**UNIVERSAL DATA DICTIONARY**

**This code tells us if a claim has ever in its lifetime been in Litigation Status. This uses the ClaimStatusHistory table. This covers both current and History status.**

  select ClaimID,

         ClaimNumber,

         Litigation = iif(c.ClaimStatusID = 3 or exists( select \* from dbo.ClaimStatusHistory as csh where csh.ClaimStatusID = 3 and csh.ClaimID = c.ClaimID ), 'yes', 'no')

from          dbo.Claim as c

**This code tells us if the Agency is a Gieco Agency**

select

\* from [dbo].[AgencyAncestry]

where AncestorAgencyId = 25886

**NOTES:**

**We only need to query Atlas Bridge database when we need quotes. Go directly to the Policy table instead.**

**Use the VIEW call ClaimDetailsAndAssignments when trying to retrieve the Adjusting Company.**

**This first changes an empty string into a NULL. Then it changes all NULL’s into the string “Unknown”. This is a trick to accommodate both empty strings and NULL’s.**

Select isnull(nullif(cl.ReportingParty, ''), 'Unknown') as ReportedBy,

**This returns Policy counts and total premium**

declare @ThisYear datetime = getdate()

declare @LastYear datetime = dateadd(year, -1, @ThisYear)

ThisYear as (

select pah.AgencyID as AgencyID,

count(\*) as Policies,

sum(pva.TotalPremium) as Premium

from dbo.[Policy] as p

inner join ( select pv.PolicyID,

min(pv.PolicyVersionID) as PolicyVersionID

from dbo.PolicyVersion as pv

group by pv.PolicyID) as pvm

inner join dbo.PolicyVersionAttributes as pva

on pva.PolicyVersionID = pvm.PolicyVersionID

on pvm.PolicyID = p.PolicyID

inner join ( select pa.PolicyID,

min(pa.PolicyAgencyHistoryID) as PolicyAgencyHistoryID

from dbo.PolicyAgencyHistory as pa

group by pa.PolicyID) as pahm

inner join dbo.PolicyAgencyHistory as pah

on pah.PolicyAgencyHistoryID = pahm.PolicyAgencyHistoryID

on pahm.PolicyID = p.PolicyID

where p.CreatedDate

between dateadd(day, -datepart(dayofyear, @ThisYear) + 1, cast(@ThisYear as date))

and @ThisYear

and p.ProductID != 2

group by pah.AgencyID

)

**\*\*VERIFIED BEST WAY\*\* This will get you the Adjuster, Examiner, AdjustingCO**

select q.QueueID,

sux.SystemUserID as ExaminerID,

isnull(sux.FirstName + ' ' + sux.LastName,'') as ExaminerDesc,

sua.SystemUserID as AdjusterID,

coalesce(sua.FirstName + ' ' + sua.LastName, co.FirstName + ' ' + co.LastName, '') as AdjusterDesc,

co.BusinessName as AdjustingCompany,

1 as RowNumber

from #Queue as q

inner join Workflow.Referrals as re with (readpast)

left join dbo.SystemUsers as sux with (readpast)

on sux.SystemUserID = dbo.GetExaminerByClaimID(re.ClaimID)

left join dbo.SystemUsers as sua with (readpast)

on sua.SystemUserID = dbo.GetAdjusterByClaimID(re.ClaimID)

left join dbo.Contact as co with (readpast)

on co.ContactID = dbo.GetAdjustingCompanyByClaimID(re.ClaimID)

on re.QueueID = q.QueueID

where q.TypeID not in (2, 5)

**(Old Way) This will get you the Adjuster, Examiner**

GetAdjusters (ClaimID, AdjustingCompanyContactID,ExaminerSystemUserID, Active)

as

(

select cd.ClaimID,cd.adjustingCompanyContactID,cd.SystemUserID as ExaminerSystemUserID, cd.Active

from ClaimDetails as cd

union

select ca.ClaimID,ca.AdjustingCompanyContactID,ca.ExaminerSystemUserID, isnull(cd.Active, 0)

from Claim.ClaimAssignment as ca

left join ClaimDetails as cd

on ca.ClaimID = cd.ClaimID

where isnull(cd.Active, 1) = 1 and ca.[RemovedDate] is null and ca.ClaimDetailTypeID = 3

),

**\*\*VERIFIED\*\*This will get you the Public Adjuster**

select c.ClaimID,co.BusinessName

from Atlas.dbo.Claim as c

inner join Atlas.dbo.ClaimContactAssoc as cca

inner join Atlas.dbo.Contact as co

inner join Atlas.dbo.ContactType as ct

on ct.ContactTypeID = co.ContactTypeID

on co.ContactID = cca.ContactID

on cca.ClaimID = c.ClaimID

where ct.ContactTypeCode = 'Claims'

and ct.ContactTypeDesc = 'Public Adjuster'

