

ANEXO 2 – PROCESO ETL

En este anexo se documentan los trabajos y transformaciones desarrolladas en Pentaho PDI para obtener, transformar y cargar los datos de la BBDD MIMIC-III a la BBDD intermedia MIMICSEL y obtener a partir de ahí los datos empleados en este estudio. En otro anexo de la presente memoria se documenta el proceso de instalación y carga del catálogo de Pentaho PDI que contiene toda la información de los procesos utilizados y que permite replicar la configuración. Se han incluido además las consultas SQL utilizadas para obtener los datos de origen de la BBDD MIMIC-III.

Contenido

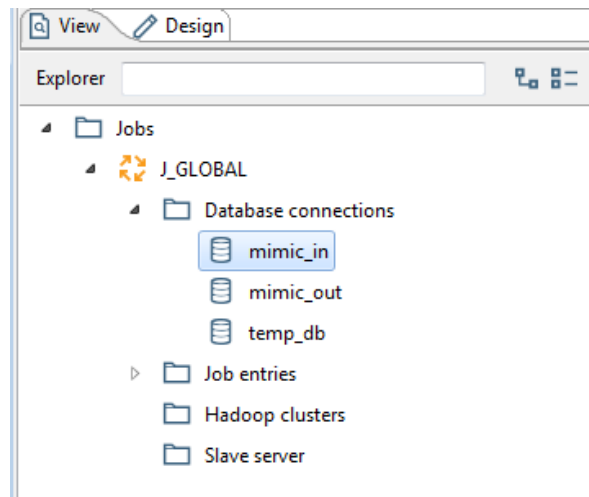
A2-1.	INTRODUCCIÓN	4
A2-2.	ESQUEMA GENERAL.....	6
A2-3.	T_ETL_LOAD_SCORES	9
A2-4.	T_ETL_ASSIGN_COMORBIDITIES	11
A2-5.	T_ETL_ASSIGN_ICD9.....	13
A2-6.	T_ETL_ICU_EVENTS.....	15
A2-7.	T_ETL_ICU_PRESC_INSULIN.....	19
A2-8.	T_ETL_ADT_SEPSIS	22
A2-9.	T_ETL_ICD9_DIAG	23
A2-10.	T_ETL_ICU_VASOPRESS.....	25
A2-11.	T_ETL_COMBINE_TMP_TABLES	27
A2-12.	T_ETL_ICU_STAYS.....	29
A2-13.	T_ETL_ICD9_PROC	34
A2-14.	T_DOC	36
A2-15.	T_ETL_MED_FEATURES.....	37
A2-16.	T_ETL_ADT_ALLERGIES.....	39
A2-17.	T_GRP_ICU_CONSOLIDE_GLUCOSE	41
A2-18.	T_GRP_ICU_GLUCOSE_VARIAB	43
A2-19.	T_GRP_ICU_GLUCOSE	45
A2-20.	T_GRP_ICU_EVENTS	49
A2-21.	T_GRP_ICU_NUTRITION	51
A2-22.	T_GRP_ICU_ADMIN_INSULIN	53
A2-23.	T_SEL_ICU	55
A2-24.	T_ETL_ICU_ADMIN_INSULIN	57
A2-25.	J_CREATE_TABLES_ADM.....	60
A2-26.	J_CREATE_TABLES_EVENTS	62
A2-27.	J_CREATE_TABLES_INSULIN.....	63
A2-28.	J_CREATE_TABLES_MED_FEATURES.....	64
A2-29.	J_CREATE_TABLES_SELECT.....	65
A2-30.	J_GROUP_DATA_GLUCOSE.....	66
A2-31.	J_CREATE_TABLES_GROUP.....	67
A2-32.	J_GROUP_DATA	69
A2-33.	J_SELECT_CASES.....	70
A2-34.	J_POST_ETL.....	71
A2-35.	J_ETL_EVENTS.....	73
A2-36.	J_GLOBAL	74

A2-37.	J_ETL_INSULIN	75
A2-38.	J_GLOBAL_ETL	76
A2-39.	J_ETL_MED_FEATURES	77
A2-40.	J_ETL_ADM	78
A2-41.	J_DOC	79

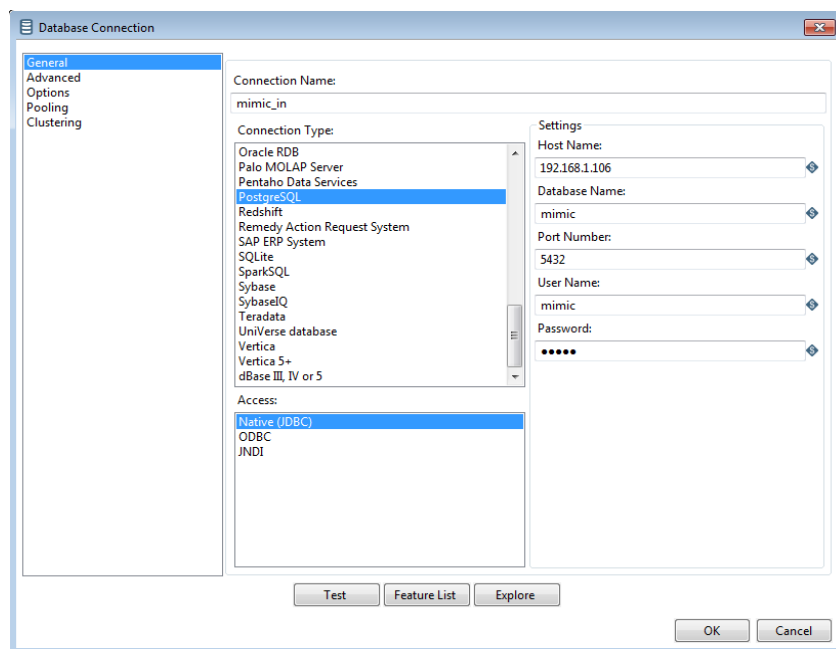
A2-1. INTRODUCCIÓN

Los catálogos de Pentaho PDI son ficheros XML obtenidos a partir de la exportación del repositorio (en BBDD o fichero) que permiten importar todas las transformaciones, trabajos y datos de configuración utilizados. Al cargar el catálogo correspondiente a estos procesos será necesario modificar las conexiones de base de datos utilizadas para adecuarse a los nuevos entornos.

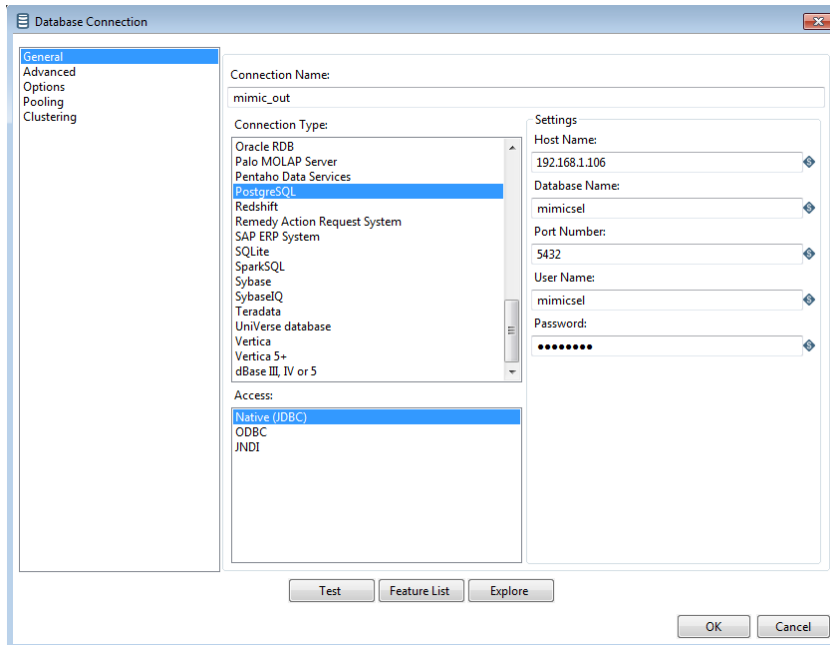
En estos procesos se han utilizado tres conexiones de BBDD que es necesario reconfigurar en base a la configuración concreta de cada entorno:



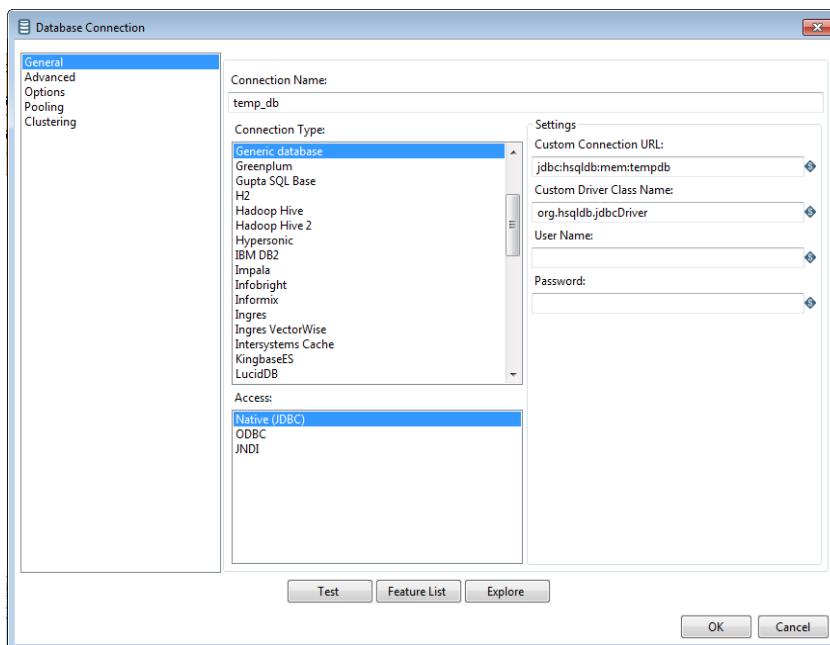
Donde **mimic_in** es la BBDD origen MIMIC-III



mimic_out es la BBDD destino MIMICSEL



Y **temp_bd** es la BBDD en memoria utilizada para almacenar tablas temporales

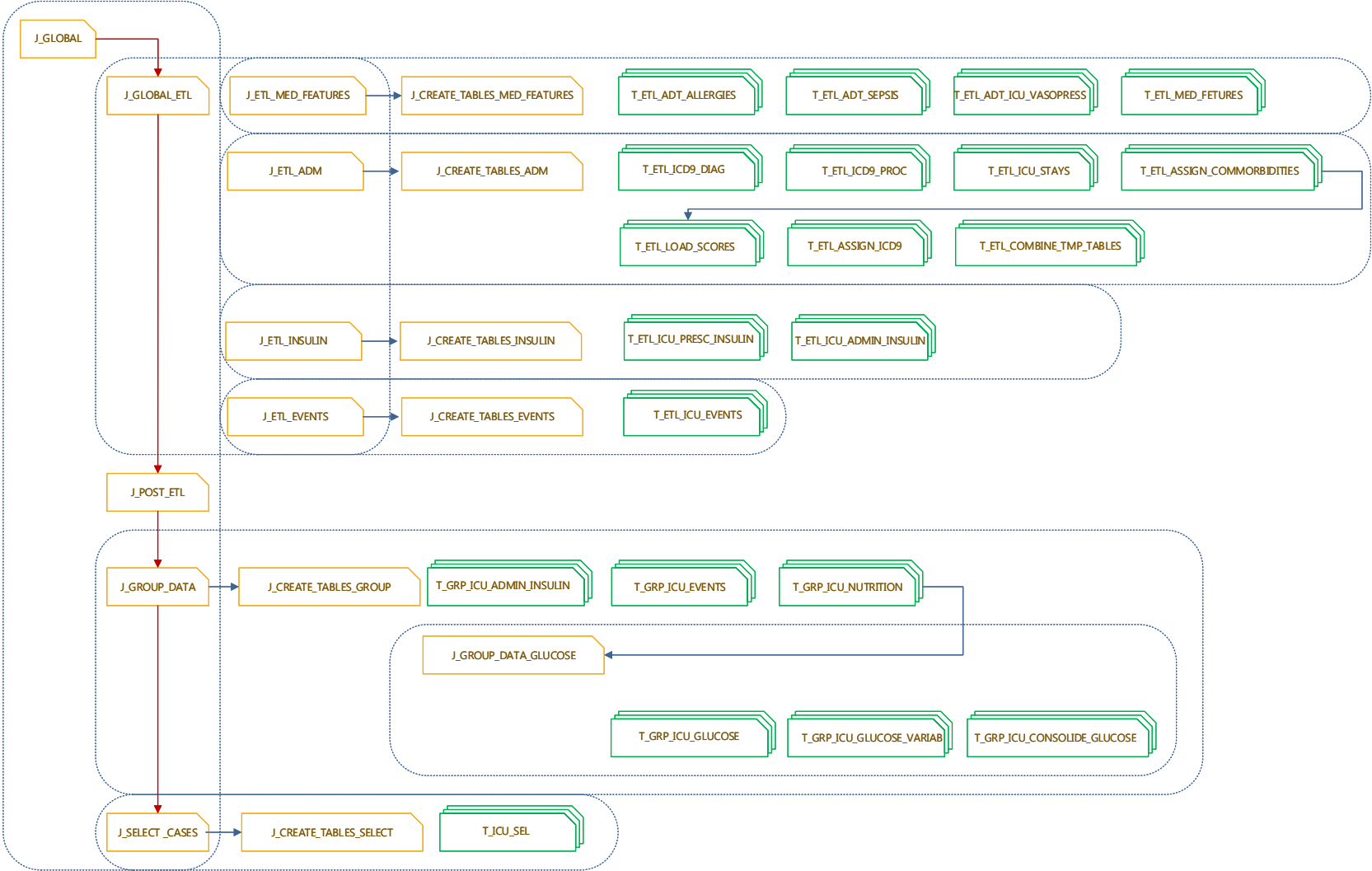


El proceso ETL está diseñado para crear las tablas necesarias en la base de datos MIMICSEL y utiliza tablas temporales en memoria cuando necesita guardar resultados parciales. Todo el proceso de extracción de datos, transformación y volcado de datos tarda unos 12 minutos y procesa millones de registros para obtener las tablas de destino.

A2-2. ESQUEMA GENERAL

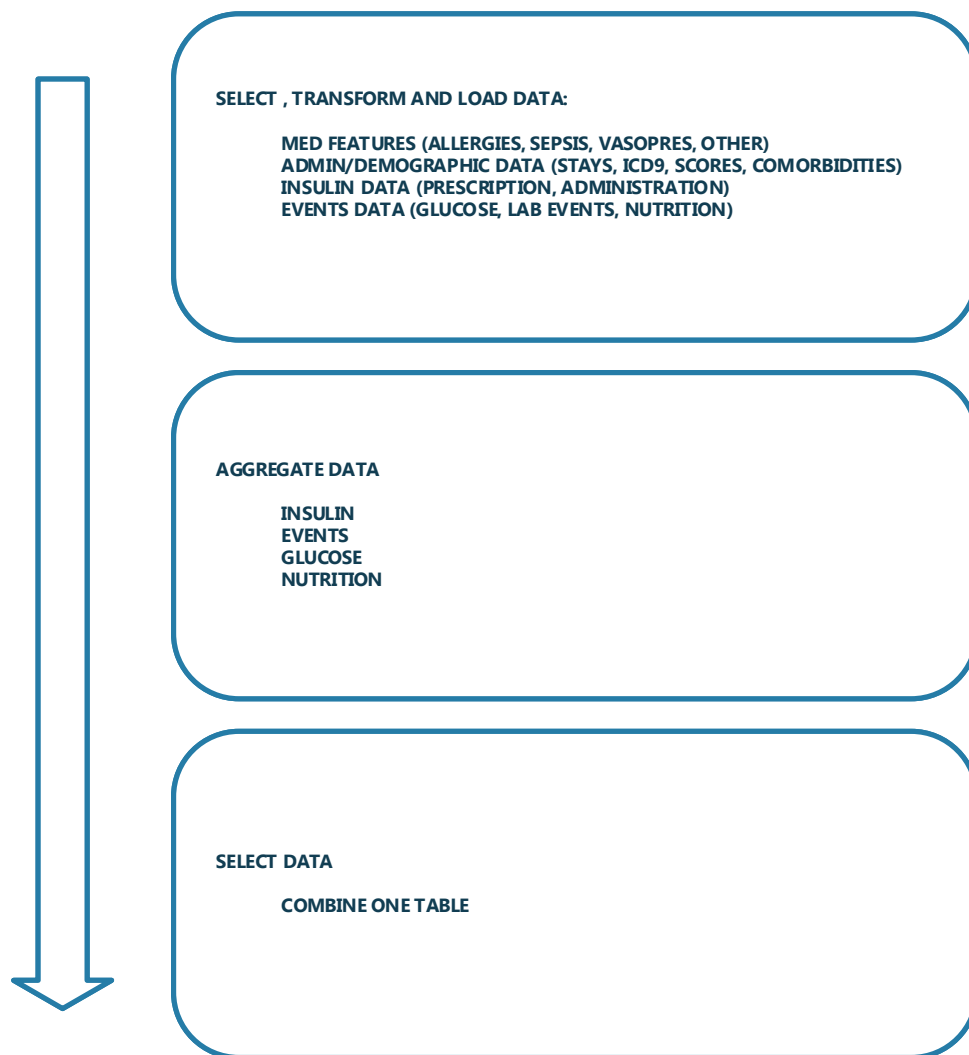
La jerarquía de trabajos y transformaciones, partiendo del trabajo inicial, se muestra el siguiente mapa:

MAP OF ETL JOBS AND TRANSFORMATIONS



El esquema general de las transformaciones ETL desarrolladas es el siguiente:

ETL

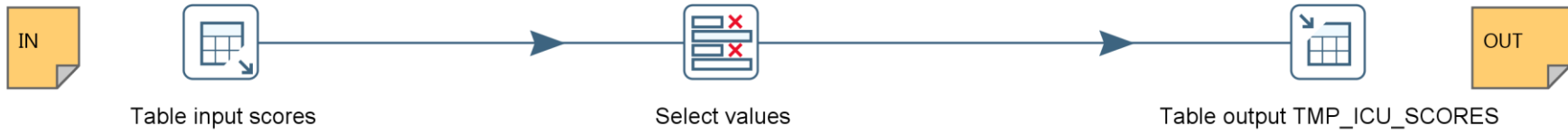


A continuación, se incluye la documentación detallada de las transformaciones y trabajos ETL.

