**Отчет о выполненной работе по разработке ПО для фиксации потерь стали при простоях и периодах сниженной производительности МНЛЗ-5**

За периоды сниженной производительности МНЛЗ подсчитывается вес стали, недополученный в связи со скоростью разливки ниже нормативной.

В запросе из базы данных 2-го уровня (см. Приложение 1) получаем сведения, сколько (в мм) недолитой длины по каждому профилю (200\*200 и 300\*360) за сутки в течение выбранного периода.

Расчёты выполнены по следующим формулам.

здесь

– недолитая длина металла,

– длина металла при нормативной скорости разливки,

– длина металла при фактической (сниженной) скорости разливки.

, где процент отклонения скорости в базе данных

Получается пропорционально для длины:

– 100%

– 100% -

Объем потерянной стали за сутки считаем произведением длины и ширины (из данных по профилю разливки за сутки) на значение за сутки.

Вес недополученной стали , где = 7800 кг/м3

При простое запросом из базы данных 2-го уровня (см. Приложение 2) определяется время начала и конца простоя, высчитывается его длительность в секундах. Далее выполняется расчет длины отлитого металла, если бы МНЛЗ работала всё это время простоя на пяти ручьях с нормативной скоростью для того профиля и марки, какой преимущественно был до простоя за текущие сутки.

Вес недополученной стали за простой аналогично считаем произведением длины и ширины (из данных по профилю разливки за сутки) на значение вычисленной длины.

На рисунке 1 представлено отображение данных по потерям стали в Системе учёта и расследования простоев.

СОГЛАСОВАНО:

Директор ДАСУТП К.С. Теличко

ПОДГОТОВЛЕНО:

Гл.специалист оАСУТП А.Н. Петрова

Приложение 1

SELECT

outtr.STOP\_DATE,

outtr.steel\_grade\_id,

outtr.practice\_id,

CCM\_GENERAL\_PROGRAM\_VARS.NUMERIC\_VALUE DENSITY,

REP\_CCM\_PRODUCT\_ORDERS.WIDTH,

REP\_CCM\_PRODUCT\_ORDERS.THICKNESS,

DELAYS.Undercast\_length *--(mm)*

FROM

(

SELECT STOP\_DATE,

MAX(innr.cnt) KEEP (DENSE\_RANK LAST ORDER BY cnt) cnt,

MAX(innr.REPORT\_COUNTER) KEEP (DENSE\_RANK LAST ORDER BY cnt) REPORT\_COUNTER,

MAX(innr.PRACTICE\_ID) KEEP (DENSE\_RANK LAST ORDER BY cnt) PRACTICE\_ID,

MAX(innr.STEEL\_GRADE\_ID) KEEP (DENSE\_RANK LAST ORDER BY cnt) STEEL\_GRADE\_ID

FROM

(

SELECT STOP\_DATE,

STEEL\_GRADE\_ID,

PRACTICE\_ID,

MIN(REPORT\_COUNTER) REPORT\_COUNTER,

COUNT(\*) cnt

FROM

(

SELECT steel\_grade\_id,

practice\_id,

TRUNC(STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute, 'DDD') as STOP\_DATE,

report\_counter

FROM REPORTS

WHERE STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute >= TO\_DATE(:startT, 'DD-MM-YYYY') and STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute < TO\_DATE(:stopT, 'DD-MM-YYYY')+ interval '1' day

)

GROUP BY STOP\_DATE, steel\_grade\_id, practice\_id

) innr

GROUP BY STOP\_DATE

) outtr

JOIN CCM\_GENERAL\_PROGRAM\_VARS

ON CCM\_GENERAL\_PROGRAM\_VARS.PRACTICE\_ID = outtr.practice\_id AND CCM\_GENERAL\_PROGRAM\_VARS.VAR\_CODE = 52003 AND AREA\_ID = 1100

JOIN REP\_CCM\_PRODUCT\_ORDERS

ON outtr.REPORT\_COUNTER = REP\_CCM\_PRODUCT\_ORDERS.REPORT\_COUNTER

JOIN

(

SELECT delayDate,

sum(Undercast\_length) as Undercast\_length *--(mm)*

FROM

(

SELECT TRUNC(startS, 'DDD') as delayDate,

startS as startShifted,

stopS as stopShifted,

startS - INTERVAL '4' hour - INTERVAL '30' minute as startTime,

stopS - INTERVAL '4' hour - INTERVAL '30' minute as stopTime ,

Undercast\_length

FROM

(

with rws as

(

select rownum r from dual connect by level <= 2

)