**This will get you the AdjustingCO**

GetAdjustingCO

as

(

select cd.ClaimID,cd.ClaimDetailTypeID, cd.CreatedDate, cd.AdjustingCompanyContactID

from dbo.ClaimDetails as cd

where --cd.ClaimID = @ClaimID and

cd.Active = 1

and cd.ClaimDetailTypeID in (1, 2)

union all

select ca.ClaimID,ca.ClaimDetailTypeID,

ca.AssignmentDate as CreatedDate,

ca.AdjustingCompanyContactID

from Claim.ClaimAssignment as ca

where --ca.ClaimID = @ClaimID and

ca.ClaimDetailTypeID = 1

and ca.RemovedBy is null

)

**This will get you the Plaintiff’s Attorney**

GetPlaintiffAttorney as (

select distinct

cl.ClaimID

from dbo.Claim as cl

inner join dbo.ClaimContactAssoc as cta

on cta.ClaimID = cl.ClaimID

and cta.RemovedBy is null

inner join dbo.Contact as co

on co.ContactID = cta.ContactID

and co.ContactTypeID in (15,16)

)

**This should get you the Adjuster?**

select

concat(su.FirstName + ' ' + su.LastName) as Adjuster

from            Claim as c

left join       dbo.SystemUsers as su

        on      su.SystemUserID = dbo.GetAdjusterByClaimID(c.ClaimID)

**This should get you the Examiner?**

select

concat(ex.FirstName + ' ' + ex.LastName) as Examiner

from            Claim as c

left join dbo.SystemUsers as ex

on ex.SystemUserID =[dbo].[GetExaminerByClaimID](c.ClaimID)

When joining on policy,

for **policy quotes** use “PolicyAttributes”

select cmb.CompanyCode,

cmb.[State],

cmb.County,

cmb.Territory,

cmb.PolicyForm,

cmb.ProductID,

Unsubmitted = isnull(count(distinct pa.PolID), 0)

into #Quoted

from Combinations as cmb

left join AtlasBridge.dbo.PolicyAttributes as pa

on cast(pa.AppSubmitDate as date)

between @StartingDate

and @EndingDate

and pa.[Retention] = cmb.CompanyID

and pa.PolicyLocationState = cmb.[State]

and pa.PolicyCounty = cmb.County

and pa.PolicyLocationTerritoryCode = cmb.Territory

and pa.BillingTotalEstPremiumAmount > 0

and pa.PolicyFormTypeTextChoice = cmb.PolicyForm

and pa.QueueStage != 'Submitted'

for **policy submitted apps** use “Policy” table

select cmb.CompanyCode,

cmb.[State],

cmb.County,

cmb.Territory,

cmb.PolicyForm,

cmb.ProductID,

Submitted = isnull(count(distinct p.PolicyID), 0),

SubmittedPremium = isnull(sum(pva.SubTotal), cast(0 as money))

into #Submitted

from Combinations as cmb

left join Atlas.dbo.Policy as p

inner join ( select pv.PolicyID,

min(pv.PolicyVersionID) as PolicyVersionID

from Atlas.dbo.PolicyVersion as pv

group by pv.PolicyID) as pvm

inner join Atlas.dbo.PolicyVersionAttributes as pva

on pva.PolicyVersionID = pvm.PolicyVersionID

on pvm.PolicyID = p.PolicyID

inner join Atlas.dbo.InsuredEntity as ie

inner join Atlas.dbo.Addresses as ad

left join Atlas.dbo.County as cn

on cn.CountyID = ad.CountyID

on ad.AddressID = ie.PropertyAddressID

on ie.PolicyID = p.PolicyID

on cast(p.BoundDate as date)

between @StartingDate

and @EndingDate

and p.CompanyID = cmb.CompanyID

and p.StateID = cmb.StateID

and cn.CountyDescription = cmb.County

and pva.TerritoryCode = cmb.Territory

and pva.FormType = cmb.PolicyForm

**IMPORTANT SPs**

[dbo].[reportDWPEarned]

* Use the Created Date instead of the Reported Date from the Claims table.
* All perils may not be a catastrophe. All catastrophes will have a peril attached to it.
* Get FormType from the dbo.PolicyVersionAttributes table
* You only need to use PolicyVersion when you need to get to the policy table
* Use the ClaimsTransaction table for subrogation data