Name: A2-3. T_ETL_LOAD_SCORES
Description: Carga los datos de las vistas de scores
Long description: SELECT ORIGEN

```
select icu.icustay_id icu_id
,apsiii, lods, mlods, oasis, qsofa,saps,sapsii,sirs, sofa
from icustays icu
join apsiii on apsiii.icustay_id=icu.icustay_id
join lods on lods.icustay_id=icu.icustay_id
join mlods on mlods.icustay_id=icu.icustay_id
join oasis on oasis.icustay_id=icu.icustay_id
join qsofa on qsofa.icustay_id=icu.icustay_id
join saps on saps.icustay_id=icu.icustay_id
join sapsii on sapsii.icustay_id=icu.icustay_id
join sofa on sofa.icustay_id=icu.icustay_id
join sirs on sirs.icustay_id=icu.icustay_id
```

ENTRADA: tabla icustays, vistas apsiii, lods, mlods, oasis, qsofa, saps, sapsii, sofa, sirs
SALIDA: tmp_icu_scores
PROCESO: obtiene los datos de las vistas de scores y los carga en una tabla temporal en memoria



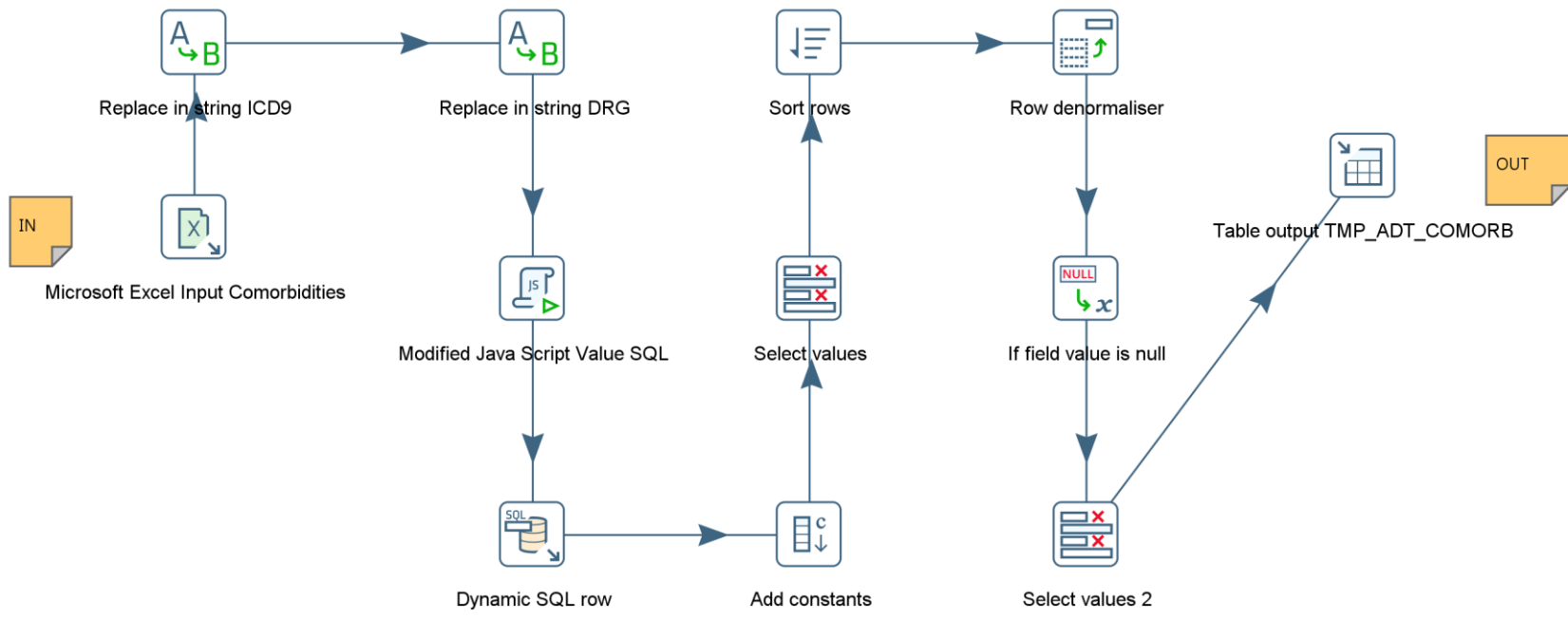
Name: A2-4. T_ETL_ASSIGN_COMORBIDITIES

Description: Asigna comorbilidades a episodios

Long description: Query dinámica

```
var SQL1 = 'select adm.hadm_id from icustays icu, admissions adm where exists (select null from
diagnoses_icd where hadm_id=adm.hadm_id and (' ;
var SQL2 = '))and not exists (select null from drgcodes where hadm_id=adm.hadm_id and (' ;
var SQL3 = ')) and adm.hadm_id=icu.hadm_id and adm.has_chartevents_data=1' ;
var SQL = SQL1 + ICD9_INCLUDE + SQL2 + DRG_EXCLUDE + SQL3;
```

ENTRADA: excel comorbidities
 SALIDA: tmp_adt_comorb
 PROCESO: Carga la excel de comorbididades y crea una consulta dinámica pra obtener para cada episodio de admisión si tiene o no una de las 30 comorbididades
 carga estas comorbididades en campos booleanos separados

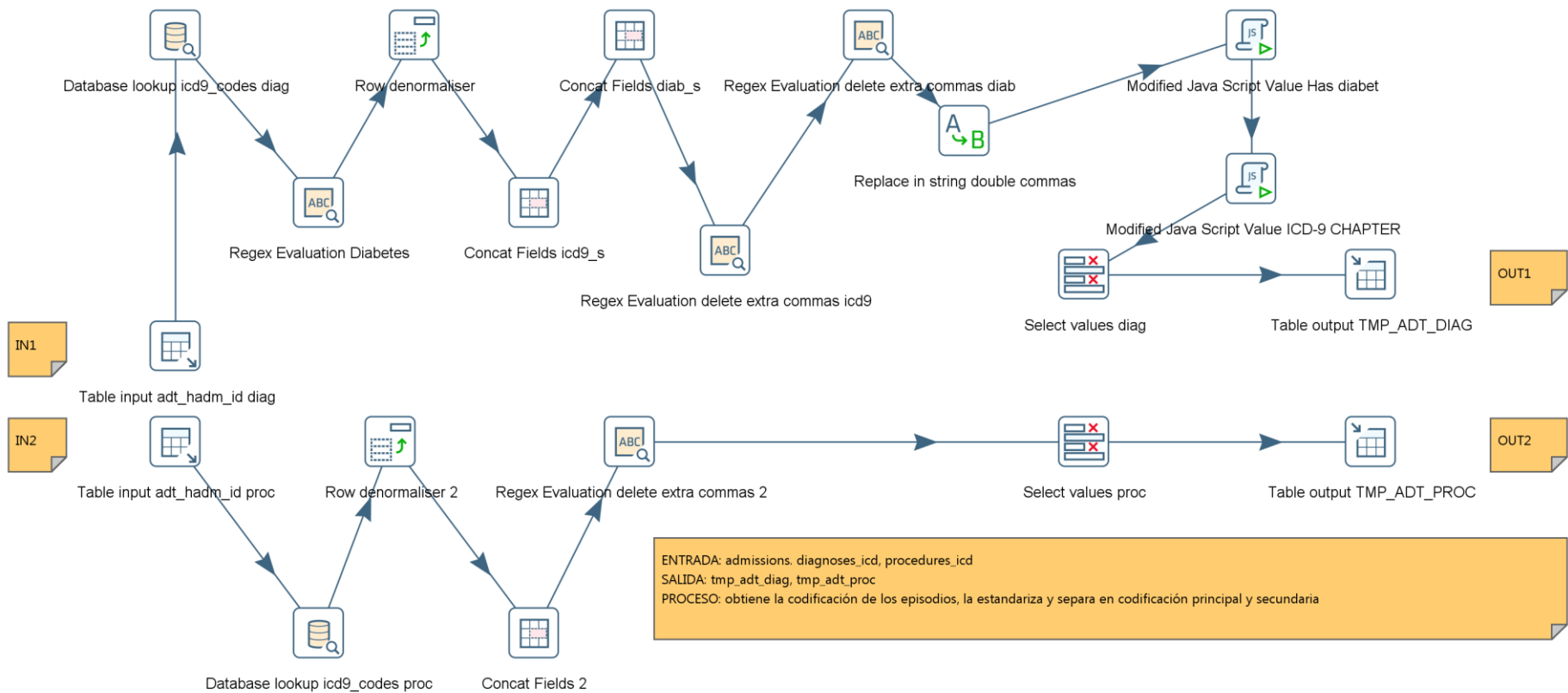


Name: A2-5. T_ETL_ASSIGN_ICD9
Description: Asigna procedimientos y diagnósticos a episodios
Long description: SELECT ORIGEN 1

```
select adt.hadm_id,diag.icd9_code,diag.seq_num  
from admissions adt  
join diagnoses_icd diag on adt.hadm_id=diag.hadm_id  
and adt.subject_id=diag.subject_id  
order by adt.hadm_id, diag.seq_num
```

SELECT ORIGEN 2

```
select adt.hadm_id,proc.icd9_code,proc.seq_num  
from admissions adt  
join procedures_icd proc on adt.hadm_id=proc.hadm_id  
and adt.subject_id=proc.subject_id  
order by adt.hadm_id, proc.seq_num
```



Name: A2-6. T_ETL_ICU_EVENTS
Description: Carga los datos de los eventos seleccionados (glucosa, analíticas y nutrición)
Long description: SELECT ORIGEN 1

```
select ce.charttime,ce.valuenum,di.itemid,di.label,icu.icustay_id,adt.hadm_id,adt.subject_id,
ce.resultstatus, ce.valueuom, icu.intime
from chartevents ce,d_items di,icustays icu, admissions adt
where ce.icustay_id=icu.icustay_id
and di.itemid=ce.itemid
and icu.hadm_id=adt.hadm_id
and adt.has_chartevents_data=1
and ce.charttime >= (icu.intime - '06:00:00'::interval hour)
and ce.error IS DISTINCT FROM 1
and di.itemid =811
AND ce.valuenum IS NOT NULL AND ce.valuenum > 0::double precision
```

SELECT ORIGEN 2

```
select ce.charttime,ce.valuenum,di.itemid,di.label,icu.icustay_id,adt.hadm_id,adt.subject_id,
ce.resultstatus, ce.valueuom, icu.intime
from chartevents ce,d_items di,icustays icu, admissions adt
where ce.icustay_id=icu.icustay_id
and di.itemid=ce.itemid
and icu.hadm_id=adt.hadm_id
and ce.charttime >= (icu.intime - '06:00:00'::interval hour)
and adt.has_chartevents_data=1
and ce.error IS DISTINCT FROM 1
and di.itemid =1529
AND ce.valuenum IS NOT NULL AND ce.valuenum > 0::double precision
```

SELECT ORIGEN 3

```
select ce.charttime,ce.valuenum,di.itemid,di.label,icu.icustay_id,adt.hadm_id,adt.subject_id,
ce.resultstatus,ce.valueuom, icu.intime
from chartevents ce,d_items di,icustays icu, admissions adt
where ce.icustay_id=icu.icustay_id
and di.itemid=ce.itemid
and icu.hadm_id=adt.hadm_id
and adt.has_chartevents_data=1
and ce.charttime >= (icu.intime - '06:00:00'::interval hour)
and ce.error IS DISTINCT FROM 1
and di.itemid in (807,3745,3744,225664,220621,226537)
AND ce.valuenum IS NOT NULL AND ce.valuenum > 0::double precision
```

SELECT ORIGEN 4

```
select
icu.icustay_id,icu.subject_id,icu.hadm_id,le.charttime,le.valuenum,le.valueuom,le.flag,le.itemid,dl.labe
l,
CASE
    WHEN le.itemid = 50802 THEN 'BASEEXCESS'::text
    WHEN le.itemid = 50811 THEN 'HEMOGLOBIN'::text
    WHEN le.itemid = 50813 THEN 'LACTATE'::text
    WHEN le.itemid = 50818 THEN 'PCO2'::text
    WHEN le.itemid = 50820 THEN 'PH'::text
    WHEN le.itemid = 50821 THEN 'PO2'::text
    WHEN le.itemid = 50882 THEN 'BICARBONATE'::text
    WHEN le.itemid = 50912 THEN 'CREATININE'::text
    WHEN le.itemid = 51222 THEN 'HEMOGLOBIN'::text
    WHEN le.itemid = 51300 THEN 'WHITEBLOOD'::text
```



```

        WHEN le.itemid = 51301 THEN 'WHITEBLOOD'::text
        ELSE NULL::text
    END as type,
    icu.intime
from icustays icu
inner JOIN labevents le ON le.subject_id = icu.subject_id AND le.hadm_id = icu.hadm_id
AND le.charttime >= (icu.intime - '06:00:00'::interval hour)
AND le.charttime <= icu.outtime
AND (le.itemid = ANY (ARRAY[50802,50811,50813,50818,50820,50821,50882,50912,51222,51300,51301]))
AND le.valuenum IS NOT NULL AND le.valuenum > 0::double precision
inner join admissions adt on icu.hadm_id=adt.hadm_id
and adt.has_chartevents_data=1
join d_labitems dl on dl.itemid=le.itemid

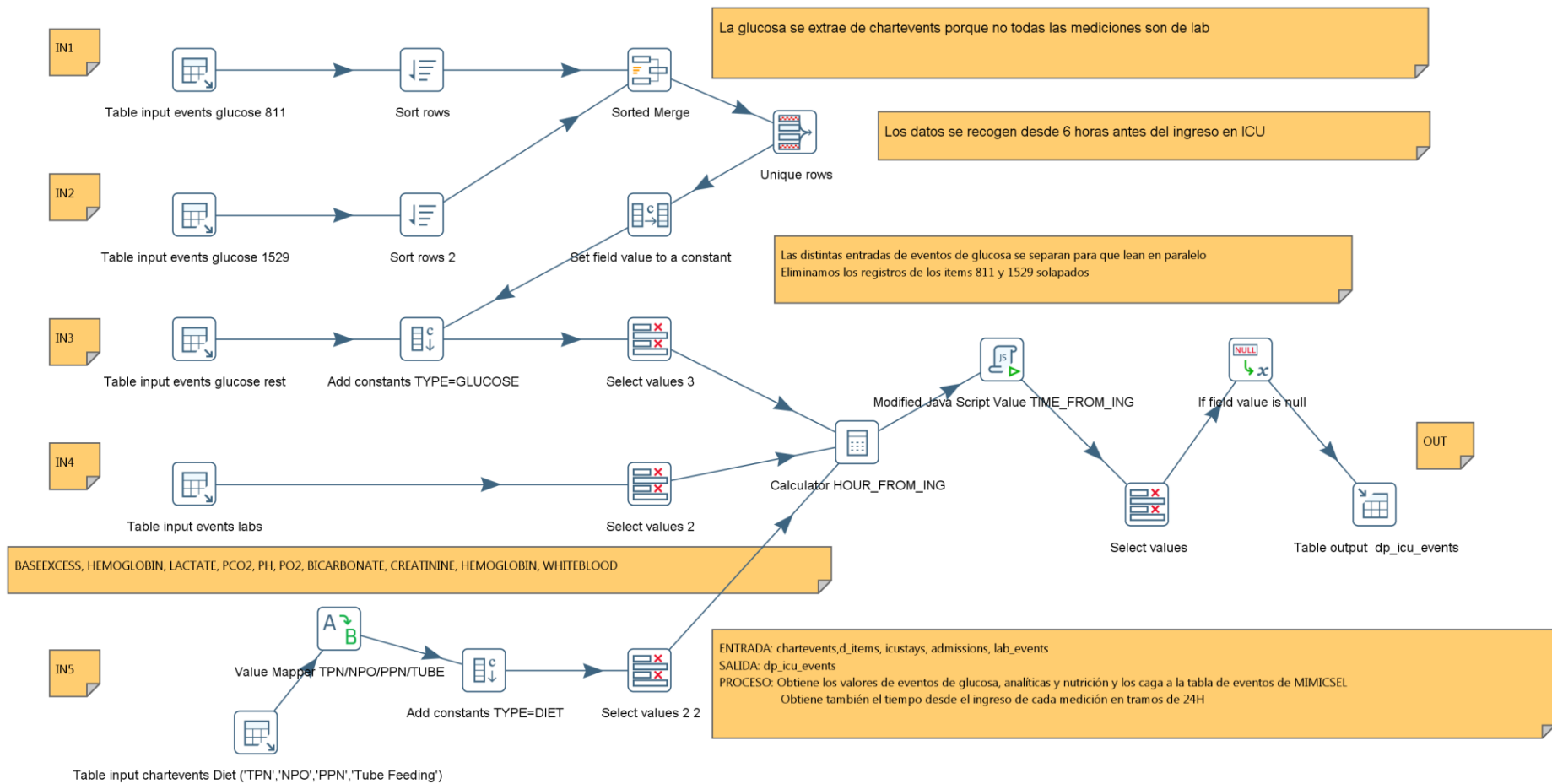
```

SELECT ORIGIN 5

```

select icu.icustay_id,icu.subject_id,icu.hadm_id,le.charttime,le.valueuom,
le.itemid,le.value as label,icu.intime
from icustays icu
inner JOIN chartevents le ON le.subject_id = icu.subject_id AND le.icustay_id = icu.icustay_id
AND le.itemid in (154,224001)
AND le.value in ('TPN','NPO','PPN','Tube Feeding')
join d_items d on le.itemid=d.itemid

```



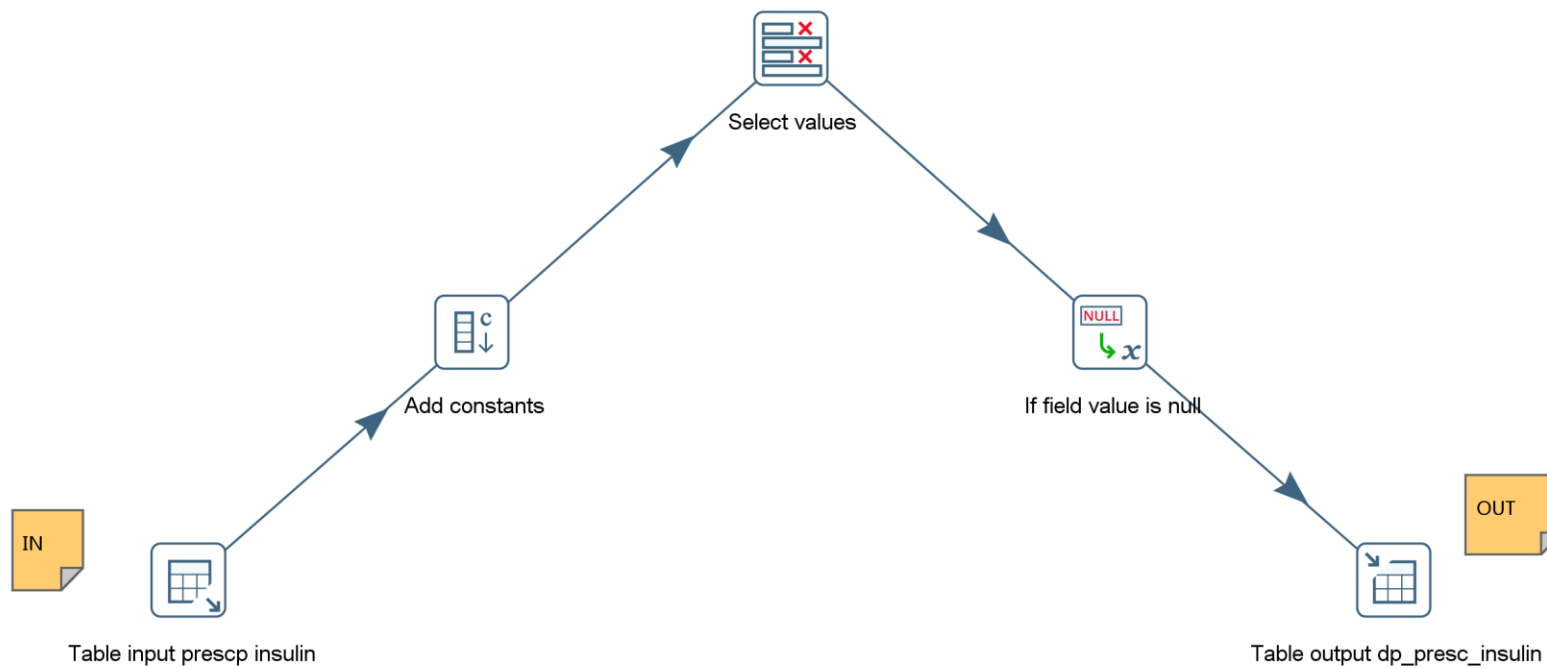
Name: A2-7. T_ETL_ICU_PRESC_INSULIN

Description: Carga las prescripciones de insulina

Long description: SELECT ORIGEN

```
with t_pres as(  
select pres.icustay_id,  
1 pres_insulin,  
case  
    when pres.prod_strength like 'Dummy%Sliding%' then 1  
    else 0  
end sliding_scale,  
case  
    when pres.route in ('SC','SCPUMP','SC') then 1  
    else 0  
end subcutan  
from prescriptions pres  
where pres.icustay_id is not null  
and (upper(pres.drug) like any ( array  
['%XULTOPHY%', '%NOVOLIN%', '%NOVOLOG%', '%TRESIBA%', '%LEVEMIR%', '%LANTUS%', '%BASAGLAR%', '%TOUJEI%', '%SOLIQ  
UA%', '%APIDRA%', '%AFREZZA%', '%HUMULIN%', '%HUMALOG%', '%INSULIN%']))  
or upper(pres.drug_name_generic) like any ( array  
['%XULTOPHY%', '%NOVOLIN%', '%NOVOLOG%', '%TRESIBA%', '%LEVEMIR%', '%LANTUS%', '%BASAGLAR%', '%TOUJEI%', '%SOLIQ  
UA%', '%APIDRA%', '%AFREZZA%', '%HUMULIN%', '%HUMALOG%', '%INSULIN%']))  
or upper(pres.drug_name_poe) like any ( array  
['%XULTOPHY%', '%NOVOLIN%', '%NOVOLOG%', '%TRESIBA%', '%LEVEMIR%', '%LANTUS%', '%BASAGLAR%', '%TOUJEI%', '%SOLIQ  
UA%', '%APIDRA%', '%AFREZZA%', '%HUMULIN%', '%HUMALOG%', '%INSULIN%']))  
)  
select count(t_pres.pres_insulin) as pres_n_insulin,  
case  
    when max(t_pres.sliding_scale)=1 then 'true'::bool
```

```
        else 'false'::bool
    end as pres_sliding_scale,
    case
        when max(t_pres.subcutan)=1 then 'true'::bool
        else 'false'::bool
    end as pres_subcutan,
    t_pres.icustay_id as icu_id,
    (select dbsource from icustays as icu where icu.icustay_id=t_pres.icustay_id) as icu_dbsource
from t_pres
group by t_pres.icustay_id
```



