**DOMO DATAFLOW TRANSFORMS**

**ClaimsInfoByDay DataFlow**

-- Rolls up PaidLosses and Outstanding by ClaimID

select

claimid, policyversionid, policyid, claimnumber, lossdate, reportedDate, losstype, catastrophecode, catastrophe, catdate, sum(PaidLosses) as PaidLosses, sum(Outstanding) as Outstanding

from claim\_detail

group by claimid, policyversionid, policyid, claimnumber, lossdate, reportedDate, losstype, catastrophecode, catastrophe, catdate

-- Counts up details by year

select

claimid, policyversionid, policyid, claimnumber, date(lossdate) as lossdate, reportedDate, losstype,

catastrophecode, catastrophe, catdate, PaidLosses, Outstanding, claimcount, claimspaid

from

(select

\*, case when @y = year(lossdate) then @d:=@d+1 else @d:=1 end as claimcount,

case when @y = year(lossdate) then @p:=@p+ifnull(PaidLosses,0) else @p:=ifnull(PaidLosses,0) end as claimspaid, @p as stuffp,

@y:= year(lossdate) as stuff

from

staging1

join

(select @d:=1, @y:=1, @p:=0) d

order by lossdate) d

-- where year(lossdate) = '2015' and lossdate > '2015-02-04'

-- order by lossdate

-- Attaches the number of claims received per day to the policy numbers by day

select

`Date`, Policies, Claims, Paid, datediff(str\_to\_date(concat(year(`Date`),'-12-31'),'%Y-%m-%d'),`Date`) as daysleft

from

(Select

Date, lossdate, `Total Inforce` as Policies,

cast(Case when (c.claims is null and @y <> year(`Date`)) then 0 else Case when c.claims is null then @f else c.claims end end as unsigned) as claims,

cast(Case when (c.paid is null and @y <> year(`Date`)) then 0 else Case when c.paid is null then @p else c.paid end end as decimal(15,2)) as paid,

Case when c.claims is null then @f else @f:=c.claims end as stuff1,

Case when c.paid is null then @p else @p:=c.paid end as stuff2,

Case when lossdate is null then @y else @y:=year(lossdate) end as stuff3

from

inforce\_policy\_trends p

left join

(select lossdate, max(claimspaid) as paid, max(claimcount) as claims from claims group by lossdate) c

on p.date = c.lossdate

join

(select @f:=0, @y:=1, @p:=0) d

order by `Date`) d

-- where year(`Date`) = '2015'

-- Figures out the Annualized # and % for each day as well as the avg. amt paid. Past years used the fixed total claim #, while current years get it projected out through the end of the year(annualized).

select

c.`Date`, Case when d.DayName = 'Friday' then 'X' end as W, Case when last\_day(c.`Date`) = c.`Date` then 'X' end as M,

c.Policies as 'Inforce Policies', c.Claims as 'Claims Filed', c.Paid as 'Claims Paid', c.Paid / c.Claims as 'Avg. Claims Paid',

Case when year(`Date`) <> year(curdate()) then m.yearmax else (c.claims / (365 - daysleft)) \* 365 end as Annualized\_number,

Case when year(`Date`) <> year(curdate()) then m.yearmax / policies else ((c.claims / (365 - daysleft)) \* 365) / policies end as Annualized\_percent

from

combined c

left join

(select year(`Date`) as 'Year', max(Claims) as yearmax from combined group by year(`Date`)) m

on year(c.`Date`) = m.`Year`

left join

calendar d

on c.`Date` = d.dt

-- where year(`Date`) = '2015'

**ClaimDetail Dataflow**

alter table claims2

add index (claimid)

select

c.claimid, c.PolicyVersionID, c.policyid, c.ClaimNumber, c.LossDate, c.ReportedDate, c.LossType, c.CatastropheCode, c.Catastrophe, c.CatDate,

t.CreatedDate, t.PaidLosses, t.Outstanding

from

claims2 c

left join

claimtransactions t

on t.claimid = c.claimid

**TRENDING DATA**

alter table inforcereporthistory

add index (ReportDate, `State`, Form, WindType, CompanyCode)

select

i.\*,

ifnull(i.PolicyCount - s.PolicyCount,0) as 'PolicyCount - Daily Change',

1 - ifnull(s.PolicyCount / i.PolicyCount,0) as 'PolicyCount - Daily % Change',

ifnull(i.Premium - s.Premium,0) as 'Premium - Daily Change',

1 - ifnull(s.Premium / i.Premium,0) as 'Premium - Daily % Change'

from

inforcereporthistory i

left join

staging s

on i.ReportDate = s.ReportDate and i.`State` = s.`State` and i.Form = s.Form and i.WindType = s.WindType and i.CompanyCode = s.CompanyCode

order by i.`State`, i.Form, i.WindType, i.CompanyCode, i.ReportDate

CLAIMS PAID

Case when year(`Date`) < year(CURDATE()) then `Annualized\_number` end

**DATE**

(CASE WHEN (MONTH(`ReportDate`) < 10) THEN Concat('0',MONTH(`ReportDate`),' - ',MONTHNAME(`ReportDate`)) ELSE Concat(MONTH(`ReportDate`),' - ',MONTHNAME(`ReportDate`)) END )

concat(quarter(`ReportDate`),'')

Year(`ReportDate`)