**JOINS USED to get to FIPS**

--This code joins AGENCY to COUNTY for FIPS

--This code joins InforcePOLICY to COUNTY for FIPS and has been WORKING SUCCESSFULLY

select isnull(c.FIPS, '') as 'FIPS'

from InforcePolicy as ly

left join InforcePolicy as ty

on ty.PolicyID = ly.PolicyID

and @PastYearAsOfDate

between ty.EffectiveDate

and ty.ExpirationDate

Join Policy as p

ON ly.PolicyID = p.PolicyID

join company cp on cp.CompanyID = p.CompanyID

join product pd on pd.ProductID = p.ProductID

join states s on s.StateID = p.StateID

join InsuredEntity ie on ie.PolicyID = p.PolicyID

join Addresses a on a.AddressId = ie.PropertyAddressID

left join county c on c.CountyID = a.CountyID

--This code joins POLICY to COUNTY for FIPS and has been WORKING SUCCESSFULLY

select isnull(c.FIPS, '') as FIPS

from policy p

join company cp on cp.CompanyID = p.CompanyID

join product pd on pd.ProductID = p.ProductID

join states s on s.StateID = p.StateID

join InsuredEntity ie on ie.PolicyID = p.PolicyID

join Addresses a on a.AddressId = ie.PropertyAddressID

left join county c on c.CountyID = a.CountyID

where cast(p.createddate as date) < cast(getdate() as date)

//////////////////////////////////////////////////////////////////////////////////

from Claim as c

inner join Atlas.dbo.PolicyVersion as pv

on pv.PolicyVersionID = c.PolicyVersionID

inner join Atlas.dbo.Policy as p

on p.PolicyID = pv.PolicyID

LEFT JOIN [dbo].[InsuredEntity] AS ie

on ie.PolicyID = p.PolicyID

LEFT JOIN [dbo].[Addresses] a

ON a.AddressId = ie.PropertyAddressID

LEFT join Atlas.dbo.States as st

on st.StateID = a.StateID

LEFT JOIN [dbo].[County] co

ON co.CountyID = a.CountyID

**CODE TO DETERMINE IF PRINTED DRAFT IS A FIELD DRAFT OR INTERNAL DRAFT**

(where ur.RoleID = 135 means it’s a Field Draft)

from ClaimDrafts cd

join PropertyStatements ps on ps.PropertyStatementId = cd.PropertyStatementId

join claim cl on cl.claimid = ps.claimid

left join claim.FastTrack ft on ft.ClaimID = cl.ClaimID and ft.RemovedOn is null

join ClaimDraftStatus cds on cds.ClaimDraftStatusID = cd.ClaimDraftStatusId

join systemusers su on su.SystemUserID = cd.CreatedBy

join UserRoles ur on ur.SystemUserID = su.SystemUserID

where ur.RoleID = 135 AND ft.ClaimID = c.ClaimID),'FieldDraft', 'InternalDraft'),

**(NEW WAY)**

**CODE TO PULL CLAIMS BY State, County,City,Zip (From CLAIM table to INSURED ENTITY)**

Select distinct a.City,co.CountyDescription

from [dbo].[Claim] c

left join [dbo].[InsuredEntity] ie

on c.InsuredEntityID = ie.InsuredEntityID

left join [dbo].[Addresses] a

on ie.PropertyAddressID = a.AddressId

left join [dbo].[States] s

on a.StateID = s.StateID

left join [dbo].[County] co

on a.CountyID = co.CountyID

**(OLD WAY)**

**CODE TO PULL CLAIMS BY Date and State (From CLAIM table to POLICY)**

select

count(c.ClaimID) AS ClaimCount, CAST(c.CreatedDate AS Date) AS CreatedDate, st.StateDescription, coalesce(co.CountyDescription, st.StateDescription) as CountyDescription

from Claim as c

inner join Atlas.dbo.PolicyVersion as pv

on pv.PolicyVersionID = c.PolicyVersionID

inner join Atlas.dbo.Policy as p

on p.PolicyID = pv.PolicyID

LEFT JOIN [dbo].[InsuredEntity] AS ie

on ie.PolicyID = p.PolicyID

LEFT JOIN [dbo].[Addresses] a

ON a.AddressId = ie.PropertyAddressID

LEFT join Atlas.dbo.States as st

on st.StateID = a.StateID

LEFT JOIN [dbo].[County] co

ON co.CountyID = a.CountyID

WHERE

c.CreatedDate >= '20150101'

--c.CreatedDate=cast(dateadd(d,-1,getdate()) as date)

group by c.CreatedDate, st.StateDescription, co.CountyDescription

from         dbo.Claim as c

inner join    dbo.PolicyVersion as pv

        inner join    dbo.Policy as p

                inner join      [dbo].[InsuredEntity] as ie

                        inner join      [dbo].[Addresses] as ad

                                left join dbo.States as st

                                        on     st.StateID = ad.StateID

                                left join       [dbo].[County] as co

                                        on              co.CountyID = ad.CountyID

                                on              ad.AddressId = ie.PropertyAddressID

                        on              ie.PolicyID = p.PolicyID

                on              p.PolicyID = pv.PolicyID

        on              pv.PolicyVersionID = c.PolicyVersionID

where           c.CreatedDate >= '20150101'

group by        cast(c.CreatedDate as date),

                        st.StateDescription,

                        coalesce(co.CountyDescription, st.StateDescription)