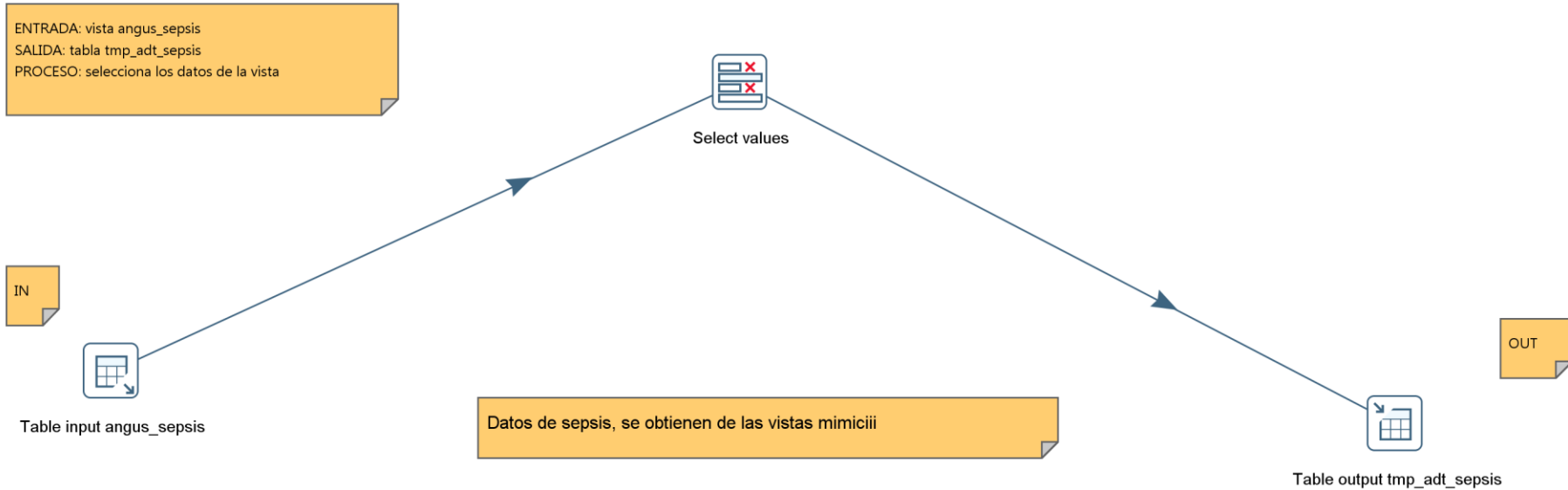
ENTRADA: prescriptions, icustays
 SALIDA: dp_presc_inculin
 PROCESO: carga los datos de las prescripciones de insulina en MIMICSEL

Name: A2-8. T_ETL_ADT_SEPSIS

Description: Carga los datos de sepsis

Long description: SELECT ORIGEN

```
select * from angus_sepsis
```



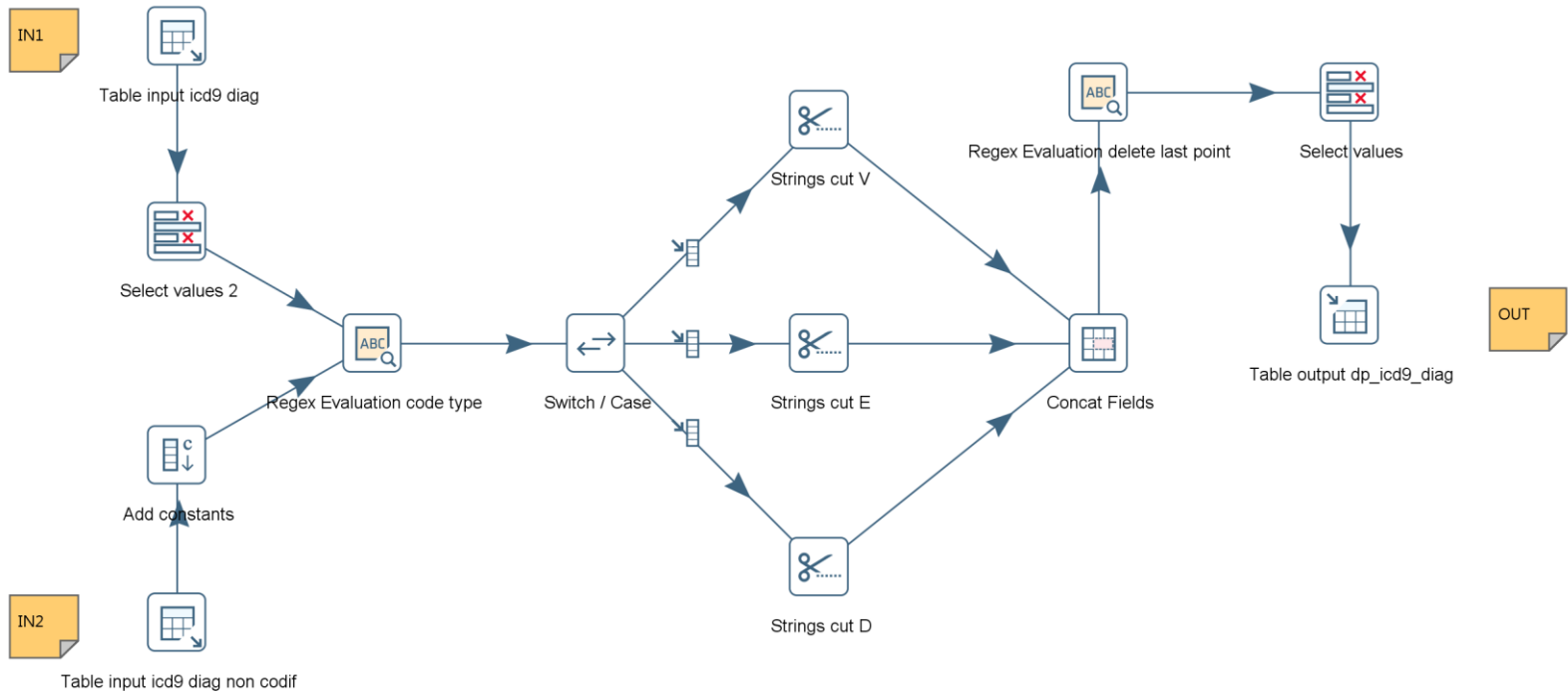
Name: **A2-9. T_ETL_ICD9_DIAG**
Description: Carga la tabla de diagnósticos CIE9
Long description: SELECT ORIGEN 1

```
SELECT * FROM d_icd_diagnoses
```

SELECT ORIGEN 2

```
select distinct(icd9_code) from diagnoses_icd diag  
where icd9_code is not null  
and not exists (select null from d_icd_diagnoses icd  
where icd.icd9_code=diag.icd9_code)
```

ENTRADA: tablas d_icd_diagnoses, diagnoses_icd
 SALIDA: tabla dp_icd9_diag
 PROCESO: selecciona los datos de diagnósticos CIE9, los clasifica en su tipo y convierte el código al estándar CIE9.
 Las codificaciones que ya no existen en la tabla maestra se marcan como NO CODIF



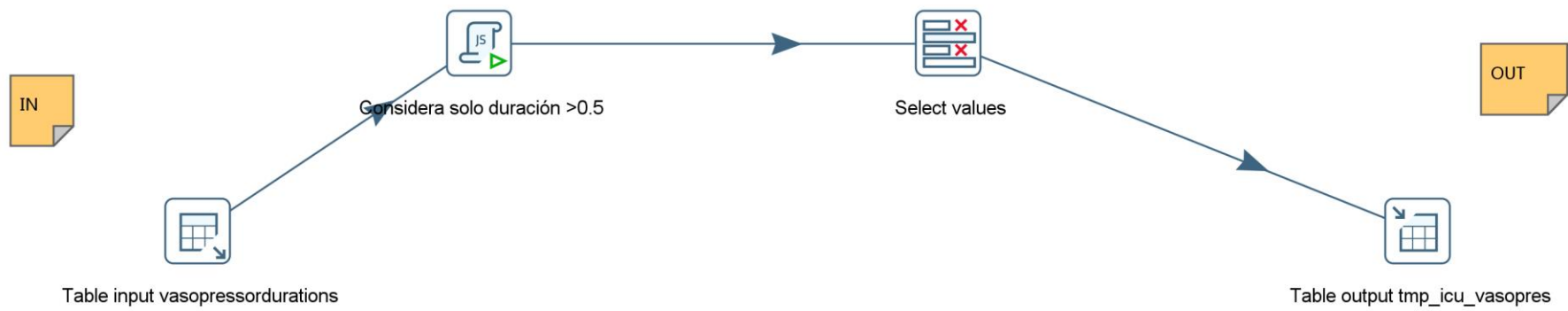
Name: A2-10.T_ETL_ICU_VASOPRESS

Description: Carga los datos de vasopresores

Long description: SELECT ORIGEN

```
select
vaso.icustay_id,
sum(round(extract(epoch from vaso.endtime::timestamp-vaso.starttime::timestamp)::numeric/3600,2))
TOTAL_VASO_DURATION,
sum(round(extract(epoch from least(vaso.endtime::timestamp,(icu.intime + interval '24
hours')::timestamp)
-
least(vaso.starttime::timestamp,(icu.intime + interval '24 hours')::timestamp))::numeric/3600,2))
FIRST24_VASO_DURATION
from vasopressordurations vaso
join icustays icu on vaso.icustay_id=icu.icustay_id
group by vaso.icustay_id,icu.intime
```

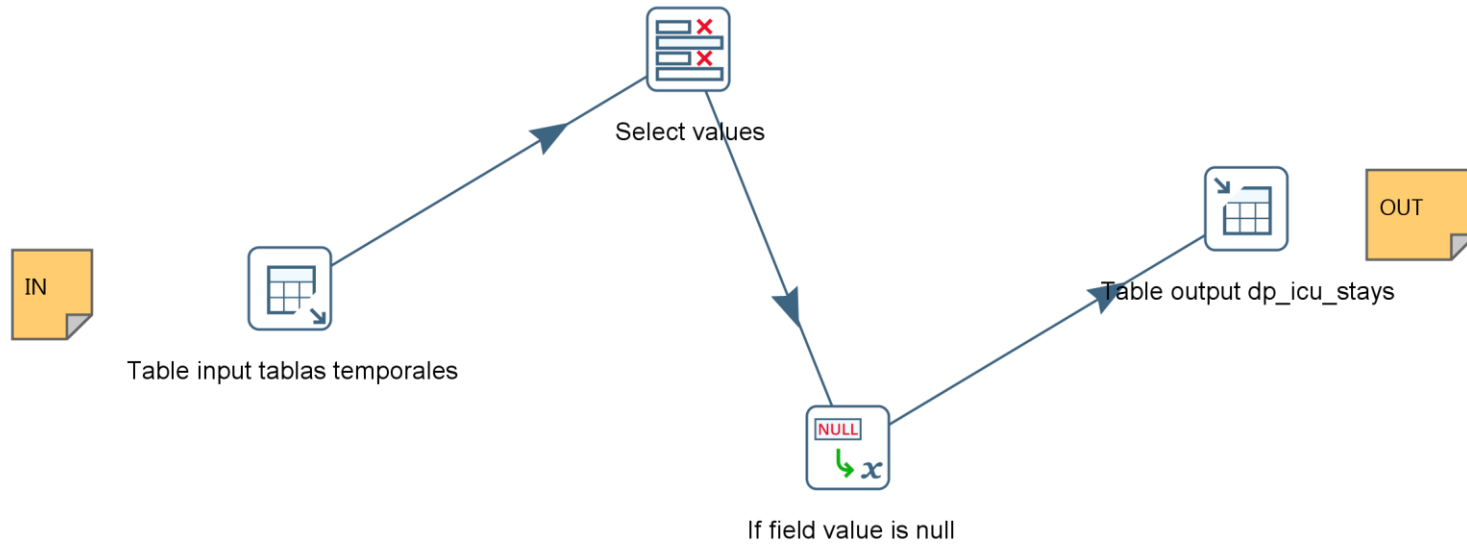
ENTRADA: vista modificada vasopressordurations_2, tabla icustays
SALIDA: tabla tmp_icu_vasopres
PROCESO: obtiene los datos de vasopresores, solo considera que se ha tratado al paciente con vasopresores si la duración es >0.5



Name: A2-11.T_ETL_COMBINE_TMP_TABLES
Description: Combina las tablas temporales en memoria y las carga en MIMICSEL
Long description: SELECT ORIGEN

```
select * from TMP_ICU_STAYS T1
left join TMP_ADT_COMORB T2 on T2.adt_id=T1.adt_id
left join TMP_ADT_DIAG T3 on T3.adt_id=T1.adt_id
left join TMP_ADT_PROC T4 on T4.adt_id=T1.adt_id
left join TMP_ICU_SCORES T5 on T5.icu_id=T1.icu_id
left join TMP_ICU_FEATURES T6 on T6.icu_id=T1.icu_id
left join TMP_ICU_VASOPRES T7 on T7.icu_id=T1.icu_id
left join TMP_ADT_ALLERGIES T8 on T8.adt_id=T1.adt_id
left join TMP_ADT_SEPSIS T9 on T9.adt_id=T1.adt_id
```

ENTRADA: TMP_ICU_STAYS , TMP_ADT_COMORB, TMP_ADT_DIAG, TMP_ADT_PROC, TMP_ICU_SCORES, TMP_ICU_FEATURES,
TMP_ICU_VASOPRES, TMP_ADT_ALLERGIES, TMP_ADT_SEPSIS
SALIDA: dp_icu_stays
PROCESO: combina todas las tablas en memoria en una única tabla en MIMICSEL



Name: A2-12.T_ETL_ICU_STAYS
Description: Carga los datos demográficos y administrativos de pacientes
Long description: SELECT ORIGEN

```
select p.subject_id as pat_id,  
admission_id as adm_id,  
icu.icustay_id as icu_id,  
admission.admittime as adm_intime,  
admission.dischtime as adm_outtime,  
p.gender as pat_gender,  
p.expire_flag=1 as pat_expire,  
date_part('years',admission.admittime::date) - date_part('years',p.dob::date)::numeric as adm_hos_age,  
date_part('years',p.dod) - date_part('years',p.dob)::numeric as pat_dod_age,  
(p.dod_hosp is not null) as pat_expire_hosp,  
dense_rank() OVER (PARTITION BY admission.subject_id ORDER BY admission.admittime) AS adm_stay_seq,  
CASE  
    WHEN dense_rank() OVER (PARTITION BY admission.subject_id ORDER BY admission.admittime) = 1 THEN 'true'::bool  
    ELSE 'false'::bool  
END AS adm_first_stay,  
(select count(*) from admissions adm2  
where adm.hadm_id<>adm2.hadm_id  
and admission.subject_id=adm2.subject_id)+1 as adm_tot_adm,  
(admission.dischtime::date-admission.admittime::date)::numeric as adm_hstays,  
coalesce(round(extract(epoch from admission.edouttime::timestamp-admission.edregtime::timestamp)::numeric/3600,2),0)  
as adm_estays,  
admission_type as adm_admission_type,  
admission_location as adm_admission_location,  
discharge_location as adm_discharge_location,  
exists (  
    select null from admissions adm2
```

```

        where adt2.subject_id=adt.subject_id
        and adt.hadm_id<>adt2.hadm_id
        and adt.admittime>adt2.disctime
        and adt2.disctime>=adt.admittime - interval '72 hours'
    ) as adt_readmit,
insurance as pat_insurance,
language as pat_language,
marital_status as pat_marital_status,
ethnicity as pat_ethnicity,
diagnosis as adt_diagnosis,
icu.dbsource as icu_dbsource,
icu.last_careunit as icu_last_care_unit,
round(extract(epoch from icu.intime::timestamp-adt.admittime::timestamp)::numeric/3600,2) as icu_intime,
round(extract(epoch from icu.outtime::timestamp-adt.admittime::timestamp)::numeric/3600,2) as
icu_outtime,
round(icu.los::numeric,4) as icu_los,
round((icu.outtime::date - icu.intime::date)::numeric, 4) AS icu_stay,
dense_rank() OVER (PARTITION BY icu.hadm_id ORDER BY icu.intime) AS icu_stay_seq,
CASE
    WHEN dense_rank() OVER (PARTITION BY icu.hadm_id ORDER BY icu.intime) = 1 THEN 'true'::bool
    ELSE 'false'::bool
END AS icu_first_stay,
CASE
    WHEN dense_rank() OVER (PARTITION BY icu.hadm_id ORDER BY icu.intime desc) = 1 THEN 'true'::bool
    ELSE 'false'::bool
END AS icu_last_stay,
(select count(*) from icustays icu2
    where icu.icustay_id<>icu2.icustay_id
    and icu.hadm_id=icu2.hadm_id)+1 as icu_tot_icu,
(select curr_service from

```

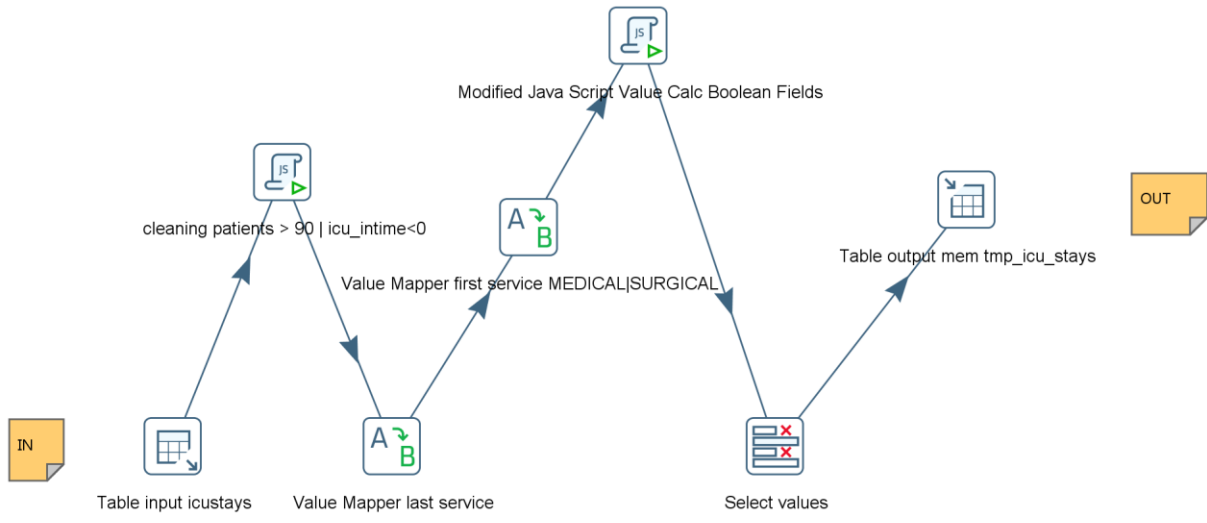
```

(select curr_service,row_number() over (partition by hadm_id order by transfertime asc)
  from services where hadm_id=adt.hadm_id) AS r
  where r.row_number=1) as icu_first_service,
(select curr_service from
  (select curr_service,row_number() over (partition by hadm_id order by transfertime desc)
    from services where hadm_id=adt.hadm_id) AS r
    where r.row_number=1) as icu_last_service,
(select count(*) from
  transfers tran where tran.hadm_id=adt.hadm_id) as adt_ctransfers,
round(he.height::numeric,2) as first_day_height,
round(we.weight::numeric,2) as first_day_weight,
vi.glucose_min as first_day_glucose_min,
vi.glucose_max as first_day_glucose_max,
round(vi.glucose_mean::numeric,2) as first_day_glucose_mean,
case
  when icu.outtime>=p.dod then true
  else false
end icu_dead,
case
  when dod_hosp is not null then greatest(0,date(dod_hosp::timestamp)-date(icu.outtime::timestamp))
  else date(dod_ssn::timestamp)-date(icu.outtime::timestamp)
end
icu_dead_days,
case
  when icu.dbsource='carevue' and exists (
    select null from inputevents_cv cv where cv.icustay_id=icu.icustay_id
    and cv.itemid in (30045,30100))
  then true
  when icu.dbsource='metavision' and exists (
    select null from inputevents_mv mv where mv.icustay_id=icu.icustay_id

