+'01'; -- 누적년시작월 @vc\_start\_yyyymm  SET @vc\_py\_act\_start\_yyyymm = @vc\_py\_yyyy + '01'; -- 전년01월  SET @vc\_py\_act\_end\_yyyymm = @vc\_py\_yyyy + SUBSTRING(@base\_yyyymm,5,2); -- 전년현재월  SET @vc\_py\_plan\_start\_yyyymm = @vc\_py\_yyyy + SUBSTRING(CONVERT(VARCHAR(6), DATEADD(m, 1, CONVERT(DATETIME,@base\_yyyymm + '01')), 112),5,2); -- 전년차월  SET @vc\_py\_plan\_end\_yyyymm = @vc\_py\_yyyy + SUBSTRING(CONVERT(VARCHAR(6), DATEADD(m, 3, CONVERT(DATETIME,@base\_yyyymm + '01')), 112),5,2); -- 전년차차차월  SET @vc\_post\_1 = CONVERT(VARCHAR(6), DATEADD(m, 1, CONVERT(DATETIME,@base\_yyyymm + '01')), 112); -- 차월  SET @vc\_post\_2 = CONVERT(VARCHAR(6), DATEADD(m, 2, CONVERT(DATETIME,@base\_yyyymm + '01')), 112); -- 차월  SET @vc\_post\_3 = CONVERT(VARCHAR(6), DATEADD(m, 3, CONVERT(DATETIME,@base\_yyyymm + '01')), 112); -- 차월  BEGIN  select \*  from (  select a.sub  ,a.prod  ,max(a.sub\_enm) AS sub\_enm  ,max(a.sub\_knm) AS sub\_knm  ,max(a.prod\_enm) AS prod\_enm  ,max(a.prod\_knm) AS prod\_knm  ,sum(a.plan\_total\_rate) AS plan\_total\_rate  ,sum(a.plan\_post\_1\_rate) AS plan\_post\_1\_rate  ,sum(a.plan\_post\_2\_rate) AS plan\_post\_2\_rate  ,sum(a.plan\_post\_3\_rate) AS plan\_post\_3\_rate  ,sum(a.act\_cur\_mon\_rate) AS act\_cur\_mon\_rate  ,sum(a.act\_cur\_base\_rate) AS act\_cur\_base\_rate  ,sum(a.act\_py\_base\_rate) AS act\_py\_base\_rate  ,sum(a.act\_py\_plan\_rate) AS act\_py\_plan\_rate  ,sum(a.cur\_mon\_13\_rate) AS cur\_mon\_13\_rate  ,sum(a.rate\_mon\_13\_rate) AS rate\_mon\_13\_rate  ,sum(a.plan\_total\_qty) AS plan\_total\_qty  ,sum(a.plan\_post\_1\_qty) AS plan\_post\_1\_qty  ,sum(a.plan\_post\_2\_qty) AS plan\_post\_2\_qty  ,sum(a.plan\_post\_3\_qty) AS plan\_post\_3\_qty  ,sum(a.act\_cur\_mon\_qty) AS act\_cur\_mon\_qty  ,sum(a.act\_cur\_base\_qty) AS act\_cur\_base\_qty  ,sum(a.act\_py\_base\_qty) AS act\_py\_base\_qty  ,sum(a.act\_py\_plan\_qty) AS act\_py\_plan\_qty  ,sum(a.cur\_mon\_13\_qty) AS cur\_mon\_13\_qty  ,sum(a.rate\_mon\_13\_qty) AS rate\_mon\_13\_qty  ,sum(a.plan\_total\_rate) - sum(a.act\_cur\_mon\_rate) AS vs\_rate  ,sum(a.plan\_total\_qty) - sum(a.act\_cur\_mon\_qty) AS vs\_qty  ,sum(a.plan\_total\_rate) - sum(a.act\_cur\_base\_rate) AS vs\_base\_rate  ,sum(a.accu\_rate) AS accu\_rate  ,sum(a.py13\_cur\_accu\_rate) AS py13\_cur\_accu\_rate  from (  select sub.display\_name AS sub  ,prod.display\_name AS prod  ,MIN(sub.display\_enm) as sub\_enm  ,MIN(sub.display\_knm) as sub\_knm  ,MIN(prod.display\_enm) as prod\_enm  ,MIN(prod.display\_knm) as prod\_knm  ,case when sum(case when a.kpi\_period\_code IN (@vc\_post\_1,@vc\_post\_2,@vc\_post\_3) then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code IN (@vc\_post\_1,@vc\_post\_2,@vc\_post\_3) then a.production\_quantity end) / sum(case when a.kpi\_period\_code IN (@vc\_post\_1,@vc\_post\_2,@vc\_post\_3) then