**CODE TO find out if it had representation**

Category = iif(exists ( select \*

from dbo.Claim as cl

inner join dbo.ClaimContactAssoc as cta

on cta.ClaimID = cl.ClaimID

and cta.RemovedBy is null

inner join dbo.Contact as co

on co.ContactID = cta.ContactID

and co.ContactTypeID in(15,16)

where cl.ClaimID = c.ClaimID), 'With Rep.', 'Without Rep.')

select          count(c.ClaimID) as ClaimCount,

                        cast(c.CreatedDate as date) as CreatedDate,

                        st.StateDescription,

                        coalesce(co.CountyDescription, st.StateDescription) as CountyDescription

from         dbo.Claim as c

inner join    dbo.PolicyVersion as pv

        inner join    dbo.Policy as p

                inner join      [dbo].[InsuredEntity] as ie

                        inner join      [dbo].[Addresses] as ad

                                left join dbo.States as st

                                        on     st.StateID = ad.StateID

                                left join       [dbo].[County] as co

                                        on              co.CountyID = ad.CountyID

                                on              ad.AddressId = ie.PropertyAddressID

                        on              ie.PolicyID = p.PolicyID

                on              p.PolicyID = pv.PolicyID

        on              pv.PolicyVersionID = c.PolicyVersionID

where           c.CreatedDate >= '20150101'

group by        cast(c.CreatedDate as date),

                        st.StateDescription,

                        coalesce(co.CountyDescription, st.StateDescription)

**CODE TO get to Adjusting Company BusinessName field ( uses a join to a function )**

SELECT isnull(co.BusinessName, '') as AdjustingCompany

from Claim as c

**left join dbo.Contact as co**

**on co.ContactID = dbo.GetAdjustingCompanyByClaimID(c.ClaimID)**

inner join PolicyVersion as pv

on pv.PolicyVersionID = c.PolicyVersionID

left join GetBODCatastrophe() as bod

inner join Catastrophe as cat

on cat.CatastropheID = bod.CatastropheID

on bod.CatastropheID = c.CatastropheID

inner join LossType as lt

on lt.LossTypeID = c.LossTypeID

where cast(c.createddate as date) < cast(getdate() as date)

from dbo.Claim as c

inner join dbo.PolicyVersion as pv

inner join dbo.PolicyVersionAttributes as pva

on pva.PolicyVersionID = pv.PolicyVersionID

inner join dbo.Policy as p

inner join dbo.InsuredEntity as ie

inner join dbo.Addresses as a

inner join dbo.States as s

on s.StateID = a.StateID

left join dbo.County cnt

on cnt.CountyID = a.CountyID

on ie.PropertyAddressID = a.AddressId

on ie.PolicyID = p.PolicyID

inner join dbo.PolicyContactAssoc as pca

inner join dbo.Contact as co

on co.ContactID = pca.ContactID

and co.ContactTypeID = 4

on pca.PolicyID = p.PolicyID

and pca.RemovedBy is null

and pca.[Primary] = 1

on p.PolicyID = pv.PolicyID

on pv.PolicyVersionID = c.PolicyVersionID

inner join dbo.LossType as lt

on lt.LossTypeID = c.LossTypeID

SummaryByDate as (

select cast(p.BoundDate as date) as [Date],

pah.AgencyID as AgencyID,

p.CompanyID,

p.StateID,

ad.CountyID,

pva.FormType as PolicyForm,

pva.TerritoryCode as Territory,

count(\*) as [Count],

sum(pva.SubTotal) as Premium

from dbo.Policy as p

inner join dbo.InsuredEntity as ie

inner join dbo.Addresses as ad

on ad.AddressID = ie.PropertyAddressID

on ie.PolicyID = p.PolicyID

inner join ( select pv.PolicyID,

min(pv.PolicyVersionID) as PolicyVersionID

from dbo.PolicyVersion as pv

group by pv.PolicyID) as pvm

inner join dbo.PolicyVersionAttributes as pva

on pva.PolicyVersionID = pvm.PolicyVersionID

on pvm.PolicyID = p.PolicyID

inner join ( select pa.PolicyID,

min(pa.PolicyAgencyHistoryID) as PolicyAgencyHistoryID

from dbo.PolicyAgencyHistory as pa

group by pa.PolicyID) as pahm

inner join dbo.PolicyAgencyHistory as pah

on pah.PolicyAgencyHistoryID = pahm.PolicyAgencyHistoryID

on pahm.PolicyID = p.PolicyID

where p.BoundDate

between @StartingDate

and @EndingDate

group by cast(p.BoundDate as date),

pah.AgencyID,

p.CompanyID,

p.StateID,

pva.FormType,

ad.CountyID,

pva.TerritoryCode

)