```

```
        and mv.itemid between 223257 and 223262)
    then true
    else false
end
icu_has_admit_insulin
FROM icustays icu
JOIN admissions adt ON icu.hadm_id = adt.hadm_id
JOIN patients p ON icu.subject_id = p.subject_id
left join heightangusy he on icu.icustay_id=he.icustay_id
left join weightfirstday we on icu.icustay_id=we.icustay_id
left join vitalsfirstday vi on icu.icustay_id=vi.icustay_id
WHERE adt.has_chartevents_data = 1
and icu.outtime is not null
```


IMPORTANT: check that temp_db support boolean datatype



ENTRADA: tablas icustays, admissions, patients, vistas heightfirstday, weightfirstday, vitalsfirstday
SALIDA: tmp_icu_stays
PROCESO: carga los datos demográficos y administrativos de pacientes, la mayor parte de la lógica se realiza en la query

Filtramos por chartevents
Filtramos también 10 registros de ICU sin fecha fin pero con fecha fin del episodio ADT

Name: A2-13.T_ETL_ICD9_PROC
Description: Carga los datos de procedimientos CIE9
Long description: SELECT ORIGEN 1

```
SELECT * FROM d_icd_procedures
```

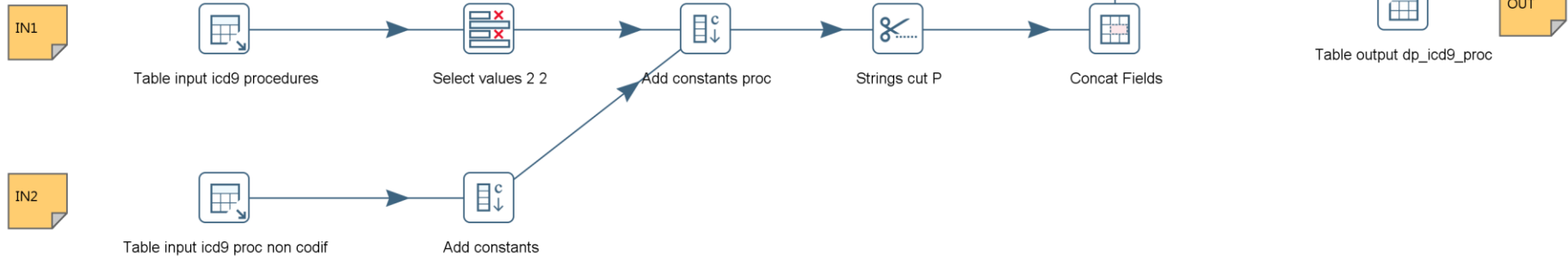
SELECT ORIGEN 2

```
select distinct(icd9_code) from procedures_icd proc  
where icd9_code is not null  
and not exists (select null from d_icd_procedures icd  
where icd.icd9_code=proc.icd9_code)
```

ENTRADA: tablas d_icd_procedures, procedures_icd

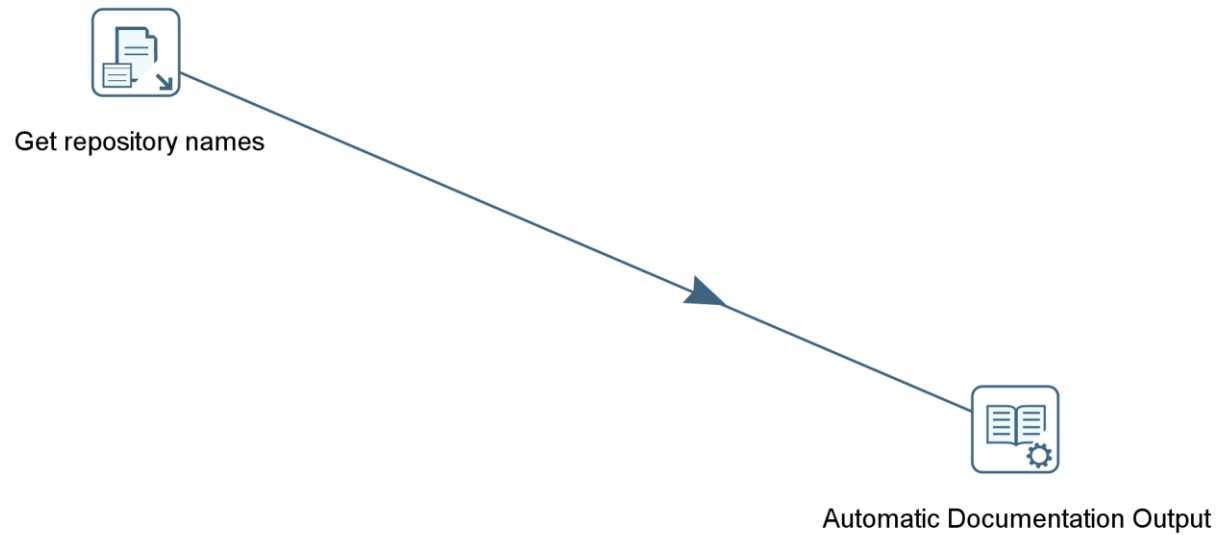
SALIDA: tabla dp_icd9_proc

PROCESO: selecciona los datos de procedimientos CIE9, los clasifica en su tipo y convierte el código al estándar CIE9. Las codificaciones que ya no existen en la tabla maestra se marcan como NO CODIF



Name: A2-14.T_DOC
Description: Trabajos de Documentación
Long description: Documenta el resto de trabajos y transformaciones

Modificar la extensión del fichero en función del tipo de salida



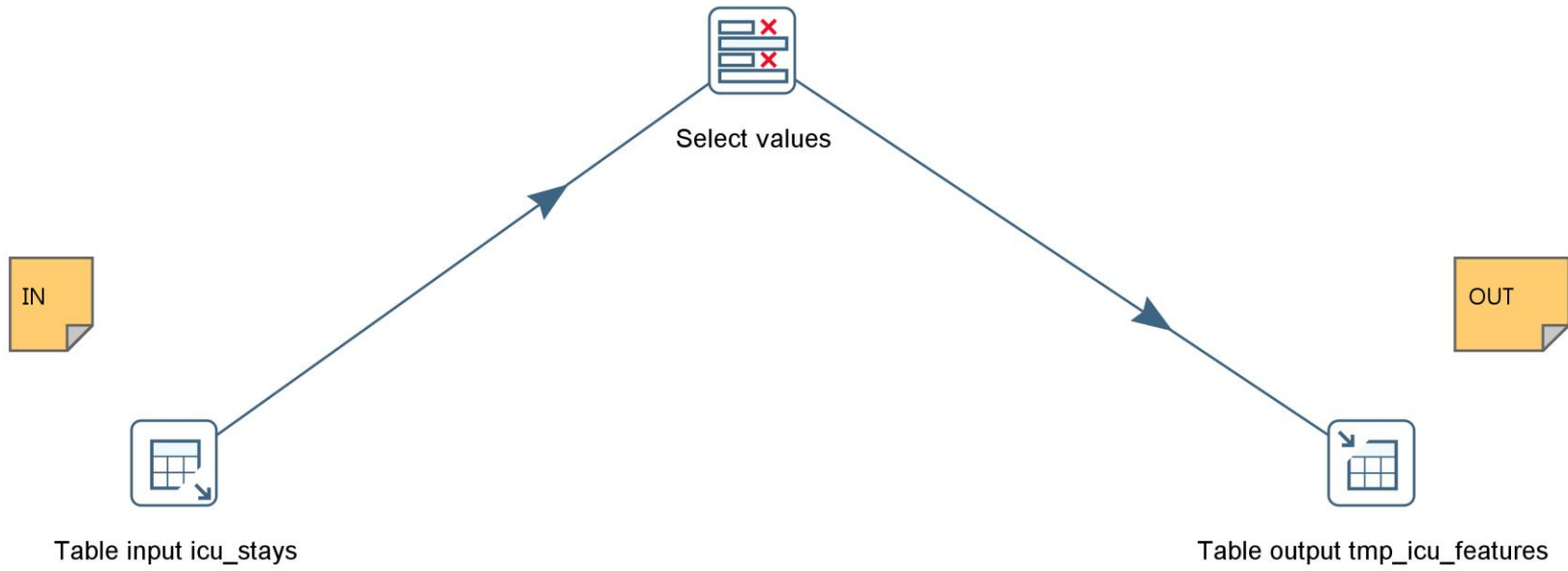
Name: A2-15.T_ETL_MED_FEATURES

Description: Carga varios datos clínicos

Long description: SELECT ORIGEN

```
select icu.icustay_id,
exists (
  select null from rrt where rrt=1 and icustay_id=icu.icustay_id
) HAS_DIALYSIS,
exists (
  select null from rrtfirstday where rrt=1 and icustay_id=icu.icustay_id
) HAS_24H_DIALYSIS,
exists (
  select null from ventdurations vd where vd.icustay_id =icu.icustay_id
) HAS_VENT,
exists (
  select null from ventdurations vd where vd.icustay_id=icu.icustay_id
  and (vd.starttime <= icu.intime AND vd.endtime >= icu.intime
  or vd.starttime >= icu.intime AND vd.starttime <= (icu.intime + '1 day'::interval day))
) HAS_24H_VENT,
exists (
  select null from prescriptions pres where (upper(drug)
  like any array['%CORTI%', '%METHASO%', '%PREDNIS%'])
  or upper(drug_name_generic) like any (array['%CORTI%', '%METHASO%', '%PREDNIS%'])
  or upper(drug_name_poe) like any (array['%CORTI%', '%METHASO%', '%PREDNIS%'])
  and icu.icustay_id=pres.icustay_id
) HAS_ESTEROID_PRESC
from icustays icu
```

ENTRADA: tablas icustays, prescriptions, vistas rtt, rttfirstday, ventdurations
SALIDA: tabla tmp_adt_allergies
PROCESO: selecciona los campos de las tablas y vistas de origen



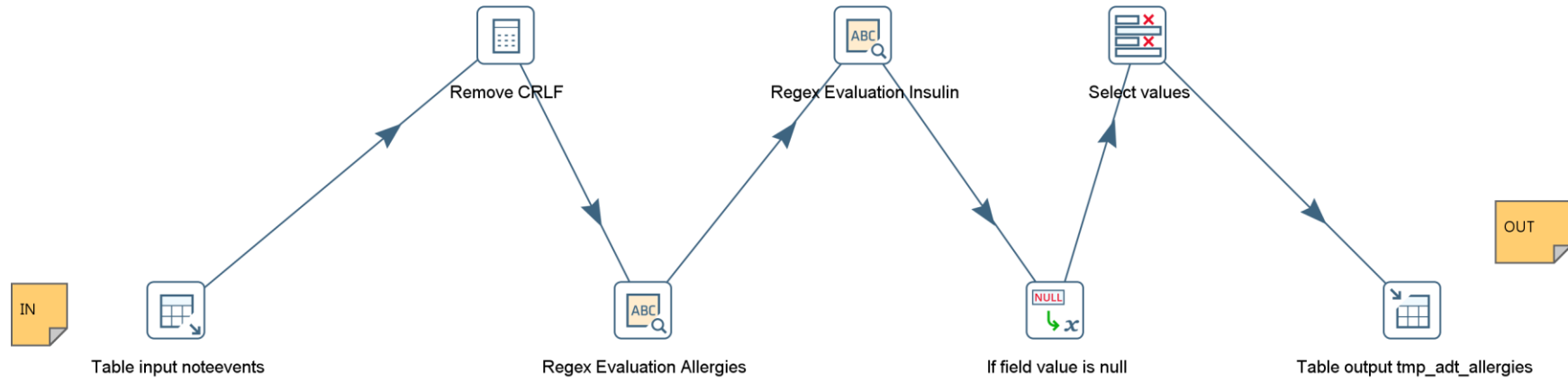
Name: A2-16.T_ETL_ADT_ALLERGIES

Description: Obtiene los datos de alergias

Long description: SELECT ORIGEN

```
select hadm_id,text from noteevents  
where category='Discharge summary' and text like '%Allergies:%' and description='Report'
```

ENTRADA: tabla noteevents
SALIDA: tabla tmp_adt_allergies
PROCESO: obtienen los datos de alergias y de alergia a insulina mediante búsqueda de patrones en texto



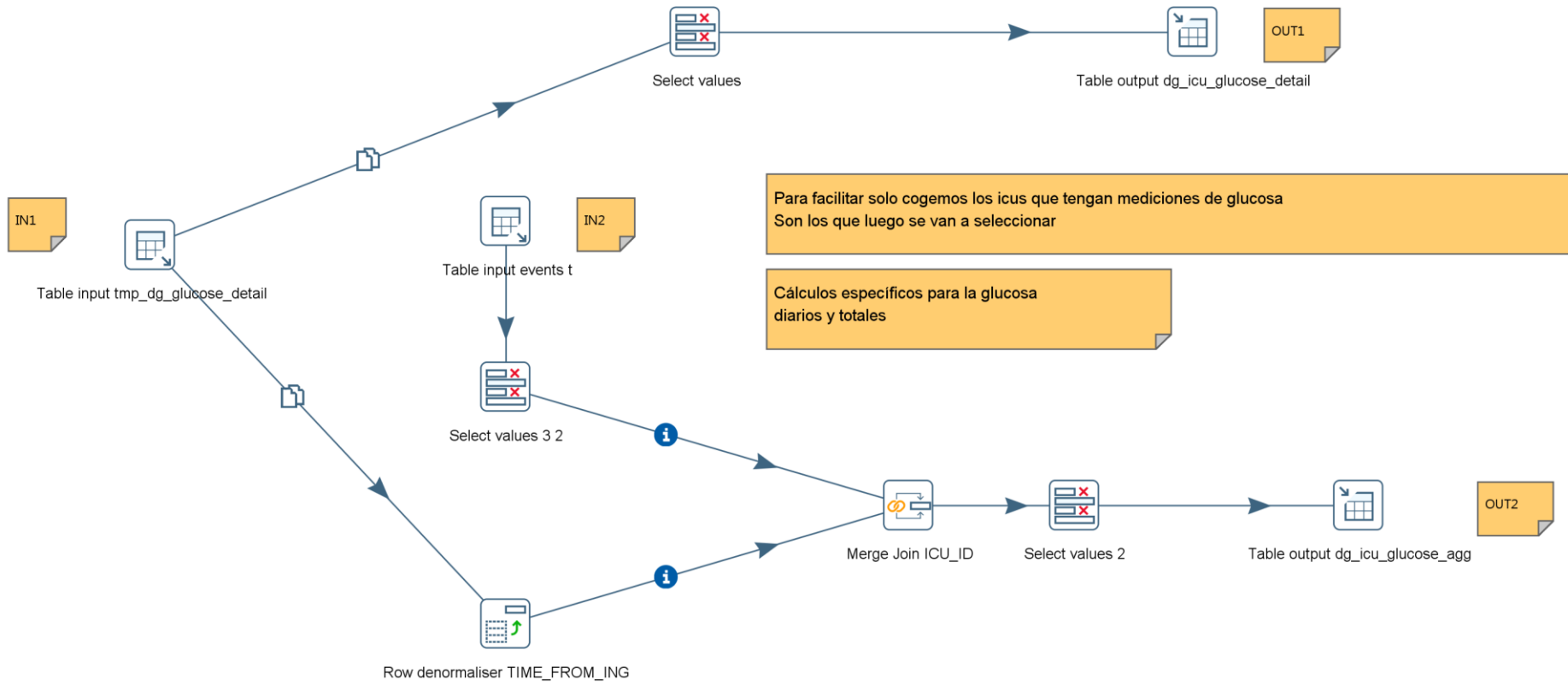
Name: A2-17.T_GRP_ICU_CONSOLIDE_GLUCOSE
Description: Consolida los datos de las tablas temporales de glucosa en las tablas de MIMICSEL
Long description: SELECT ORIGEN 1

```
select *  
from tmp_dg_glucose_detail_s1 tmp1, tmp_dg_glucose_detail_s2 tmp2  
where tmp1.icu_id=tmp2.icu_id  
and tmp1.time_from_ing=tmp2.time_from_ing  
order by tmp1.icu_id,tmp1.time_from_ing
```

SELECT ORIGEN 2

```
select tmp1.*,tmp2.*  
from tmp_dg_glucose_detail_t1 tmp1, tmp_dg_glucose_detail_t2 tmp2  
where tmp1.icu_id=tmp2.icu_id  
order by tmp1.icu_id
```

ENTRADA: tmp_dg_glucose_detail_s1, tmp_dg_glucose_detail_s2, tmp_dg_glucose_detail_t1, tmp_dg_glucose_detail_t2
 SALIDA: dg_icu_glucose_detail, dg_icu_glucose_agg
 PROCESO: agrupa los datos por tramos de 24H/48H/72H + los totales por icu a partir de las tablas temporales en memoria y los guarda en las tablas de detalle y agregados de glucosa



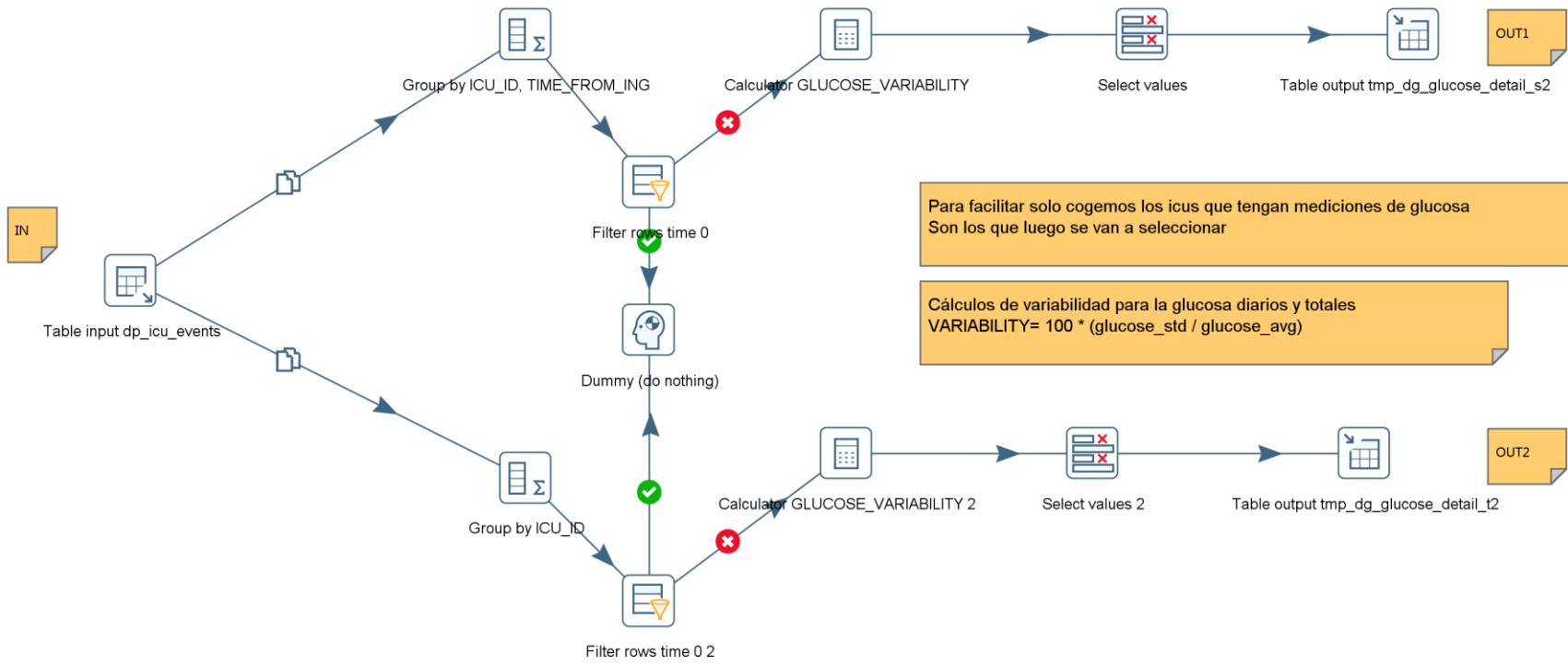
Name: A2-18.T_GRP_ICU_GLUCOSE_VARIAB

Description: Cálculo de la variabilidad de la glucosa

Long description: SELECT ORIGEN

```
select icu_id, time_from_ing,event_value,1 as n
from dp_icu_events
where event_type='GLUCOSE'
order by icu_id,time_from_ing,event_time
```