SELECT Undercast\_length,

case r

when 1 then TRUNC(delay\_stop\_shift, 'DDD')

else delay\_start\_shift

end as startS,

case r

when 2 then

CASE

WHEN EXTRACT(DAY from cast(delay\_start\_shift as timestamp)) <> EXTRACT(DAY from cast(delay\_stop\_shift as timestamp))

THEN TRUNC(delay\_stop\_shift, 'DDD')

else delay\_stop\_shift

END

else delay\_stop\_shift

end as stopS

FROM (SELECT (length\_at\_end - length\_at\_beginning) \* ( -0.01 \*percentage\_of\_deviation / (1-0.01 \* percentage\_of\_deviation)) as Undercast\_length,

start\_downtime,

end\_downtime,

start\_downtime + INTERVAL '4' hour + INTERVAL '30' minute as delay\_start\_shift,

end\_downtime + INTERVAL '4' hour + INTERVAL '30' minute as delay\_stop\_shift

FROM REP\_CCM\_DOWNTIME

WHERE start\_downtime + INTERVAL '4' hour + INTERVAL '30' minute >= TO\_DATE(:startT, 'DD-MM-YYYY')

AND start\_downtime + INTERVAL '4' hour + INTERVAL '30' minute < TO\_DATE(:stopT, 'DD-MM-YYYY')+ interval '1' day)

JOIN rws on case

when EXTRACT(DAY from cast(delay\_start\_shift as timestamp)) = EXTRACT(DAY from cast(delay\_stop\_shift as timestamp))

then 2

else 1

end <= rws.r

)

)

GROUP BY delayDate

) DELAYS

ON DELAYS.delayDate = STOP\_DATE

ORDER BY STOP\_DATE

Приложение 2

SELECT

outtr.STOP\_DATE,

outtr.steel\_grade\_id,

outtr.practice\_id,

CCM\_GENERAL\_PROGRAM\_VARS.NUMERIC\_VALUE DENSITY,

REP\_CCM\_PRODUCT\_ORDERS.WIDTH,

REP\_CCM\_PRODUCT\_ORDERS.THICKNESS,

DELAYS.timeinterval

FROM

(

SELECT STOP\_DATE,

MAX(innr.cnt) KEEP (DENSE\_RANK LAST ORDER BY cnt) cnt,

MAX(innr.REPORT\_COUNTER) KEEP (DENSE\_RANK LAST ORDER BY cnt) REPORT\_COUNTER,

MAX(innr.PRACTICE\_ID) KEEP (DENSE\_RANK LAST ORDER BY cnt) PRACTICE\_ID,

MAX(innr.STEEL\_GRADE\_ID) KEEP (DENSE\_RANK LAST ORDER BY cnt) STEEL\_GRADE\_ID

FROM

(

SELECT STOP\_DATE,

STEEL\_GRADE\_ID,

PRACTICE\_ID,

MIN(REPORT\_COUNTER) REPORT\_COUNTER,

COUNT(\*) cnt

FROM

(

SELECT steel\_grade\_id,

practice\_id,

TRUNC(STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute, 'DDD') as STOP\_DATE,

report\_counter

FROM REPORTS

WHERE STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute >= TO\_DATE(:startT, 'DD-MM-YYYY') and STOP\_DATE + INTERVAL '4' hour + INTERVAL '30' minute < TO\_DATE(:stopT, 'DD-MM-YYYY')

)

GROUP BY STOP\_DATE, steel\_grade\_id, practice\_id

) innr

GROUP BY STOP\_DATE

) outtr

JOIN CCM\_GENERAL\_PROGRAM\_VARS

ON CCM\_GENERAL\_PROGRAM\_VARS.PRACTICE\_ID = outtr.practice\_id AND CCM\_GENERAL\_PROGRAM\_VARS.VAR\_CODE = 52003 AND AREA\_ID = 1100

JOIN REP\_CCM\_PRODUCT\_ORDERS

ON outtr.REPORT\_COUNTER = REP\_CCM\_PRODUCT\_ORDERS.REPORT\_COUNTER

JOIN

(

SELECT delayDate,

sum(timeinterval) as timeinterval

FROM

(

SELECT TRUNC(startS, 'DDD') as delayDate,

startS as startShifted,

stopS as stopShifted,

startS - INTERVAL '4' hour - INTERVAL '30' minute as startTime,

stopS - INTERVAL '4' hour - INTERVAL '30' minute as stopTime,

(stopS - startS) \* 24 \* 60 \* 60 as timeinterval

FROM

(

with rws as

(

select rownum r from dual connect by level <= 2

)