**FastTrack**

with FastTrackClaims as (

select c.claimid

, ca.AdjusterSystemUserID

, ca.CreatedDate as AssignedOn

,ft.InspectedOn

,ft.ContactedOn

,ca.ClaimAssignmentID

,ft.FastTrackID

,ft.RemovedOn as FTRemovedOn

,case when ft.FastTrackID is not null and c.ClaimStatusID not in (2, 9, 10, 11, 12, 17) and coalesce(ft.PendingEMSInvoice,0) = 1 then 1 else 0 end as PendingEMSInvoice

,c.ClaimStatusID

,c.LossTypeID

,ROW\_NUMBER() OVER(PARTITION BY c.ClaimID,ca.AdjusterSystemUserID ORDER BY c.ClaimID) AS DuplicateCount

from dbo.ClaimDetailsAndAssignments ca

left join claim c on c.claimid = ca.claimid

left join claim.FastTrack ft on ft.claimid = c.claimid

left join systemusers su on su.SystemUserID = ca.AdjusterSystemUserID

where cast(ca.CreatedDate as date) between @StartDate and @EndDate and **ca.AdjustingCompanyContactID = 2500806** and ca.ClaimDetailTypeID = 1 and ca.Active = 1

**/\*2500806 is our internal Adjusting Co account\*/ (same as UAC)**

**MISC JOINS**

from AtlasBridge.dbo.PolicyAttributes as pa

--inner join dbo.States as st

-- on st.StateCode = pa.PolicyLocationState

inner join AtlasBridge.dbo.PolicyRelationships as pr

inner join AtlasBridge.dbo.Producers as pd

inner join dbo.Agency as ag

--inner join Marketing.dbo.AgencyReps as ar

-- on ar.AgencyID = ag.AgencyID

on ag.AgencyID = pd.AgencyID

on pd.userID = pr.ProducerID

on pr.PolID = pa.PolID

**OVER PARTITION**

**RANK**

rank() over (partition by b.ClaimID order by cd.ClaimDetailID) as SortOrder

rank() over (partition by datepart(year, pa.AppSubmitDate), datename(month, pa.AppSubmitDate) order by count(pa.PolID) desc , sum(pa.BillingTotalEstPremiumAmount)desc) as RankByMonthPolicys

**ROW NUMBER**

row\_number() over (partition by c.ClaimID, ca.AdjusterSystemUserID order by c.ClaimID) as DuplicateCount

row\_number() over (partition by ar.RepID order by sum(pva.TotalPremium) desc) as Ranking

**RATIOS**

**LOSS RATIO**

case when isnull(l.NonCatastrophe, 0) = 0 then 0 else

case when isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0) = 0 then 1

else l.NonCatastrophe / (isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0))

end end as Non\_Cat\_Loss\_Ratio,

case when isnull(l.Catastrophe, 0) = 0 then 0 else

case when isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0) = 0 then 1

else l.Catastrophe / (isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0))

end end as Cat\_Loss\_Ratio,

case when isnull(l.NonCatastrophe, 0) + isnull(l.Catastrophe, 0) = 0 then 0 else

case when isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0) = 0 then 1

else (l.NonCatastrophe + l.Catastrophe) / (isnull(pue.Unearned, 0) + isnull(dwp.Premium, 0) - isnull(cue.Unearned, 0))

end end as Loss\_Ratio

**IMPORTANT TABLES**

**ATLAS**

**Main Tables:**

* **Claims**
* **ClaimTransactions**
* **Policy**
* **PolicyVersion**
* **PolicyVersionAttributes**
  + Use **PolicyVersionAttributes.SubTotal** field when calculating New Premium for Marketing
* **WrittenPremium**

**IMPORTANT VIEWS**

* **ClaimDetailsAndAssignments**
  + **Pulls from ClaimDetails and Symbility for complete list of Adjusting companies.**

**Lookup Tables**

* **LossType**
* **States**
* **County**
* **Catastrophe**
* **ClaimStatus**
* **ClaimTransactionTypes**

**MiscTables**

* **Reports**
* **ReportDetail**
* **ReportType**

**CRM Tables**

* **Contact**
* **ContactType**
* **Address**

**COMMON SPs**

* + **dbo.ReportGetClaimsCount**

**COMMON FUNCTIONS**

* **GetExaminerByClaimID (Returns name of internal Examiners)**

**DATES**

* **ReportedDate**
* **CreatedDate**
* **LossDate**
* **BoundDate**
* **ModifiedDate**
* **PrintedDate**
* **DateDraftWasClearedOrVoided**
* **DateAssignedToAdjuster**
* **OpenedDate**