ENTRADA: dp_icu_events
 SALIDA: tmp_dg_glucose_detail_s1, tmp_dg_glucose_detail_s2
 PROCESO: obtiene los datos de glucosa de la tabla intermedia de eventos y calcula la variabilidad por día y por ICU



Name: A2-19.T_GRP_ICU_GLUCOSE
Description: Calcula los agrupadores de los datos de glucosa
Long description: SELECT ORIGEN

```
with inner_q as
(select
gev.icu_id, gev.time_from_ing,gev.event_time,
coalesce(lag(gev.event_time) over (partition by gev.icu_id,gev.event_type order by gev.event_time desc),
icu.adt_intime + interval '1h'*icu.icu_outtime) event_time_next,
round(extract(epoch from (coalesce(lag(gev.event_time) over (partition by gev.icu_id,gev.event_type
order by gev.event_time desc),
icu.adt_intime + interval '1h'*icu.icu_outtime))-gev.event_time::timestamp)::numeric/3600,3) diff_time,
gev.event_value,
case
when gev.event_value<=45 then 1
else 0
end hipo_critical,
case
when gev.event_value>45 and gev.event_value<=60 then 1
else 0
end hipo_moderate,
case
when gev.event_value>150 then 1
else 0
end hyper_gluce,
case
when gev.event_value>110 and gev.event_value<=150 then 1
else 0
end range_protocol,
case
```

```

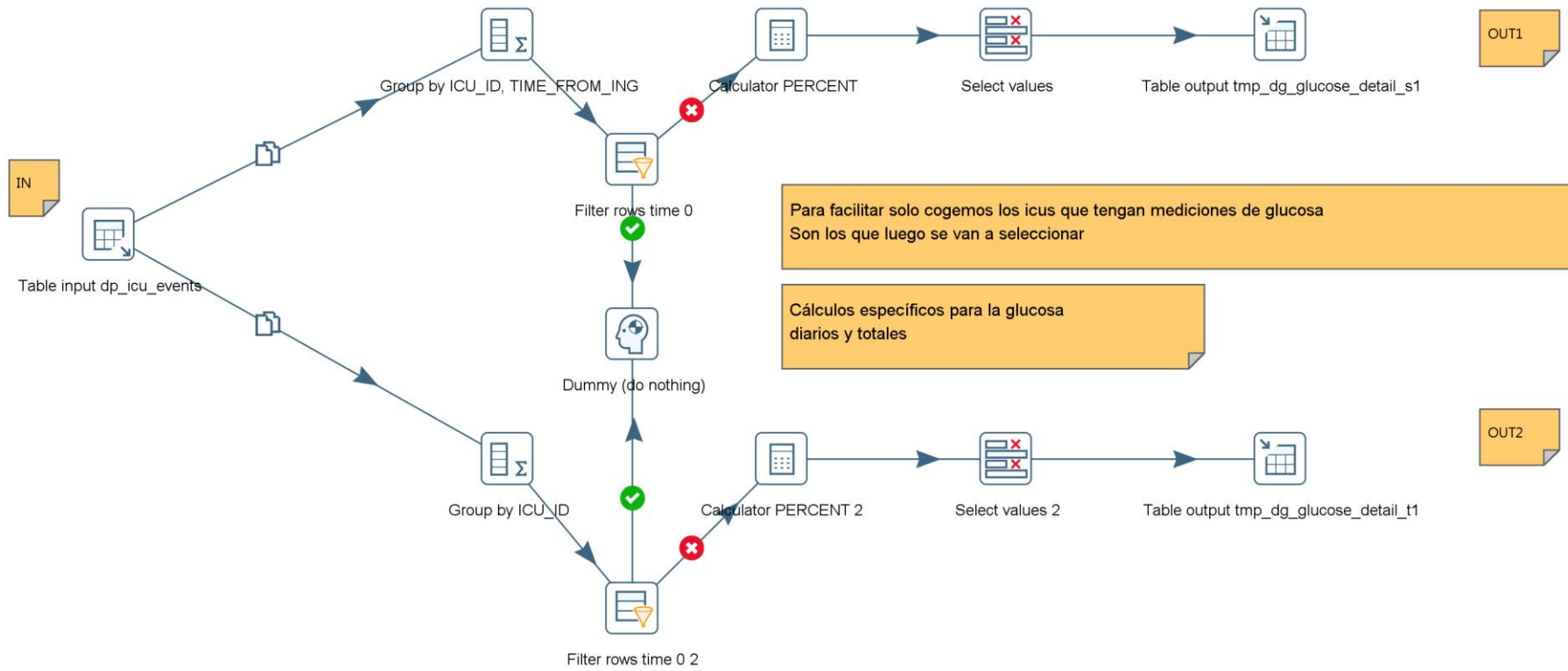
        when gev.event_value>80 and gev.event_value<=150 then 1
        else 0
    end range_clinic
from dp_icu_events gev
join dp_icu_stays icu on icu.icu_id=gev.icu_id
and icu.icu_has_event_glucose is true
and gev.event_type='GLUCOSE'
)
select icu_id, time_from_ing,diff_time,event_time,event_time_next,
hipo_critical,
case
    when hipo_critical=1 then coalesce(diff_time,0)
    else 0
end
time_hipo_critical,
hipo_moderate,
case
    when hipo_moderate=1 then coalesce(diff_time,0)
    else 0
end
time_hipo_moderate,
hyper_glucose,
case
    when hyper_glucose=1 then coalesce(diff_time,0)
    else 0
end
time_hyper_glucose,
range_protocol,
case
    when range_protocol=1 then coalesce(diff_time,0)

```

```
        else 0
    end

    time_range_protocol,
    range_clinic,
    case
        when range_clinic=1 then coalesce(diff_time,0)
        else 0
    end
    time_range_clinic
from inner_q
order by icu_id,time_from_ing,event_time
```

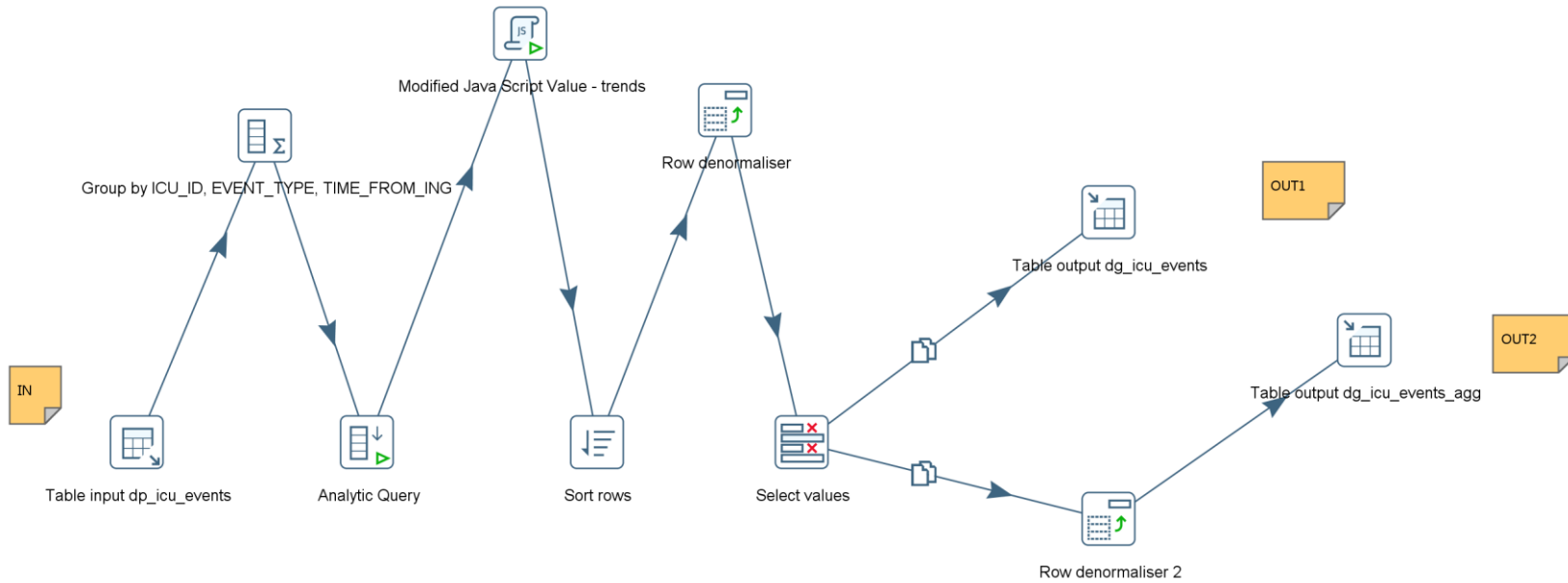
ENTRADA: dp_icu_events
 SALIDA: tmp_dg_glucose_detail_s1, tmp_dg_glucose_detail_t1
 PROCESO: obtiene los datos de glucosa de la tabla intermedia de eventos y los agrupa por 24H/48H/72H + ICU, se guardan en tablas temporales que se consolidarán posteriormente



Name: A2-20.T_GRP_ICU_EVENTS
Description: Cálculo de los agrupadores por ICU y tramo de 24H de los datos de eventos
Long description: SELECT ORIGEN

```
select eve.*  
from dp_icu_events eve  
join dp_icu_stays icu on icu.icu_id=eve.icu_id  
and icu.icu_has_event_glucose is true  
order by eve.icu_id,eve.event_type,eve.time_from_ing
```

ENTRADA: dp_icu_events
 SALIDA: dg_icu_events, dg_icu_events_agg
 PROCESO: obtiene los datos de dp_icu_events, calcula los agrupadores min, max, avg, std, median, number, first non-null por tramo de ingreso 24H748H/72H y por icu

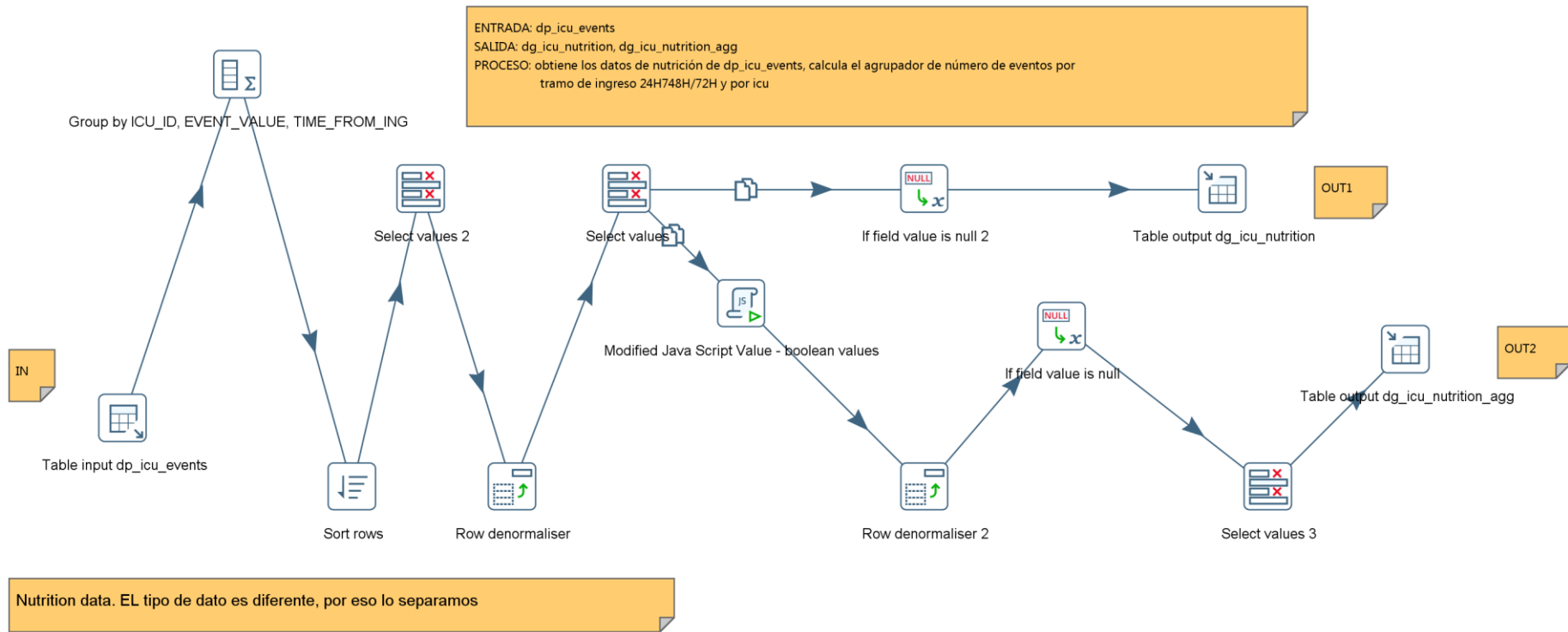


Para facilitar solo cogemos los icus que tengan mediciones de glucosa
 Son los que luego se van a seleccionar

Los datos específicos de glucosa se calculan aparte,
 es más sencillo calcularlos con SQL
 Y necesitamos todos los datos de la estancia

Name: A2-21.T_GRP_ICU_NUTRITION
Description: Datos de nutrición, agrupados por ICU y por tramo de 24H
Long description: SELECT ORIGEN

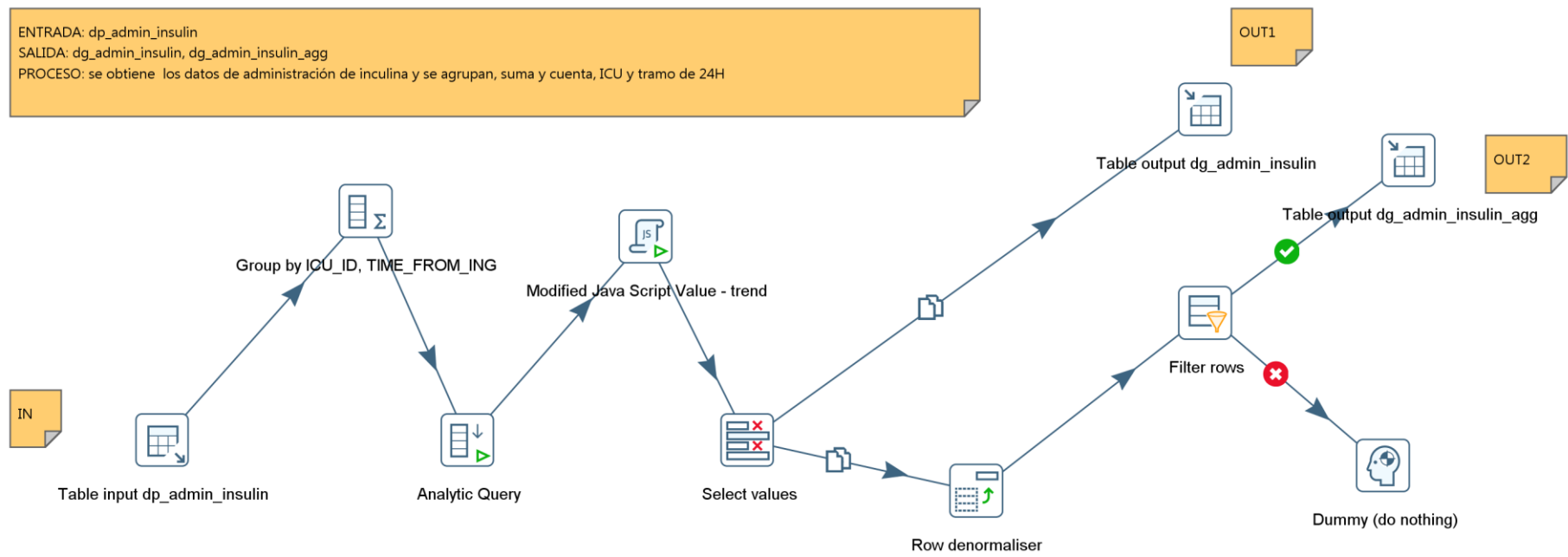
```
select eve.*  
from dp_icu_events eve  
where event_type='DIET'  
order by eve.icu_id,eve.event_value,eve.time_from_ing
```



Name: A2-22.T_GRP_ICU_ADMIN_INSULIN
Description: Agrupa los datos de administración de insulina por ICU y tramo de 24H
Long description: SELECT ORIGEN

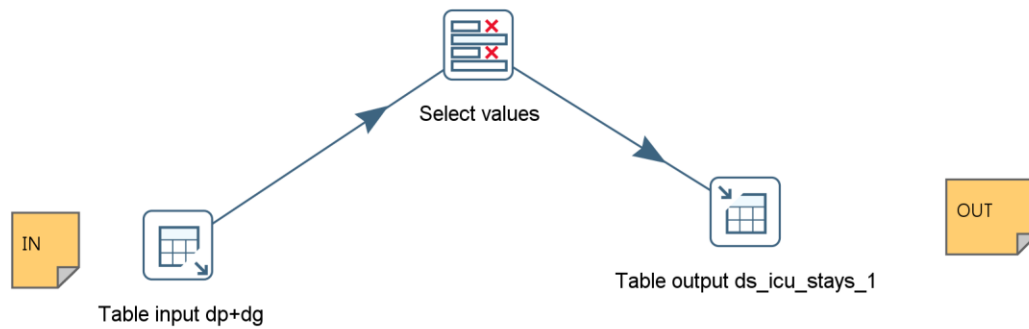
```
select * from dp_admin_insulin  
order by icu_id,time_from_ing, starttime
```

ENTRADA: dp_admin_insulin
SALIDA: dg_admin_insulin, dg_admin_insulin_agg
PROCESO: se obtiene los datos de administración de insulina y se agrupan, suma y cuenta, ICU y tramo de 24H



Name: A2-23.T_SEL_ICU
Description: Trabajo que obtiene los datos de las tablas de MIMICSEL y crea la tabla para análisis con los criterios indicados
Long description: SELECT ORIGEN

```
select *
from dp_icu_stays icu
left join dp_presc_insulin pres on icu.icu_id=pres.icu_id
left join dg_admin_insulin_agg admin on icu.icu_id=admin.icu_id
left join dg_icu_events_agg event on icu.icu_id=event.icu_id
left join dg_icu_glucose_agg gluco on icu.icu_id=gluco.icu_id
left join dg_icu_nutrition_agg nut on icu.icu_id=nut.icu_id
where icu.pat_adult=true
and icu.icu_los>=2
and gluco.glucose_value_n_24h>=3
and gluco.glucose_value_n_48h>=3
and icu.icd9d_m != '-'
```



SELECCIÓN DE CASOS

DS_ICU_STAYS_1

- * CON DIAGNOSTICO PPAL AL MENOS
- * ADULTO
- * > 3 MEDIDAS GLUCOSA 24+48
- * > 2 DIAS STAY ICU

ENTRADA: dp_icu_stays icu, dp_presc_insulin, dg_admin_insulin_agg, dg_icu_events_agg, dg_icu_glucose_agg gluco, dg_icu_nutrition_agg