a.production\_capa end) end AS plan\_total\_rate    ,case when sum(case when a.kpi\_period\_code = @vc\_post\_1 then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code = @vc\_post\_1 then a.production\_quantity end) / sum(case when a.kpi\_period\_code = @vc\_post\_1 then a.production\_capa end) end AS plan\_post\_1\_rate    ,case when sum(case when a.kpi\_period\_code = @vc\_post\_2 then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code = @vc\_post\_2 then a.production\_quantity end) / sum(case when a.kpi\_period\_code = @vc\_post\_2 then a.production\_capa end) end AS plan\_post\_2\_rate    ,case when sum(case when a.kpi\_period\_code = @vc\_post\_3 then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code = @vc\_post\_3 then a.production\_quantity end) / sum(case when a.kpi\_period\_code = @vc\_post\_3 then a.production\_capa end) end AS plan\_post\_3\_rate    ,case when sum(case when a.kpi\_period\_code = @base\_yyyymm then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code = @base\_yyyymm then a.production\_quantity end) / sum(case when a.kpi\_period\_code = @base\_yyyymm then a.production\_capa end) end AS act\_cur\_mon\_rate    ,case when sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_quantity end) / sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_capa end) end AS act\_cur\_base\_rate    ,case when sum(case when a.kpi\_period\_code between @vc\_py\_act\_start\_yyyymm and @vc\_py\_act\_end\_yyyymm then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code between @vc\_py\_act\_start\_yyyymm and @vc\_py\_act\_end\_yyyymm then a.production\_quantity end) / sum(case when a.kpi\_period\_code between @vc\_py\_act\_start\_yyyymm and @vc\_py\_act\_end\_yyyymm then a.production\_capa end) end AS act\_py\_base\_rate    ,case when sum(case when a.kpi\_period\_code between @vc\_py\_plan\_start\_yyyymm and @vc\_py\_plan\_end\_yyyymm then a.production\_capa end) = 0 then 0  else sum(case when a.kpi\_period\_code between @vc\_py\_plan\_start\_yyyymm and @vc\_py\_plan\_end\_yyyymm then a.production\_quantity end) / sum(case when a.kpi\_period\_code between @vc\_py\_plan\_start\_yyyymm and @vc\_py\_plan\_end\_yyyymm then a.production\_capa end) end AS act\_py\_plan\_rate    ,case when sum(a.production\_capa) = 0 then 0  else sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_quantity end) / sum(a.production\_capa) end AS cur\_mon\_13\_rate    ,case when sum(a.production\_capa) = 0 then 0  else sum(a.production\_quantity) / sum(a.production\_capa) end AS rate\_mon\_13\_rate    ,SUM(case when a.kpi\_period\_code IN (@vc\_post\_1,@vc\_post\_2,@vc\_post\_3) then a.production\_quantity end)/3 AS plan\_total\_qty  ,sum(case when a.kpi\_period\_code = @vc\_post\_1 then a.production\_quantity end) AS plan\_post\_1\_qty  ,sum(case when a.kpi\_period\_code = @vc\_post\_2 then