**Adjusting Companies**

procedure [dbo].[reportGetAdjustingCompanies]

**BusinessName AdjustingCoID**

**All 0**

**Aplin Peer & Associates 10265**

**Insurance Claims Adjusters, LP 4227272**

**Noble Claims Services 1123**

**Downes and Associates 22656**

**Coastal 22709**

**Bayside 22711**

**Specialty Claims Consulting 22712**

**UAC 2500806**

**Crawford & Company 3273336**

**J.F. Adjusting 4042739**

**ICA, Inc 4124068**

**Property Loss Specialist, LLC 4167779**

**CPR Insurance 4167799**

**Advanced Adjusting Ltd. 4167815**

**Zengler & Associates, Inc. 4208238**

**Dominguez All Claims Services, Inc 4217737**

**Sue L. Weiss Consulting Co. 4251657**

**Insurance Appraisal Service, Inc 4356457**

**BELFOR Property Restoration 4364411**

**ContactTypeID**

**ContactTypeID ContactTypeCode ContactTypeDesc**

**1 Employee Employee**

**2 Agency Producer**

**3 AdditionalInterest Additional Insured**

**4 Insured Insured**

**5 AdditionalInterest Mortgagee**

**6 Agency Agency Principal**

**7 Other Other**

**8 AdditionalInterest Finance Company**

**9 Claims 3rd Party Claimant**

**10 Claims Witness**

**11 Claims Adjusting Company**

**12 Claims Spouse**

**13 Claims Parent/Guardian**

**14 Claims Relative**

**15 Attorney Plaintiff's Attorney**

**16 Claims Public Adjuster**

**17 AdditionalInterest Condo Assocation**

**18 Inspection Home Inspector**

**19 AdditionalInterestOld 2nd Mortgagee**

**20 Claims Vendor**

**21 Attorney Defense Attorney**

**22 Claims Transcription Company**

**23 Claims Water Extraction**

**24 Claims Mold remediation**

**25 Claims Contractor Company**

**26 U/W Wind Mit Inspection Company**

**27 Contest Contestant**

**28 Accounting 1099 Miscellaneous Recipient**

**29 Accounting 1099 Dividend Recipient**

**30 Accounting 1096 Filer**

**32 Reinsurance Intermediary**

**33 Reinsurance Reinsurer**

**34 AdditionalInterest Additional Interest**

**ClaimAccountingTransactionTypeId**

**ClaimAccountingTransactionTypeId Description ClaimAccountId Comments**

**1 Open Loss Reserve Loss Reserve For opening loss reserve for the first time (increments Reported Claim Count).**

**2 Change Loss Reserve Loss Reserve For adjusting reserves except for opening or closing. NOTE: may not reduce loss reserve <= 0.**

**3 Close Loss Reserve Loss Reserve For closing loss reserve (increments Closed with/without Payment Count). Must decrease loss reserve to zero.**