SALIDA: ds_icu_stays_1

PROCESO: selecciona de las diferentes tablas de MIMICSEL los datos según los criterios indicados

Name: A2-24.T_ETL_ICU_ADMIN_INSULIN
Description: Carga de los datos de administración de insulina
Long description: SELECT ORIGEN 1

```
select ie.icustay_id,ie.itemid,de.label,  
ie.charttime-time '01:00' starttime,  
ie.charttime endtime,icu.intime as icu_intime,  
case  
    when ie.amount is null then 0  
    else ie.amount  
end amount,  
(ie.charttime-time '01:00')::timestamp::date startdate,  
charttime::timestamp::date enddate,  
originalroute,  
'U' amountuom,  
'carevue' dbsource  
from inputevents_cv ie, icustays icu, admissions adt, d_items de  
where de.itemid in (30045,30100)  
and de.itemid=ie.itemid  
and ie.icustay_id=icu.icustay_id  
and icu.hadm_id=adt.hadm_id  
and adt.has_chartevents_data=1  
and ie.stopped is null  
and ie.originalroute like 'Intravenous%'
```

SELECT ORIGEN 2

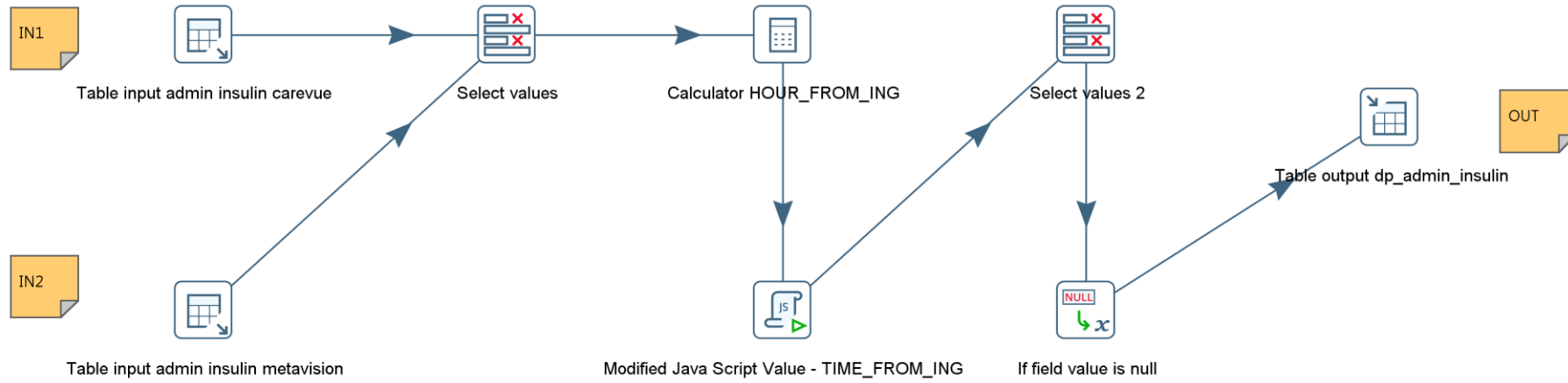
```
select ie.icustay_id,ie.itemid,de.label,ie.starttime,ie.endtime,icu.intime as icu_intime,  
case  
    when ie.amount is null then 0
```

```
    else ie.amount
end amount,starttime::timestamp::date startdate,endtime::timestamp::date enddate,
ordercategorydescription originalroute,
'U' amountuom,
'metavision' dbsource
from inputevents_mv ie, icustays icu, admissions adt, d_items de
where de.itemid between 223257 and 223262
and de.itemid=ie.itemid
and ie.icustay_id=icu.icustay_id
and icu.hadm_id=adt.hadm_id
and adt.has_chartevents_data=1
and ie.statusdescription!='rewritten'
and ie.ordercategoryname not in ('06-Insulin (Non IV)', '05-Med Bolus')
```

ENTRADA: inputevents_cv, inputevents_mv, icustays icu, admissions, d_items

SALIDA: dp_admin_insulin

PROCESO: obtiene los datos de administración de insulina de las tablas de inputevents de carevue y metavision y carga los datos en MIMICSEL incluyendo el tramo de 24H desde el ingreso, utilizado para agrupar posteriormente



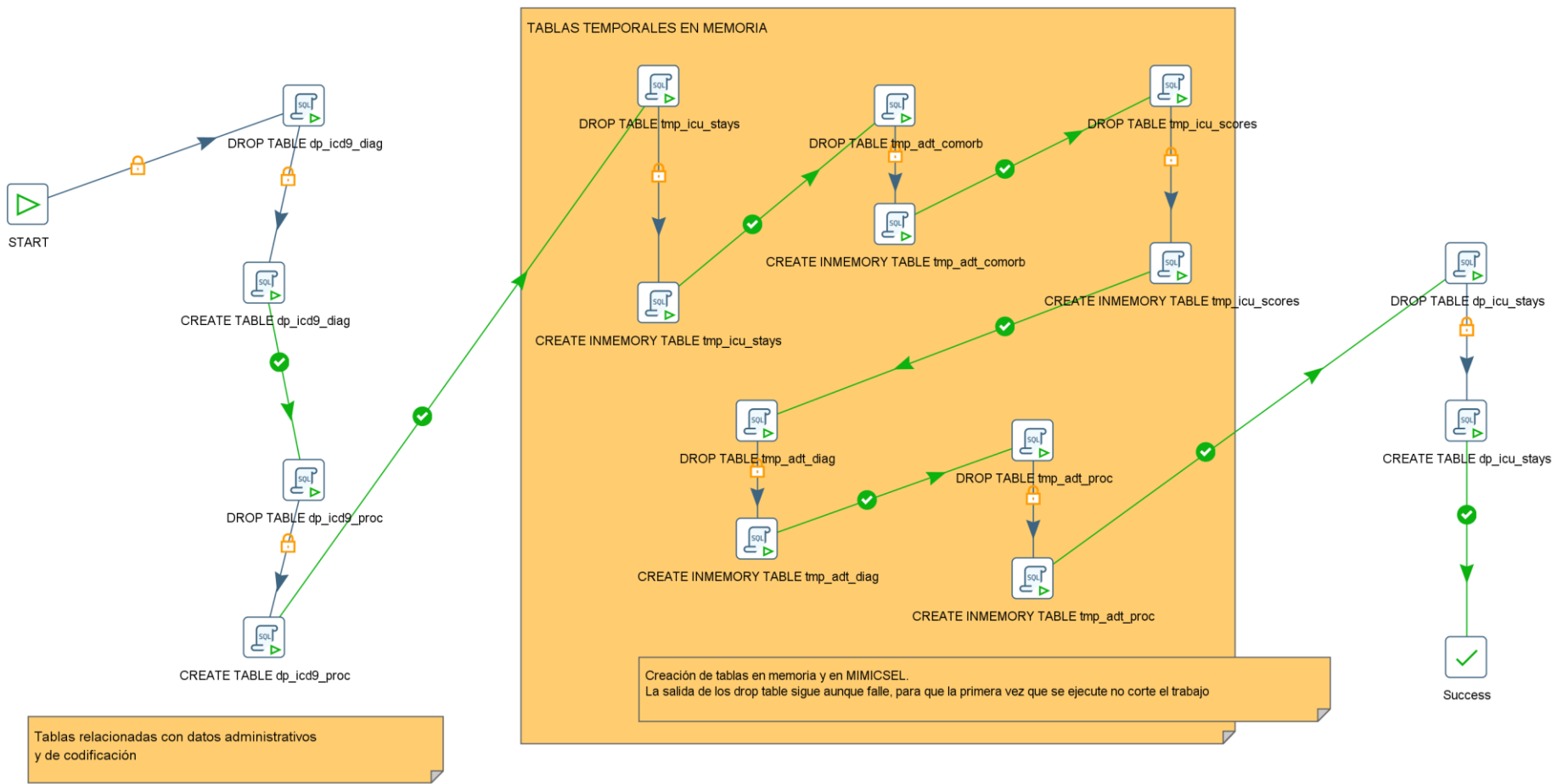
Name: A2-25.J_CREATE_TABLES_ADM
Description: Crea las tablas para los datos administrativos y de codificación
Long description: Parte de las tablas se crea en memoria para consolidarlas en trabajos posteriores, las de codificación se crean en MIMICSEL

Tablas creadas en memoria:

tmp_icu_stays
tmp_adt_comorb
tmp_icu_scores
tmp_adt_diag
tmp_adt_proc

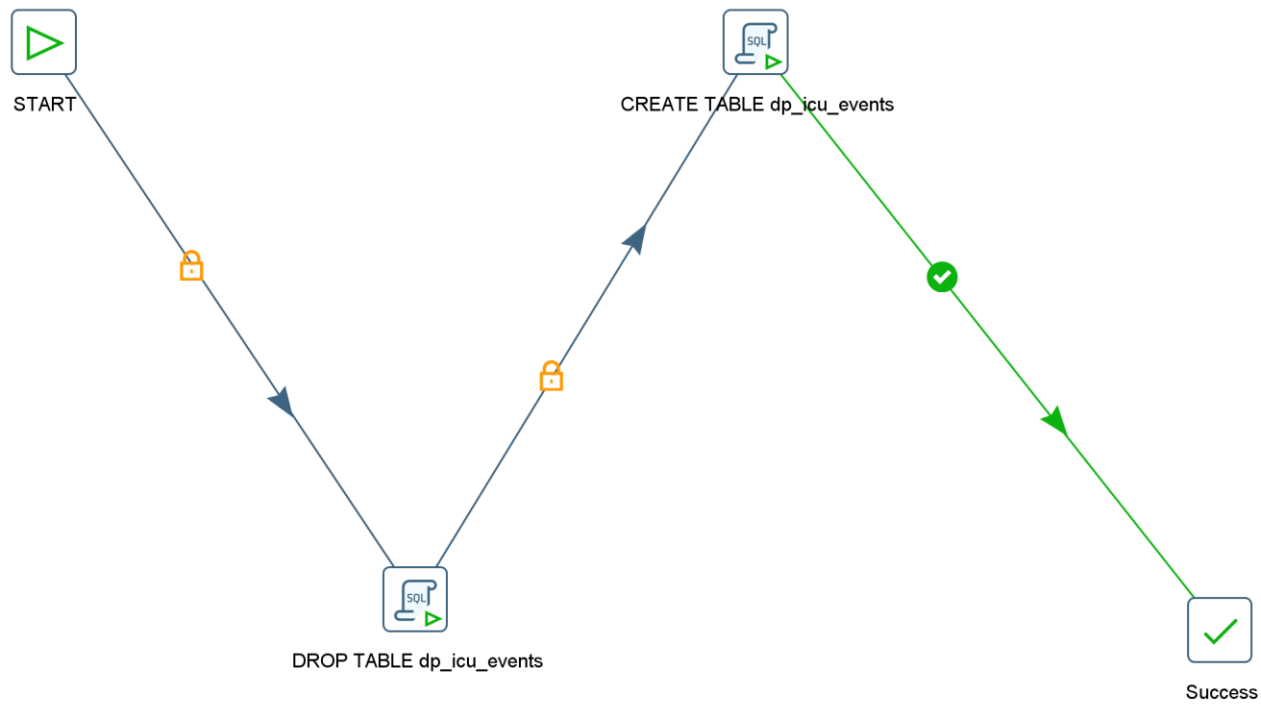
Tablas creadas en MIMICSEL:

dp_icd9_proc
dp_icd9_diag



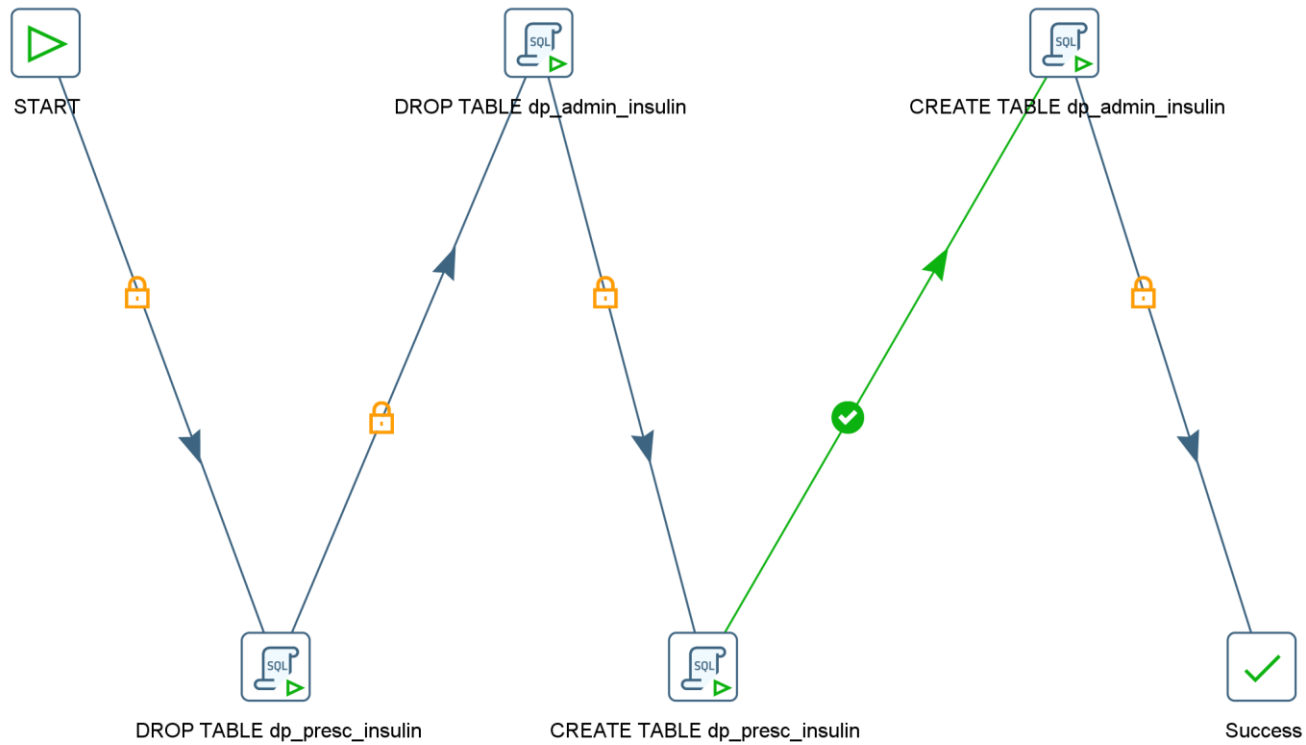
Name: A2-26.J_CREATE_TABLES_EVENTS
Description: Crea la tabla de eventos seleccionados
Long description: Crea la tabla de eventos seleccionados en la BBDD MIMICSEL

Creación de tablas en MIMICSEL
La salida de los drop table sigue aunque falle, para que la primera vez que se ejecute no corte el trabajo



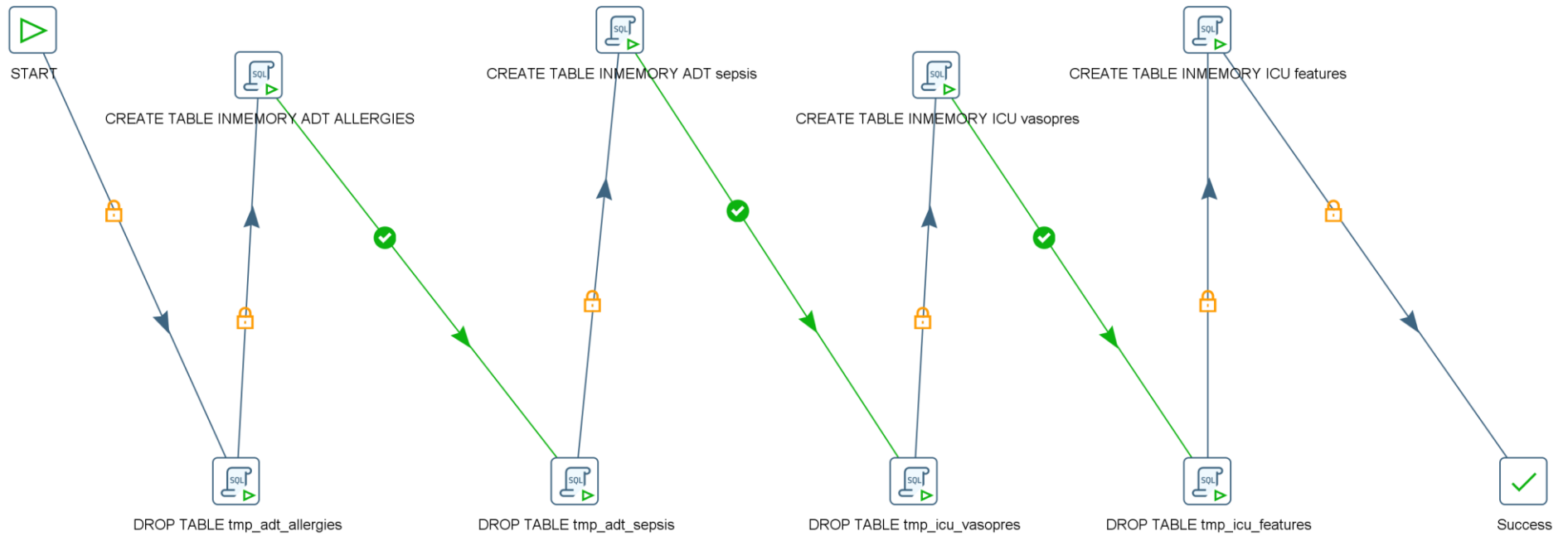
Name: A2-27.J_CREATE_TABLES_INSULIN
Description: Creación de tablas para datos de insulina
Long description: Crea tablas en MIMISEL:
dp_admin_insulin, dp_presc_insulin

Creación de tablas en MIMICSEL
La salida de los drop table sigue aunque falle, para que la primera vez que se ejecute no corte el trabajo



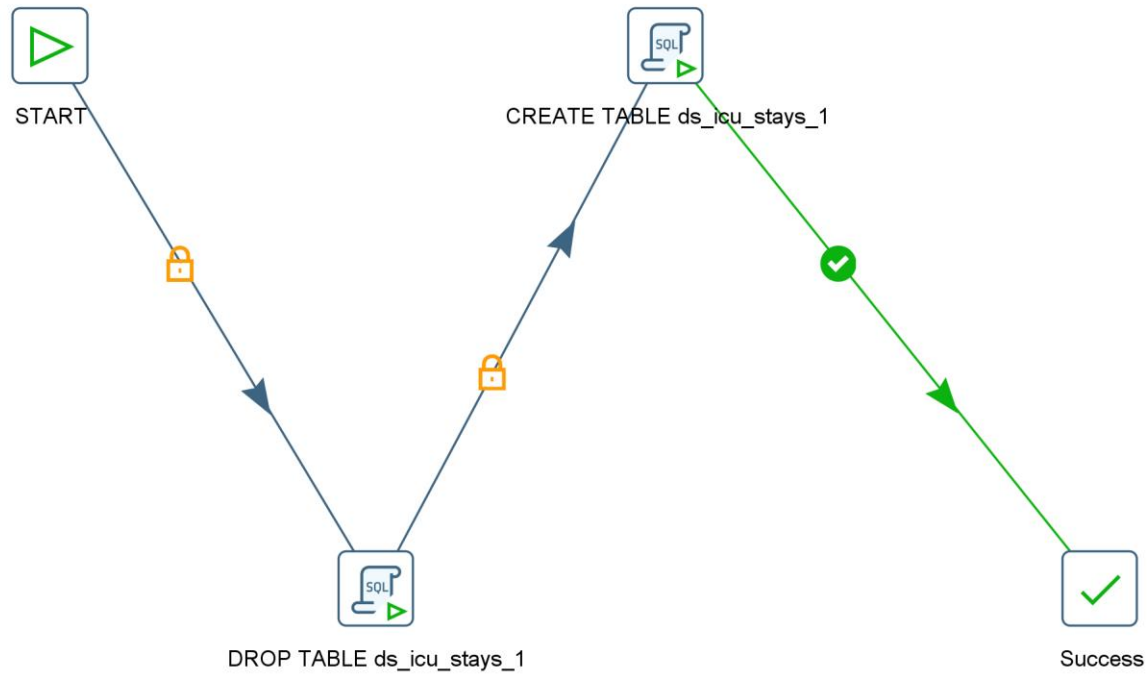
Name: A2-28.J_CREATE_TABLES_MED_FEATURES
Description: Creación de tablas en memoria para MED_FEATURES
Long description: Crea tablas temporales en memoria:
tmp_adt_allergies, tmp_adt_sepsis, tmp_icu_vasopresores, tmp_icu_features