a.production\_quantity end) AS plan\_post\_2\_qty  ,sum(case when a.kpi\_period\_code = @vc\_post\_3 then a.production\_quantity end) AS plan\_post\_3\_qty  ,sum(case when a.kpi\_period\_code = @base\_yyyymm then a.production\_quantity end) AS act\_cur\_mon\_qty  ,sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_quantity end) AS act\_cur\_base\_qty  ,sum(case when a.kpi\_period\_code between @vc\_py\_act\_start\_yyyymm and @vc\_py\_act\_end\_yyyymm then a.production\_quantity end) AS act\_py\_base\_qty  ,sum(case when a.kpi\_period\_code between @vc\_py\_plan\_start\_yyyymm and @vc\_py\_plan\_end\_yyyymm then a.production\_quantity end) AS act\_py\_plan\_qty  ,sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_quantity end) AS cur\_mon\_13\_qty  ,sum(a.production\_quantity) AS rate\_mon\_13\_qty  ,case when max(pqty.production\_quantity) = 0 then 0  else sum(case when a.kpi\_period\_code between @vc\_start\_yyyymm and @base\_yyyymm then a.production\_quantity end) / max(pqty.production\_quantity) end AS accu\_rate    ,case when SUM(case when a.kpi\_period\_code BETWEEN @vc\_pre\_13 AND @base\_yyyymm THEN a.production\_capa end) = 0 then 0  else SUM(case when a.kpi\_period\_code BETWEEN @vc\_pre\_13 AND @base\_yyyymm THEN a.production\_quantity end) / SUM(case when a.kpi\_period\_code BETWEEN @vc\_pre\_13 AND @base\_yyyymm THEN a.production\_capa end) end AS py13\_cur\_accu\_rate    FROM m\_opsmr\_tb\_op\_rate(nolock) a  ,m\_opsmr\_tb\_op\_rate\_sub\_mst(nolock) sub  ,m\_opsmr\_tb\_op\_rate\_prod\_mst(nolock) prod  ,(SELECT GBU\_CODE,  SUM(PRODUCTION\_QUANTITY) AS production\_quantity  FROM M\_OPSMR\_TB\_OP\_RATE(NOLOCK)  WHERE OPSMR\_TYPE = 'STD'  AND KPI\_PERIOD\_CODE BETWEEN @vc\_start\_yyyymm AND @base\_yyyymm  GROUP BY GBU\_CODE) pqty  where a.opsmr\_type = @opsmr\_type  and a.base\_yyyymm = @base\_yyyymm  and a.kpi\_period\_code BETWEEN @vc\_py\_act\_start\_yyyymm AND @vc\_post\_3  AND a.factory\_region1 = sub.mapping\_code  AND a.gbu\_code = prod.mapping\_code  AND sub.use\_flag = 'Y'  AND prod.use\_flag = 'Y'  AND a.gbu\_code = pqty.gbu\_code  GROUP BY sub.display\_name  ,prod.display\_name  ) a  where a.plan\_total\_rate < @base\_rate\*0.01  group by a.sub  ,a.prod  ) a  where a.plan\_total\_rate <> 0  ;  END |
| **오퍼레이션** |  |
| **에러처리** | 에러 발생시 프로그램 내에서 처리하지 않고 공통모듈 프로그램에서 처리함 |
| **사후 조건** |  |

## 

## 2.2 Return Value

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Argument | Prompt | Value Set | Default Type | Default Value | Option |
|  |  |  |  |  |  |
|  |  |  |  |  |  |

## 

## 2.3 Table and View Usage

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Table Name | Select | Insert | Update | Delete |
| M\_OPSMR\_TB\_OP\_RATE | Y |  |  |  |
| M\_OPSMR\_TB\_OP\_RATE\_PROD\_MST | Y |  |  |  |
| M\_OPSMR\_TB\_OP\_RATE\_SUB\_MST | Y |  |  |  |
|  |  |  |  |  |