**4 Re-open Loss Reserve Loss Reserve For re-opening loss reserve (does not increment Reported Claim Count).**

**5 Loss Paid Loss Paid Increases Paid Losses.**

**6 Void Loss Paid Loss Paid Decreases Paid Losses.**

**7 LAE Paid Loss Expense Paid Increases LAE Paid.**

**8 VOID LAE Paid Loss Expense Paid Decreases LAE Paid.**

**9 Refund Loss Paid Loss Paid Decreases Paid Losses (monies paid to us).**

**10 Returned Refund Loss Paid Loss Paid Increases Paid Losses (e.g. bad check).**

**11 Refund LAE Paid Loss Expense Paid Decreases LAE Paid.**

**12 Returned Refund LAE Paid Loss Expense Paid Increases LAE Paid (e.g. bad check).**

**13 Subrogation Received Loss Paid Decreases Paid Loss.**

**14 Returned Subrogation Received Loss Paid Increases Paid Loss.**

**15 Open Subrogation Reserve Subrogation Reserve Indicates claim is in active subrogation.**

**16 Adjust Subrogation Reserve Subrogation Reserve Used when partial subrogation received.**

**17 Close Subrogation Reserve Subrogation Reserve Indicates claim no longer in subrogation.**

**18 Salvage Received Loss Paid Reduces paid loss like subrogation received.**

**19 Returned Salvage Received Loss Paid Yet another bounced check transaction.**

**20 OPEN LAE RESERVE Lae Reserve For opening lae reserve for the first time**

**21 CHANGE LAE RESERVE Lae Reserve For adjusting lae reserves except for opening or closing.**

**22 CLOSE LAE RESERVE Lae Reserve For closing lae reserve.**

**23 REOPEN LAE RESERVE Lae Reserve For re-opening lae reserve.**

**24 DELETE LOSS RESERVE Loss Reserve deletes the loss reserve**

**25 REVERSE VOID LOSS PAID Loss Paid reverses a voided loss payment**

**26 REVERSE VOID LAE PAID Loss Expense Paid reverses a voided lae payment**

**27 RETURN LOSS PAID Loss Paid Mark loss payment as returned**

**28 PENDING RETURN LOSS PAID Loss Paid Mark loss payment as pending return**

**29 RETURN LAE PAID Loss Expense Paid Mark lae payment as returned**

**30 PENDING RETURN LAE PAID Loss Expense Paid Mark lae payment as pending return**

**31 APPLIED PARTIAL REFUND Loss Paid Apply partial refund to loss payment**

**32 APPLIED FULL REFUND Loss Paid Apply full refund to loss payment**

**33 REISSUED LOSS PAID Loss Paid Mark loss payment as reissued.**

**ClaimStatusID Description (**dbo.ClaimStatusHistory table)

**1 Open**

**2 Closed**

**3 Litigation**

**4 Appraisal**

**5 Subrogation**

**6 Supplement**

**7 Mediation**

**8 Reinspection**

**9 Withdrawn**

**10 Denied**

**11 Below Deductible**

**12 Erroneous**

**13 SIU**

**14 EUO**

**15 SIU - MM**

**16 Settled**

**17 Lack of Interest**

**18 SIU - 626**

**DRAFT STATUS DESC (dbo.ClaimDraftStatus)**

**ClaimDraftStatusID Description**

**1 PENDING**

**2 QUEUED**

**3 PRINTED**

**4 VOIDED**

**5 CLEARED**

**6 RETURNED**

**7 PENDING RETURN**

**8 REQUIRES REVIEW**

**9 FULLY REFUNDED**

**10 PARTIALLY REFUNDED**

**11 ON HOLD - PENDING REVIEW**

**12 REISSUED**

**13 PRESENTED FOR PAYMENT**

**CLAIM DETAIL TYPES**

**ClaimDetailTypeId Description**

**1 Initial Assignment**

**2 Reassignment**

**3 Sub Assignment**

**4 Quality Review**

**5 Internal Assignment**

**TRANSACTION TYPES**

**ClaimTransactionTypeID Description Type**

**1 Building Advance P**

**2 Building Final P**

**3 Liability Expense P**

**4 Supplement P**

**5 Contents Advance P**

**6 Contents Final P**

**7 Emergency Repairs P**

**8 LAE E**

**9 ALE P**

**10 Initial Reserve R**

**11 Reserve Adjustment R**

**12 Dwelling Advance P**

**13 Dwelling Final P**

**14 EMS P**

**15 Final Settlement P**

**16 Attorney Services E**

**17 Engineer Services E**

**18 Rec. Dep. P**

**19 Appraisal Services E**

**20 Loss Assessment P**

**21 Property Damage P**

**22 Other (Claim) P**

**23 Other (LAE) E**

**24 Final Reserve R**

**25 Rec. Dep. Reserve D**

**26 Loss of Use P**

**27 Loss of Rent P**

**28 Mold Remediation P**

**29 Other Structures P**

**30 Business Advance P**

**31 Other (Advance) P**

**32 Unknown X**

**33 Medical Payment P**

**34 Other (Mold Testing) E**

**35 Subrogation Reserve S**

**36 Subrogation Payment S**

**37 Umpire Fee E**

**38 Law & Ordinance P**

**39 Professional Service E**

**40 Deductible Refund P**

**41 Building Undisputed Amount P**

**42 Contents Undisputed Amount P**

**43 Appraisal Settlement P**

**44 Mediation Settlement P**

**45 Mediation Services E**

**46 Overhead & Profit P**

**47 Expense Refund E**

**48 Partial Settlement P**

**49 Re-Open Loss Reserve R**

**50 Umpire Fee (Other) P**

**51 Appraisal Settlement - Dwelling P**

**52 Appraisal Settlement - Other Structures P**

**53 Appraisal Settlement - Contents P**

**54 Appraisal Settlement - Loss of Use P**

**ClaimReserveTypeID ( ClaimReserve table)**

* **There is no ClaimReserve description table**