Creación de tablas en memoria.
La salida de los drop table sigue aunque falle, para que la primera vez que se ejecute no corte el trabajo



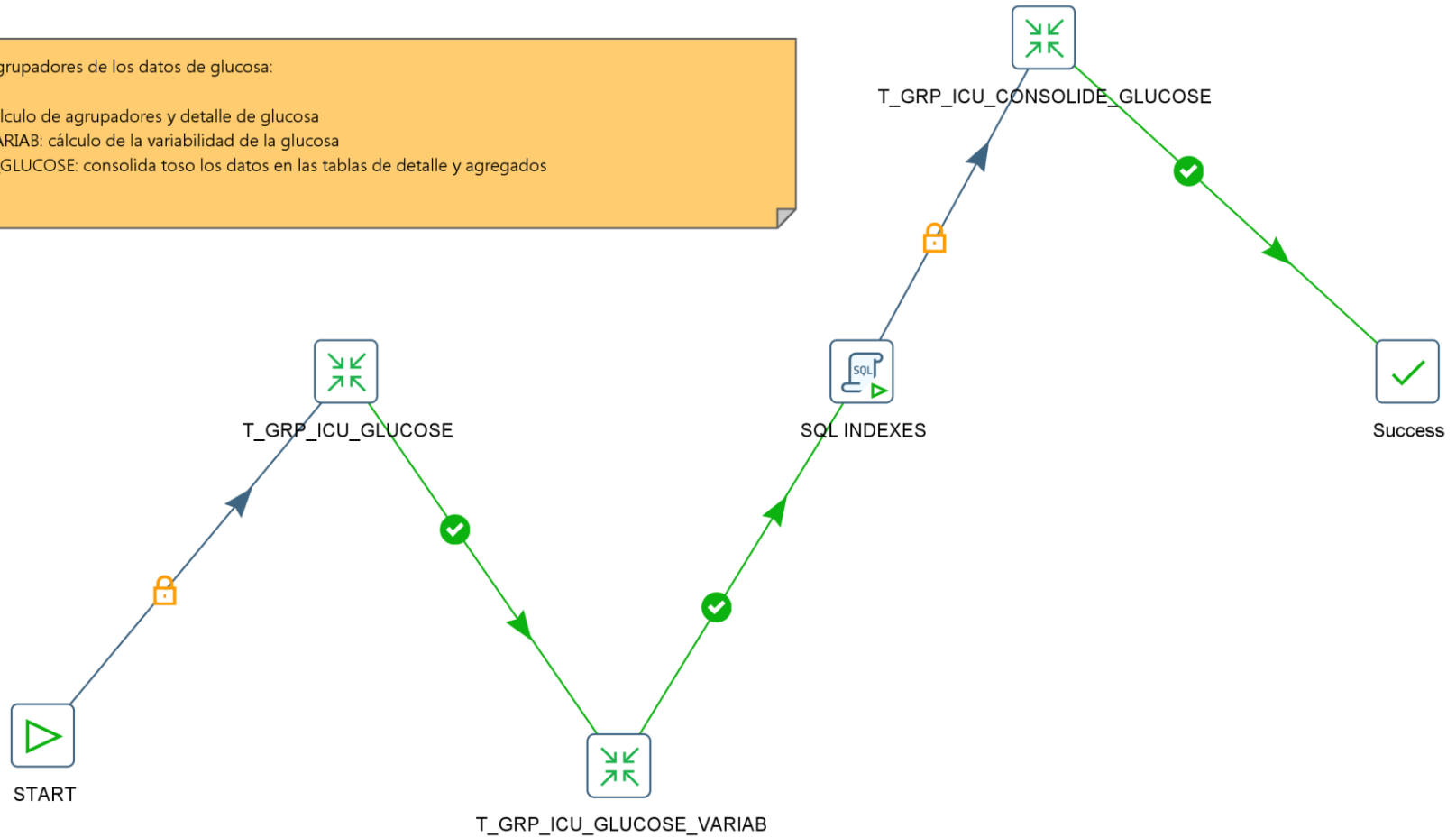
Name: A2-29.J_CREATE_TABLES_SELECT
Description: Creación de tablas para análisis de datos
Long description: Crea en MIMICSEL la tabla:
ds_icu_stays_1

Creación de tablas en MIMICSEL
La salida de los drop table sigue aunque falle, para que la primera vez que se ejecute no corte el trabajo



Name: A2-30.J_GROUP_DATA_GLUCOSE
Description: Trabajo para obtener los agrupadores de glucosa
Long description: Obtiene las tablas de agrupados y detalle de las mediciones de glucosa

Trabajo para lanzar los agrupadores de los datos de glucosa:
T_GRP_ICU_GLUCOSE: cálculo de agrupadores y detalle de glucosa
T_GRP_ICU_GLUCOSE_VARIAB: cálculo de la variabilidad de la glucosa
T_GRP_ICU_CONSOLIDE_GLUCOSE: consolida toso los datos en las tablas de detalle y agregados



Name: A2-31.J_CREATE_TABLES_GROUP
Description: Creación de tablas de agregados
Long description: Crea las tablas para cargar los agregados y para consolidar los datos de las tablas temporales intermedias

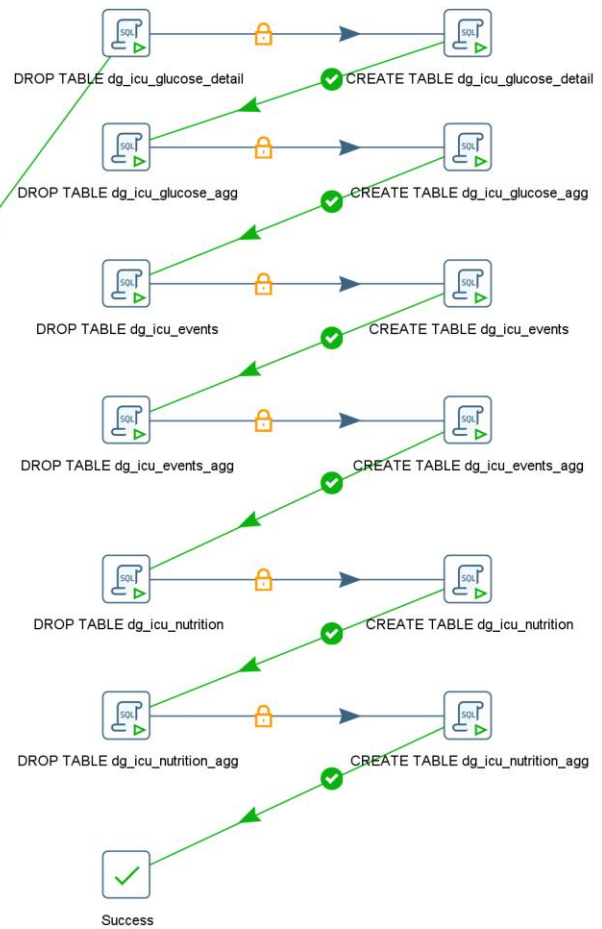
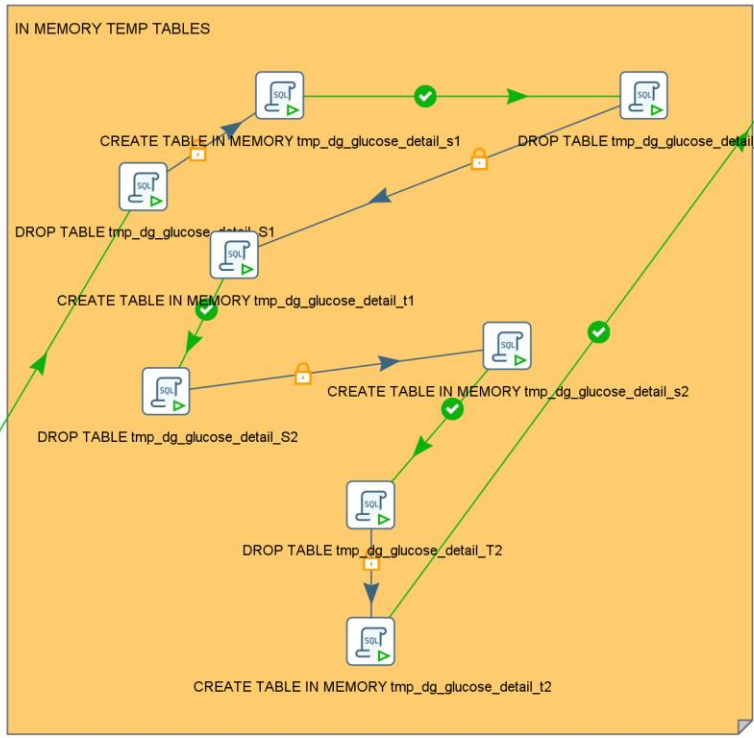
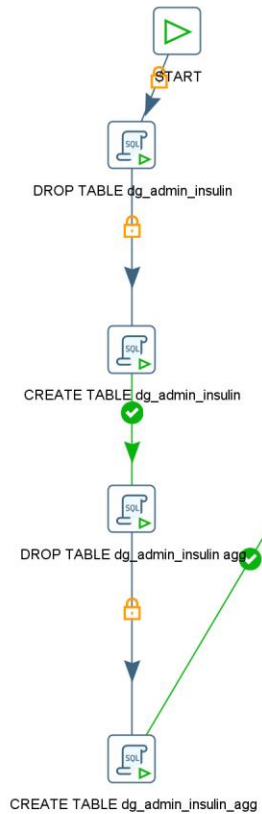
Tablas creadas en memoria:

tmp_dg_glucose_detail_s1
tmp_dg_glucose_detail_s2
tmp_dg_glucose_detail_t1
tmp_dg_glucose_detail_t2

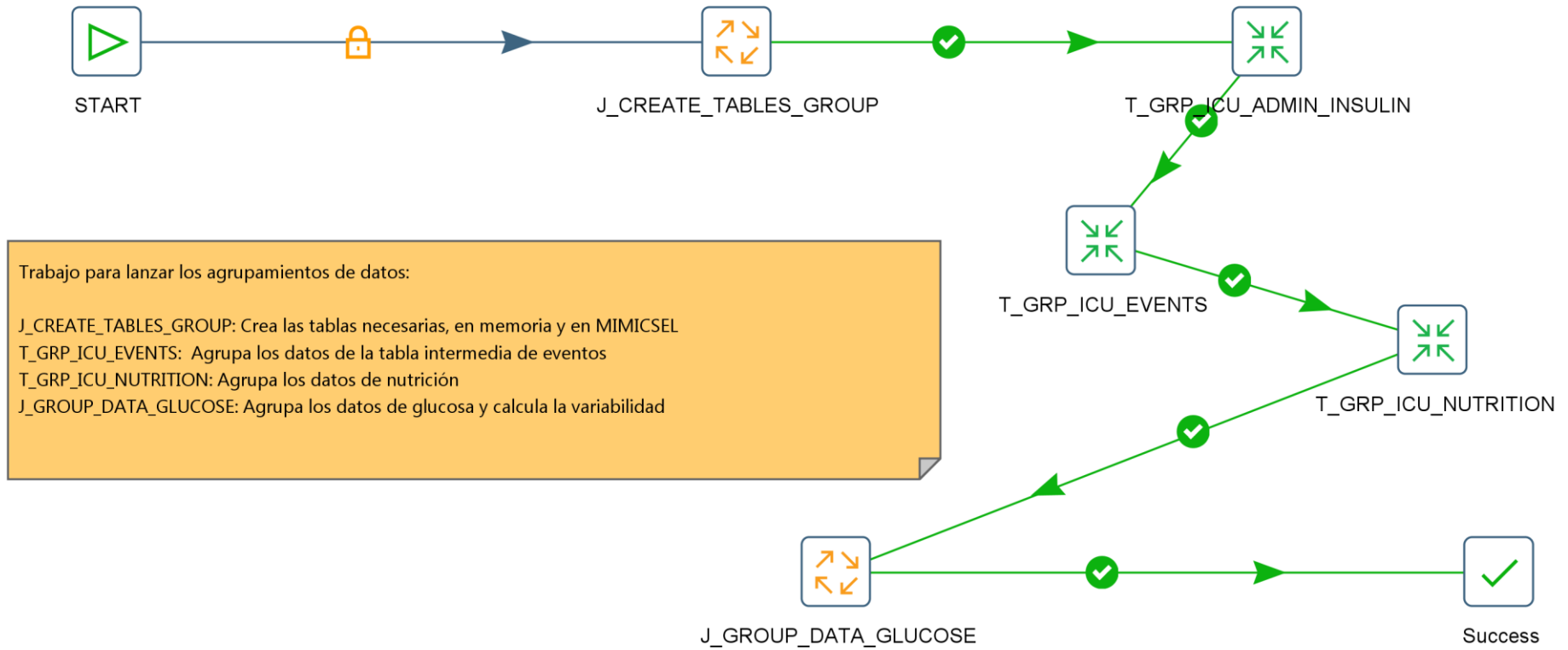
Tablas creadas en BBDD MIMICSEL

dg_admin_insulin
dg_admin_insulin_agg
dg_icu_glucose_detail
dg_icu_glucose_agg
dg_icu_events
dg_icu_events_agg
dg_icu_nutrition
dg_icu_nutrition_agg

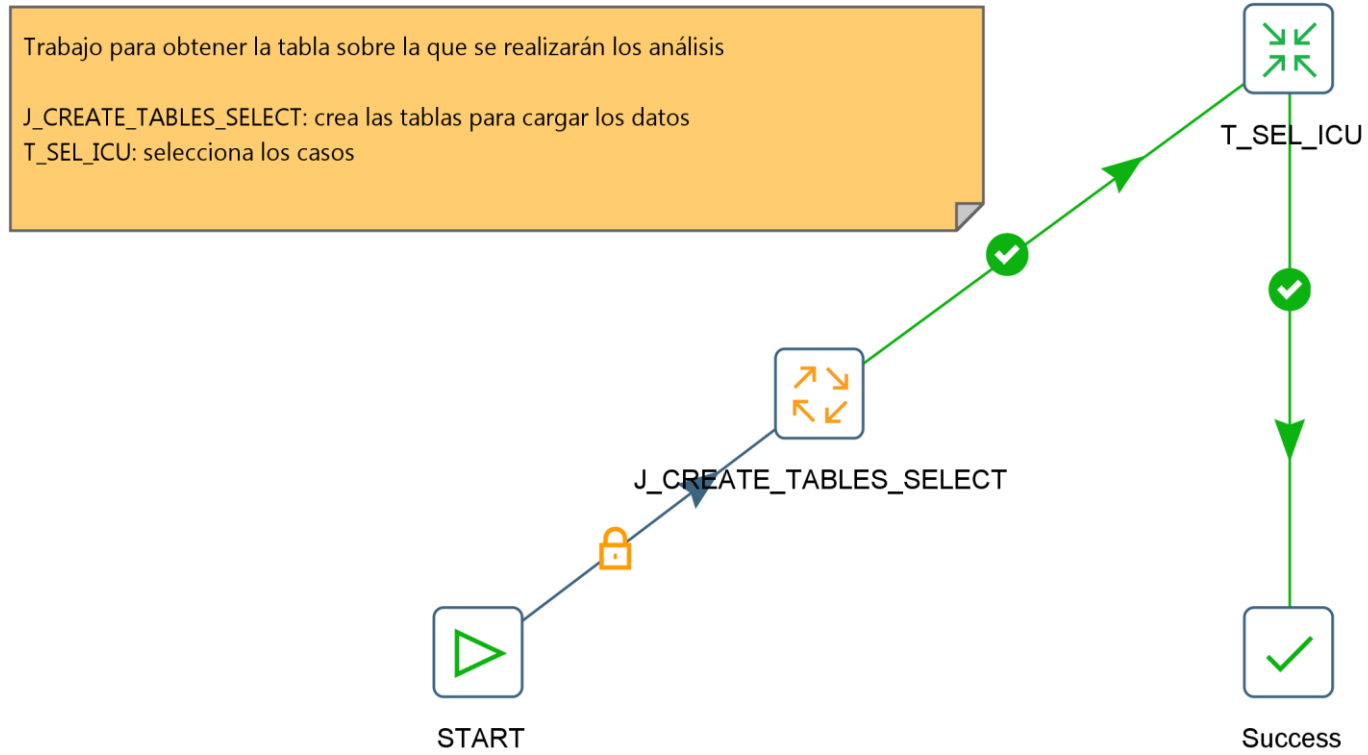
Creación de tablas en memoria y en MIMICSEL.
 La salida de los drop table sigue aunque falle, para que la primera vez que se ejecute no corte el trabajo



Name: A2-32.J_GROUP_DATA
Description: Trabajo principal para realizar la agrupación de los datos
Long description: Agrupación de datos de eventos, administración de insulina, nutrición y glucosa



Name: A2-33.J_SELECT_CASES
Description: Selección de casos
Long description: Selecciona los datos de las tablas intermedias y filtra los casos según los criterios para el análisis



Name: A2-34.J_POST_ETL
Description: Carga de datos POSTETL
Long description: Obtiene los datos de prescripción de insulina, administración iv de insulina y eventos de glucosa de las tablas intermedias

ADMINIV_INSULIN

```
update dp_icu_stays di set icu_has_admiv_insulin=true
where exists(
    select null from dp_admin_insulin dp where dp.icu_id=di.icu_id);
```

```
update dp_icu_stays set icu_has_admiv_insulin=false
where icu_has_admiv_insulin is null;
```

PRESC_INSULIN

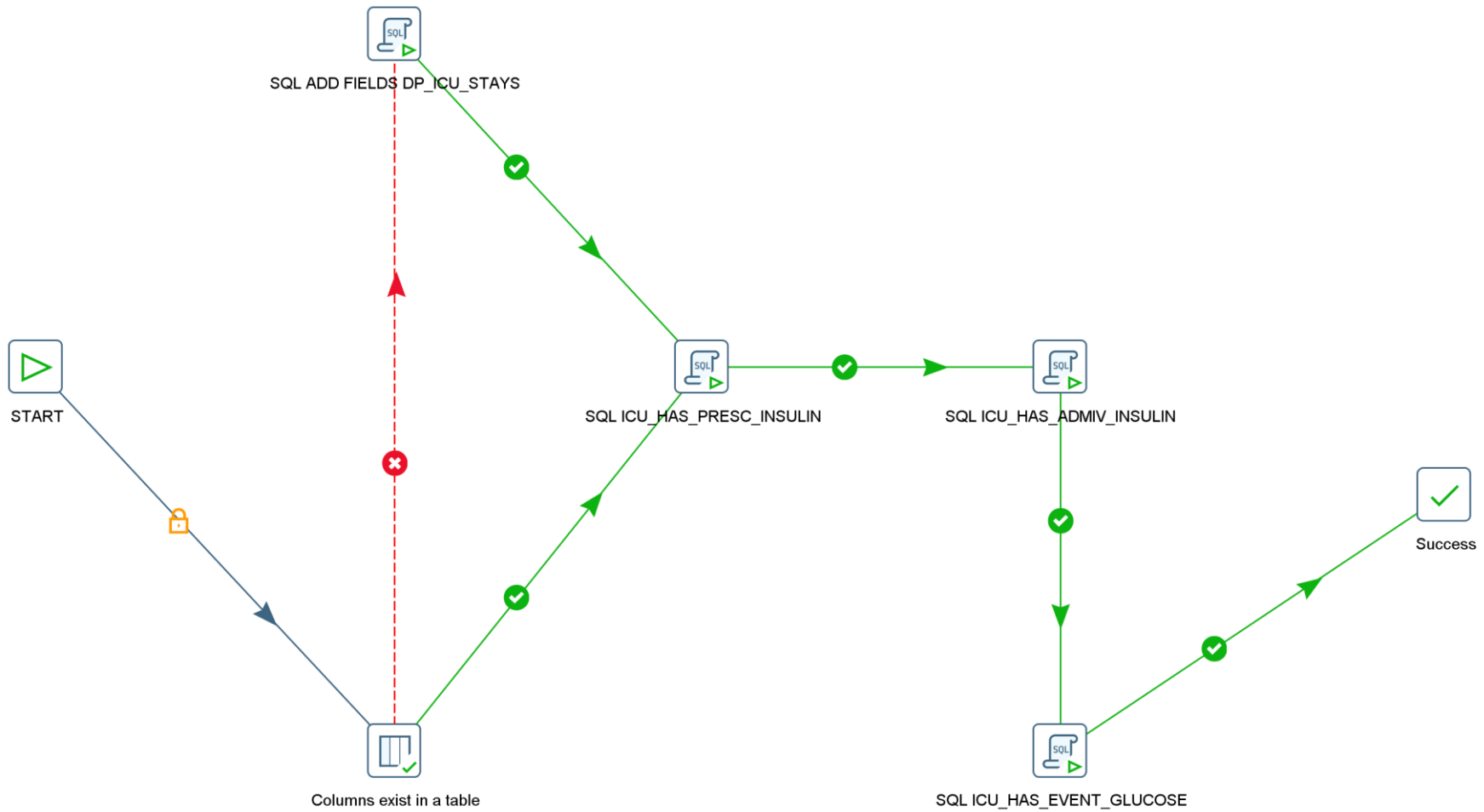
```
update dp_icu_stays di set icu_has_presc_insulin=true
where exists(
    select null from dp_presc_insulin dp where dp.icu_id=di.icu_id);
```

```
update dp_icu_stays set icu_has_presc_insulin=false
where icu_has_presc_insulin is null;
```