SELECT case r

when 1 then TRUNC(delay\_stop\_shift, 'DDD')

else delay\_start\_shift

end as startS,

case r

when 2 then

CASE

WHEN EXTRACT(DAY from cast(delay\_start\_shift as timestamp)) <> EXTRACT(DAY from cast(delay\_stop\_shift as timestamp))

THEN TRUNC(delay\_stop\_shift, 'DDD')

else delay\_stop\_shift

END

else delay\_stop\_shift

end as stopS

FROM (SELECT delay\_start\_time,

delay\_stop\_time,

delay\_start\_time + INTERVAL '4' hour + INTERVAL '30' minute as delay\_start\_shift,

delay\_stop\_time + INTERVAL '4' hour + INTERVAL '30' minute as delay\_stop\_shift

FROM REP\_DELAYS

WHERE delay\_start\_time + INTERVAL '4' hour + INTERVAL '30' minute >= TO\_DATE(:startT, 'DD-MM-YYYY')

AND delay\_start\_time + INTERVAL '4' hour + INTERVAL '30' minute < TO\_DATE(:stopT, 'DD-MM-YYYY'))

JOIN rws on case

when EXTRACT(DAY from cast(delay\_start\_shift as timestamp)) = EXTRACT(DAY from cast(delay\_stop\_shift as timestamp))

then 2

else 1

end <= rws.r

)

)

GROUP BY delayDate

) DELAYS

ON DELAYS.delayDate = STOP\_DATE

ORDER BY STOP\_DATE

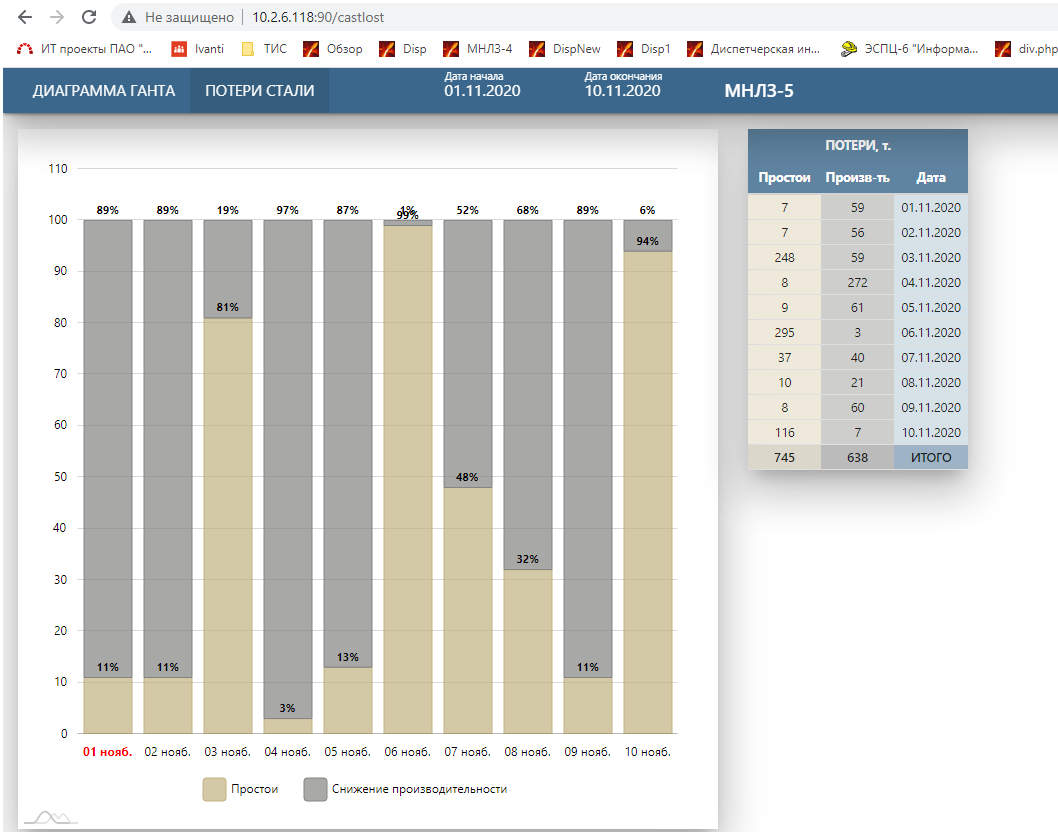


Рисунок 1 – Потери стали в Системе учета и расследования простоев