**WORKFLOW CONCERN CODES**

**ConcernCodeID TypeID Description**

**13 3 Other**

**14 3 PI Verified**

**15 3 Report due/7/15/30/Final**

**16 3 Awaiting final report**

**17 3 Awaiting contractor estimates**

**18 3 Awaiting Public Adjuster Estimates**

**19 3 Awaiting POL**

**20 3 Awaiting correspondence Response**

**21 3 Awaiting Expert Report**

**22 3 Awaiting Revision**

**23 3 Diary Review**

**24 3 Remind Me**

**25 3 Remind Me: 7/15/30/FINAL REPORT DUE**

**26 3 Remind Me: Check For Status Update**

**27 3 Remind Me: Check For Executed Release and Dismissal**

**28 3 Remind Me: Check Correspondence Response**

**29 3 Remind Me: Diary Review**

**30 3 Remind Me: Follow up**

**32 4 SIU Referral**

**33 4 SIU Mis-Rep Referral**

**34 3 Reserve Request (OBSOLETE. Please use the reserve requests under Referrals)**

**35 4 ADR - Re-Inspection Request**

**36 4 ISO Request**

**38 3 Missing Condo Association Information**

**39 4 SIU Sinkhole Referral**

**40 3 SIU/MM File returned to Adjuster**

**43 4 Draft Request - Void & Re-Issue**

**44 4 Draft Request - Void ONLY**

**45 4 Draft Request - Undisputed Draft**

**46 4 Draft Request - Proof of Losses**

**47 4 Draft Request - Ready To Print**

**48 4 SIU 626**

**49 4 EMS Review**

**50 4 SIU Claim Processing**

**51 4 EUO Request**

**52 3 10% SinkHole Claim Requires Management Release**

**53 4 Subrogation**

**55 4 PA MIDSTREAM REVIEW**

**56 3 45-day Claim Age Alert**

**57 3 Lawsuit Response**

**58 3 Demand Response**

**59 3 Policy Disclosure Response**

**60 4 Draft Request - Draft Status**

**61 3 Claims Help**

**62 4 Draft Request - Courtesy Call**

**63 4 Draft Request - Action Required**

**64 4 U/W - Correct Address**

**65 4 U/W - Correct Dog Breed**

**66 4 U/W - Correct Endorsement**

**67 4 U/W - Correct Name**

**68 4 U/W - Correct Mortgagee**

**69 4 U/W - Correct Policy**

**70 4 U/W - Multiple Claims**

**71 4 U/W - Unacceptable breed of dogs on premise**

**72 4 U/W - No Insurable Interest**

**73 4 U/W - Failure to mitigate damages**

**74 4 U/W - Unoccupied/Vacant Property**

**75 4 U/W - Pre-Existing Damage**

**76 4 U/W - Under the authority of a homeowner or co-op**

**77 4 U/W - Business Exposure/Business being operated on premise**

**78 4 U/W - Relinquished per settlement**

**79 4 U/W - Substantial Change in Risk**

**80 4 U/W - Deferred maintenance**

**81 4 U/W - Incorrect Statement on the Application**

**83 4 ADR - Appraisal Demand**

**84 4 ADR - Mediation Demand**

**85 4 ADR - Civil Remedy Notice (CRN)**

**86 4 ADR - DFS Complaint**

**87 4 ADR - BBB Complaint**

**88 4 Claim Processing Queue - PAYMENT**

**89 4 Claim Processing Queue - NON PAYMENT**

**90 4 Document Processing Queue - FINAL**

**91 4 Document Processing Queue - INTERIM**

**92 4 Draft Req - Attorney Invoice**

**93 4 Draft Req - Vendor Invoice**

**94 4 Draft Req - EMS Invoice**

**96 4 Field Assignment Request**

**97 4 Document Processing Queue - FINAL (FastTrack)**

**99 4 Claim Processing Queue - FAST TRACK**

**100 4 Document Processing Queue - INTERIM (FastTrack)**

**104 4 Reserve Request ( 5 - 10K )**

**105 4 Reserve Request ( 10 - 25K )**

**106 4 Reserve Request ( Over 25K )**

**107 4 Supervisor Denial Review**

**108 4 Letter Request - Contact / Document**

**109 4 Letter Request - Lack Of Interest**

**110 4 Supervisor Payment Review / Reserves**

**111 5 New Contest Producer**

**112 5 Updated Contest Producer Information**

**113 4 Document Processing Queue - FINAL (Represented)**

**114 4 Document Processing Queue - INTERIM (Represented)**

**115 5 New Contest Entry**

**116 5 Updated Contest Entry Information**

**117 5 New Contest Agency**

**118 5 Updated Contest Agency Information**

**119 5 New Claim Vendor**

**120 5 Updated Claim Vendor Information**

**121 4 Document Processing Queue - Supplement**

**122 4 Represented Unit- Reopen/Negotiate Settlement**

**123 4 Director Payment Review / Reserves**

**124 4 VP Claims Payment Review / Reserves**

**125 5 Contest Producer W9 Requirement**

**126 4 Premium Refund Verification**

**127 5 Contest Producer Validation Error**

**128 4 SIU PA/ATTNY Late Reporting**