EVENT_GLUCOSE

```
update dp_icu_stays di set icu_has_event_glucose=true
where exists(
    select null from dp_icu_events de
        where de.icu_id=di.icu_id
        and de.event_type='GLUCOSE');
```

```
update dp_icu_stays set icu_has_event_glucose=false
where icu_has_event_glucose is null;
```

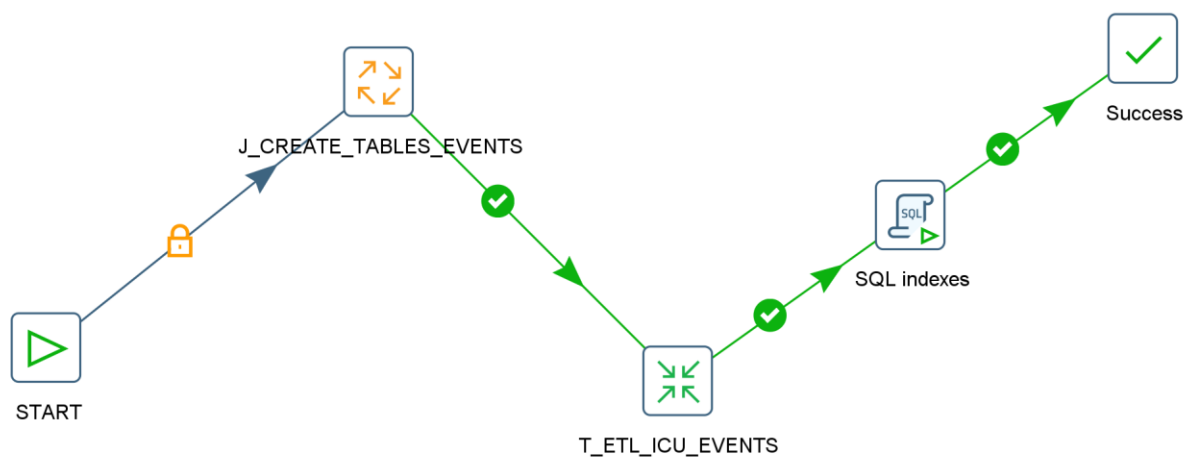


Name: A2-35.J_ETL_EVENTS
Description: Carga de los datos de eventos
Long description: Crea las tablas necesarias y realiza la carga de los datos de eventos

Trabajo para lanzar la carga de los datos de eventos:

J_CREATE_TABLES_EVENTS: Crea la tabla de eventos en MIMICSEL

T_ETL_ICU_EVENTS: Carga los datos de eventos seleccionados de las tablas de MIMIC-III a las de MIMICSEL

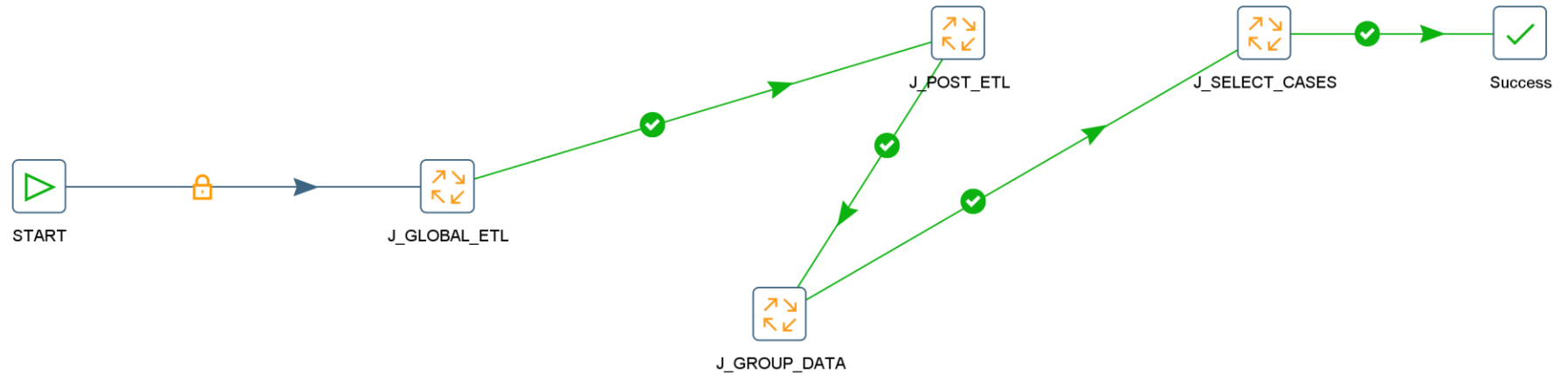


El job de EVENTS se encarga de extraer datos variables en el tiempo: mediciones de laboratorio, mediciones de glucosa, etc

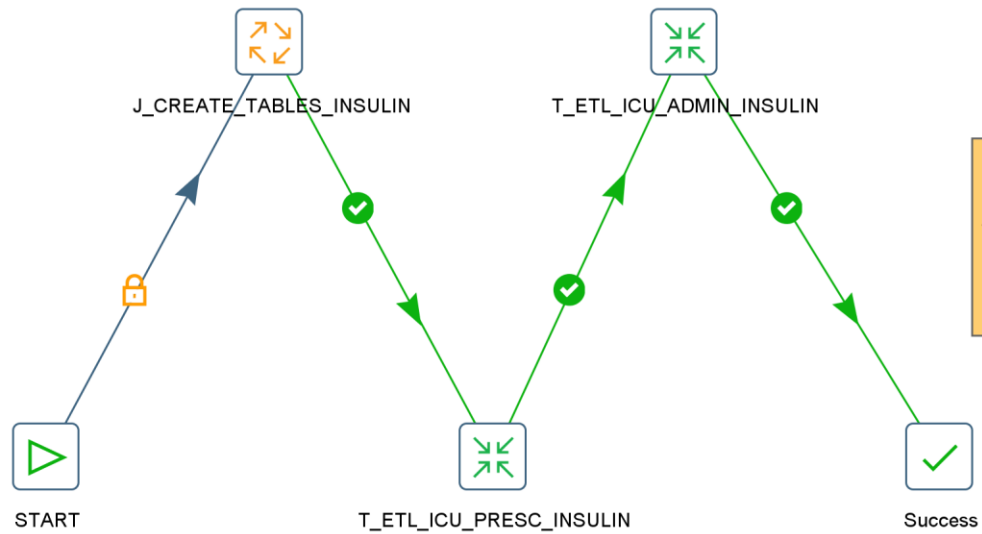
Name: A2-36.J_GLOBAL
Description: Trabajo inicial
Long description: Trabajo encargado de encadenar el resto de trabajos.
Lanzar para ejecutar todo el proceso

Trabajo inicial que lanza el resto de trabajos y transformaciones:

- J_GLOBAL_ETL: carga de tablas intermedias
- J_POST_ETL: transformaciones posteriores a la carga
- J_GROUP_DATA: crea tablas de datos agrupadas
- J_SELECT_CASES: selecciona los datos para el estudio y crea una tabla única



Name: A2-37.J_ETL_INSULIN
Description: Carga de datos de insulina
Long description: Crea y carga los datos de administración y prescripción de insulina



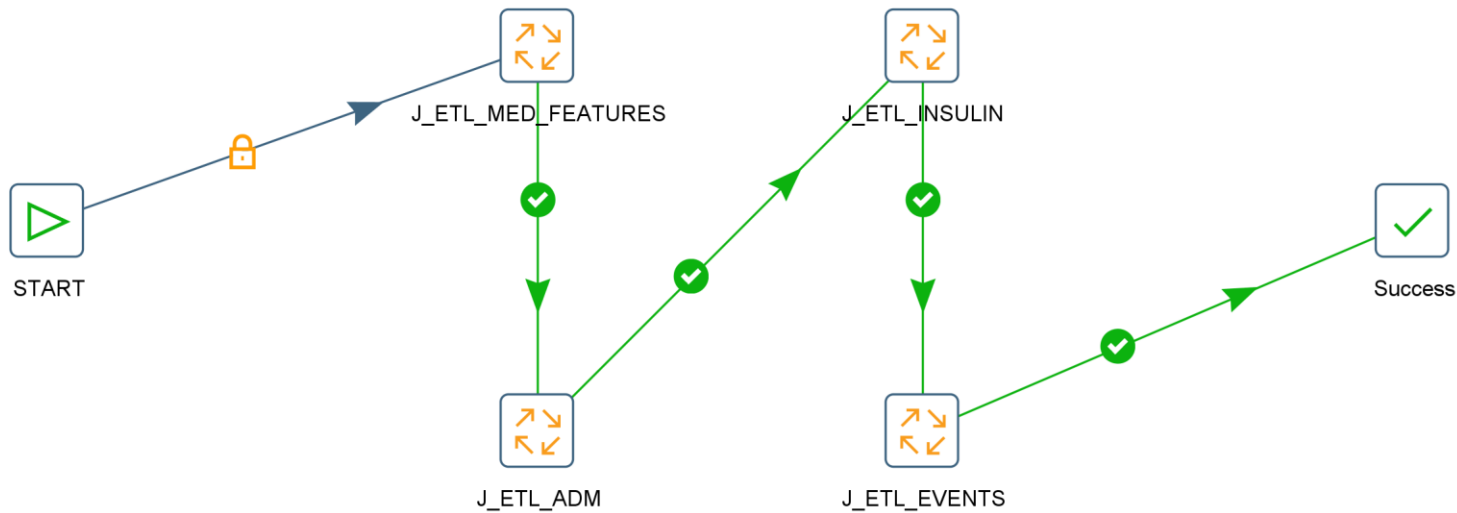
Trabajo para lanzar la carga de los datos de insulina:

- J_CREATE_TABLES_INSULIN: Crea las tablas necesarias en MIMICSEL
- T_ETL_ICU_PRESC_INSULIN: Carga los datos de prescripción de insulina
- T_ETL_ICU_ADMIN_INSULIN: Carga los datos de administración de insulina

Name: A2-38.J_GLOBAL_ETL
Description: Trabajo global para ETL
Long description: Obtiene los datos de las distintas tablas de la BBDD MIMIC-III y los carga en una BBDD intermedia

Trabajo para lanzar la carga de las tablas de la BBDD intermedia:

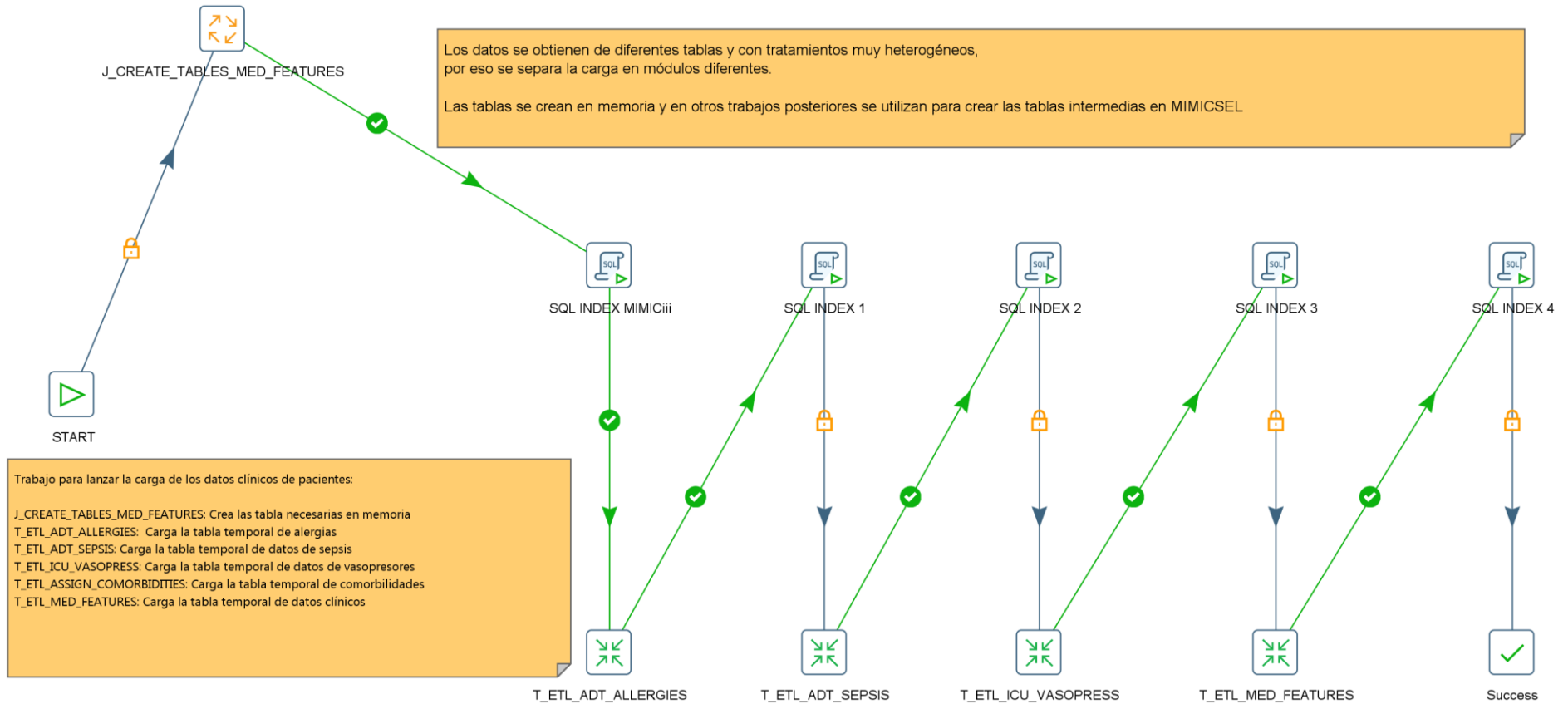
J_ETL_MED_FEATURES: Selecciona y carga los indicadores clínicos
J_ETL_ADM: Selecciona y carga los datos administrativos de pacientes y episodios
J_ETL_INSULIN: Selecciona y carga los datos de prescripción y administración de insulina
J_ETL_EVENTS: Selecciona y carga los datos de eventos y mediciones



Name: A2-39.J_ETL_MED_FEATURES

Description: Job principal para la carga de diferentes indicadores clínicos

Long description: Las tablas necesarias se crean al inicio del trabajo, así como los índices necesarios después de cada carga. Las tablas de salida se crean en memoria. En este trabajo se obtienen los datos de alergias, sepsis, vasopresores y varios indicadores.



Name:

A2-40.J_ETL_ADM

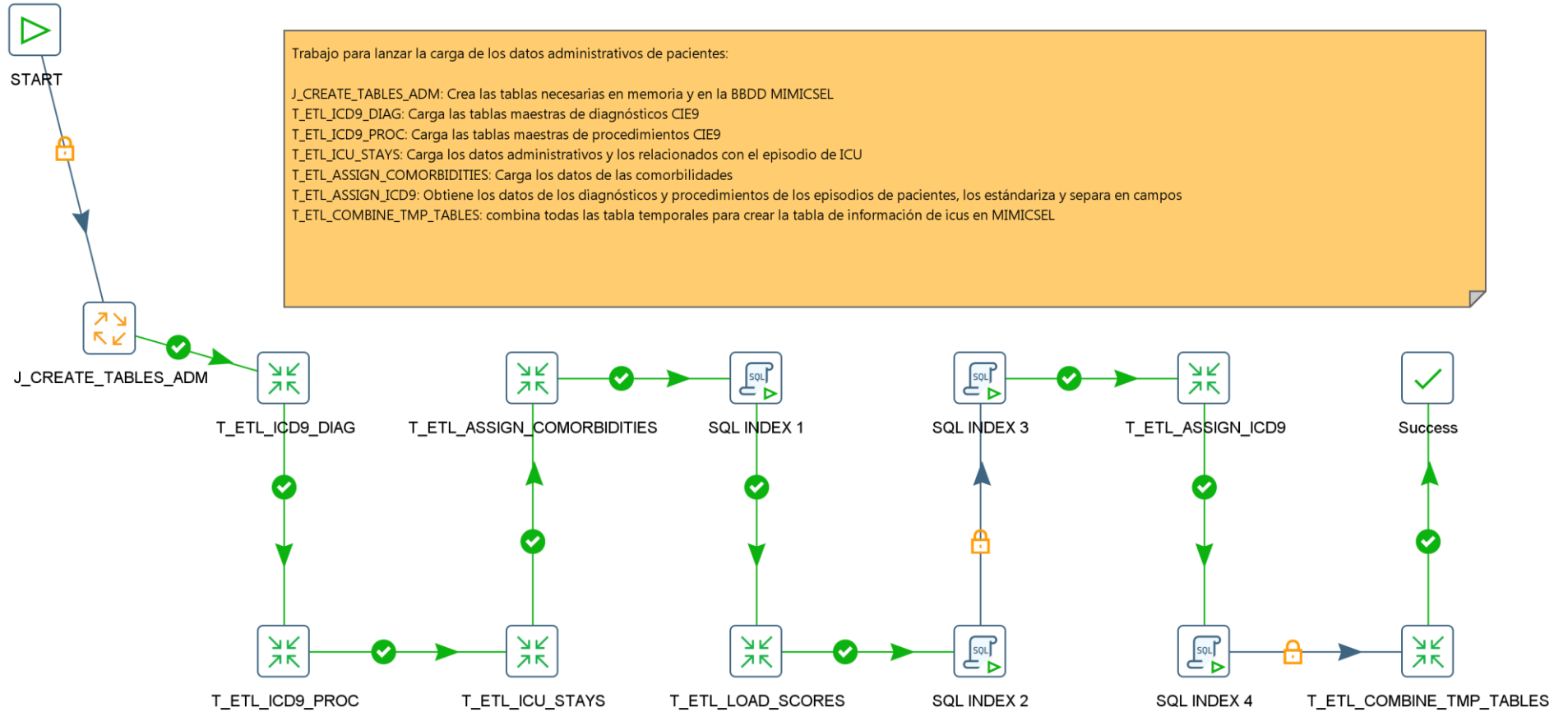
Description:

Trabajo principal para la carga de datos administrativos

Long description:

Creación de las tablas necesarias y obtención de los datos de codificación, estancias ICU, comorbilidades y scores

Creación también de los índices necesarios y combinación de todas las tablas en memoria temporal en las tablas de la BBDD MIMICSEL



Name: A2-41.J_DOC
Description: Documentación automática
Long description:

