CSA Design

Contents

[Introduction 7](#_Toc161135226)

[Design Consideration for Programming 7](#_Toc161135227)

[Technology Use Guidelines: 7](#_Toc161135228)

[Version Control 7](#_Toc161135229)

[Clone 7](#_Toc161135230)

[Branching 7](#_Toc161135231)

[Merging (Pull Request) 8](#_Toc161135232)

[Frontend Framework (Click for More Info) 8](#_Toc161135233)

[CSA Server Side Design 8](#_Toc161135234)

[HTTP Request Life Cycle 8](#_Toc161135235)

[DAL layer example 8](#_Toc161135236)

[BLL layer example 9](#_Toc161135237)

[API Controller layer example 9](#_Toc161135238)

[Entities (Classes used to store query result) 10](#_Toc161135239)

[SaleInfoFull 10](#_Toc161135240)

[SaleInfoFullExpenseFl 11](#_Toc161135241)

[UpdatedSaleInfo 11](#_Toc161135242)

[ModifiedBy 12](#_Toc161135243)

[Occupancy 12](#_Toc161135244)

[LeaseInfo 12](#_Toc161135245)

[Lease 13](#_Toc161135246)

[PropertyInfo 15](#_Toc161135247)

[MailingName 15](#_Toc161135248)

[PropChar(From CSA Database) 16](#_Toc161135249)

[PropCharVacantLand(VacantLand) 16](#_Toc161135250)

[RentInfo 17](#_Toc161135251)

[NRAOccupancy 17](#_Toc161135252)

[APN 17](#_Toc161135253)

[IncomeExpenseAnalysis 17](#_Toc161135254)

[CSAExpenseCode 18](#_Toc161135255)

[CSAExpenseEntity 18](#_Toc161135256)

[RptSource 19](#_Toc161135257)

[CSARptAdjustments 19](#_Toc161135258)

[CSAComment 20](#_Toc161135259)

[AdjustmentsForAllAdjustmentCodes 20](#_Toc161135260)

[FieldCode 20](#_Toc161135261)

[AllAPNBooks 20](#_Toc161135262)

[AntiUseCodesNotForLandUse 21](#_Toc161135263)

[APN 21](#_Toc161135264)

[CSASourceData 21](#_Toc161135265)

[IncomeAnalysis 22](#_Toc161135266)

[CSAAmount 23](#_Toc161135267)

[ExpenseTotal 23](#_Toc161135268)

[PotentialGrossIncome 23](#_Toc161135269)

[SourceDesc 23](#_Toc161135270)

[WorkerInfo 23](#_Toc161135271)

[VLPropetyInfo 23](#_Toc161135272)

[PCBuilding 24](#_Toc161135273)

[PCEvent 25](#_Toc161135274)

[VLPropChar 26](#_Toc161135275)

[LeaseComment 26](#_Toc161135276)

[LeaseUseCategory 27](#_Toc161135277)

[LeaseUseSubCat 27](#_Toc161135278)

[wksAPN 27](#_Toc161135279)

[Models (Used to store in and out object, not directly database query result) 27](#_Toc161135280)

[UpdatedSaleInfoWithCIAnalysis 27](#_Toc161135281)

[UpdatedIncomeAnalysis 28](#_Toc161135282)

[CSASaleInfo 28](#_Toc161135283)

[CSAProperty 29](#_Toc161135284)

[CSAUpdatedSourceData 29](#_Toc161135285)

[CSASaleExpenses 30](#_Toc161135286)

[CSAExpense 31](#_Toc161135287)

[CSASaleAdjustments 32](#_Toc161135288)

[CSAAdjustment 33](#_Toc161135289)

[IncAnalysisValInd 33](#_Toc161135290)

[LeaseBO 34](#_Toc161135291)

[LeaseDetailBO 36](#_Toc161135292)

[LeaseCommentBO 36](#_Toc161135293)

[RPCSaleInfo 36](#_Toc161135294)

[RPCPropChar 37](#_Toc161135295)

[RPCVLPropChar 37](#_Toc161135296)

[wkSheetNewImp 38](#_Toc161135297)

[Stored Procedure Queries 38](#_Toc161135298)

[IE\_GET\_CSA\_SALE\_INFO\_FULL (P\_CSA\_ID) 38](#_Toc161135299)

[IE\_GET\_CSA\_PAGE\_TITLE(P\_CSA\_TYPE), mention field code. 42](#_Toc161135300)

[IE\_GET\_CSA\_TOTAL\_NRA(P\_CSA\_ID) 42](#_Toc161135301)

[IE\_GET\_CSA\_SQFT\_LEASED(P\_CSA\_ID) 42](#_Toc161135302)

[IE\_GET\_CSA\_AMOUNT(P\_CSA\_ID, P\_AMOUNT\_CD) 42](#_Toc161135303)

[IE\_GET\_CSA\_EXPENSE\_CODE(P\_CSA\_ID) 43](#_Toc161135304)

[IE\_GET\_CSA\_AMOUNT\_ACTUAL\_EXPENSE(P\_CSA\_ID) 43](#_Toc161135305)

[IE\_GET\_CSA\_PROP\_INFO(P\_CSA\_ID) 44](#_Toc161135306)

[IE\_GET\_CSA\_COMMENTS(P\_CSA\_ID) 46](#_Toc161135307)

[IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE(P\_PROPERTY\_ID, P\_AS\_OF\_DATE) 46](#_Toc161135308)

[IE\_GET\_CSA\_TAX\_RATE(P\_PROPERTY\_ID, P\_EVENT\_DATE, P\_ROLL\_YEAR) 46](#_Toc161135309)

[IE\_GET\_CSA\_LEASE\_INFO(P\_CSA\_ID) 47](#_Toc161135310)

[IE\_GET\_CSA\_TOTAL\_RENT(P\_CSA\_ID) 49](#_Toc161135311)

[IE\_GET\_CSA\_TOTAL\_RENT\_EXC\_LEASE 50](#_Toc161135312)

[IE\_GET\_CSA\_NRA\_OCCUPANCY(P\_CSA\_ID) 50](#_Toc161135313)

[IE\_DEL\_CSA\_LEASE(P\_LEASE\_ID) 50](#_Toc161135314)

[IE\_GET\_CSA\_ANALYSIS(P\_CSA\_ID) 50](#_Toc161135315)

[IE\_GET\_CSA\_ALL\_APN\_IN\_DOC(P\_CSA\_ID) 51](#_Toc161135316)

[IE\_GET\_CSA\_PROP\_CHAR(P\_CSA\_ID) 51](#_Toc161135317)

[IE\_GET\_CSA\_INCOME\_EXPENSE\_ANALYSIS(P\_CSA\_ID) 52](#_Toc161135318)

[IE\_GET\_CSA\_EXPENSE(P\_CSA\_ID) 55](#_Toc161135319)

[IE\_GET\_CSA\_RPT\_SOURCE(P\_CSA\_ID) 56](#_Toc161135320)

[IE\_GET\_CSA\_RPT\_ADJ(P\_CSA\_ID) 57](#_Toc161135321)

[IE\_GET\_CSA\_ADJ\_FOR\_ALL\_ADJ\_CODES(P\_CSA\_ID) 60](#_Toc161135322)

[IE\_GET\_CSA\_AMOUNT\_CODES(P\_AMOUNT\_TYPE) 61](#_Toc161135323)

[IE\_GET\_FIELD\_CODE(P\_FIELD\_ID) 61](#_Toc161135324)

[IE\_GET\_CSA\_ANTICIPATED\_USE\_CODES\_NOT\_FOR\_LAND\_USE 61](#_Toc161135325)

[IE\_GET\_CSA\_ALL\_APN\_IN\_DOC(P\_CSA\_ID) 61](#_Toc161135326)

[IE\_GET\_CSA\_SOURCE(P\_CSA\_ID) 62](#_Toc161135327)

[IE\_GET\_CSA\_INCOME\_ANALYSIS(P\_CSA\_ID) 63](#_Toc161135328)

[IE\_GET\_CSA\_ADJUST\_AMOUNT 64](#_Toc161135329)

[IE\_GET\_CSA\_PCT\_OCCUP\_THRESHLD 65](#_Toc161135330)

[IE\_GET\_CSA\_EXPENSE\_TOTAL(P\_CSA\_ID) 65](#_Toc161135331)

[IE\_GET\_CSA\_POTENTIAL\_INCOME(P\_CSA\_ID) 65](#_Toc161135332)

[IE\_GET\_CSA\_ALL\_WORKER 65](#_Toc161135333)

[IE\_GET\_CSA\_PROP\_INFO\_VACANT\_LAND(P\_CSA\_ID) 66](#_Toc161135334)

[IE\_GET\_CSA\_LEASE(P\_LEASE\_ID) 67](#_Toc161135335)

[IE\_GET\_CSA\_LEASE\_COMMENTS(P\_LEASE\_ID) 70](#_Toc161135336)

[IE\_GET\_CSA\_LEASE\_USE\_CATEGORY() 70](#_Toc161135337)

[IE\_GET\_CSA\_LEASE\_USE\_SUB\_BY\_USE\_CAT(P\_USE\_CAT) 71](#_Toc161135338)

[IE\_GET\_PC\_BUILDING\_AS\_OF\_DATE(int[] PropId, DateTime AsofDate) 71](#_Toc161135339)

[IE\_GET\_PCE\_AS\_OF\_DT 73](#_Toc161135340)

[IE\_GET\_USE\_CD\_AS\_OF\_DATE(P\_PROP\_ID, P\_AS\_OF\_DATE) 75](#_Toc161135341)

[IE\_GET\_SITUS\_CITY(P\_PROP\_ID) 75](#_Toc161135342)

[IE\_GET\_VL\_PROP\_CHAR\_AS\_OF\_DATE(P\_PROPERTY\_ID, P\_AS\_OF\_DATE) 76](#_Toc161135343)

[IE\_DEL\_CSA\_WKS\_APN(P\_CSA\_WKS\_NUM) 76](#_Toc161135344)

[IE\_GET\_CSA\_WKS\_APN\_LIST(P\_CSA\_WKS\_NUM) 76](#_Toc161135345)

[Helper Class 78](#_Toc161135346)

[DB2HelperCustom Class 78](#_Toc161135347)

[DB2Helpers Class 78](#_Toc161135348)

[Helpers Class 78](#_Toc161135349)

[DAL Layer: 79](#_Toc161135350)

[CSASale Class 79](#_Toc161135351)

[CSAProperty Class 82](#_Toc161135352)

[CSALease Class 91](#_Toc161135353)

[CSASourceInput Class 92](#_Toc161135354)

[CSAVacantLandInput Class 93](#_Toc161135355)

[CSAShared Class 93](#_Toc161135356)

[CSAIncAnalysisValInd Class 94](#_Toc161135357)

[CSALeaseInfo Class 95](#_Toc161135358)

[CSALease Class 96](#_Toc161135359)

[wkSheetNewImps Class 97](#_Toc161135360)

[BLL Layer 98](#_Toc161135361)

[Note: 98](#_Toc161135362)

[CSASale Class 99](#_Toc161135363)

[CSAProperty class 105](#_Toc161135364)

[CSALease class 106](#_Toc161135365)

[CSASource Class 108](#_Toc161135366)

[CSAShared Class 110](#_Toc161135367)

[IncAnalysisValIndBLL Class 111](#_Toc161135368)

[CSAVacantLandInput Class 121](#_Toc161135369)

[CSALeaseInfo Class 121](#_Toc161135370)

[wkSheetNewImps Class 122](#_Toc161135371)

[API Controllers 123](#_Toc161135372)

[For CI Sales Analysis (Controllers => CISalesAnalysisController.cs) 123](#_Toc161135373)

[For CSALeaseInfo 124](#_Toc161135374)

[IE\_GET\_CSA\_LEASE 124](#_Toc161135375)

[IE\_ADD\_CSA\_LEASE\_NO\_RENT\_COMP 124](#_Toc161135376)

[For CSAPropertyInfo 124](#_Toc161135377)

[IE\_GET\_CSA\_PROP\_INFO\_VACANT\_LAND 124](#_Toc161135378)

[For wkSheet 125](#_Toc161135379)

[IE\_GET\_CSA\_WKS\_APN\_LIST 125](#_Toc161135380)

[Exception handling and error logging 125](#_Toc161135381)

[Exception handling 125](#_Toc161135382)

[Error logging 125](#_Toc161135383)

[New Exceptions 127](#_Toc161135384)

[CSA Client Side Design 127](#_Toc161135385)

[Core Technologies 127](#_Toc161135386)

[Angular Components 127](#_Toc161135387)

[PBPage (Testing page) 127](#_Toc161135388)

[Default page (Loading page) 127](#_Toc161135389)

[ToolBar 128](#_Toc161135390)

[popSubjectAPN page 128](#_Toc161135391)

[popModifyReason page 128](#_Toc161135392)

[CISalesAnalysis page 128](#_Toc161135393)

[VacantLandInput page 144](#_Toc161135394)

[Title Bar 145](#_Toc161135395)

[Control Bar 145](#_Toc161135396)

[CSAPropertyInfo 146](#_Toc161135397)

[Comments 147](#_Toc161135398)

[Loading Overlay page 147](#_Toc161135399)

[CSALeaseEdit page 148](#_Toc161135400)

# Introduction

The purpose of this Software Design Document (SDD) is to provide a comprehensive and detailed overview of the design and architecture of the Sales Analysis system of the CSA. This document serves as a blueprint for the development team, guiding the implementation of the software and ensuring a clear understanding of its structure and functionality.

# Design Consideration for Programming

**What to do:**

* Keep separation of concerns, write single responsibility modules.
* Try to take advantage of Async programming feature of .NET especially for I/O and DB querying.

**Not to do:**

* SQL direct injections
* Directly commit on Github Main branch. Please branch out and use PR to merge.

# Technology Use Guidelines:

IDE: Visual Studio 2022 (preferred) should be used.

Source Control: Github

Web Language: Angular 16.0.6

API Language: Microsoft ASP.NET Core 7.0 (RESTful)

Database: DB2

Error logging backend: MS Sql Server

# Version Control

Github is used for version control and Github Action will be used to implement necessary work flows. Below are setup requirements as well as rules to follow to ensure a relatively linear version history.

GitHub Desktop is mandatory to install.

## Clone

Remember to Clone repo into a NON CLOUD SYNC folder (keep in mind to avoid the iCloud backed up folders).

## Branching

Branching out when working with code base. AVOID directly working and committing to “Main” branch at all time.

## Merging (Pull Request)

Before start a PR, make sure to first pull from the origin/main to stay up to date, in Github Desktop click Branch -> Update From Main, then resolve conflicts if any on current branch. Once conflicts are resolved, create a PR by going Branch -> Create Pull Request. Feel free to compare difference by preview pull request.

In PR creation Github page, make sure to request teammates to do a code view prior to merge to main.

# [Frontend Framework (Click for More Info)](https://acgovt.sharepoint.com/:w:/r/sites/IMPROVENET/Design/CSA%20Frontend%20Framework%20Design.docx?d=w007cb72ff81b42d58e0871136e12416c&csf=1&web=1&e=e8G8X8)

# CSA Server Side Design

## HTTP Request Life Cycle

\*Entities will be used as DTOs out as well as store database extractions. Models will be used as DTOs in to gather HTTP request data.

\*Try the best to apply asynchronous programming in order to improve overall program performance.

\*Register BLL class objects and DAL class objects to IoC containers to achieve loose coupling standard.

### DAL layer example

**Each function within a DAL layer class should be referring to one stored procedure and one only.**

public Task<SaleInfoTable> GetSaleInfo(int pCSAId)

{

IDB2HelperCustom \_db2HelperCustom = new DB2HelperCustom();

DB2Connection dB2Connection = \_db2HelperCustom.Connection;

List<DB2Parameter> parameterList = new List<DB2Parameter>();

DB2Parameter param1 = new DB2Parameter("P\_CSA\_ID", DB2Type.Integer);

param1.Value = pCSAId;

parameterList.Add(param1);

try

{

//\_db2HelperCustom is a data helper service that will be added to IoC container

DB2DataReader dataReader = \_db2HelperCustom.ExecuteReader(Properties.Resources. IE\_CSA\_GET\_CSA\_SALE\_INFO, System.Data.CommandType.StoredProcedure, dB2Connection, parameterList);

DB2Helpers dB2Helpers = new DB2Helpers();

SaleInfoTable tableDetail = await dB2Helpers.GetEntityFromReader<SaleInfoTable>(ref dataReader);

dataReader.Close();

return tableDetail;

}

catch

{

throw;

}

finally

{

\_db2HelperCustom.CloseConnection(dB2Connection);

}

}

\*GetEntityFromReader is a helper function to assign values extracted from stored procedures to each entity class members.

### BLL layer example

public Task<SaleInfoTable> GetSaleInfo(int pCSAId)

{

SaleInfoTable SaleInfo = null;

//Use try catch block for data reading actions to catch any exceptions

try

{

SaleInfo = await CSA.DAL.CISaleAnalysis.GetSaleInfo(pCSAId);

}

catch

{

throw;

}

//TODO: Other business logic if required

Return SaleInfo;

}

### API Controller layer example

[HttpGet]

[Route("api/[controller]/GetCSASaleInfo")]

public async Task<IActionResult> GetAssessmentValueDetail([FromBody] SaleInfoModel saleInfoModel)

{

int rowCount = 0;

DateTime startTime = DateTime.Now;

string eventId = "";

try

{

//This CSASaleBLL should be injected in the IoC container

var task = await CSASaleBLL.GetSaleInfo(saleInfoModel);

rowCount = 1;

return Ok(task);

}

catch (Exception ex)

{

eventId = string.Empty;

eventId = \_loggerService.WriteExceptionLog(ref ex, null, HttpContext, this.HttpContext.Request.Path.ToString(), Utility.GetKeyValuePairs(valueDetailModel));

return this.StatusCode(555, eventId);

}

finally

{

DateTime endTime = DateTime.Now;

\_loggerService.LogApiSpCall(LoggingEnums.LogType.API\_Call, startTime, endTime, this.HttpContext.Request.Path.ToString(), rowCount, \_commonService.GetUserFromToken(this.HttpContext.Request.Headers["Authorization"].ToString()), eventId, Utility.GetKeyValuePairs(valueDetailModel));

}

}

## Entities (Classes used to store query result)

To increase reusability and decrease code redundancy. Some entities are reused multiple times, for an update stored procedure (MOD). If the entity contains extra variable that was not supposed to be updated. Set it to null and the data will not be used as parameter when using DB2HelperCustom.ExecuteNonQuery. If the value needs to be updated in the database to null, set the value to “**DBNull**” before applying DB2HelperCustom’s ExcuteNonQuery function.

**CISalesAnalysis**

### SaleInfoFull

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| CSA\_TYPE | System.Int16 | SMALLINT |
| DOC\_PREFIX | System.String | CHAR |
| DOC\_SERIES | System.Int32 | INTEGER |
| PROPERTY\_ID | System.Int32 | INTEGER |
| USE\_CD | System.Int16 | SMALLINT |
| CSA\_PROP\_USE\_CD | System.Int16 | SMALLINT |
| CSA\_PROP\_USE\_DETL | System.String | VARCHAR |
| ANTICIPATED\_USE\_CD | System.Int16 | SMALLINT |
| COND\_AT\_SALE\_CD | System.Int16 | SMALLINT |
| FULL\_EXPENSES\_FL | System.String | CHAR |
| RELEASE\_DT | System.DateTime | DATE |
| BROKER\_INVOLVED\_FL | System.String | CHAR |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| TOT\_LOT\_SIZE | System.Int32 | INTEGER |
| SHAPE\_CD | System.Int16 | SMALLINT |
| BENCHMARK\_RATE\_CD | System.Int16 | SMALLINT |
| ZONING\_CD | System.String | VARCHAR |
| ZONING\_AGENCY\_CD | System.Int16 | SMALLINT |
| BUY\_SELL\_REL\_FL | System.String | CHAR |
| BUY\_SELL\_REL\_DESC | System.String | CHAR |
| PUR\_PREDATE\_BY\_OPT | System.String | CHAR |
| PUR\_PRED\_CONT\_SALE | System.String | CHAR |
| PREDATE\_CONT\_DATE | System.DateTime | DATE |
| PCT\_OWNER\_OCCUP | System.Int32 | INTEGER |
| SUPRV\_APPROVED\_FL | System.String | CHAR |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| EVENT\_TS | System.DateTime | TIMESTMP |
| PRINT\_PARCEL | System.String | CHAR |
| PROP\_USE\_TEXT? | System.String | VARCHAR |
| COND\_AT\_SALE\_TEXT? | System.String | VARCHAR |
| BENCHMARK\_NAME? | System.String | VARCHAR |
| CSA\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| SUPRV\_APPR\_WKR | System.String | VARCHAR |
| ANTICIPATED\_USE\_CD | System.Int16 | SMALLINT |
| SUPRV\_APPROVED\_FL | System.String | CHAR |
| SUPRV\_APPR\_TEXT | System.String | CHAR |
| SUPRV\_APPR\_UPD\_WKR | System.String | CHAR |
| SUPRV\_APPR\_UPD\_TS | System.DateTime | TIMESTMP |
| ANTI\_USE\_CD\_NAME | System.String | CHAR |
| USE\_NAME\_SHORT | System.String | CHAR |
| LAND\_USE\_FL | System.String | CHAR |
| USE\_AS\_IMPROVE\_FL | System.String | CHAR |
| BUYER\_NAME | System.String | VARCHAR |
| SELLER\_NAME | System.String | VARCHAR |

### SaleInfoFullExpenseFl

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| UPDATE\_WORKER | System.String | CHAR |
| FULL\_EXPENSE\_FL | System.String | CHAR |

### UpdatedSaleInfo

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| ANTICIPATED\_USE\_CD | System.Int16 | SMALLINT |
| CSA\_PROP\_USE\_DETL | System.String | VARCHAR |
| PCT\_OWNER\_OCCUP | System.Int32 | INTEGER |
| COND\_AT\_SALE\_CD | System.Int16 | SMALLINT |
| BROKER\_INVOLVED\_FL | System.String | CHAR |
| BUY\_SELL\_REL\_FL | System.String | CHAR |
| BUY\_SELL\_REL\_DESC | System.String | CHAR |
| PUR\_PREDATE\_BY\_OPT | System.String | CHAR |
| PUR\_PRED\_CONT\_SALE | System.String | CHAR |
| PREDATE\_CONT\_DATE | System.DateTime | DATE |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |

### ModifiedBy

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| LAST\_MODIFY\_PAGE | System.String | CHAR |
| SOURCE\_MODIFIED\_BY (Worker ID) | System.String | CHAR |

### Occupancy

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| UPDATE\_WORKER | System.String | CHAR |
| PCT\_OCCUPANCY | System.Decimal | DECIMAL |
| CSA\_NET\_RENT\_AREA | System.Int32 | INTEGER |

UpdatedCSAAmount

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| UPDATE\_WORKER | System.String | CHAR |
| AMOUNT\_CD | System.Int16 | SMALLINT |
| AMOUNT\_CSA | System.Decimal | DECIMAL |
| AMOUNT\_OWNERS | System.Decimal | DECIMAL |
| AMOUNT\_TENANT | System.Decimal | DECIMAL |
| AMOUNT\_DESCRIPTION | System.String | CHAR |

SaleInfoSuprvApprove

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| SUPRV\_APPR\_UPD\_WKR (Worker ID) | System.String | CHAR |
| SUPRV\_APPROVED\_FL | System.String | CHAR |

### LeaseInfo

|  |  |  |
| --- | --- | --- |
| LEASE\_ID | System.Int32 | INTEGER |
| CSA\_ID | System.Int32 | INTEGER |
| PROPERTY\_ID | System.Int32 | INTEGER |
| LEASE\_LOCATION | System.String | VARCHAR |
| SITUS\_CITY\_NAME | System.String | CHAR |
| LESSEE\_NAME | System.String | VARCHAR |
| VACANT\_FL | System.String | CHAR |
| NET\_RENTABLE\_AREA | System.Int32 | INTEGER |
| LEASE\_START\_DT | System.DateTime | DATE |
| LEASE\_TERM | System.Int16 | SMALLINT |
| LEASE\_OPTION | System.String | VARCHAR |
| INITIAL\_RENT | System.Decimal | DECIMAL |
| INITIAL\_RENT\_PSF | System.Decimal | DECIMAL |
| STABIL\_RENT | System.Decimal | DECIMAL |
| STABIL\_RENT\_PSF | System.Decimal | DECIMAL |
| CURRENT\_RENT | System.Decimal | DECIMAL |
| CURRENT\_RENT\_PSF | System.Decimal | DECIMAL |
| ANTICIPATED\_RENT | System.Decimal | DECIMAL |
| ANTI\_RENT\_PSF | System.Decimal | DECIMAL |
| RENT\_ADJUSTMENTS | System.Decimal | DECIMAL |
| LEASE\_SUBCATEGORY | System.Int16 | SMALLINT |
| LEASE\_TYPE\_CD | System.Int16 | SMALLINT |
| LEASE\_TYPE\_TEXT | System.String | VARCHAR |
| EXPENSE\_TYPE\_CD | System.Int16 | SMALLINT |
| EXPENSE\_TYPE\_TEXT | System.String | VARCHAR |
| TENANCY\_CD | System.Int16 | SMALLINT |
| INFO\_SOURCE\_CD | System.Int16 | SMALLINT |
| LEASE\_PARKING | System.String | VARCHAR |
| SPRINKLER\_FL | System.String | CHAR |
| LEASE\_CREATED\_BY | System.String | CHAR |
| LEASE\_CREATE\_DT | System.DateTime | DATE |
| LEASE\_MODIFY\_BY | System.String | CHAR |
| LEASE\_MODIFY\_DT | System.DateTime | DATE |
| USE\_FOR\_COMP\_FL | System.String | CHAR |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| WKS\_EXIST\_FL | System.String | CHAR |
| RENT\_PSF | System.Decimal | DECIMAL |
| LEASE\_OPTION\_FL | System.String | CHAR |
| PRINT\_PARCEL | System.String | CHAR |
| SITUS\_STREET\_NUM | System.String | VARCHAR |
| SITUS\_PRE\_DIRECT | System.String | CHAR |
| SITUS\_STREET\_NAME | System.String | VARCHAR |
| SITUS\_STREET\_SFX | System.String | CHAR |
| SITUS\_POST\_DIRECT | System.String | CHAR |
| SITUS\_UNIT\_DESIG | System.String | CHAR |
| SITUS\_UNIT\_NUM | System.String | VARCHAR |
| SITUS\_STATE | System.String | CHAR |
| SITUS\_ZIP\_CD | System.String | CHAR |

### Lease

|  |  |  |
| --- | --- | --- |
| PRINT\_PARCEL | System.String | CHAR |
| LEASE\_ID | System.Int32 | INTEGER |
| PROPERTY\_ID | System.Int32 | INTEGER |
| LEASE\_LOCATION | System.String | VARCHAR |
| SITUS\_CITY\_NAME | System.String | CHAR |
| LESSEE\_NAME | System.String | VARCHAR |
| VACANT\_FL | System.String | CHAR |
| NET\_RENTABLE\_AREA | System.Int32 | INTEGER |
| LEASE\_START\_DT | System.DateTime | DATE |
| LEASE\_TERM | System.Int16 | SMALLINT |
| LEASE\_OPTION | System.String | VARCHAR |
| INITIAL\_RENT | System.Decimal | DECIMAL |
| INITIAL\_RENT\_PSF | System.Decimal | DECIMAL |
| STABIL\_RENT | System.Decimal | DECIMAL |
| STABIL\_RENT\_PSF | System.Decimal | DECIMAL |
| ANTICIPATED\_RENT | System.Decimal | DECIMAL |
| USE\_FOR\_COMP\_FL | System.String | CHAR |
| ANTI\_RENT\_PSF | System.Decimal | DECIMAL |
| RENT\_ADJUSTMENTS | System.Decimal | DECIMAL |
| TENANT\_IMPS\_PSF | System.Decimal | DECIMAL |
| SUBCATEGORY\_NAME | System.Int16 | SMALLINT |
| LEASE\_TYPE\_TEXT | System.Int16 | SMALLINT |
| EXPENSE\_TYPE\_TEXT | System.Int16 | SMALLINT |
| TENANCY\_TEXT | System.Int16 | SMALLINT |
| INFO\_SOURCE\_CD | System.Int16 | SMALLINT |
| LEASE\_PARKING | System.String | VARCHAR |
| SPRINKLER\_FL | System.String | CHAR |
| LEASE\_CREATED\_BY | System.String | CHAR |
| LEASE\_CREATED\_NAME | System.String | VARCHAR |
| LEASE\_CREATE\_DT | System.DateTime | DATE |
| LEASE\_MODIFY\_BY | System.String | CHAR |
| LEASE\_MODIFY\_NAME | System.String | VARCHAR |
| LEASE\_MODIFY\_DT | System.DateTime | DATE |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| CSA\_ID | System.Int32 | INTEGER |
| LEASE\_USE\_NAME | System.Int16 | SMALLINT |
| CURRENT\_RENT | System.Decimal | DECIMAL |
| CURRENT\_RENT\_PSF | System.Decimal | DECIMAL |
| RENT\_ADJ\_TEXT | System.String | CHAR |
| EXPENSE\_BASE\_YR | System.Int16 | SMALLINT |
| TI\_ALLOWANCE | System.Decimal | DECIMAL |
| TI\_TYPE\_CD | System.Int16 | SMALLINT |
| TI\_TYPE\_TEXT | System.String | VARCHAR |
| LEASE\_PARKING\_2 | System.String | CHAR |
| INFO\_SOURCE\_TEXT | System.String | CHAR |
| SUPRV\_APPROVED\_FL | System.String | CHAR |
| SUPRV\_APPR\_UPD\_WKR | System.String | CHAR |
| SUPRV\_APPR\_WKR\_NAME | System.String | VARCHAR |
| SUPRV\_APPR\_UPD\_TS | System.DateTime | TIMESTMP |
| DetLeaseId | System.Int32 | INTEGER |
| IND\_CLR\_HGT | System.Decimal | DECIMAL |
| IND\_OFFC\_AREA\_PCT | System.Decimal | DECIMAL |
| IND\_LAB\_PCT | System.Decimal | DECIMAL |
| IND\_MFG\_PCT | System.Decimal | DECIMAL |
| IND\_WAREHOUSE\_PCT | System.Decimal | DECIMAL |
| IND\_RETAIL\_PCT | System.Decimal | DECIMAL |
| IND\_OTHER\_PCT | System.Decimal | DECIMAL |
| OFFC\_LOAD\_FACTOR | System.Decimal | DECIMAL |
| OFFC\_EXPENSE\_STOP | System.Decimal | DECIMAL |
| RETAIL\_DBA | System.String | VARCHAR |
| RETAIL\_RENT\_PCT | System.Decimal | DECIMAL |
| RENT\_BREAKPOINT | System.Decimal | DECIMAL |
| RETAIL\_STORE\_OCC | System.Int16 | SMALLINT |

### PropertyInfo

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| DOC\_PREFIX | System.String | CHAR |
| DOC\_SERIES | System.Int32 | INTEGER |
| PROPERTY\_ID | System.Int32 | INTEGER |
| USE\_CD | System.Int16 | SMALLINT |
| PRINT\_PARCEL | System.String | CHAR |
| SORT\_PARCEL | System.String | CHAR |
| address | System.String | VARCHAR |
| SITUS\_CITY\_NAME | System.String | VARCHAR |
| SITUS\_STATE | System.String | CHAR |
| ZIP\_CD | System.String | CHAR |
| DOC\_PARCEL\_CNT | System.Int16 | SMALLINT |
| EVENT\_TS | System.DateTime | TIMESTMP |
| IND\_PUR\_PRICE | System.Decimal | DECIMAL |
| ADJ\_SALES\_PRICE | System.Decimal | DECIMAL |
| TRAN\_TAX\_PRICE | System.Decimal | DECIMAL |
| USE\_NAME | System.Int16 | SMALLINT |

### MailingName

|  |  |  |
| --- | --- | --- |
| ADDR\_ID | System.Int32 | INTEGER |
| STD\_ADDR\_CD | System.Int16 | SMALLINT |
| MAILING\_NAME | System.String | VARCHAR |
| CARE\_OF\_NAME | System.String | VARCHAR |
| ATTENTION\_NAME | System.String | VARCHAR |

### PropChar(From CSA Database)

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| ZONING\_AGENCY\_CD | System.Int16 | SMALLINT |
| ZONING\_CD | System.String | VARCHAR |
| ZONING\_NAME | System.String | VARCHAR |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| TOT\_LOT\_SIZE | System.Int32 | INTEGER |
| SBE\_CLASS\_LOW | System.String | CHAR |
| SBE\_CLASS\_HI | System.String | CHAR |
| MVS\_CLASS\_LOW | System.String | CHAR |
| MVS\_CLASS\_HI | System.String | CHAR |
| BLDG\_BUILT\_YR\_LOW | System.Int16 | SMALLINT |
| BLDG\_BUILT\_YR\_HI | System.Int16 | SMALLINT |
| BLDG\_EFF\_YR\_LOW | System.Int16 | SMALLINT |
| BLDG\_EFF\_YR\_HI | System.Int16 | SMALLINT |
| PCT\_BUILT\_OUT\_LOW | System.Decimal | DECIMAL |
| PCT\_BUILT\_OUT\_HI | System.Decimal | DECIMAL |
| NUM\_BUILDINGS\_LOW | System.Decimal | DECIMAL |
| NUM\_BUILDINGS\_HI | System.Decimal | DECIMAL |
| NUM\_STORIES\_LOW | System.Decimal | DECIMAL |
| NUM\_STORIES\_HI | System.Decimal | DECIMAL |
| AVG\_ACTUAL\_HT\_LOW | System.Decimal | DECIMAL |
| AVG\_ACTUAL\_HT\_HI | System.Decimal | DECIMAL |
| LAND\_IMP\_PATIO\_LOW | System.Decimal | DECIMAL |
| LAND\_IMP\_PATIO\_HI | System.Decimal | DECIMAL |
| FAR\_LOW | System.Decimal | DECIMAL |
| FAR\_HI | System.Decimal | DECIMAL |
| TOT\_PARKING | System.Int32 | INTEGER |

### PropCharVacantLand(VacantLand)

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| TOT\_LOT\_SIZE | System.Int32 | INTEGER |
| TOT\_ACRES | System.Decimal | DECIMAL |
| SHAPE\_CD | System.Int16 | SMALLINT |
| SHAPE\_DESC | System.String | VARCHAR |
| ZONING\_CD | System.String | VARCHAR |
| ZONING\_AGENCY\_CD | System.Int16 | SMALLINT |
| AGENCY\_NAME | System.String | VARCHAR |
| VIEW\_CD | System.Int16 | SMALLINT |
| VIEW\_DESC | System.String | VARCHAR |
| SLOPE\_CD | System.Int16 | SMALLINT |
| SLOPE\_DESC | System.String | VARCHAR |
| TOPO\_CD | System.Int16 | SMALLINT |
| TOPO\_DESC | System.String | VARCHAR |
| PAVED\_ST\_FL | System.String | CHAR |
| UTILITIES\_FL | System.String | CHAR |
| ST\_FRONTAGE\_FL | System.String | CHAR |
| FLAG\_LOT\_FL | System.String | CHAR |
| LANDLOCKED\_FL | System.String | CHAR |
| UTILITIES\_FL | System.String | CHAR |
| WELL\_FL | System.String | CHAR |
| SEPTIC\_TANK\_FL | System.String | CHAR |
| RR\_ACCESS\_FL | System.String | CHAR |
| OTH\_VAC\_LAND\_CHAR | System.String | CHAR |
| SOIL\_GOOD\_FL | System.String | CHAR |
| SOIL\_UNSTABLE\_FL | System.String | CHAR |
| SOIL\_LANDSLIDE\_FL | System.String | CHAR |
| SOIL\_FILL\_FL | System.String | CHAR |
| SOIL\_FAULT\_ZONE\_FL | System.String | CHAR |
| SOIL\_CONTAMINAT\_FL | System.String | CHAR |
| BEAR\_CAPACITY\_CD | System.Int16 | SMALLINT |
| DRAINAGE\_CD | System.Int16 | SMALLINT |
| NEED\_FILL\_FL | System.String | CHAR |
| NEED\_FILL\_FEET | System.Int32 | INTEGER |
| NEED\_EXCAVATE\_FL | System.String | CHAR |
| NEED\_EXCAVATE\_FEET | System.Int32 | INTEGER |
| NEED\_LEVELING\_FL | System.String | CHAR |
| NEED\_RETAIN\_WAL\_FL | System.String | CHAR |

### RentInfo

|  |  |  |
| --- | --- | --- |
| NRA\_SUM | System.Int32 | INTEGER |
| OCCUPIED\_SQ\_FT | System.Int32 | INTEGER |
| CURRENT\_RENT\_SUM | System.Decimal | DECIMAL |
| ANTICIP\_RENT\_SUM | System.Decimal | DECIMAL |
| TOTAL\_RENT | System.Decimal | DECIMAL |

### NRAOccupancy

|  |  |  |
| --- | --- | --- |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| PCT\_OCCUPANCY | System.Decimal | DECIMAL |

### APN

|  |  |  |
| --- | --- | --- |
| PROPERTY\_ID | System.Int32 | INTEGER |
| PRINT\_PARCEL | System.String | CHAR |
| SORT\_PARCEL | System.String | CHAR |
| SEQ\_NUM | System.Int16 | SMALLINT |

### IncomeExpenseAnalysis

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| PCT\_PGI | System.Decimal | DECIMAL |
| ADDITION\_NET\_INCOM | System.Decimal | DECIMAL |
| TENANT\_REIMBURSE | System.Decimal | DECIMAL |
| ANTI\_PROPERTY\_TAX | System.Decimal | DECIMAL |
| OTHER\_TAXES\_DESC | System.String | CHAR |
| OTHER\_TAXES\_AMOUNT | System.Decimal | DECIMAL |
| TAX\_RATE | System.Decimal | DECIMAL |
| PCT\_OCCUPANCY | System.Decimal | DECIMAL |
| PCT\_OCCUP\_MINIMUM | System.Decimal | DECIMAL |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| CSA\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| PGI | System.Decimal | DECIMAL |
| EXP\_OWNERS\_SUBTOTL | System.Decimal | DECIMAL |
| EXP\_TENANT\_SUBTOTL | System.Decimal | DECIMAL |
| EXP\_OWNERS\_TOTAL | System.Decimal | DECIMAL |
| EXP\_TENANT\_TOTAL | System.Decimal | DECIMAL |
| CSA\_ADJUST\_TOTAL | System.Decimal | DECIMAL |

### CSAExpenseCode

|  |  |  |
| --- | --- | --- |
| CMC\_AMT\_CD | System.Int16 | SMALLINT |
| SORT\_SEQ | System.Int16 | SMALLINT |
| CMC\_AMT\_TYPE | System.String | CHAR |
| CMC\_AMT\_DESC | System.String | CHAR |
| CMC\_UPDATE\_DESC\_FL | System.String | CHAR |
| CSA\_ID | System.Int32 | INTEGER |
| CAM\_AMT\_CD | System.Int16 | SMALLINT |
| CSA\_AMOUNT\_SEQ\_NUM | System.Int16 | SMALLINT |
| CAM\_AMT\_TYPE | System.String | CHAR |
| CAM\_AMT\_DESC | System.String | CHAR |
| CAM\_UPDATE\_DESC\_FL | System.String | CHAR |
| AMOUNT\_OWNERS | System.Decimal | DECIMAL |
| AMOUNT\_TENANT | System.Decimal | DECIMAL |

### CSAExpenseEntity

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| CSA\_AMOUNT\_SEQ\_NUM | System.Int16 | SMALLINT |
| AMOUNT\_TYPE | System.String | CHAR |
| AMOUNT\_CD | System.Int16 | SMALLINT |
| AMT\_DISPLAY\_TEXT | System.String | CHAR |
| AMOUNT\_OWNERS | System.Decimal | DECIMAL |
| OWNER\_DLR\_PER\_SQFT | System.Decimal | DECIMAL |
| AMOUNT\_TENANT | System.Decimal | DECIMAL |
| TENT\_DLR\_PER\_SQFT | System.Decimal | DECIMAL |
| AMOUNT\_CSA | System.Decimal | DECIMAL |
| UPDATE\_DESC\_FL | System.String | CHAR |
| GROUP\_BY\_TYPE | System.String | CHAR |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| CSA\_NET\_RENT\_AREA | System.Int32 | INTEGER |

### RptSource

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| SOURCE\_CD\_PRI | System.Int16 | SMALLINT |
| SOURCE\_NAME\_PRI | System.String | CHAR |
| SOURCE\_TITLE\_PRI | System.String | CHAR |
| SOURCE\_COMPANY\_PRI | System.String | CHAR |
| SOURCE\_PHONE\_PRI | System.Decimal | DECIMAL |
| SOURCE\_CD\_SEC | System.Int16 | SMALLINT |
| SOURCE\_NAME\_SEC | System.String | CHAR |
| SOURCE\_TITLE\_SEC | System.String | CHAR |
| SOURCE\_COMPANY\_SEC | System.String | CHAR |
| SOURCE\_PHONE\_SEC | System.Decimal | DECIMAL |
| SOURCE\_OTHER\_DESC | System.String | CHAR |
| SRCE\_IQ\_FL | System.String | CHAR |
| SRCE\_AAB\_FL | System.String | CHAR |
| SRCE\_PCOR\_COS\_FL | System.String | CHAR |
| SRCE\_RFR\_FL | System.String | CHAR |
| SRCE\_PH\_EX\_PRI | System.Int32 | INTEGER |
| SRCE\_ALT\_PH\_PRI | System.Decimal | DECIMAL |
| SRCE\_ALT\_PH\_EX\_PRI | System.Int32 | INTEGER |
| SRCE\_EMAIL\_PRI | System.String | VARCHAR |
| SOURCE\_CREATED\_BY | System.String | CHAR |
| SOURCE\_MODIFIED\_BY | System.String | CHAR |
| SOURCE\_MODIFIED\_DT | System.DateTime | DATE |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| SRCE\_MODIFY\_RSN\_CD | System.Int16 | SMALLINT |
| SOURCE\_PRI\_TEXT | System.String | VARCHAR |
| SOURCE\_SEC\_TEXT | System.String | VARCHAR |
| MODIFY\_RSN\_TEXT | System.String | VARCHAR |
| CREATED\_WORKER | System.String | VARCHAR |
| MODIFIED\_WORKER | System.String | VARCHAR |

### CSARptAdjustments

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| SORT\_SEQ | System.Int16 | SMALLINT |
| LEFT\_ROW\_NO | System.String | CHAR |
| RIGHT\_ROW\_NO | System.String | CHAR |
| CSA\_AMOUNT\_SEQ\_NUM | System.Int16 | SMALLINT |
| AMOUNT\_CD | System.Int16 | SMALLINT |
| AMT\_TEXT | System.String | CHAR |
| AMTCD\_TEXT | System.String | CHAR |
| AMT\_DISPLAY\_TEXT | System.String | CHAR |
| AMOUNT\_CSA\_LEFT | System.Decimal | DECIMAL |
| AMOUNT\_CSA | System.Decimal | DECIMAL |
| ROW\_TYPE | System.String | CHAR |

### CSAComment

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| SEQ\_NUM | System.Int16 | SMALLINT |
| COMMENT\_TEXT | System.String | VARCHAR |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |

### AdjustmentsForAllAdjustmentCodes

|  |  |  |
| --- | --- | --- |
| CMC\_AMT\_CD | System.Int16 | SMALLINT |
| SORT\_SEQ | System.Int16 | SMALLINT |
| CMC\_AMT\_TYPE | System.String | CHAR |
| CMC\_AMT\_DESC | System.String | CHAR |
| CMC\_UPDATE\_DESC\_FL | System.String | CHAR |
| CSA\_ID | System.Int32 | INTEGER |
| CAM\_AMT\_CD | System.Int16 | SMALLINT |
| CSA\_AMOUNT\_SEQ\_NUM | System.Int16 | SMALLINT |
| CAM\_AMT\_TYPE | System.String | CHAR |
| CAM\_UPDATE\_DESC\_FL | System.String | CHAR |
| AMOUNT\_CSA | System.Decimal | DECIMAL |

### FieldCode

|  |  |  |
| --- | --- | --- |
| CD\_ID | System.Int16 | SMALLINT |
| FIELD\_ID | System.Int16 | SMALLINT |
| CD\_SHORT\_NAME | System.String | VARCHAR |
| CD\_LONG\_NAME | System.String | VARCHAR |
| ASSR\_UPDATE\_FL | System.String | CHAR |
| EFFECTIVE\_DT | System.DateTime | DATE |
| END\_DT | System.DateTime | DATE |
| ROW\_CHANGE\_TS | System.DateTime | TIMESTMP |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_USER | System.String | VARCHAR |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.String | CHAR |
| UPDATE\_WORKER | System.String | CHAR |
| SORT\_SEQ | System.Int16 | SMALLINT |
| ALCAP\_SHORT\_NAME | System.String | CHAR |

### AllAPNBooks

|  |  |  |
| --- | --- | --- |
| APN\_BOOK | System.Int16 | SMALLINT |
| VALUE | System.Int32 | INT |

### AntiUseCodesNotForLandUse

|  |  |  |
| --- | --- | --- |
| ANTICIPATED\_USE\_CD | System.Int16 | SMALLINT |
| USE\_NAME | System.String | CHAR |
| USE\_NAME\_SHORT | System.String | CHAR |

### APN

|  |  |  |
| --- | --- | --- |
| PROPERTY\_ID | System.Int32 | INTEGER |
| PRINT\_PARCEL | System.String | CHAR |
| SORT\_PARCEL | System.String | CHAR |
| SEQ\_NUM | System.Int16 | SMALLINT |

### CSASourceData

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| SOURCE\_CD\_PRI | System.Int16? | SMALLINT |
| SOURCE\_NAME\_PRI | System.String | CHAR |
| SOURCE\_TITLE\_PRI | System.String | CHAR |
| SOURCE\_COMPANY\_PRI | System.String | CHAR |
| SOURCE\_PHONE\_PRI | System.Decimal? | DECIMAL |
| SOURCE\_CD\_SEC | System.Int16? | SMALLINT |
| SOURCE\_NAME\_SEC | System.String | CHAR |
| SOURCE\_TITLE\_SEC | System.String | CHAR |
| SOURCE\_COMPANY\_SEC | System.String | CHAR |
| SOURCE\_PHONE\_SEC | System.Decimal? | DECIMAL |
| SOURCE\_OTHER\_DESC | System.String | CHAR |
| SOURCE\_CREATED\_BY | System.String | CHAR |
| SOURCE\_CREATE\_DT | System.DateTime? | DATE |
| SOURCE\_MODIFIED\_BY | System.String | CHAR |
| SOURCE\_MODIFIED\_DT | System.DateTime | DATE |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| SRCE\_PH\_EX\_PRI | System.Int32 | INTEGER |
| SRCE\_ALT\_PH\_PRI | System.Decimal | DECIMAL |
| SRCE\_ALT\_PH\_EX\_PRI | System.Int32 | INTEGER |
| SRCE\_EMAIL\_PRI | System.String | VARCHAR |
| SRCE\_PH\_EX\_SEC | System.Int32? | INTEGER |
| SRCE\_ALT\_PH\_SEC | System.Decimal? | DECIMAL |
| SRCE\_ALT\_PH\_EX\_SEC | System.Int32? | INTEGER |
| SRCE\_EMAIL\_SEC | System.String | VARCHAR |
| SRCE\_IQ\_FL | System.String | CHAR |
| SRCE\_AAB\_FL | System.String | CHAR |
| SRCE\_PCOR\_COS\_FL | System.String | CHAR |
| SRCE\_RFR\_FL | System.String | CHAR |
| CD\_SHORT\_NAME | System.String | VARCHAR |
| SOURCE\_DESC\_SEC | System.String |  |
| CMP\_CREATED\_BY | System.String |  |
| CMP\_MODIFIED\_BY | System.String |  |
| SRCE\_MODIFY\_RSN\_CD | System.Int32? | SMALLINT |
| RELEASE\_DT | System.DateTime | DATE |
| SUPRV\_APPR\_UPD\_WKR | System.String | CHAR |
| SUPRV\_APPR\_UPD\_TS | System.DateTime | TIMESTMP |
| SUPRV\_APPROVED\_FL | System.String | CHAR |
| SUPRV\_APPR\_TEXT | System.String | CHAR |

### IncomeAnalysis

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| PCT\_PGI | System.Decimal | DECIMAL |
| ADDITION\_NET\_IMCOM | System.Decimal | DECIMAL |
| TENANT\_REIMBURSE | System.Decimal | DECIMAL |
| ANTI\_PROPERTY\_TAX | System.Decimal | DECIMAL |
| OTHER\_TAXES\_DESC | System.String | CHAR |
| OTHER\_TAXES\_AMOUNT | System.Decimal | DECIMAL |
| TAX\_RATE | System.Decimal | DECIMAL |
| CSA\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| PCT\_OCCUPANCY | System.Decimal | DECIMAL |
| PCT\_OCCUP\_MINIMUM | System.Decimal | DECIMAL |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |
| PROPERTY\_ID | System.Int32 | INTEGER |
| FULL\_EXPENSES\_FL | System.String | CHAR |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| BENCHMARK\_RATE\_CD | System.Int16 | SMALLINT |
| PCT\_OWNER\_OCCUP | System.Int32 | INTEGER |
| EVENT\_TS | System.DateTime | TIMESTMP |
| ADJ\_SALES\_PRICE | System.Decimal | DECIMAL |
| IND\_PUR\_PRICE | System.Decimal | DECIMAL |
| BM\_DESC | System.String | VARCHAR |
| CMP\_OAR\_DESC | System.Int32 | INTEGER |
| CMP\_POT\_GROSS\_INC | System.Int32 | INTEGER |
| CMP\_VAC\_COLL\_LOSS | System.Int32 | INTEGER |
| CMP\_EFF\_GROSS\_INC | System.Int32 | INTEGER |
| CMP\_TOT\_EXPENSES | System.Int32 | INTEGER |
| CMP\_NET\_OP\_INC | System.Int32 | INTEGER |
| CMP\_EGI\_PCT | System.Int32 | INTEGER |
| CMP\_NOI\_ADJ\_SALE | System.Int32 | INTEGER |
| CMP\_ADJ\_SALE\_PGI | System.Int32 | INTEGER |
| CMP\_ADJ\_SALE\_NRA | System.Int32 | INTEGER |
| CMP\_RENT\_NRA | System.Int32 | INTEGER |
| CMP\_NOI\_NRA | System.Int32 | INTEGER |
| CMP\_EXPENSE\_NRA | System.Int32 | INTEGER |
| CMP\_TOT\_XPENS\_NRA | System.Int32 | INTEGER |
| PCT\_OCCUP\_THRESHLD | System.Int32 | INTEGER |

### CSAAmount

|  |  |  |
| --- | --- | --- |
| CMC\_AMT\_CD | System.Int16 | SMALLINT |
| SORT\_SEQ | System.Int16 | SMALLINT |
| CMC\_AMT\_TYPE | System.String | CHAR |
| CMC\_AMT\_DESC | System.String | CHAR |
| CMC\_UPDATE\_DESC\_FL | System.String | CHAR |
| CSA\_ID | System.Int32 | INTEGER |
| CAM\_AMT\_CD | System.Int16 | SMALLINT |
| CSA\_AMOUNT\_SEQ\_NUM | System.Int16 | SMALLINT |
| CAM\_AMT\_TYPE | System.String | CHAR |
| CAM\_UPDATE\_DESC\_FL | System.String | CHAR |

### ExpenseTotal

|  |  |  |
| --- | --- | --- |
| TOT\_EXPENSE\_OWNER | System.Decimal | DECIMAL |
| TOT\_EXPENSE\_TENANT | System.Decimal | DECIMAL |
| TOT\_EXPENSES | System.Decimal | DECIMAL |

### PotentialGrossIncome

|  |  |  |
| --- | --- | --- |
| TOT\_CURRENT\_RENT | System.Decimal | DECIMAL |
| TOT\_ANTI\_RENT | System.Decimal | DECIMAL |

### SourceDesc

|  |  |  |
| --- | --- | --- |
| FIELD\_ID | System.Int16 | SMALLINT |
| CD\_ID | System.Int16 | SMALLINT |
| CD\_SHORT\_NAME | System.String | VARCHAR |
| CD\_LONG\_NAME | System.String | VARCHAR |

### WorkerInfo

|  |  |  |
| --- | --- | --- |
| WORKER\_ID | System.String | CHAR |
| WORKER\_NAME | System.String | VARCHAR |

**Vacant Land**

### VLPropetyInfo

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| DOC\_PREFIX | System.String | CHAR |
| DOC\_SERIES | System.Int32 | INTEGER |
| PROPERTY\_ID | System.Int32 | INTEGER |
| USE\_CD | System.Int16 | SMALLINT |
| TOT\_LOT\_SIZE | System.Int32 | INTEGER |
| PCT\_OWNER\_OCCUP | System.Int32 | INTEGER |
| PRINT\_PARCEl | System.String | CHAR |
| ADDRESS | System.String | VARCHAR |
| SITUS\_CITY\_NAME | System.String | VARCHAR |
| SITUS\_STATE | System.String | CHAR |
| ZIP\_CD | System.String | CHAR |
| DOC\_PARCEL\_CNT | System.Int16 | SMALLINT |
| EVENT\_TS | System.DateTime | TIMESTMP |
| IND\_PUR\_PRICE | System.Decimal | DECIMAL |
| ADJ\_SALES\_PRICE | System.Decimal | DECIMAL |
| TRAN\_TAX\_PRICE | System.Decimal | DECIMAL |
| USE\_NAME | System.Int16 | SMALLINT |
| TOT\_NET\_RENT\_AREA | System.Int32 | INTEGER |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |

**PropChar**

### PCBuilding

|  |  |  |
| --- | --- | --- |
| PROPERTY\_ID | System.Int32 | INTEGER |
| BUILDING\_NUM | System.Int16 | SMALLINT |
| EFFECTIVE\_DT | System.DateTime | DATE |
| BUILT\_YR | System.Int16 | SMALLINT |
| EFFECTIVE\_YR | System.Int16 | SMALLINT |
| NUM\_STORIES | System.Decimal | DECIMAL |
| NUM\_BEDROOMS | System.Int16 | SMALLINT |
| NUM\_BATHS\_FULL | System.Int16 | SMALLINT |
| NUM\_BATHS\_HALF | System.Int16 | SMALLINT |
| NUM\_ROOMS | System.Int16 | SMALLINT |
| NUM\_UNITS | System.Int16 | SMALLINT |
| COND\_CD | System.Int16 | SMALLINT |
| COND\_YR | System.Int16 | SMALLINT |
| BUILDING\_AREA | System.Int32 | INTEGER |
| ADDITIONS\_AREA | System.Int32 | INTEGER |
| MISC\_AREA | System.Int32 | INTEGER |
| NET\_RENTABLE\_AREA | System.Int32 | INTEGER |
| PCT\_OFFICE | System.Decimal | DECIMAL |
| AVG\_ACTUAL\_HT | System.Decimal | DECIMAL |
| SBE\_CONST\_TYPE\_CD | System.Int16 | SMALLINT |
| SBE\_QUALITY | System.Decimal | DECIMAL |
| SBE\_SHAPE\_CD | System.Int16 | SMALLINT |
| CONDO\_TYPE\_CD | System.Int16 | SMALLINT |
| CONDO\_UNIT\_FLOOR | System.Int16 | SMALLINT |
| CONFORMITY\_CD | System.Int16 | SMALLINT |
| REMODEL\_CD | System.Int16 | SMALLINT |
| REMODEL\_YR | System.Int16 | SMALLINT |
| FIRE\_SPRINKLER\_FL | System.String | CHAR |
| NUM\_ELEVATOR | System.Int16 | SMALLINT |
| END\_DT | System.DateTime | DATE |
| NUM\_KITCHENS | System.Int16 | SMALLINT |
| ADDITION\_YR | System.Int16 | SMALLINT |
| FINISHED\_AREA | System.Int32 | INTEGER |
| BASEMENT\_AREA | System.Int32 | INTEGER |
| AVG\_UNIT\_AREA | System.Int32 | INTEGER |
| PCT\_BUILT\_OUT | System.Decimal | DECIMAL |
| AVG\_CLEAR\_HT | System.Decimal | DECIMAL |
| MVS\_CONST\_TYPE\_CD | System.Int16 | SMALLINT |
| MVS\_PERIMETER | System.Int32 | INTEGER |
| MVS\_QUALITY | System.Decimal | DECIMAL |
| MVS\_SHAPE\_CD | System.Int16 | SMALLINT |
| SEISMIC\_RETRO\_FL | System.String | CHAR |
| SEISMIC\_RETRO\_YR | System.Int16 | SMALLINT |
| OCC\_CD | System.Int16 | SMALLINT |
| DEMOLISHED\_FL | System.String | CHAR |
| BUILDING\_LABEL | System.String | VARCHAR |
| SBE\_CONST\_TYPE | System.String | CHAR |
| SBE\_SHAPE | System.String | CHAR |
| MVS\_CONST\_TYPE | System.String | CHAR |
| MVS\_SHAPE | System.String | CHAR |
| SBE\_CODE | System.String | CHAR |
| MVS\_CODE | System.String | CHAR |
| SELECT\_FL | System.String | CHAR |

### PCEvent

|  |  |  |
| --- | --- | --- |
| CSA\_ID | System.Int32 | INTEGER |
| EVENT\_TS | System.DateTime | TIMESTMP |
| PRINT\_PARCEL | System.String | CHAR |
| NEIGH\_CD | System.String | CHAR |
| PROPERTY\_ID | System.String | CHAR |
| DOC\_PARCEL\_CNT | System.Int16 | SMALLINT |
| DOC\_PREFIX | System.String | CHAR |
| DOC\_SERIES | System.Int32 | INTEGER |
| SEQ\_NUM | System.Int16 | SMALLINT |
| USE\_CD | System.Int16 | SMALLINT |
| SBE\_CONST\_TYPE | System.String | CHAR |
| SBE\_QUALITY | System.Decimal | DECIMAL |
| SBE\_SHAPE | System.String | CHAR |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| BUILT\_YR | System.Int16 | SMALLINT |
| EFFECTIVE\_YR | System.Int16 | SMALLINT |
| TOT\_ROOMS | System.Int16 | SMALLINT |
| TOT\_BEDROOMS | System.Int16 | SMALLINT |
| TOT\_BATHS\_FULL | System.Int16 | SMALLINT |
| TOT\_BATHS\_HALF | System.Int16 | SMALLINT |
| LOT\_SIZE | System.Decimal | DECIMAL |
| LOT\_SIZE\_CD | System.Int16 | SMALLINT |
| LOT\_SIZE\_SQ\_FT | System.Int32 | INTEGER |
| COND\_CD | System.Int16 | SMALLINT |
| REMODEL\_FL | System.String | CHAR |
| TOT\_NUM\_UNITS | System.Int32 | INTEGER |
| NUM\_BUILDINGS | System.Int16 | SMALLINT |
| VIEW\_CD | System.Int16 | SMALLINT |
| SLOPE\_CD | System.Int16 | SMALLINT |
| LAND\_IMP\_RATIO | System.Decimal | DECIMAL |
| AMENITIES\_FL | System.String | CHAR |
| HAZARD\_FL | System.String | CHAR |
| TOT\_PARKING | System.Int32 | INTEGER |
| POOL\_FL | System.String | CHAR |
| NUM\_STORIES | System.Decimal | DECIMAL |
| BUILDING\_NUM | System.Int16 | SMALLINT |
| CONDO\_UNIT\_FLOOR | System.Int16 | SMALLINT |
| PARKING\_TYPE\_CD | System.Int16 | SMALLINT |
| COND\_YR | System.Int16 | SMALLINT |
| REMODEL\_YR | System.Int16 | SMALLINT |
| POOL\_TYPE\_CD | System.Int16 | SMALLINT |
| POOL\_BUILT\_YR | System.Int16 | SMALLINT |
| LAND\_PARKING | System.Int32 | INTEGER |
| MVS\_CONST\_TYPE | System.String | CHAR |
| MVS\_QUALITY | System.Decimal | DECIMAL |
| MVS\_SHAPE | System.String | CHAR |
| FLOOR\_AREA\_RATIO | System.Decimal | DECIMAL |
| SITUS\_STREET\_NUM | System.String | VARCHAR |
| SITUS\_PRE\_DIRECT | System.String | CHAR |
| SITUS\_STREET\_NAME | System.String | VARCHAR |
| SITUS\_STREET\_SFX | System.String | CHAR |
| SITUS\_POST\_DIRECT | System.String | CHAR |
| SITUS\_UNIT\_DESIG | System.String | CHAR |
| SITUS\_UNIT\_NUM | System.String | VARCHAR |
| SITUS\_CITY\_NAME | System.String | VARCHAR |
| SITUS\_ZIP\_CD | System.String | CHAR |
| SITUS\_ZIP\_PLUS\_4 | System.String | CHAR |
| APN\_BOOK | System.String | CHAR |
| APN\_BLOCK | System.Int16 | SMALLINT |
| SORT\_PARCEL | System.String | CHAR |

### VLPropChar

|  |  |  |
| --- | --- | --- |
| VIEW\_CD | System.Int16 | SMALLINT |
| SLOPE\_CD | System.Int16 | SMALLINT |
| WELL\_FL | System.String | CHAR |
| SEPTIC\_FL | System.String | CHAR |

**Lease**

### LeaseComment

|  |  |  |
| --- | --- | --- |
| LEASE\_ID | System.Int32 | INTEGER |
| SEQ\_NUM | System.Int16 | SMALLINT |
| COMMENT\_TEXT | System.String | VARCHAR |
| ENTRY\_TS | System.DateTime | TIMESTMP |
| ENTRY\_WORKER | System.String | CHAR |
| UPDATE\_TS | System.DateTime | TIMESTMP |
| UPDATE\_WORKER | System.String | CHAR |

### LeaseUseCategory

|  |  |  |
| --- | --- | --- |
| USE\_CATEGORY\_CD | System.String | CHAR |
| USE\_NAME | System.String | CHAR |
| END\_DT | System.DateTime | DATE |

### LeaseUseSubCat

|  |  |  |
| --- | --- | --- |
| LEASE\_SUBCATEGORY | System.String | CHAR |
| SUBCATEGORY\_NAME | System.String | CHAR |
| END\_DT | System.DateTime | DATE |

**wkSheet**

### wksAPN

|  |  |  |
| --- | --- | --- |
| DOC\_PREFIX | System.String | CHAR |
| DOC\_SERIES | System.Int32 | INTEGER |
| SEQ\_NUM | System.Int16 | SMALLINT |
| PRINT\_PARCEL | System.String | CHAR |
| SORT\_PARCEL | System.String | CHAR |
| PROPERTY\_ID | System.Int32 | INTEGER |
| PRIMARY\_APN\_FL | System.String | CHAR |
| CSA\_SUBJECT | System.String | CHAR |
| USE\_NAME\_SHORT | System.String | VARCHAR |
| TOT\_BUILDING\_AREA | System.Int32 | INTEGER |
| TOT\_LOT\_SIZE | System.Decimal | DECIMAL |
| CSA\_WKS\_NUM | System.Int32 | INTEGER |

## Models (Used to store in and out object, not directly database query result)

### UpdatedSaleInfoWithCIAnalysis

|  |  |
| --- | --- |
| csa\_id | System.Int32 |
| anticipated\_use\_cd | System.Int16? |
| csa\_prop\_use\_detl | System.String |
| cond\_at\_sale\_cd | System.Int32? |
| broker\_involved\_fl | System.String |
| buy\_sell\_rel\_fl | System.String |
| buy\_sell\_rel\_desc | System.String |
| pur\_predate\_by\_opt | System.String |
| pur\_pred\_cont\_sale | System.String |
| predate\_cont\_date | System.DateTime? |
| event\_ts | System.DateTime |
| lastmodifypage | System.String |
| workerid | System.String |
| retmsg | System.String? |

### UpdatedIncomeAnalysis

|  |  |
| --- | --- |
| csa\_id | System.Int32 |
| pct\_pgi | System.Decimal? |
| addition\_net\_incom | System.Decimal? |
| tenant\_reimburse | System.Decimal? |
| anti\_property\_tax | System.Decimal? |
| other\_taxes\_desc | System.String |
| other\_taxes\_amount | System.Decimal? |
| pct\_occup\_minimum | System.Decimal? |

### CSASaleInfo

|  |  |
| --- | --- |
| doc\_prefix | System.String |
| situs\_city | System.String |
| broker\_involved\_fl | System.String |
| zoning\_cd | System.String |
| buy\_sell\_rel\_fl | System.String |
| buy\_sell\_rel\_desc | System.String |
| pur\_predate\_by\_opt | System.String |
| pur\_pred\_cont\_sale | System.String |
| full\_expenses\_fl | System.String |
| entry\_worker | System.String |
| situs\_state | System.String |
| update\_worker | System.String |
| csa\_prop\_use\_detl | System.String |
| benchmark\_rate\_cd\_name | System.String |
| anticipated\_use\_cd | System.Int16? |
| csa\_id | System.Int32 |
| csa\_type | System.Int32 |
| doc\_series | System.Int32 |
| property\_id | System.Int32 |
| benchmark\_rate\_cd | System.Int32? |
| cond\_at\_sale\_cd | System.Int32? |
| csa\_prop\_use\_cd | System.Int32? |
| shape\_cd | System.Int32? |
| tot\_building\_area | System.Int32? |
| tot\_lot\_size | System.Int32? |
| tot\_net\_rent\_area | System.Int32? |
| use\_cd | System.Int32? |
| zoning\_agency\_cd | System.Int32? |
| csa\_net\_rent\_area | System.Int32? |
| pct\_owner\_occup | System.Int32? |
| event\_ts | System.DateTime |
| entry\_ts | System.DateTime |
| predate\_cont\_date | System.DateTime? |
| release\_dt | System.DateTime? |
| update\_ts | System.DateTime? |
| print\_parcel | System.String |
| buyer\_name | System.String |
| seller\_name | System.String |
| anticipated\_use\_cd\_name | System.String |
| suprv\_approved\_fl | System.String |
| suprv\_approved\_fl\_text | System.String |

### CSAProperty

|  |  |
| --- | --- |
| doc\_prefix | System.String |
| print\_parcel | System.String |
| situs\_address | System.String |
| situs\_city | System.String |
| situs\_state | System.String |
| situs\_zip\_cd | System.String |
| csa\_id | System.Int32 |
| doc\_parcel\_cnt | System.Int32 |
| doc\_series | System.Int32 |
| property\_id | System.Int32 |
| use\_cd | System.Int32 |
| ind\_pur\_price | System.Decimal |
| event\_ts | System.DateTime |

### CSAUpdatedSourceData

|  |  |
| --- | --- |
| csa\_id | System.Int32 |
| source\_cd\_pri | System.Int32? |
| source\_name\_pri | System.String |
| source\_title\_pri | System.String |
| source\_company\_pri | System.String |
| srce\_email\_pri | System.String |
| source\_phone\_pri | System.Decimal? |
| srce\_ph\_ex\_pri | System.Int32? |
| srce\_alt\_ph\_pri | System.Decimal? |
| srce\_alt\_ph\_ex\_pri | System.Int32? |
| source\_cd\_sec | System.Int32? |
| source\_name\_sec | System.String |
| source\_title\_sec | System.String |
| source\_company\_sec | System.String |
| srce\_email\_sec | System.String |
| source\_phone\_sec | System.Decimal? |
| srce\_ph\_ex\_sec | System.Int32? |
| srce\_alt\_ph\_sec | System.Decimal? |
| srce\_alt\_ph\_ex\_sec | System.Int32? |
| srce\_iq\_fl | System.String |
| srce\_aab\_fl | System.String |
| srce\_pcor\_cos\_fl | System.String |
| srce\_rfr\_fl | System.String |
| srce\_modify\_rsn\_cd | System.Int32? |
| source\_other\_desc | System.String |
| source\_created\_by | System.String |
| source\_create\_dt | System.DateTime? |
| suprv\_approved\_fl | System.String |
| \_LastModifyPage | System.String |
| worker\_id | System.String |

### CSASaleExpenses

For this entity:

ExpensesForAllExpenseCodesBool default value should be set to true.

Full\_Expenses\_Fl\_Bool is meant to return a bool type of the Full\_Expenses\_Fl, depending on its value is “Y” or “N”.

|  |  |
| --- | --- |
| csa\_id | System.Int32 |
| csa\_net\_rent\_area | System.Int32? |
| full\_expenses\_fl | System.String |
| full\_expenses\_fl\_bool | System.Boolean |
| is\_for\_all\_expense\_codes | System.Boolean |
| tot\_building\_area | System.Int32? |
| tot\_net\_rent\_area | System.Int32? |
| csa\_net\_rent\_area | System.Int32? |
| list\_of\_expenses | List<CSAExpense> |

For is\_for\_all\_expense\_codes property its value should be initialize as **true**, whenever a new instance is created.

You will only set is\_for\_all\_expense\_codes false when validate if existIncomesExpenses is true. Which will be covered in the Sale Info tab in the web application session.

For constructor of this model class

public CSASaleExpenses(int p\_csa\_id, bool p\_expense\_bool = true)

{

    this.csa\_id = p\_csa\_id;

    this.is\_for\_all\_expense\_codes = p\_expense\_bool

    this.list\_of\_expenses = \_CSASaleBll.GetExpenses(this.csa\_id,

                            this.expenses\_for\_all\_expense\_codes\_bool);

}

For **list\_of\_expenses** property make sure to setup its get method as below:

public List<CSAExpense> list\_of\_expenses

{

    get

    {

        if (list\_of\_expenses == null)

        {

            // CSASaleBll should be depandancy injected to this model class

            if (is\_for\_all\_expense\_codes)

            {

                // Get expense amounts for all expense codes

                list\_of\_expenses = \_CSASaleBll.GetExpensesForAllExpenseCodes (this.csa\_id);

            }

            else

            {

                // Get expense amounts for actual CSA amount rows only for the given CSA ID

                list\_of\_expenses = \_CSASaleBll.GetActualExpenses(this.csa\_id);

            }

        }

        return list\_of\_expenses;

    }

}

### CSAExpense

|  |  |
| --- | --- |
| cam\_update\_desc\_fl | System.String |
| cam\_amount\_desc | System.String |
| cam\_amount\_type | System.String |
| cmc\_update\_desc\_fl | System.String |
| cmc\_amount\_desc | System.String |
| cmc\_amount\_type | System.String |
| expense\_description | System.String |
| csa\_id | System.Int32 |
| cmc\_amount\_cd | System.Int32 |
| sort\_seq | System.Int32 |
| cam\_amount\_cd | System.Int32? |
| csa\_amount\_seq\_num | System.Int32? |
| amount\_owners | System.Decimal? |
| amount\_tenant | System.Decimal? |
| is\_tax | System.Boolean? |
| is\_valid | System.Boolean |
| errors | System.Collection |

For cam\_update\_desc\_fl and cmc\_update\_desc\_fl property, use below template for properties:

set

{

    if (value == null || value.Trim().Length == 0)

    {

Set to DBNull so that when using DB2 helper function this value will be assigned to corresponding field in the database (since null will be ignored from update call)

        this.cam\_update\_desc\_fl = DBNull;

    }

    else if (value.Trim().ToUpper() == "Y" || value.Trim().ToUpper() == "N")

    {

        this.cam\_update\_desc\_fl = value;

    }

}

Remember to perform below functions once all other values are filled using CSAExpenseCode entity. (UpdateExpenseDesc() and Validate())

For expense\_description

public void UpdateExpenseDesc()

{

    if (cam\_update\_desc\_fl != null)

    {

        if (cam\_update\_desc\_fl == "Y")

        {

            this.expense\_description = cam\_amount\_desc;

        }

        else

        {

            this.expense\_description = cmc\_amount\_desc;

        }

    }

    else

    {

        this.expense\_description = cmc\_amount\_desc;

    }

}

### CSASaleAdjustments

|  |  |
| --- | --- |
| csa\_id | System.Int32 |
| adj\_sales\_price | System.Decimal |
| ind\_pur\_price | System.Decimal |
| total\_not\_user\_defined\_not\_bonds\_adjustments | System.Decimal |
| total\_user\_defined\_adjustments | System.Decimal |
| is\_mismatched\_ind\_pur\_price | System.Boolean |
| is\_mismatched\_adj\_sales\_price | System.Boolean |
| list\_of\_adjustments | List<CSAAdjustment> |

Constructor

public CSASaleAdjustments(int p\_csa\_id)

{

this.csa\_id = p\_csa\_id

    this.list\_of\_adjustments = GetAdjustmentsForAllAdjustmentCodes(p\_csa\_id)

}

### CSAAdjustment

|  |  |
| --- | --- |
| cam\_update\_desc\_fl | System.String |
| cam\_amount\_desc | System.String |
| cam\_amount\_type | System.String |
| cmc\_update\_desc\_fl | System.String |
| cmc\_amount\_desc | System.String |
| cmc\_amount\_type | System.String |
| csa\_id | System.Int32 |
| cmc\_amount\_cd | System.Int32 |
| sort\_seq | System.Int32 |
| cam\_amount\_cd | System.Int32? |
| csa\_amount\_seq\_num | System.Int32? |
| amount\_csa | System.Decimal? |
| isvalid | System.Boolean? |
| errors | System.Collection |

For cam\_update\_desc\_fl and cmc\_update\_desc\_fl use the below template:

public string cam\_update\_desc\_fl

{

    set

    {

        if (value == null || value.Trim().Length == 0)

        {

            cam\_update\_desc\_fl = null;

        }

        else if (value.Trim().ToUpper() == "Y" || value.Trim().ToUpper() == "N")

        {

            cam\_update\_desc\_fl = value;

        }

    }

}

### IncAnalysisValInd

|  |  |
| --- | --- |
| csa\_net\_rent\_area | System.Int32? |
| cmp\_tot\_lease\_nra | System.Int32? |
| csa\_id | System.Int32 |
| pct\_pgi | System.Decimal? |
| addition\_net\_incom | System.Decimal? |
| tenant\_reimburse | System.Decimal? |
| anti\_property\_tax | System.Decimal? |
| other\_taxes\_amount | System.Decimal? |
| tax\_rate | System.Decimal? |
| pct\_occupancy | System.Decimal? |
| cmp\_eff\_gross\_inc | System.Decimal? |
| cmp\_tot\_expenses | System.Decimal? |
| cmp\_net\_op\_inc | System.Decimal? |
| cmp\_egi\_pct | System.Decimal? |
| cmp\_noi\_adj\_sale | System.Decimal? |
| cmp\_adj\_sale\_pgi | System.Decimal? |
| cmp\_adj\_sale\_nra | System.Decimal? |
| cmp\_rent\_nra | System.Decimal? |
| cmp\_noi\_nra | System.Decimal? |
| cmp\_expense\_nra | System.Decimal? |
| cmp\_tot\_xpens\_nra | System.Decimal? |
| cmp\_pot\_gross\_inc | System.Decimal? |
| cmp\_vac\_coll\_loss | System.Decimal? |
| cmp\_pct\_occup\_threshld | System.Decimal? |
| pct\_occup\_minimum | System.Decimal? |
| pct\_owner\_occup | System.Int32? |
| cmp\_adj\_sale\_price | System.Decimal? |
| bm\_desc | System.String |
| other\_taxes\_desc | System.String |
| oar\_desc | System.String |
| full\_expenses\_fl | System.String |
| tot\_net\_rent\_area | System.Int32 |
| tot\_building\_area | System.Int32 |
| benchmark\_rate\_cd | System.Int16? |

### LeaseBO

|  |  |
| --- | --- |
| lease\_id | System.Int32 |
| property\_id | System.Int32 |
| lease\_location | System.String |
| situs\_city\_name | System.String |
| lessee\_name | System.String |
| vacant\_fl = “N” | System.String |
| net\_rentable\_area | System.Int32 |
| lease\_start\_dt | System.DateTime? |
| lease\_term | System.Int32? |
| lease\_option | System.String |
| initial\_rent | System.Decimal? |
| initial\_rent\_psf | System.Decimal? |
| stabil\_rent | System.Decimal? |
| stabil\_rent\_psf | System.Decimal? |
| anticipated\_rent | System.Decimal? |
| tenant\_imps\_psf | System.Decimal? |
| lease\_subcategory | System.Int32? |
| lease\_type\_cd | System.Int32? |
| expense\_type\_cd | System.Int32? |
| tenancy\_cd | System.Int32? |
| info\_source\_cd | System.Int32? |
| lease\_parking | System.String |
| sprinkler\_fl | System.String |
| lease\_created\_by | System.String |
| lease\_create\_dt | System.DateTime? |
| lease\_modify\_by | System.String |
| lease\_modify\_dt | System.DateTime? |
| entry\_ts | System.DateTime |
| entry\_worker | System.String |
| update\_ts | System.DateTime? |
| update\_worker | System.String |
| csa\_id | System.Int32? |
| csa\_id\_specified | System.Boolean |
| anti\_rent\_psf | System.Decimal? |
| use\_for\_comp\_fl = “Y” | System.String |
| use\_category\_cd | System.Int32? |
| current\_rent | System.Decimal? |
| current\_rent\_psf | System.Decimal? |
| rent\_adj\_textfield | System.String |
| expense\_base\_yr | System.Int32? |
| ti\_allowance | System.Decimal? |
| ti\_type\_cd | System.Int32? |
| lease\_parking\_2 | System.String |
| info\_source\_text | System.String |
| suprv\_approved\_fl = “N” | System.String |
| suprv\_appr\_upd\_wkr | System.String |
| suprv\_appr\_upd\_ts | System.DateTime? |
| lease\_detail | LeaseDetailBO |
| apn | System.String |
| lease\_created\_name | System.String |
| lease\_modify\_name | System.String |
| suprv\_appr\_wkr\_name | System.String |
| lease\_use\_name | System.String |
| subcategory\_name | System.String |
| lease\_type\_text | System.String |
| expense\_type\_text | System.String |
| tenancy\_text | System.String |
| ti\_type\_text | System.String |
| isleasedetempty | System.Boolean |
| isleasedetmodified | System.Boolean |

### LeaseDetailBO

|  |  |
| --- | --- |
| lease\_id | System.Int32 |
| ind\_clr\_hgt | System.Decimal? |
| ind\_clr\_hgt\_specified | System.Boolean |
| ind\_offc\_area\_pct | System.Decimal? |
| ind\_offc\_area\_pct\_specified | System.Boolean |
| ind\_lab\_pct | System.Decimal? |
| ind\_lab\_pct\_specified | System.Boolean |
| ind\_mfg\_pct | System.Decimal? |
| ind\_mfg\_pct\_specified | System.Boolean |
| ind\_warehouse\_pct | System.Decimal? |
| ind\_warehouse\_pct\_specified | System.Boolean |
| ind\_retail\_pct | System.Decimal? |
| ind\_retail\_pct\_specified | System.Boolean |
| ind\_other\_pct | System.Decimal? |
| ind\_other\_pct\_specified | System.Boolean |
| offc\_load\_factor | System.Decimal? |
| offc\_load\_factor\_specified | System.Boolean |
| offc\_expense\_stop | System.Decimal? |
| offc\_expense\_stop\_specified | System.Boolean |
| retail\_dba | System.String |
| retail\_rent\_pct | System.Decimal? |
| retail\_rent\_pct\_specified | System.Boolean |
| rent\_breakpoint | System.Decimal? |
| rent\_breakpoint\_specified | System.Boolean |
| retail\_store\_occ | System.Int16? |
| retail\_store\_occ\_specified | System.Boolean |
| entry\_ts | System.DateTime |
| entry\_worker | System.String |
| update\_ts | System.DateTime? |
| update\_ts\_specified | System.Boolean |
| update\_worker | System.String |

### LeaseCommentBO

|  |  |
| --- | --- |
| lease\_id | System.Int32 |
| seq\_num | System.Int16 |
| comment\_text | System.String |
| entry\_ts | System.DateTime |
| entry\_worker | System.String |
| update\_ts | System.DateTime? |
| update\_ts\_specified | System.Boolean |
| update\_worker | System.String |

### RPCSaleInfo

|  |  |
| --- | --- |
| use\_cd | System.Int32? |
| situs\_city\_name | System.String |
| tot\_building\_area | System.int32? |
| tot\_net\_rent\_area | System.Int32? |
| tot\_lot\_size | System.Int32? |
| zoning\_cd | System.String |
| zoning\_agency\_cd | System.Int32? |
| update\_worker | System.String |
| adj\_sales\_price | System.Decimal? |
| cs\_id | System.Int32? |

### RPCPropChar

|  |  |
| --- | --- |
| sbe\_class\_low | System.String |
| sbe\_class\_hi | System.String |
| mvs\_class\_low | System.String |
| mvs\_class\_hi | System.String |
| bldg\_built\_yr\_low | System.int32? |
| bldg\_built\_yr\_hi | System.int32? |
| bldg\_eff\_yr\_low | System.int32? |
| bldg\_eff\_yr\_hi | System.int32? |
| pct\_built\_out\_low | System.Decimal? |
| pct\_built\_out\_hi | System.Decimal? |
| pct\_office\_low | System.Decimal? |
| pct\_office\_hi | System.Decimal? |
| num\_buildings\_low | System.Decimal? |
| num\_buildings\_hi | System.Decimal? |
| num\_stories\_low | System.Decimal? |
| num\_stories\_hi | System.Decimal? |
| avg\_actual\_ht\_low | System.Decimal? |
| avg\_actual\_ht\_hi | System.Decimal? |
| land\_imp\_ratio\_low | System.Decimal? |
| land\_imp\_ratio\_hi | System.Decimal? |
| far\_low | System.Decimal? |
| far\_hi | System.Decimal? |
| tot\_parking | System.int32? |
| update\_worker | System.String |
| csa\_id | System.int32 |

### RPCVLPropChar

|  |  |
| --- | --- |
| view\_cd | System.int32? |
| slope\_cd | System.int32? |
| topo\_cd | System.int32? |
| septic\_tank\_fl | System.String |
| well\_fl | System.String |
| update\_worker | System.String |
| csa\_id | System.int32 |

### wkSheetNewImp

|  |  |
| --- | --- |
| Subject | String |
| Primary | Bool |
| APN | String |
| propertyID | Integer |
| Use | String |
| LotSize | String |
| BldgArea | String |

## Stored Procedure Queries

**Note for transactions, be on alert if timeout happens, based on situation might need to revise the specific design for the transaction.**

For database querying, rewrite the original query string into stored procedures to be called in order to increase performance and security. Stored procedures are created under schema “IE\_COLLECTION”. For naming use “IE\_CSA\_SPName”. Below is an example of an original query string:

### IE\_GET\_CSA\_SALE\_INFO\_FULL (P\_CSA\_ID)

SELECT

    csi.CSA\_ID,

    csi.CSA\_TYPE,

    csi.DOC\_PREFIX,

    csi.DOC\_SERIES,

    csi.PROPERTY\_ID,

    csi.USE\_CD,

    csi.CSA\_PROP\_USE\_CD,

    csi.CSA\_PROP\_USE\_DETL,

    csi.ANTICIPATED\_USE\_CD,

    csi.COND\_AT\_SALE\_CD,

    csi.FULL\_EXPENSES\_FL,

    csi.RELEASE\_DT,

    csi.BROKER\_INVOLVED\_FL,

    csi.TOT\_BUILDING\_AREA,

    csi.TOT\_NET\_RENT\_AREA,

    csi.TOT\_LOT\_SIZE,

    csi.SHAPE\_CD,

    csi.BENCHMARK\_RATE\_CD,

    csi.ZONING\_CD,

    csi.ZONING\_AGENCY\_CD,

    csi.BUY\_SELL\_REL\_FL,

    csi.BUY\_SELL\_REL\_DESC,

    csi.PUR\_PREDATE\_BY\_OPT,

    csi.PUR\_PRED\_CONT\_SALE,

    csi.PREDATE\_CONT\_DATE,

    csi.PCT\_OWNER\_OCCUP,

    csi.SUPRV\_APPROVED\_FL,

    csi.ENTRY\_TS,

    csi.ENTRY\_WORKER,

    csi.UPDATE\_TS,

    csi.UPDATE\_WORKER,

    IEAOWN.IE\_CHG\_OWNER.EVENT\_TS,

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    fcd.CD\_LONG\_NAME AS PROP\_USE\_TEXT,

    fcd2.CD\_LONG\_NAME AS COND\_AT\_SALE\_TEXT,

    fcd3.CD\_LONG\_NAME AS BENCHMARK\_NAME,

    ana.CSA\_NET\_RENT\_AREA,

    (

        WK1.LAST\_NAME ||

        CASE WHEN IFNULL(WK1.FIRST\_NAME, '') = '' THEN '' ELSE ', ' || WK1.FIRST\_NAME END ||

        CASE WHEN IFNULL(WK1.MIDDLE\_NAME, '') = '' THEN '' ELSE ' ' || WK1.MIDDLE\_NAME END

    ) AS SUPRV\_APPR\_WKR,

    csi.ANTICIPATED\_USE\_CD,

    csi.SUPRV\_APPROVED\_FL,

    CASE csi.SUPRV\_APPROVED\_FL WHEN 'Y' THEN 'Yes' ELSE 'No' END AS SUPRV\_APPR\_TEXT,

    csi.SUPRV\_APPR\_UPD\_WKR,

    csi.SUPRV\_APPR\_UPD\_TS,

    ant.USE\_NAME AS anti\_use\_cd\_name,

    ant.USE\_NAME\_SHORT,

    ant.LAND\_USE\_FL,

    ant.USE\_AS\_IMPROVE\_FL

FROM

    IEAOWN.IE\_CSA\_SALE\_INFO csi

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE fcd ON csi.CSA\_PROP\_USE\_CD = fcd.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE fcd2 ON csi.COND\_AT\_SALE\_CD = fcd2.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE fcd3 ON csi.BENCHMARK\_RATE\_CD = fcd3.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_ANALYSIS ana ON csi.CSA\_ID = ana.CSA\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_ANTICIP\_USE ant ON csi.ANTICIPATED\_USE\_CD = ant.ANTICIPATED\_USE\_CD

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK1 ON WK1.WORKER\_ID = csi.SUPRV\_APPR\_UPD\_WKR,

    IEAOWN.IE\_CHG\_OWNER,

    IEAOWN.IE\_REAL\_PROPERTY

WHERE

    csi.CSA\_ID = **P\_CSA\_ID**

    AND IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX = csi.DOC\_PREFIX

    AND IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES = csi.DOC\_SERIES

    AND IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = csi.PROPERTY\_ID;

Below is the example to create the stored procedure to complete the above query:

**CREATE PROCEDURE** IE\_CSA\_GET\_CSA\_SALE\_INFO (**IN** CSAId **INTEGER**)

**VERSION** V1

**ISOLATION** **LEVEL** CS

RESULT SETS 1

ALLOW DEBUG **MODE**

WLM ENVIRONMENT **FOR** DEBUG **MODE** DB2DWDBG

**PACKAGE** OWNER IEAPLND

ASUTIME LIMIT 600000

P1: **BEGIN**

-- Declare cursor

**DECLARE** cursor1 **CURSOR** **WITH** RETURN **FOR**

**SELECT**

        csi.CSA\_ID,

        csi.CSA\_TYPE,

        csi.DOC\_PREFIX,

        csi.DOC\_SERIES,

        csi.PROPERTY\_ID,

        csi.USE\_CD,

        csi.CSA\_PROP\_USE\_CD,

        csi.CSA\_PROP\_USE\_DETL,

        csi.ANTICIPATED\_USE\_CD,

        csi.COND\_AT\_SALE\_CD,

        csi.FULL\_EXPENSES\_FL,

        csi.RELEASE\_DT,

        csi.BROKER\_INVOLVED\_FL,

        csi.TOT\_BUILDING\_AREA,

        csi.TOT\_NET\_RENT\_AREA,

        csi.TOT\_LOT\_SIZE,

        csi.SHAPE\_CD,

        csi.BENCHMARK\_RATE\_CD,

        csi.ZONING\_CD,

        csi.ZONING\_AGENCY\_CD,

        csi.BUY\_SELL\_REL\_FL,

        csi.BUY\_SELL\_REL\_DESC,

        csi.PUR\_PREDATE\_BY\_OPT,

        csi.PUR\_PRED\_CONT\_SALE,

        csi.PREDATE\_CONT\_DATE,

        csi.PCT\_OWNER\_OCCUP,

        csi.SUPRV\_APPROVED\_FL,

        csi.ENTRY\_TS,

        csi.ENTRY\_WORKER,

        csi.UPDATE\_TS,

        csi.UPDATE\_WORKER,

        IEAOWN.IE\_CHG\_OWNER.EVENT\_TS,

        IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

        fcd.CD\_LONG\_NAME **AS** PROP\_USE\_TEXT,

        fcd2.CD\_LONG\_NAME **AS** COND\_AT\_SALE\_TEXT,

        fcd3.CD\_LONG\_NAME **AS** BENCHMARK\_NAME,

        ana.CSA\_NET\_RENT\_AREA,

        (

            WK1.LAST\_NAME ||

**CASE WHEN IFNULL**(WK1.FIRST\_NAME, '') = '' **THEN** '' **ELSE** ', ' || WK1.FIRST\_NAME **END** ||

**CASE WHEN IFNULL**(WK1.MIDDLE\_NAME, '') = '' **THEN** '' **ELSE** ' ' || WK1.MIDDLE\_NAME **END**

        ) **AS** SUPRV\_APPR\_WKR,

        csi.ANTICIPATED\_USE\_CD,

        csi.SUPRV\_APPROVED\_FL,

**CASE** csi.SUPRV\_APPROVED\_FL **WHEN** 'Y' **THEN** 'Yes' **ELSE** 'No' **END AS** SUPRV\_APPR\_TEXT,

        csi.SUPRV\_APPR\_UPD\_WKR,

        csi.SUPRV\_APPR\_UPD\_TS,

        ant.USE\_NAME **AS** anti\_use\_cd\_name,

        ant.USE\_NAME\_SHORT,

        ant.LAND\_USE\_FL,

        ant.USE\_AS\_IMPROVE\_FL

**FROM**

        IEAOWN.IE\_CSA\_SALE\_INFO csi

**LEFT OUTER JOIN**

        IEAOWN.IE\_FIELD\_CODE fcd **ON** csi.CSA\_PROP\_USE\_CD = fcd.CD\_ID

**LEFT OUTER JOIN**

        IEAOWN.IE\_FIELD\_CODE fcd2 **ON** csi.COND\_AT\_SALE\_CD = fcd2.CD\_ID

**LEFT OUTER JOIN**

        IEAOWN.IE\_FIELD\_CODE fcd3 **ON** csi.BENCHMARK\_RATE\_CD = fcd3.CD\_ID

**LEFT OUTER JOIN**

        IEAOWN.IE\_CSA\_ANALYSIS ana **ON** csi.CSA\_ID = ana.CSA\_ID

**LEFT OUTER JOIN**

        IEAOWN.IE\_CSA\_ANTICIP\_USE ant **ON** csi.ANTICIPATED\_USE\_CD = ant.ANTICIPATED\_USE\_CD

**LEFT OUTER JOIN**

        IEAOWN.IE\_WORKER WK1 **ON** WK1.WORKER\_ID = csi.SUPRV\_APPR\_UPD\_WKR,

        IEAOWN.IE\_CHG\_OWNER,

        IEAOWN.IE\_REAL\_PROPERTY

**WHERE**

        csi.CSA\_ID = **P\_CSA\_ID**

**AND** IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX = csi.DOC\_PREFIX

**AND** IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES = csi.DOC\_SERIES

**AND** IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = csi.PROPERTY\_ID;

-- Cursor left open for client application

**OPEN** cursor1;

**END** P1

### IE\_GET\_CSA\_PAGE\_TITLE(P\_CSA\_TYPE), mention field code.

SELECT

    IEAOWN.IE\_FIELD\_CODE.CD\_LONG\_NAME

FROM

    IEAOWN.IE\_FIELD\_CODE

WHERE

    IEAOWN.IE\_FIELD\_CODE.CD\_ID = **P\_CSA\_TYPE**

### IE\_GET\_CSA\_TOTAL\_NRA(P\_CSA\_ID)

SELECT SUM(IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA) AS TOTAL\_SQ\_FT\_LEASED

FROM IEAOWN.IE\_CSA\_LEASE

WHERE IEAOWN.IE\_CSA\_LEASE.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_SQFT\_LEASED(P\_CSA\_ID)

SELECT SUM(IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA) AS TOTAL\_SQ\_FT\_LEASED

FROM IEAOWN.IE\_CSA\_LEASE

WHERE IEAOWN.IE\_CSA\_LEASE.VACANT\_FL <> 'Y'

  AND IEAOWN.IE\_CSA\_LEASE.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_AMOUNT(P\_CSA\_ID, P\_AMOUNT\_CD)

SELECT

  IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_CD AS cmc\_amt\_cd,

  IEAOWN.IE\_CSA\_AMOUNT\_CODE.SORT\_SEQ,

  IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_TYPE AS cmc\_amt\_type,

  IEAOWN.IE\_CSA\_AMOUNT\_CODE.UPDATE\_DESC\_FL AS cmc\_update\_desc\_fl,

  IEAOWN.IE\_CSA\_AMOUNT.CSA\_ID,

  IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_CD AS cam\_amt\_cd,

  IEAOWN.IE\_CSA\_AMOUNT.CSA\_AMOUNT\_SEQ\_NUM,

  IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_TYPE AS cam\_amt\_type,

  IEAOWN.IE\_CSA\_AMOUNT.UPDATE\_DESC\_FL AS cam\_update\_desc\_fl

FROM

  IEAOWN.IE\_CSA\_AMOUNT\_CODE cmc

  LEFT OUTER JOIN IEAOWN.IE\_CSA\_AMOUNT cam ON cmc.AMOUNT\_CD = cam.AMOUNT\_CD AND cam.CSA\_ID = **P\_CSA\_ID**

WHERE

  IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_CD = **P\_AMOUNT\_CD**;

### IE\_GET\_CSA\_EXPENSE\_CODE(P\_CSA\_ID)

SELECT

    cmc.AMOUNT\_CD AS CMC\_AMT\_CD,

    cmc.SORT\_SEQ,

    cmc.AMOUNT\_TYPE AS CMC\_AMT\_TYPE,

    cmc.AMOUNT\_DESCRIPTION AS CMC\_AMT\_DESC,

    cmc.UPDATE\_DESC\_FL AS CMC\_UPDATE\_DESC\_FL,

    cam.CSA\_ID,

    cam.AMOUNT\_CD AS CAM\_AMT\_CD,

    cam.CSA\_AMOUNT\_SEQ\_NUM,

    cam.AMOUNT\_TYPE AS CAM\_AMT\_TYPE,

    cam.AMOUNT\_DESCRIPTION AS CAM\_AMT\_DESC,

    cam.UPDATE\_DESC\_FL AS CAM\_UPDATE\_DESC\_FL,

    cam.AMOUNT\_OWNERS,

    cam.AMOUNT\_TENANT

FROM

    IEAOWN.IE\_CSA\_AMOUNT\_CODE cmc

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_AMOUNT cam ON cmc.AMOUNT\_CD = cam.AMOUNT\_CD AND cam.CSA\_ID = **P\_CSA\_ID**

WHERE

    cmc.AMOUNT\_TYPE = 'E'

    AND cmc.AMOUNT\_CD <> 15

ORDER BY

    cmc.SORT\_SEQ;

### IE\_GET\_CSA\_AMOUNT\_ACTUAL\_EXPENSE(P\_CSA\_ID)

SELECT

    cmc.AMOUNT\_CD AS CMC\_AMT\_CD,

    cmc.SORT\_SEQ,

    cmc.AMOUNT\_TYPE AS CMC\_AMT\_TYPE,

    cmc.AMOUNT\_DESCRIPTION AS CMC\_AMT\_DESC,

    cmc.UPDATE\_DESC\_FL AS CMC\_UPDATE\_DESC\_FL,

    cam.CSA\_ID,

    cam.CSA\_AMOUNT\_SEQ\_NUM,

    cam.AMOUNT\_TYPE AS CAM\_AMT\_TYPE,

    cam.AMOUNT\_CD AS CAM\_AMT\_CD,

    cam.AMOUNT\_DESCRIPTION AS CAM\_AMT\_DESC,

    cam.AMOUNT\_OWNERS,

    cam.AMOUNT\_TENANT,

    cam.UPDATE\_DESC\_FL AS CAM\_UPDATE\_DESC\_FL

FROM

    IEAOWN.IE\_CSA\_AMOUNT cam

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_AMOUNT\_CODE cmc

ON

    cam.AMOUNT\_CD = cmc.AMOUNT\_CD

WHERE

    cam.AMOUNT\_TYPE = 'E'

    AND cam.AMOUNT\_CD <> 15

    AND cam.CSA\_ID = **P\_CSA\_ID**

ORDER BY

    cmc.SORT\_SEQ;

### IE\_GET\_CSA\_PROP\_INFO(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID,

    IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX,

    IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES,

    IEAOWN.IE\_CSA\_SALE\_INFO.PROPERTY\_ID,

    IEAOWN.IE\_CSA\_SALE\_INFO.USE\_CD,

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    IEAOWN.IE\_REAL\_PROPERTY.SORT\_PARCEL,

    STRIP(

        CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM

        END

    ) AS address,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_CITY\_NAME,

    'CA' AS SITUS\_STATE,

    STRIP(

        IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_CD

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4, '') = '' THEN ''

            ELSE '-' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4

        END

    ) AS ZIP\_CD,

    IEAOWN.IE\_DOCUMENT.DOC\_PARCEL\_CNT,

    IEAOWN.IE\_CHG\_OWNER.EVENT\_TS,

    IEAOWN.IE\_CHG\_OWNER.IND\_PUR\_PRICE,

    IEAOWN.IE\_CHG\_OWNER.ADJ\_SALES\_PRICE,

    IFNULL(IEAOWN.IE\_CHG\_OWNER.XFER\_TAX\_PRICE\_CO, IEAOWN.IE\_CHG\_OWNER.XFER\_TAX\_PRICE\_CTY) AS TRAN\_TAX\_PRICE,

    IEAOWN.IE\_USE\_CODE.USE\_NAME

FROM

    IEAOWN.IE\_CSA\_SALE\_INFO csa

LEFT OUTER JOIN

    IEAOWN.IE\_USE\_CODE usc ON csa.USE\_CD = usc.USE\_CD,

    IEAOWN.IE\_REAL\_PROPERTY,

    IEAOWN.IE\_DOCUMENT,

    IEAOWN.IE\_CHG\_OWNER

WHERE

    IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX = IEAOWN.IE\_DOCUMENT.DOC\_PREFIX

    AND IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES = IEAOWN.IE\_DOCUMENT.DOC\_SERIES

    AND IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX = IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX

    AND IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES = IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES

    AND IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = IEAOWN.IE\_CSA\_SALE\_INFO.PROPERTY\_ID

    AND IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_COMMENTS(P\_CSA\_ID)

SELECT CSA\_ID, SEQ\_NUM, STRIP(COMMENT\_TEXT) AS COMMENT\_TEXT, ENTRY\_TS, ENTRY\_WORKER, UPDATE\_TS, UPDATE\_WORKER

FROM IEAOWN.IE\_CSA\_COMMENT

WHERE CSA\_ID = **P\_CSA\_ID**

ORDER BY SEQ\_NUM ASC;

### IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE(P\_PROPERTY\_ID, P\_AS\_OF\_DATE)

SELECT

    IEAOWN.IE\_ADDRESS.ADDR\_ID,

    IEAOWN.IE\_ADDRESS.STD\_ADDR\_CD,

    IEAOWN.IE\_ADDRESS.MAILING\_NAME,

    IEAOWN.IE\_ADDRESS.CARE\_OF\_NAME,

    IEAOWN.IE\_ADDRESS.ATTENTION\_NAME

FROM

    IEAOWN.IE\_PROP\_ID\_ADDRESS

INNER JOIN

    IEAOWN.IE\_ADDRESS ON IEAOWN.IE\_ADDRESS.ADDR\_ID = IEAOWN.IE\_PROP\_ID\_ADDRESS.ADDR\_ID

WHERE

    IEAOWN.IE\_PROP\_ID\_ADDRESS.PROPERTY\_ID = **P\_PROPERTY\_ID**

    AND IEAOWN.IE\_ADDRESS.ADDR\_TYPE\_CD = 4

    AND IEAOWN.IE\_ADDRESS.EFFECTIVE\_TS <= **P\_AS\_OF\_DATE**

    AND (IEAOWN.IE\_ADDRESS.END\_TS >= **P\_AS\_OF\_DATE** OR IEAOWN.IE\_ADDRESS.END\_TS IS NULL);

### IE\_GET\_CSA\_TAX\_RATE(P\_PROPERTY\_ID, P\_EVENT\_DATE, P\_ROLL\_YEAR)

SELECT

    IEAOWN.IE\_TAX\_RATE.TAX\_RATE

FROM

    IEAOWN.IE\_TAX\_RATE

INNER JOIN

    IEAOWN.IE\_TRA\_HISTORY ON IEAOWN.IE\_TAX\_RATE.TRA\_PRIME = IEAOWN.IE\_TRA\_HISTORY.TRA\_PRIME

    AND IEAOWN.IE\_TAX\_RATE.TRA\_SEC = IEAOWN.IE\_TRA\_HISTORY.TRA\_SEC

WHERE

    IEAOWN.IE\_TRA\_HISTORY.PROPERTY\_ID = **P\_PROPERTY\_ID**

    AND IEAOWN.IE\_TRA\_HISTORY.EFFECTIVE\_DT <= **P\_EVENT\_DATE**

    AND (IEAOWN.IE\_TRA\_HISTORY.END\_DT IS NULL OR IEAOWN.IE\_TRA\_HISTORY.END\_DT >= **P\_EVENT\_DATE**)

    AND IEAOWN.IE\_TAX\_RATE.ROLL\_YR = **P\_ROLL\_YEAR**;

### IE\_GET\_CSA\_LEASE\_INFO(P\_CSA\_ID)

SELECT DISTINCT

    IEAOWN.IE\_CSA\_LEASE.LEASE\_ID,

    IEAOWN.IE\_CSA\_LEASE.CSA\_ID,

    IEAOWN.IE\_CSA\_LEASE.PROPERTY\_ID,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_LOCATION,

    IEAOWN.IE\_CSA\_LEASE.SITUS\_CITY\_NAME,

    IEAOWN.IE\_CSA\_LEASE.LESSEE\_NAME,

    IEAOWN.IE\_CSA\_LEASE.VACANT\_FL,

    IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_START\_DT,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_TERM,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_OPTION,

    IEAOWN.IE\_CSA\_LEASE.INITIAL\_RENT,

    IEAOWN.IE\_CSA\_LEASE.INITIAL\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.STABIL\_RENT,

    IEAOWN.IE\_CSA\_LEASE.STABIL\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT,

    IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT,

    IEAOWN.IE\_CSA\_LEASE.ANTI\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.RENT\_ADJUSTMENTS,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_SUBCATEGORY,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_TYPE\_CD,

    FCD1.CD\_LONG\_NAME AS LEASE\_TYPE\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.EXPENSE\_TYPE\_CD,

    FCD2.CD\_LONG\_NAME AS EXPENSE\_TYPE\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.TENANCY\_CD,

    IEAOWN.IE\_CSA\_LEASE.INFO\_SOURCE\_CD,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_PARKING,

    IEAOWN.IE\_CSA\_LEASE.SPRINKLER\_FL,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_CREATED\_BY,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_CREATE\_DT,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_MODIFY\_BY,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_MODIFY\_DT,

    IEAOWN.IE\_CSA\_LEASE.USE\_FOR\_COMP\_FL,

    IEAOWN.IE\_CSA\_LEASE.ENTRY\_TS,

    IEAOWN.IE\_CSA\_LEASE.ENTRY\_WORKER,

    IEAOWN.IE\_CSA\_LEASE.UPDATE\_TS,

    IEAOWN.IE\_CSA\_LEASE.UPDATE\_WORKER,

    CASE

        WHEN IEAOWN.IE\_CSA\_WKS\_INCOME.CSA\_WKS\_NUM IS NULL THEN 'N'

        ELSE 'Y'

    END AS WKS\_EXIST\_FL,

    CASE

        WHEN IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT\_PSF IS NULL THEN IEAOWN.IE\_CSA\_LEASE.ANTI\_RENT\_PSF

        ELSE IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT\_PSF

    END AS RENT\_PSF,

    CASE

        WHEN LENGTH(RSTRIP(IEAOWN.IE\_CSA\_LEASE.LEASE\_OPTION)) > 0 THEN 'Y'

        ELSE 'N'

    END AS LEASE\_OPTION\_FL,

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    STRIP(

        CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM, '')

            WHEN '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT, '')

            WHEN '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT)

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME, '')

            WHEN '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX, '')

            WHEN '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX)

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT, '')

            WHEN '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT)

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG, '')

            WHEN '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG)

        END

        || CASE

            IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM, '')

            WHEN '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM

        END

    ) AS address,

    'CA' AS SITUS\_STATE,

    STRIP(

        IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_CD

        || '-' || COALESCE(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4, '')

    ) AS ZIP\_CD

FROM

    IEAOWN.IE\_CSA\_LEASE

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE FCD1 ON IEAOWN.IE\_CSA\_LEASE.LEASE\_TYPE\_CD = FCD1.CD\_ID

LEFT OUTER JOIN IEAOWN.IE\_CSA\_WKS\_INCOME ON IEAOWN.IE\_CSA\_LEASE.LEASE\_ID = IEAOWN.IE\_CSA\_WKS\_INCOME.LEASE\_ID

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE FCD2 ON IEAOWN.IE\_CSA\_LEASE.EXPENSE\_TYPE\_CD = FCD2.CD\_ID,

IEAOWN.IE\_REAL\_PROPERTY

WHERE

    IEAOWN.IE\_CSA\_LEASE.PROPERTY\_ID = IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID

    AND IEAOWN.IE\_CSA\_LEASE.CSA\_ID = **P\_CSA\_ID**

ORDER BY

    IEAOWN.IE\_CSA\_LEASE.LEASE\_ID;

### IE\_GET\_CSA\_TOTAL\_RENT(P\_CSA\_ID)

SELECT

    SUM(IFNULL(IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA, 0)) AS NRA\_SUM,

    SUM(

        CASE

            WHEN IEAOWN.IE\_CSA\_LEASE.VACANT\_FL = 'Y' THEN 0

            ELSE IFNULL(IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA, 0)

        END

    ) AS OCCUPIED\_SQ\_FT,

    SUM(IFNULL(IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT, 0)) AS CURRENT\_RENT\_SUM,

    SUM(IFNULL(IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT, 0)) AS ANTICIP\_RENT\_SUM,

    IFNULL(SUM(IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT), 0) + IFNULL(SUM(IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT), 0) AS TOTAL\_RENT

FROM

    IEAOWN.IE\_CSA\_LEASE

WHERE

    IEAOWN.IE\_CSA\_LEASE.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_TOTAL\_RENT\_EXC\_LEASE

SELECT

    SUM(IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA) AS NRA\_SUM,

    SUM(IFNULL(IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT, 0)) AS CURRENT\_RENT\_SUM,

    SUM(IFNULL(IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT, 0)) AS ANTICIP\_RENT\_SUM,

    IFNULL(SUM(IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT), 0) + IFNULL(SUM(IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT), 0) AS TOTAL\_RENT

FROM

    IEAOWN.IE\_CSA\_LEASE

WHERE

    IEAOWN.IE\_CSA\_LEASE.CSA\_ID = **P\_CSA\_ID**

    AND IEAOWN.IE\_CSA\_LEASE.LEASE\_ID <> **P\_EXCLUDEDLEASEID\_ID**;

### IE\_GET\_CSA\_NRA\_OCCUPANCY(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_CSA\_SALE\_INFO.TOT\_NET\_RENT\_AREA,

    IEAOWN.IE\_CSA\_ANALYSIS.PCT\_OCCUPANCY

FROM

    IEAOWN.IE\_CSA\_SALE\_INFO

JOIN

    IEAOWN.IE\_CSA\_ANALYSIS ON IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = IEAOWN.IE\_CSA\_ANALYSIS.CSA\_ID

WHERE

    IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = **P\_CSA\_ID**;

### IE\_DEL\_CSA\_LEASE(P\_LEASE\_ID)

DELETE FROM IEAOWN.IE\_CSA\_LEASE

WHERE IEAOWN.IE\_CSA\_LEASE.LEASE\_ID = **P\_LEASE\_ID**

### IE\_GET\_CSA\_ANALYSIS(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_CSA\_ANALYSIS.CSA\_ID,

    IEAOWN.IE\_CSA\_ANALYSIS.PCT\_PGI,

    IEAOWN.IE\_CSA\_ANALYSIS.ADDITION\_NET\_INCOM,

    IEAOWN.IE\_CSA\_ANALYSIS.TENANT\_REIMBURSE,

    IEAOWN.IE\_CSA\_ANALYSIS.ANTI\_PROPERTY\_TAX,

    IEAOWN.IE\_CSA\_ANALYSIS.OTHER\_TAXES\_DESC,

    IEAOWN.IE\_CSA\_ANALYSIS.OTHER\_TAXES\_AMOUNT,

    IEAOWN.IE\_CSA\_ANALYSIS.TAX\_RATE,

    IEAOWN.IE\_CSA\_ANALYSIS.PCT\_OCCUPANCY,

    IEAOWN.IE\_CSA\_ANALYSIS.ENTRY\_TS,

    IEAOWN.IE\_CSA\_ANALYSIS.ENTRY\_WORKER,

    IEAOWN.IE\_CSA\_ANALYSIS.UPDATE\_TS,

    IEAOWN.IE\_CSA\_ANALYSIS.UPDATE\_WORKER,

    IEAOWN.IE\_CSA\_ANALYSIS.PCT\_OCCUP\_MINIMUM

FROM

    IEAOWN.IE\_CSA\_ANALYSIS

WHERE

    IEAOWN.IE\_CSA\_ANALYSIS.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_ALL\_APN\_IN\_DOC(P\_CSA\_ID)

SELECT IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL

FROM IEAOWN.IE\_DOCUMENT

INNER JOIN IEAOWN.IE\_PARCEL\_DET ON

    IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_PARCEL\_DET.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_PARCEL\_DET.DOC\_SERIES

INNER JOIN IEAOWN.IE\_REAL\_PROPERTY ON

    IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL = IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL

INNER JOIN IEAOWN.IE\_CSA\_SALE\_INFO ON

    IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES

WHERE

    IEAOWN.IE\_PARCEL\_DET.END\_DT IS NULL

    AND IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = @CSAId

ORDER BY IEAOWN.IE\_PARCEL\_DET.SEQ\_NUM;

### IE\_GET\_CSA\_PROP\_CHAR(P\_CSA\_ID)

SELECT

    csa.CSA\_ID,

    csa.ZONING\_AGENCY\_CD,

    csa.ZONING\_CD,

    zoc.ZONING\_NAME,

    csa.TOT\_BUILDING\_AREA,

    csa.TOT\_NET\_RENT\_AREA,

    csa.TOT\_LOT\_SIZE,

    char.SBE\_CLASS\_LOW,

    char.SBE\_CLASS\_HI,

    char.MVS\_CLASS\_LOW,

    char.MVS\_CLASS\_HI,

    char.BLDG\_BUILT\_YR\_LOW,

    char.BLDG\_BUILT\_YR\_HI,

    char.BLDG\_EFF\_YR\_LOW,

    char.BLDG\_EFF\_YR\_HI,

    char.PCT\_BUILT\_OUT\_LOW,

    char.PCT\_BUILT\_OUT\_HI,

    char.PCT\_OFFICE\_LOW,

    char.PCT\_OFFICE\_HI,

    char.NUM\_BUILDINGS\_LOW,

    char.NUM\_BUILDINGS\_HI,

    char.NUM\_STORIES\_LOW,

    char.NUM\_STORIES\_HI,

    char.AVG\_ACTUAL\_HT\_LOW,

    char.AVG\_ACTUAL\_HT\_HI,

    char.LAND\_IMP\_RATIO\_LOW,

    char.LAND\_IMP\_RATIO\_HI,

    char.FAR\_LOW,

    char.FAR\_HI,

    char.TOT\_PARKING

FROM

    IEAOWN.IE\_CSA\_SALE\_INFO csa

LEFT OUTER JOIN

    IEAOWN.IE\_ZONING\_CODE zoc ON

    csa.ZONING\_AGENCY\_CD = zoc.AGENCY\_CD

    AND csa.ZONING\_CD = zoc.ZONING\_CD

JOIN

    IEAOWN.IE\_CSA\_PROP\_CHAR char ON

    csa.CSA\_ID = char.CSA\_ID

WHERE

    csa.CSA\_ID =**P\_CSA\_ID**;

### IE\_GET\_CSA\_INCOME\_EXPENSE\_ANALYSIS(P\_CSA\_ID)

Swapped out subqueries and use temp table instead of subqueries

Create temp table for LEASE subquery

CREATE TEMPORARY TABLE temp\_lease\_pgi AS

SELECT

    CSA\_ID,

    (IFNULL(SUM(CURRENT\_RENT), 0) + IFNULL(SUM(ANTICIPATED\_RENT), 0)) \* 12 AS PGI

FROM

    IEAOWN.IE\_CSA\_LEASE

WHERE

    CSA\_ID = P\_CSA\_ID

GROUP BY

    CSA\_ID;

Create temp table for AMOUNT1 subquery

CREATE TEMPORARY TABLE temp\_amount1 AS

SELECT

    CSA\_ID,

    IFNULL(SUM(AMOUNT\_OWNERS), 0) AS EXP\_OWNERS\_SUBTOTL,

    IFNULL(SUM(AMOUNT\_TENANT), 0) AS EXP\_TENANT\_SUBTOTL

FROM

    IEAOWN.IE\_CSA\_AMOUNT

WHERE

    CSA\_ID = P\_CSA\_ID

    AND CSA\_AMOUNT\_SEQ\_NUM < 14

GROUP BY

    CSA\_ID;

Create temp table for AMOUNT2 subquery

CREATE TEMPORARY TABLE temp\_amount2 AS

SELECT

    CSA\_ID,

    IFNULL(SUM(AMOUNT\_OWNERS), 0) AS EXP\_OWNERS\_TOTAL,

    IFNULL(SUM(AMOUNT\_TENANT), 0) AS EXP\_TENANT\_TOTAL

FROM

    IEAOWN.IE\_CSA\_AMOUNT

WHERE

    CSA\_ID = P\_CSA\_ID

GROUP BY

    CSA\_ID;

Create temp table for AMOUNT3 subquery

CREATE TEMPORARY TABLE temp\_amount3 AS

SELECT

    CSA\_ID,

    IFNULL(SUM(AMOUNT\_CSA), 0) AS CSA\_ADJUST\_TOTAL

FROM

    IEAOWN.IE\_CSA\_AMOUNT

WHERE

    CSA\_ID = P\_CSA\_ID

    AND AMOUNT\_CD IN (20, 22, 23)

GROUP BY

    CSA\_ID;

Main query using temp tables

SELECT

    ANA.CSA\_ID,

    ANA.PCT\_PGI,

    ANA.ADDITION\_NET\_INCOM,

    ANA.TENANT\_REIMBURSE,

    ANA.ANTI\_PROPERTY\_TAX,

    ANA.OTHER\_TAXES\_DESC,

    ANA.OTHER\_TAXES\_AMOUNT,

    ANA.TAX\_RATE,

    ANA.PCT\_OCCUPANCY,

    ANA.PCT\_OCCUP\_MINIMUM,

    ANA.ENTRY\_TS,

    ANA.ENTRY\_WORKER,

    ANA.UPDATE\_TS,

    ANA.UPDATE\_WORKER,

    ANA.CSA\_NET\_RENT\_AREA,

    LEASE.PGI,

    AMOUNT1.EXP\_OWNERS\_SUBTOTL,

    AMOUNT1.EXP\_TENANT\_SUBTOTL,

    AMOUNT2.EXP\_OWNERS\_TOTAL,

    AMOUNT2.EXP\_TENANT\_TOTAL,

    AMOUNT3.CSA\_ADJUST\_TOTAL

FROM

    IEAOWN.IE\_CSA\_ANALYSIS ANA

LEFT OUTER JOIN

    temp\_lease\_pgi LEASE ON ANA.CSA\_ID = LEASE.CSA\_ID

JOIN

    temp\_amount1 AMOUNT1 ON ANA.CSA\_ID = AMOUNT1.CSA\_ID

JOIN

    temp\_amount2 AMOUNT2 ON ANA.CSA\_ID = AMOUNT2.CSA\_ID

JOIN

    temp\_amount3 AMOUNT3 ON ANA.CSA\_ID = AMOUNT3.CSA\_ID

WHERE

    ANA.CSA\_ID = AMOUNT1.CSA\_ID

    AND ANA.CSA\_ID = AMOUNT2.CSA\_ID

    AND ANA.CSA\_ID = AMOUNT3.CSA\_ID

    AND ANA.CSA\_ID = P\_CSA\_ID;

DROP TEMPORARY TABLE IF EXISTS temp\_lease\_pgi;

DROP TEMPORARY TABLE IF EXISTS temp\_amount1;

DROP TEMPORARY TABLE IF EXISTS temp\_amount2;

DROP TEMPORARY TABLE IF EXISTS temp\_amount3;

### IE\_GET\_CSA\_EXPENSE(P\_CSA\_ID)

SELECT

    AMT.CSA\_ID,

    AMT.CSA\_AMOUNT\_SEQ\_NUM,

    AMT.AMOUNT\_TYPE,

    AMT.AMOUNT\_CD,

    CASE

        WHEN AMTCD.UPDATE\_DESC\_FL = 'Y' THEN AMT.AMOUNT\_DESCRIPTION

        ELSE AMTCD.AMOUNT\_DESCRIPTION

    END AS AMT\_DISPLAY\_TEXT,

    AMT.AMOUNT\_OWNERS,

    CASE

        WHEN CSAANA.CSA\_NET\_RENT\_AREA > 0 THEN AMT.AMOUNT\_OWNERS / CSAANA.CSA\_NET\_RENT\_AREA

        ELSE

            CASE

                WHEN SALEINFO.TOT\_BUILDING\_AREA > 0 THEN AMT.AMOUNT\_OWNERS / SALEINFO.TOT\_BUILDING\_AREA

                ELSE NULLIF(0.0, 0.0)

            END

    END AS OWNER\_DLR\_PER\_SQFT,

    AMT.AMOUNT\_TENANT,

    CASE

        WHEN CSAANA.CSA\_NET\_RENT\_AREA > 0 THEN AMT.AMOUNT\_TENANT / CSAANA.CSA\_NET\_RENT\_AREA

        ELSE

            CASE

                WHEN SALEINFO.TOT\_BUILDING\_AREA > 0 THEN AMT.AMOUNT\_TENANT / SALEINFO.TOT\_BUILDING\_AREA

                ELSE NULLIF(0.0, 0.0)

            END

    END AS TENT\_DLR\_PER\_SQFT,

    AMT.AMOUNT\_CSA,

    AMT.UPDATE\_DESC\_FL,

    CASE

        WHEN AMT.AMOUNT\_CD < 14 THEN '1'

        ELSE '2'

    END AS GROUP\_BY\_TYPE,

    AMT.ENTRY\_TS,

    AMT.ENTRY\_WORKER,

    AMT.UPDATE\_TS,

    AMT.UPDATE\_WORKER,

    SALEINFO.TOT\_BUILDING\_AREA,

    SALEINFO.TOT\_NET\_RENT\_AREA,

    CSAANA.CSA\_NET\_RENT\_AREA

FROM

    IEAOWN.IE\_CSA\_AMOUNT AMT,

    IEAOWN.IE\_CSA\_SALE\_INFO SALEINFO,

    IEAOWN.IE\_CSA\_ANALYSIS CSAANA,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE AMTCD

WHERE

    SALEINFO.CSA\_ID = AMT.CSA\_ID

    AND SALEINFO.CSA\_ID = CSAANA.CSA\_ID

    AND AMT.AMOUNT\_CD = AMTCD.AMOUNT\_CD

    AND AMTCD.AMOUNT\_TYPE = 'E'

    AND AMT.CSA\_ID = @CSAId

ORDER BY

    AMTCD.SORT\_SEQ;

### IE\_GET\_CSA\_RPT\_SOURCE(P\_CSA\_ID)

SELECT

    CSS.CSA\_ID,

    CSS.SOURCE\_CD\_PRI,

    CSS.SOURCE\_NAME\_PRI,

    CSS.SOURCE\_TITLE\_PRI,

    CSS.SOURCE\_COMPANY\_PRI,

    CSS.SOURCE\_PHONE\_PRI,

    CSS.SOURCE\_CD\_SEC,

    CSS.SOURCE\_NAME\_SEC,

    CSS.SOURCE\_TITLE\_SEC,

    CSS.SOURCE\_COMPANY\_SEC,

    CSS.SOURCE\_PHONE\_SEC,

    CSS.SOURCE\_OTHER\_DESC,

    CSS.SRCE\_IQ\_FL,

    CSS.SRCE\_AAB\_FL,

    CSS.SRCE\_PCOR\_COS\_FL,

    CSS.SRCE\_RFR\_FL,

    CSS.SRCE\_PH\_EX\_PRI,

    CSS.SRCE\_ALT\_PH\_PRI,

    CSS.SRCE\_ALT\_PH\_EX\_PRI,

    CSS.SRCE\_EMAIL\_PRI,

    CSS.SRCE\_PH\_EX\_SEC,

    CSS.SRCE\_ALT\_PH\_SEC,

    CSS.SRCE\_ALT\_PH\_EX\_SEC,

    CSS.SRCE\_EMAIL\_SEC,

    CSS.SOURCE\_CREATED\_BY,

    CSS.SOURCE\_CREATE\_DT,

    CSS.SOURCE\_MODIFIED\_BY,

    CSS.SOURCE\_MODIFIED\_DT,

    CSS.ENTRY\_TS,

    CSS.ENTRY\_WORKER,

    CSS.UPDATE\_TS,

    CSS.UPDATE\_WORKER,

    CSS.SRCE\_MODIFY\_RSN\_CD,

    FCD1.CD\_LONG\_NAME AS SOURCE\_PRI\_TEXT,

    FCD2.CD\_LONG\_NAME AS SOURCE\_SEC\_TEXT,

    FCD3.CD\_LONG\_NAME AS MODIFY\_RSN\_TEXT,

    (WK1.LAST\_NAME || CASE IFNULL(WK1.FIRST\_NAME, '')

        WHEN '' THEN '' ELSE ', ' || WK1.FIRST\_NAME END

        || CASE IFNULL(WK1.MIDDLE\_NAME, '')

            WHEN '' THEN '' ELSE ' ' || WK1.MIDDLE\_NAME END) AS CREATED\_WORKER,

    (WK2.LAST\_NAME || CASE IFNULL(WK2.FIRST\_NAME, '')

        WHEN '' THEN '' ELSE ', ' || WK2.FIRST\_NAME END

        || CASE IFNULL(WK2.MIDDLE\_NAME, '')

            WHEN '' THEN '' ELSE ' ' || WK2.MIDDLE\_NAME END) AS MODIFIED\_WORKER

FROM

    IEAOWN.IE\_CSA\_SOURCE CSS

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD1 ON FCD1.CD\_ID = CSS.SOURCE\_CD\_PRI

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD2 ON FCD2.CD\_ID = CSS.SOURCE\_CD\_SEC

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD3 ON FCD3.CD\_ID = CSS.SRCE\_MODIFY\_RSN\_CD

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK1 ON WK1.WORKER\_ID = CSS.SOURCE\_CREATED\_BY

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK2 ON WK2.WORKER\_ID = CSS.SOURCE\_MODIFIED\_BY

WHERE

    CSS.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_RPT\_ADJ(P\_CSA\_ID)

SELECT

    AMT.CSA\_ID,

    AMTCD.SORT\_SEQ,

    '' AS LEFT\_ROW\_NO,

    CASE AMT.AMOUNT\_CD

        WHEN 16 THEN '(a)'

        WHEN 17 THEN '(b)'

        WHEN 18 THEN '(c)'

        WHEN 19 THEN '(d)'

        WHEN 20 THEN '(f)'

        WHEN 22 THEN '(g)'

        WHEN 23 THEN '(h)'

    END AS RIGHT\_ROW\_NO,

    AMT.CSA\_AMOUNT\_SEQ\_NUM,

    AMT.AMOUNT\_CD,

    AMT.AMOUNT\_DESCRIPTION AS AMT\_TEXT,

    AMTCD.AMOUNT\_DESCRIPTION AS AMTCD\_TEXT,

    CASE

        WHEN AMTCD.UPDATE\_DESC\_FL = 'Y' THEN AMT.AMOUNT\_DESCRIPTION

        ELSE ''

    END AS AMT\_DISPLAY\_TEXT,

    CASE

        WHEN AMTCD.AMOUNT\_CD = 21 THEN NULLIF(0, 0)

        ELSE AMT.AMOUNT\_CSA

    END AS AMOUNT\_CSA\_LEFT,

    AMT.AMOUNT\_CSA AS AMOUNT\_CSA,

    CASE

        WHEN AMT.AMOUNT\_CD = 21 THEN 'BONDS'

        ELSE 'ADJ'

    END AS ROW\_TYPE

FROM

    IEAOWN.IE\_CSA\_AMOUNT AMT,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE AMTCD

WHERE

    AMT.AMOUNT\_CD = AMTCD.AMOUNT\_CD

    AND AMTCD.AMOUNT\_TYPE = 'A'

    AND AMT.CSA\_ID = **P\_CSA\_ID**

UNION

SELECT

    AMT.CSA\_ID,

    5 AS SORT\_SEQ,

    '(e)' AS LEFT\_ROW\_NO,

    'Total (a thru d)' AS RIGHT\_ROW\_NO,

    0 AS CSA\_AMOUNT\_SEQ\_NUM,

    0 AS AMOUNT\_CD,

    '' AS AMT\_TEXT,

    'Ind Pur Price' AS AMTCD\_TEXT,

    '' AS AMT\_DISPLAY\_TEXT,

    NULLIF(0, 0) AS AMOUNT\_CSA\_LEFT,

    SUM(AMT.AMOUNT\_CSA) AS AMOUNT\_CSA,

    'PUR' AS ROW\_TYPE

FROM

    IEAOWN.IE\_CSA\_AMOUNT AMT,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE AMTCD

WHERE

    AMT.AMOUNT\_CD = AMTCD.AMOUNT\_CD

    AND AMTCD.AMOUNT\_CD IN (16, 17, 18, 19)

    AND AMT.CSA\_ID = **P\_CSA\_ID**

GROUP BY

    AMT.CSA\_ID

UNION

SELECT

    AMT.CSA\_ID,

    5.5 AS SORT\_SEQ,

    '(e)' AS LEFT\_ROW\_NO,

    '(e)' AS RIGHT\_ROW\_NO,

    0 AS CSA\_AMOUNT\_SEQ\_NUM,

    0 AS AMOUNT\_CD,

    '' AS AMT\_TEXT,

    'PCOR Ind Pur Price' AS AMTCD\_TEXT,

    '' AS AMT\_DISPLAY\_TEXT,

    NULLIF(0, 0) AS AMOUNT\_CSA\_LEFT,

    CO.IND\_PUR\_PRICE AS AMOUNT\_CSA,

    'PUR' AS ROW\_TYPE

FROM

    IEAOWN.IE\_CSA\_AMOUNT AMT,

    IEAOWN.IE\_CSA\_SALE\_INFO CSI,

    IEAOWN.IE\_CHG\_OWNER CO

WHERE

    CSI.DOC\_PREFIX = CO.DOC\_PREFIX

    AND CSI.DOC\_SERIES = CO.DOC\_SERIES

    AND AMT.CSA\_ID = CSI.CSA\_ID

    AND AMT.CSA\_ID = **P\_CSA\_ID**

UNION

SELECT

    AMT.CSA\_ID,

    11 AS SORT\_SEQ,

    '(i)' AS LEFT\_ROW\_NO,

    'Total (e thru h)' AS RIGHT\_ROW\_NO,

    0 AS CSA\_AMOUNT\_SEQ\_NUM,

    0 AS AMOUNT\_CD,

    '' AS AMT\_TEXT,

    'Adj Sales Price' AS AMTCD\_TEXT,

    '' AS AMT\_DISPLAY\_TEXT,

    NULLIF(0, 0) AS AMOUNT\_CSA\_LEFT,

    SUM(AMT.AMOUNT\_CSA) AS AMOUNT\_CSA,

    'SALES' AS ROW\_TYPE

FROM

    IEAOWN.IE\_CSA\_AMOUNT AMT,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE AMTCD

WHERE

    AMT.AMOUNT\_CD = AMTCD.AMOUNT\_CD

    AND AMTCD.AMOUNT\_CD IN (20, 22, 23)

    AND AMT.CSA\_ID = **P\_CSA\_ID**

GROUP BY

    AMT.CSA\_ID

ORDER BY

    SORT\_SEQ;

### IE\_GET\_CSA\_ADJ\_FOR\_ALL\_ADJ\_CODES(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_CD AS cmc\_amt\_cd,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.SORT\_SEQ,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_TYPE AS cmc\_amt\_type,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_DESCRIPTION AS cmc\_amt\_desc,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.UPDATE\_DESC\_FL AS cmc\_update\_desc\_fl,

    IEAOWN.IE\_CSA\_AMOUNT.CSA\_ID,

    IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_CD AS cam\_amt\_cd,

    IEAOWN.IE\_CSA\_AMOUNT.CSA\_AMOUNT\_SEQ\_NUM,

    IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_TYPE AS cam\_amt\_type,

    IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_DESCRIPTION AS cam\_amt\_desc,

    IEAOWN.IE\_CSA\_AMOUNT.UPDATE\_DESC\_FL AS cam\_update\_desc\_fl,

    IEAOWN.IE\_CSA\_AMOUNT.AMOUNT\_CSA

FROM

    IEAOWN.IE\_CSA\_AMOUNT\_CODE cmc

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_AMOUNT cam

ON

    cmc.AMOUNT\_CD = cam.AMOUNT\_CD

    AND cam.CSA\_ID = **P\_CSA\_ID**

WHERE

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_TYPE = 'A'

ORDER BY

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.SORT\_SEQ;

### IE\_GET\_CSA\_AMOUNT\_CODES(P\_AMOUNT\_TYPE)

SELECT

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_CD,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_TYPE,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.AMOUNT\_DESCRIPTION,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.SORT\_SEQ,

    IEAOWN.IE\_CSA\_AMOUNT\_CODE.UPDATE\_DESC\_FL

FROM

    IEAOWN.IE\_CSA\_AMOUNT\_CODE

WHERE

    AMOUNT\_TYPE = **P\_AMOUNT\_TYPE**

ORDER BY

    SORT\_SEQ;

### IE\_GET\_FIELD\_CODE(P\_FIELD\_ID)

This is an existing SP and should be used as is.

The entity should reflect the returned columns

Parameters to this table should be added with command in ADO.net

### IE\_GET\_CSA\_ANTICIPATED\_USE\_CODES\_NOT\_FOR\_LAND\_USE

SELECT

    IEAOWN.IE\_CSA\_ANTICIP\_USE.ANTICIPATED\_USE\_CD,

    IEAOWN.IE\_CSA\_ANTICIP\_USE.USE\_NAME,

    IEAOWN.IE\_CSA\_ANTICIP\_USE.USE\_NAME\_SHORT

FROM

    IEAOWN.IE\_CSA\_ANTICIP\_USE

WHERE

    (LAND\_USE\_FL IS NULL OR LAND\_USE\_FL <> 'Y')

ORDER BY

    CAST(UPPER(USE\_NAME) AS VARCHAR(50) CCSID ASCII);

### IE\_GET\_CSA\_ALL\_APN\_IN\_DOC(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID,

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    IEAOWN.IE\_REAL\_PROPERTY.SORT\_PARCEL,

    IEAOWN.IE\_PARCEL\_DET.SEQ\_NUM

FROM

    IEAOWN.IE\_DOCUMENT,

    IEAOWN.IE\_PARCEL\_DET,

    IEAOWN.IE\_REAL\_PROPERTY,

    IEAOWN.IE\_CSA\_SALE\_INFO

WHERE

    IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_PARCEL\_DET.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_PARCEL\_DET.DOC\_SERIES

    AND IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES

    AND IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL = IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL

    AND (IEAOWN.IE\_PARCEL\_DET.END\_DT IS NULL)

    AND IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = **P\_CSA\_ID**

ORDER BY

    IEAOWN.IE\_PARCEL\_DET.SEQ\_NUM;

### IE\_GET\_CSA\_SOURCE(P\_CSA\_ID)

SELECT

    css.CSA\_ID,

    css.SOURCE\_CD\_PRI,

    css.SOURCE\_NAME\_PRI,

    css.SOURCE\_TITLE\_PRI,

    css.SOURCE\_COMPANY\_PRI,

    css.SOURCE\_PHONE\_PRI,

    css.SOURCE\_CD\_SEC,

    css.SOURCE\_NAME\_SEC,

    css.SOURCE\_TITLE\_SEC,

    css.SOURCE\_COMPANY\_SEC,

    css.SOURCE\_PHONE\_SEC,

    css.SOURCE\_OTHER\_DESC,

    css.SOURCE\_CREATED\_BY,

    css.SOURCE\_CREATE\_DT,

    css.SOURCE\_MODIFIED\_BY,

    css.SOURCE\_MODIFIED\_DT,

    css.ENTRY\_TS,

    css.ENTRY\_WORKER,

    css.UPDATE\_TS,

    css.UPDATE\_WORKER,

    css.SRCE\_PH\_EX\_PRI,

    css.SRCE\_ALT\_PH\_PRI,

    css.SRCE\_ALT\_PH\_EX\_PRI,

    css.SRCE\_EMAIL\_PRI,

    css.SRCE\_PH\_EX\_SEC,

    css.SRCE\_ALT\_PH\_SEC,

    css.SRCE\_ALT\_PH\_EX\_SEC,

    css.SRCE\_EMAIL\_SEC,

    CASE css.SRCE\_IQ\_FL WHEN 'Y' THEN '1' ELSE '0' END AS SRCE\_IQ\_FL,

    CASE css.SRCE\_AAB\_FL WHEN 'Y' THEN '1' ELSE '0' END AS SRCE\_AAB\_FL,

    CASE css.SRCE\_PCOR\_COS\_FL WHEN 'Y' THEN '1' ELSE '0' END AS SRCE\_PCOR\_COS\_FL,

    CASE css.SRCE\_RFR\_FL WHEN 'Y' THEN '1' ELSE '0' END AS SRCE\_RFR\_FL,

    fcd.cd\_short\_name,

    '' AS source\_desc\_Sec,

    '' AS cmp\_created\_by,

    '' AS cmp\_modified\_by,

    css.SRCE\_MODIFY\_RSN\_CD,

    csi.RELEASE\_DT,

    csi.SUPRV\_APPR\_UPD\_WKR,

    csi.SUPRV\_APPR\_UPD\_TS,

    csi.SUPRV\_APPROVED\_FL,

    CASE csi.SUPRV\_APPROVED\_FL WHEN 'Y' THEN 'Yes' ELSE 'No' END AS SUPRV\_APPR\_TEXT

FROM

    ieaown.ie\_csa\_source css

LEFT OUTER JOIN

    ieaown.ie\_field\_code fcd ON fcd.cd\_id = css.source\_cd\_pri,

    ieaown.ie\_csa\_sale\_info csi

WHERE

    css.CSA\_ID = **P\_CSA\_ID**

    AND csi.CSA\_ID = css.CSA\_ID;

### IE\_GET\_CSA\_INCOME\_ANALYSIS(P\_CSA\_ID)

SELECT

    csa.csa\_id,

    csa.pct\_pgi,

    csa.addition\_net\_incom,

    csa.tenant\_reimburse,

    csa.anti\_property\_tax,

    csa.other\_taxes\_desc,

    csa.other\_taxes\_amount,

    csa.tax\_rate,

    csa.csa\_net\_rent\_area,

    csa.pct\_occupancy,

    csa.pct\_occup\_minimum,

    csa.update\_ts,

    csa.update\_worker,

    csi.property\_id,

    csi.full\_expenses\_fl,

    csi.tot\_net\_rent\_area,

    csi.tot\_building\_area,

    csi.benchmark\_rate\_cd,

    csi.pct\_owner\_occup,

    chg.event\_ts,

    chg.adj\_sales\_price,

    chg.ind\_pur\_price,

    bm\_desc.cd\_long\_name AS bm\_desc,

    '   ' AS cmp\_OAR\_desc,

    0 AS cmp\_pot\_gross\_inc,

    0 AS cmp\_vac\_coll\_loss,

    0 AS cmp\_eff\_gross\_inc,

    0 AS cmp\_tot\_expenses,

    0 AS cmp\_net\_op\_inc,

    0 AS cmp\_egi\_pct,

    0 AS cmp\_noi\_adj\_sale,

    0 AS cmp\_adj\_sale\_pgi,

    0 AS cmp\_adj\_sale\_nra,

    0 AS cmp\_rent\_nra,

    0 AS cmp\_noi\_nra,

    0 AS cmp\_expense\_nra,

    0 AS cmp\_tot\_xpens\_nra,

    0 AS pct\_occup\_threshld

FROM

    ieaown.ie\_csa\_analysis csa

INNER JOIN

    ieaown.ie\_chg\_owner chg ON csa.csa\_id = chg.csa\_id

INNER JOIN

    ieaown.ie\_csa\_sale\_info csi ON csa.csa\_id = csi.csa\_id

LEFT OUTER JOIN

    ieaown.ie\_field\_code bm\_desc ON bm\_desc.cd\_id = csi.benchmark\_rate\_cd

WHERE

    csa.csa\_id = **P\_CSA\_ID**

    AND csi.doc\_prefix = chg.doc\_prefix

    AND csi.doc\_series = chg.doc\_series;

### IE\_GET\_CSA\_ADJUST\_AMOUNT

SELECT

    SUM(amount\_csa) AS tot\_adj\_csa\_amt

FROM

    ieaown.ie\_csa\_amount

WHERE

    amount\_type = 'A'

    AND amount\_cd IN (20, 22, 23)

    AND csa\_id = **P\_CSA\_ID**;

### IE\_GET\_CSA\_PCT\_OCCUP\_THRESHLD

SELECT

    pct\_occup\_threshld

FROM

    IEAOWN.IE\_GLOBAL\_CONTROL;

### IE\_GET\_CSA\_EXPENSE\_TOTAL(P\_CSA\_ID)

SELECT

    SUM(amount\_owners) AS tot\_expense\_owner,

    SUM(amount\_tenant) AS tot\_expense\_tenant,

    SUM(IFNULL(amount\_owners, 0) + IFNULL(amount\_tenant, 0)) AS tot\_expenses

FROM

    ieaown.ie\_csa\_amount

WHERE

    amount\_type = 'E'

    AND amount\_cd <= 14

    AND csa\_id = **P\_CSA\_ID**;

### IE\_GET\_CSA\_POTENTIAL\_INCOME(P\_CSA\_ID)

SELECT

    SUM(current\_rent) AS tot\_current\_rent,

    SUM(anticipated\_rent) AS tot\_anti\_rent

FROM

    ieaown.ie\_csa\_lease

WHERE

    csa\_id = **P\_CSA\_ID**;

### IE\_GET\_CSA\_ALL\_WORKER

SELECT

    worker\_id,

    (last\_name || COALESCE(', ' || first\_name, '') || COALESCE(' ' || middle\_name, '')) AS worker\_name

FROM

    ieaown.ie\_worker

ORDER BY

    worker\_name;

### IE\_GET\_CSA\_PROP\_INFO\_VACANT\_LAND(P\_CSA\_ID)

SELECT

    IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID,

    IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX,

    IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES,

    IEAOWN.IE\_CSA\_SALE\_INFO.PROPERTY\_ID,

    IEAOWN.IE\_CSA\_SALE\_INFO.USE\_CD,

    IEAOWN.IE\_CSA\_SALE\_INFO.TOT\_LOT\_SIZE,

    IEAOWN.IE\_CSA\_SALE\_INFO.PCT\_OWNER\_OCCUP,

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    STRIP(

        CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG, '') = '' THEN ''

            ELSE ' ' || STRIP(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG)

        END

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM, '') = '' THEN ''

            ELSE ' ' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM

        END

    ) AS address,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_CITY\_NAME,

    'CA' AS SITUS\_STATE,

    STRIP(

        IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_CD

        || CASE

            WHEN IFNULL(IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4, '') = '' THEN ''

            ELSE '-' || IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4

        END

    ) AS ZIP\_CD,

    IEAOWN.IE\_DOCUMENT.DOC\_PARCEL\_CNT,

    IEAOWN.IE\_CHG\_OWNER.EVENT\_TS,

    IEAOWN.IE\_CHG\_OWNER.IND\_PUR\_PRICE,

    IEAOWN.IE\_CHG\_OWNER.ADJ\_SALES\_PRICE,

    IFNULL(IEAOWN.IE\_CHG\_OWNER.XFER\_TAX\_PRICE\_CO, IEAOWN.IE\_CHG\_OWNER.XFER\_TAX\_PRICE\_CTY) AS TRAN\_TAX\_PRICE,

    UPPER(IEAOWN.IE\_USE\_CODE.USE\_NAME\_SHORT) AS USE\_NAME,

    IEAOWN.IE\_CSA\_SALE\_INFO.TOT\_NET\_RENT\_AREA,

    IEAOWN.IE\_CSA\_SALE\_INFO.TOT\_BUILDING\_AREA

FROM

    IEAOWN.IE\_CSA\_SALE\_INFO

LEFT OUTER JOIN ieaown.ie\_use\_code ON IEAOWN.IE\_CSA\_SALE\_INFO.use\_cd = ieaown.ie\_use\_code.use\_cd,

IEAOWN.IE\_REAL\_PROPERTY,

IEAOWN.IE\_DOCUMENT,

IEAOWN.IE\_CHG\_OWNER

WHERE

    IEAOWN.IE\_CSA\_SALE\_INFO.PROPERTY\_ID = IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID

    AND (IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX = IEAOWN.IE\_DOCUMENT.DOC\_PREFIX)

    AND (IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES = IEAOWN.IE\_DOCUMENT.DOC\_SERIES)

    AND (IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX = IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX)

    AND (IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES = IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES)

    AND (IEAOWN.IE\_CSA\_SALE\_INFO.USE\_CD = IEAOWN.IE\_USE\_CODE.USE\_CD)

    AND IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = **P\_CSA\_ID**;

### IE\_GET\_CSA\_LEASE(P\_LEASE\_ID)

SELECT

    IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_ID,

    IEAOWN.IE\_CSA\_LEASE.PROPERTY\_ID,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_LOCATION,

    IEAOWN.IE\_CSA\_LEASE.SITUS\_CITY\_NAME,

    IEAOWN.IE\_CSA\_LEASE.LESSEE\_NAME,

    IEAOWN.IE\_CSA\_LEASE.VACANT\_FL,

    IEAOWN.IE\_CSA\_LEASE.NET\_RENTABLE\_AREA,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_START\_DT,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_TERM,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_OPTION,

    IEAOWN.IE\_CSA\_LEASE.INITIAL\_RENT,

    IEAOWN.IE\_CSA\_LEASE.INITIAL\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.STABIL\_RENT,

    IEAOWN.IE\_CSA\_LEASE.STABIL\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.ANTICIPATED\_RENT,

    IEAOWN.IE\_CSA\_LEASE.USE\_FOR\_COMP\_FL,

    IEAOWN.IE\_CSA\_LEASE.ANTI\_RENT\_PSF,

    IEAOWN.IE\_CSA\_LEASE.RENT\_ADJUSTMENTS,

    IEAOWN.IE\_CSA\_LEASE.TENANT\_IMPS\_PSF,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_SUBCATEGORY,

    SUBCAT.SUBCATEGORY\_NAME AS SUBCATEGORY\_NAME,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_TYPE\_CD,

    FCD1.CD\_LONG\_NAME AS LEASE\_TYPE\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.EXPENSE\_TYPE\_CD,

    FCD2.CD\_LONG\_NAME AS EXPENSE\_TYPE\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.TENANCY\_CD,

    FCD3.CD\_LONG\_NAME AS TENANCY\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.INFO\_SOURCE\_CD,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_PARKING,

    IEAOWN.IE\_CSA\_LEASE.SPRINKLER\_FL,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_CREATED\_BY,

    (WK1.LAST\_NAME || CASE IFNULL(WK1.FIRST\_NAME,'')

         WHEN '' THEN '' ELSE ', ' ||WK1.FIRST\_NAME END

         || CASE IFNULL(WK1.MIDDLE\_NAME,'')

         WHEN '' THEN '' ELSE ' ' ||WK1.MIDDLE\_NAME END) AS  LEASE\_CREATED\_NAME,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_CREATE\_DT,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_MODIFY\_BY,

    (WK2.LAST\_NAME || CASE IFNULL(WK2.FIRST\_NAME,'')

         WHEN '' THEN '' ELSE ', ' ||WK2.FIRST\_NAME END

         || CASE IFNULL(WK2.MIDDLE\_NAME,'')

         WHEN '' THEN '' ELSE ' ' ||WK2.MIDDLE\_NAME END) AS  LEASE\_MODIFY\_NAME,

    IEAOWN.IE\_CSA\_LEASE.LEASE\_MODIFY\_DT,

    IEAOWN.IE\_CSA\_LEASE.ENTRY\_TS,

    IEAOWN.IE\_CSA\_LEASE.ENTRY\_WORKER,

    IEAOWN.IE\_CSA\_LEASE.UPDATE\_TS,

    IEAOWN.IE\_CSA\_LEASE.UPDATE\_WORKER,

    IEAOWN.IE\_CSA\_LEASE.CSA\_ID,

    IEAOWN.IE\_CSA\_LEASE.USE\_CATEGORY\_CD,

    USE\_CAT.USE\_NAME AS LEASE\_USE\_NAME,

    IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT,

    IEAOWN.IE\_CSA\_LEASE.CURRENT\_RENT\_PSF,

    RTRIM(IEAOWN.IE\_CSA\_LEASE.RENT\_ADJ\_TEXT) AS RENT\_ADJ\_TEXT,

    IEAOWN.IE\_CSA\_LEASE.EXPENSE\_BASE\_YR,

    IEAOWN.IE\_CSA\_LEASE.TI\_ALLOWANCE,

    IEAOWN.IE\_CSA\_LEASE.TI\_TYPE\_CD,

    FCD4.CD\_LONG\_NAME AS TI\_TYPE\_TEXT,

    RTRIM(IEAOWN.IE\_CSA\_LEASE.LEASE\_PARKING\_2) AS LEASE\_PARKING\_2,

    RTRIM(IEAOWN.IE\_CSA\_LEASE.INFO\_SOURCE\_TEXT) AS INFO\_SOURCE\_TEXT,

    COALESCE(IEAOWN.IE\_CSA\_LEASE.SUPRV\_APPROVED\_FL,'N') AS SUPRV\_APPROVED\_FL,

    IEAOWN.IE\_CSA\_LEASE.SUPRV\_APPR\_UPD\_WKR,

    (WK3.LAST\_NAME || CASE IFNULL(WK3.FIRST\_NAME,'')

         WHEN '' THEN '' ELSE ', ' ||WK3.FIRST\_NAME END

         || CASE IFNULL(WK3.MIDDLE\_NAME,'')

         WHEN '' THEN '' ELSE ' ' ||WK3.MIDDLE\_NAME END) AS  SUPRV\_APPR\_WKR\_NAME,

    IEAOWN.IE\_CSA\_LEASE.SUPRV\_APPR\_UPD\_TS,

    IEAOWN.IE\_CSA\_LEASE\_DET.LEASE\_ID AS DetLeaseId,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_CLR\_HGT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_OFFC\_AREA\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_LAB\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_MFG\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_WAREHOUSE\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_RETAIL\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.IND\_OTHER\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.OFFC\_LOAD\_FACTOR,

    IEAOWN.IE\_CSA\_LEASE\_DET.OFFC\_EXPENSE\_STOP,

    IEAOWN.IE\_CSA\_LEASE\_DET.RETAIL\_DBA,

    IEAOWN.IE\_CSA\_LEASE\_DET.RETAIL\_RENT\_PCT,

    IEAOWN.IE\_CSA\_LEASE\_DET.RENT\_BREAKPOINT,

    IEAOWN.IE\_CSA\_LEASE\_DET.RETAIL\_STORE\_OCC

FROM

    IEAOWN.IE\_CSA\_LEASE lse

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_LEASE\_DET det ON lse.LEASE\_ID = det.LEASE\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_LEASE\_USE\_CAT USE\_CAT ON lse.USE\_CATEGORY\_CD = USE\_CAT.USE\_CATEGORY\_CD

LEFT OUTER JOIN

    IEAOWN.IE\_LEASE\_SUBCAT SUBCAT ON lse.LEASE\_SUBCATEGORY = SUBCAT.LEASE\_SUBCATEGORY

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD1 ON lse.LEASE\_TYPE\_CD = FCD1.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD2 ON lse.EXPENSE\_TYPE\_CD = FCD2.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD3 ON lse.TENANCY\_CD = FCD3.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_FIELD\_CODE FCD4 ON lse.TI\_TYPE\_CD = FCD4.CD\_ID

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK1 ON WK1.WORKER\_ID=lse.LEASE\_CREATED\_BY

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK2 ON WK2.WORKER\_ID=lse.LEASE\_MODIFY\_BY

LEFT OUTER JOIN

    IEAOWN.IE\_WORKER WK3 ON WK3.WORKER\_ID=lse.SUPRV\_APPR\_UPD\_WKR

LEFT OUTER JOIN

    IEAOWN.IE\_REAL\_PROPERTY ON IEAOWN.IE\_CSA\_LEASE.PROPERTY\_ID = IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID

WHERE

    IEAOWN.IE\_CSA\_LEASE.LEASE\_ID = **P\_LEASE\_ID**

### IE\_GET\_CSA\_LEASE\_COMMENTS(P\_LEASE\_ID)

SELECT

    IEAOWN.IE\_CSA\_LEASE\_CMNT.LEASE\_ID,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.SEQ\_NUM,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.COMMENT\_TEXT,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.ENTRY\_TS,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.ENTRY\_WORKER,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.UPDATE\_TS,

    IEAOWN.IE\_CSA\_LEASE\_CMNT.UPDATE\_WORKER

FROM

    IEAOWN.IE\_CSA\_LEASE\_CMNT

WHERE

    IEAOWN.IE\_CSA\_LEASE\_CMNT.LEASE\_ID = **P\_LEASE\_ID**

### IE\_GET\_CSA\_LEASE\_USE\_CATEGORY()

SELECT

    IEAOWN.IE\_LEASE\_USE\_CAT.USE\_CATEGORY\_CD,

    RTRIM(LTRIM(IEAOWN.IE\_LEASE\_USE\_CAT.USE\_NAME)) AS USE\_NAME,

    END\_DT

FROM

    IEAOWN.IE\_LEASE\_USE\_CAT

ORDER BY

    IEAOWN.IE\_LEASE\_USE\_CAT.USE\_NAME

### IE\_GET\_CSA\_LEASE\_USE\_SUB\_BY\_USE\_CAT(P\_USE\_CAT)

SELECT

    IEAOWN.IE\_LEASE\_SUBCAT.LEASE\_SUBCATEGORY,

    LTRIM(RTRIM(IEAOWN.IE\_LEASE\_SUBCAT.SUBCATEGORY\_NAME)) AS SUBCATEGORY\_NAME,

    END\_DT

FROM

    IEAOWN.IE\_LEASE\_SUBCAT

WHERE

    IEAOWN.IE\_LEASE\_SUBCAT.USE\_CATEGORY\_CD = **P\_USE\_CAT**

ORDER BY

    IEAOWN.IE\_LEASE\_SUBCAT.SUBCATEGORY\_NAME

### IE\_GET\_PC\_BUILDING\_AS\_OF\_DATE(int[] PropId, DateTime AsofDate)

SELECT

    IEAOWN.IE\_BUILDING.PROPERTY\_ID,

    IEAOWN.IE\_BUILDING.BUILDING\_NUM,

    IEAOWN.IE\_BUILDING.EFFECTIVE\_DT,

    IEAOWN.IE\_BUILDING.BUILT\_YR,

    IEAOWN.IE\_BUILDING.EFFECTIVE\_YR,

    IEAOWN.IE\_BUILDING.NUM\_STORIES,

    IEAOWN.IE\_BUILDING.NUM\_BEDROOMS,

    IEAOWN.IE\_BUILDING.NUM\_BATHS\_FULL,

    IEAOWN.IE\_BUILDING.NUM\_BATHS\_HALF,

    IEAOWN.IE\_BUILDING.NUM\_ROOMS,

    IEAOWN.IE\_BUILDING.NUM\_UNITS,

    IEAOWN.IE\_BUILDING.COND\_CD,

    IEAOWN.IE\_BUILDING.COND\_YR,

    IEAOWN.IE\_BUILDING.BUILDING\_AREA,

    IEAOWN.IE\_BUILDING.ADDITIONS\_AREA,

    IEAOWN.IE\_BUILDING.MISC\_AREA,

    IEAOWN.IE\_BUILDING.NET\_RENTABLE\_AREA,

    IEAOWN.IE\_BUILDING.PCT\_OFFICE,

    IEAOWN.IE\_BUILDING.AVG\_ACTUAL\_HT,

    IEAOWN.IE\_BUILDING.SBE\_CONST\_TYPE\_CD,

    IEAOWN.IE\_BUILDING.SBE\_QUALITY,

    IEAOWN.IE\_BUILDING.SBE\_SHAPE\_CD,

    IEAOWN.IE\_BUILDING.CONDO\_TYPE\_CD,

    IEAOWN.IE\_BUILDING.CONDO\_UNIT\_FLOOR,

    IEAOWN.IE\_BUILDING.CONFORMITY\_CD,

    IEAOWN.IE\_BUILDING.REMODEL\_CD,

    IEAOWN.IE\_BUILDING.REMODEL\_YR,

    IEAOWN.IE\_BUILDING.FIRE\_SPRINKLER\_FL,

    IEAOWN.IE\_BUILDING.NUM\_ELEVATOR,

    IEAOWN.IE\_BUILDING.END\_DT,

    IEAOWN.IE\_BUILDING.NUM\_KITCHENS,

    IEAOWN.IE\_BUILDING.ADDITION\_YR,

    IEAOWN.IE\_BUILDING.FINISHED\_AREA,

    IEAOWN.IE\_BUILDING.BASEMENT\_AREA,

    IEAOWN.IE\_BUILDING.AVG\_UNIT\_AREA,

    IEAOWN.IE\_BUILDING.PCT\_BUILT\_OUT,

    IEAOWN.IE\_BUILDING.AVG\_CLEAR\_HT,

    IEAOWN.IE\_BUILDING.MVS\_CONST\_TYPE\_CD,

    IEAOWN.IE\_BUILDING.MVS\_PERIMETER,

    IEAOWN.IE\_BUILDING.MVS\_QUALITY,

    IEAOWN.IE\_BUILDING.MVS\_SHAPE\_CD,

    IEAOWN.IE\_BUILDING.SEISMIC\_RETRO\_FL,

    IEAOWN.IE\_BUILDING.SEISMIC\_RETRO\_YR,

    IEAOWN.IE\_BUILDING.OCC\_CD,

    IEAOWN.IE\_BUILDING.DEMOLISHED\_FL,

    IEAOWN.IE\_BUILDING.BUILDING\_LABEL,

    F1.CD\_SHORT\_NAME AS SBE\_CONST\_TYPE,

    F2.CD\_SHORT\_NAME AS SBE\_SHAPE,

    F3.CD\_SHORT\_NAME AS MVS\_CONST\_TYPE,

    F4.CD\_SHORT\_NAME AS MVS\_SHAPE,

    CONCAT(

        IFNULL(F1.CD\_SHORT\_NAME, ''),

        CASE

            WHEN SBE\_QUALITY IS NULL THEN ''

            WHEN SBE\_QUALITY < 10 THEN RIGHT(CHAR(SBE\_QUALITY), 3)

            ELSE LTRIM(CHAR(SBE\_QUALITY))

        END,

        IFNULL(F2.CD\_SHORT\_NAME, '')

    ) AS SBE\_CODE,

    CONCAT(

        IFNULL(F3.CD\_SHORT\_NAME, ''),

        CASE

            WHEN MVS\_QUALITY IS NULL THEN ''

            WHEN MVS\_QUALITY < 10 THEN RIGHT(CHAR(MVS\_QUALITY), 3)

            ELSE LTRIM(CHAR(MVS\_QUALITY))

        END,

        IFNULL(F4.CD\_SHORT\_NAME, '')

    ) AS MVS\_CODE,

    'Y' AS SELECT\_FL

FROM IEAOWN.IE\_BUILDING

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE F1 ON IEAOWN.IE\_BUILDING.SBE\_CONST\_TYPE\_CD = F1.CD\_ID

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE F2 ON IEAOWN.IE\_BUILDING.SBE\_SHAPE\_CD = F2.CD\_ID

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE F3 ON IEAOWN.IE\_BUILDING.MVS\_CONST\_TYPE\_CD = F3.CD\_ID

LEFT OUTER JOIN IEAOWN.IE\_FIELD\_CODE F4 ON IEAOWN.IE\_BUILDING.MVS\_SHAPE\_CD = F4.CD\_ID

WHERE IEAOWN.IE\_BUILDING.PROPERTY\_ID IN (@prop\_id\_array)

AND (

    IEAOWN.IE\_BUILDING.EFFECTIVE\_DT <= @as\_of\_date

    AND (IEAOWN.IE\_BUILDING.END\_DT >= @as\_of\_date OR IEAOWN.IE\_BUILDING.END\_DT IS NULL)

    AND (IEAOWN.IE\_BUILDING.DEMOLISHED\_FL <> 'Y' OR IEAOWN.IE\_BUILDING.DEMOLISHED\_FL IS NULL)

)

ORDER BY

    IEAOWN.IE\_BUILDING.PROPERTY\_ID,

    IEAOWN.IE\_BUILDING.BUILDING\_NUM,

    IEAOWN.IE\_BUILDING.EFFECTIVE\_DT DESC

### IE\_GET\_PCE\_AS\_OF\_DT

SELECT

    IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID,

    IEAOWN.IE\_CHG\_OWNER.EVENT\_TS,

    IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL,

    IEAOWN.IE\_REAL\_PROPERTY.NEIGH\_CD,

    IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID,

    IEAOWN.IE\_DOCUMENT.DOC\_PARCEL\_CNT,

    IEAOWN.IE\_DOCUMENT.DOC\_PREFIX,

    IEAOWN.IE\_DOCUMENT.DOC\_SERIES,

    IEAOWN.IE\_PARCEL\_DET.SEQ\_NUM,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.USE\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.SBE\_CONST\_TYPE,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.SBE\_QUALITY,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.SBE\_SHAPE,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_BUILDING\_AREA,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.BUILT\_YR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.EFFECTIVE\_YR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_ROOMS,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_BEDROOMS,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_BATHS\_FULL,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_BATHS\_HALF,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE\_CD,

    CASE IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE\_CD

        WHEN 62 THEN IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE \* 43560

        ELSE IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE

    END AS LOT\_SIZE\_SQ\_FT,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.COND\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.REMODEL\_FL,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_NUM\_UNITS,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.NUM\_BUILDINGS,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.VIEW\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.SLOPE\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.LAND\_IMP\_RATIO,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.AMENITIES\_FL,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.HAZARD\_FL,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_PARKING,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.POOL\_FL,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.NUM\_STORIES,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.BUILDING\_NUM,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.CONDO\_UNIT\_FLOOR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.PARKING\_TYPE\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.COND\_YR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.REMODEL\_YR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.POOL\_TYPE\_CD,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.POOL\_BUILT\_YR,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.LAND\_PARKING,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.MVS\_CONST\_TYPE,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.MVS\_QUALITY,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.MVS\_SHAPE,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.FLOOR\_AREA\_RATIO,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NUM,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_PRE\_DIRECT,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_NAME,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_STREET\_SFX,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_POST\_DIRECT,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_DESIG,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_UNIT\_NUM,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_CITY\_NAME,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_CD,

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_ZIP\_PLUS\_4,

    IEAOWN.IE\_REAL\_PROPERTY.APN\_BOOK,

    IEAOWN.IE\_REAL\_PROPERTY.APN\_BLOCK,

    IEAOWN.IE\_REAL\_PROPERTY.SORT\_PARCEL

FROM

    IEAOWN.IE\_CHG\_OWNER,

    IEAOWN.IE\_DOCUMENT,

    IEAOWN.IE\_PARCEL\_DET,

    IEAOWN.IE\_PROP\_CHAR\_EVENT,

    IEAOWN.IE\_REAL\_PROPERTY,

    IEAOWN.IE\_CSA\_SALE\_INFO

WHERE

    IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES

    AND IEAOWN.IE\_CHG\_OWNER.DOC\_PREFIX = IEAOWN.IE\_PARCEL\_DET.DOC\_PREFIX

    AND IEAOWN.IE\_CHG\_OWNER.DOC\_SERIES = IEAOWN.IE\_PARCEL\_DET.DOC\_SERIES

    AND IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL = IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL

    AND IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = IEAOWN.IE\_PROP\_CHAR\_EVENT.PROPERTY\_ID

    AND IEAOWN.IE\_PROP\_CHAR\_EVENT.EFFECTIVE\_DT = (

        SELECT MAX(B.EFFECTIVE\_DT)

        FROM IEAOWN.IE\_PROP\_CHAR\_EVENT B

        WHERE B.EFFECTIVE\_DT <= DATE(IEAOWN.IE\_CHG\_OWNER.EVENT\_TS)

        AND B.PROPERTY\_ID = IEAOWN.IE\_PROP\_CHAR\_EVENT.PROPERTY\_ID

    )

    AND IEAOWN.IE\_DOCUMENT.DOC\_PREFIX = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_PREFIX

    AND IEAOWN.IE\_DOCUMENT.DOC\_SERIES = IEAOWN.IE\_CSA\_SALE\_INFO.DOC\_SERIES

    AND IEAOWN.IE\_CSA\_SALE\_INFO.CSA\_ID = @csa\_id

### IE\_GET\_USE\_CD\_AS\_OF\_DATE(P\_PROP\_ID, P\_AS\_OF\_DATE)

SELECT

    IEAOWN.IE\_PROP\_CHAR\_EVENT.USE\_CD

FROM

    IEAOWN.IE\_PROP\_CHAR\_EVENT

WHERE

    IEAOWN.IE\_PROP\_CHAR\_EVENT.PROPERTY\_ID = **P\_PROP\_ID**

    AND IEAOWN.IE\_PROP\_CHAR\_EVENT.EFFECTIVE\_DT = (

        SELECT MAX(B.EFFECTIVE\_DT)

        FROM IEAOWN.IE\_PROP\_CHAR\_EVENT B

        WHERE B.EFFECTIVE\_DT <= **P\_AS\_OF\_DATE**

        AND B.PROPERTY\_ID = IEAOWN.IE\_PROP\_CHAR\_EVENT.PROPERTY\_ID

    )

### IE\_GET\_SITUS\_CITY(P\_PROP\_ID)

SELECT

    IEAOWN.IE\_REAL\_PROPERTY.SITUS\_CITY\_NAME

FROM

    IEAOWN.IE\_REAL\_PROPERTY

WHERE

    IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = **P\_PROP\_ID**

### IE\_GET\_VL\_PROP\_CHAR\_AS\_OF\_DATE(P\_PROPERTY\_ID, P\_AS\_OF\_DATE)

SELECT

    IEAOWN.IE\_LAND.VIEW\_CD,

    IEAOWN.IE\_LAND.SLOPE\_CD,

    IEAOWN.IE\_LAND.WELL\_FL,

    IEAOWN.IE\_LAND.SEPTIC\_FL

FROM

    IEAOWN.IE\_LAND

WHERE

    IEAOWN.IE\_LAND.PROPERTY\_ID = **P\_PROPERTY\_ID**

    AND IEAOWN.IE\_LAND.EFFECTIVE\_DT = (

        SELECT MAX(B.EFFECTIVE\_DT)

        FROM IEAOWN.IE\_LAND B

        WHERE B.EFFECTIVE\_DT <= **P\_AS\_OF\_DATE**

            AND B.PROPERTY\_ID = IEAOWN.IE\_LAND.PROPERTY\_ID)

### IE\_DEL\_CSA\_WKS\_APN(P\_CSA\_WKS\_NUM)

DELETE FROM ieaown.ie\_csa\_wks\_apn

WHERE csa\_wks\_num = **P\_CSA\_WKS\_NUM**;

### IE\_GET\_CSA\_WKS\_APN\_LIST(P\_CSA\_WKS\_NUM)

WITH MaxEffectiveDate AS (

    SELECT

        PROPERTY\_ID,

        MAX(EFFECTIVE\_DT) AS MAX\_EFFECTIVE\_DT

    FROM

        IEAOWN.IE\_PROP\_CHAR\_EVENT

    WHERE

        CHANGE\_APPROVED\_FL = 'Y'

    GROUP BY

        PROPERTY\_ID

)

SELECT

    IEAOWN.IE\_PARCEL\_DET.DOC\_PREFIX,

    IEAOWN.IE\_PARCEL\_DET.DOC\_SERIES,

    IEAOWN.IE\_PARCEL\_DET.SEQ\_NUM,

    IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL,

    IEAOWN.IE\_REAL\_PROPERTY.SORT\_PARCEL,

    IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID,

    IEAOWN.IE\_CSA\_WKS\_APN.PRIMARY\_APN\_FL,

    CASE

        WHEN IEAOWN.IE\_CSA\_WKS\_APN.PROPERTY\_ID IS NULL THEN 'N'

        ELSE 'Y'

    END AS CSA\_SUBJECT,

    IEAOWN.IE\_USE\_CODE.USE\_NAME\_SHORT,

    IEAOWN.IE\_PROP\_CHAR\_EVENT.TOT\_BUILDING\_AREA,

    CASE

        WHEN IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE\_CD = 62 THEN IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE \* 43560

        WHEN IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE\_CD = 63 THEN IEAOWN.IE\_PROP\_CHAR\_EVENT.LOT\_SIZE

    END AS TOT\_LOT\_SIZE,

    IEAOWN.IE\_CSA\_WKS.CSA\_WKS\_NUM

FROM

    IEAOWN.IE\_PARCEL\_DET

INNER JOIN

    IEAOWN.IE\_CSA\_WKS ON IEAOWN.IE\_PARCEL\_DET.DOC\_PREFIX = IEAOWN.IE\_CSA\_WKS.DOC\_PREFIX

                        AND IEAOWN.IE\_PARCEL\_DET.DOC\_SERIES = IEAOWN.IE\_CSA\_WKS.DOC\_SERIES

LEFT OUTER JOIN

    IEAOWN.IE\_REAL\_PROPERTY ON IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL = IEAOWN.IE\_REAL\_PROPERTY.PRINT\_PARCEL

LEFT OUTER JOIN

    IEAOWN.IE\_CSA\_WKS\_APN ON IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID = IEAOWN.IE\_CSA\_WKS\_APN.PROPERTY\_ID

                           AND IEAOWN.IE\_CSA\_WKS\_APN.CSA\_WKS\_NUM = **P\_CSA\_WKS\_NUM**

INNER JOIN

    IEAOWN.IE\_PROP\_CHAR\_EVENT ON IEAOWN.IE\_PROP\_CHAR\_EVENT.USE\_CD = IEAOWN.IE\_USE\_CODE.USE\_CD

                               AND IEAOWN.IE\_PROP\_CHAR\_EVENT.PROPERTY\_ID = IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID

                               AND IEAOWN.IE\_PROP\_CHAR\_EVENT.EFFECTIVE\_DT = MaxEffectiveDate.MAX\_EFFECTIVE\_DT

                               AND MaxEffectiveDate.PROPERTY\_ID = IEAOWN.IE\_REAL\_PROPERTY.PROPERTY\_ID

WHERE

    NOT (IEAOWN.IE\_PARCEL\_DET.PRINT\_PARCEL LIKE '0-%')

    AND IEAOWN.IE\_PARCEL\_DET.END\_DT IS NULL

    AND IEAOWN.IE\_CSA\_WKS.CSA\_WKS\_NUM = **P\_CSA\_WKS\_NUM**;

## Helper Class

Helper class files should be included in the DAL folder.

### DB2HelperCustom Class

#### ExecuteReader

This is used to return corresponding data table with stored procedure, then use functions of DB2Helpers Class to assign to correct entity

#### ExecuteNonQuery

Remember to assign the update variable DBNull in order for it to be updated in database to DBNull, if variable value is null, it will be skipped from update query

### DB2Helpers Class

#### GetEntitiesFromReader

This is used to convert DB2DataReader object into any entity object selected. Directly apply below code without changing anything. As for how to obtain the DB2DataReader object, check above

\*\*\* Note that if a value type in entity is a string and DB2DataReader return an empty value. This string will be assigned an **empty string.**

#### GetEntitieFromReader

Return a single row as result and assign it to corresponding entity. DBNull will be assigned as null if variable is not a string, otherwise will be assigned as an empty string.

#### GetStringFromReader

This is used to return a single row single column database return and convert to a string

### Helpers Class

#### GetBaseYear

public static int GetBaseYear(DateTime eventDate)

{

    int baseYear;

    if (eventDate <= DateTime.Parse("1975-02-28"))

    {

        // on or before Feb 28, 1975

        baseYear = 1975;

    }

    else if (eventDate >= DateTime.Parse("1975-03-01") && eventDate <= DateTime.Parse("1983-02-28"))

    {

        // between March 1, 1975 and Feb 28, 1983

        if (eventDate.Month <= 2)

        {

            baseYear = eventDate.Year;

        }

        else

        {

            baseYear = eventDate.Year + 1;

        }

    }

    else if (eventDate >= DateTime.Parse("1983-03-01") && eventDate <= DateTime.Parse("1984-06-30"))

    {

        // between March 1, 1983 and June 30, 1984

        baseYear = 1984;

    }

    else

    {

        // after June 30, 1984

        if (eventDate.Month <= 6)

        {

            baseYear = eventDate.Year;

        }

        else

        {

            baseYear = eventDate.Year + 1;

        }

    }

    return baseYear;

}

## DAL Layer:

Create these functions, each corresponding to one stored procedure and one only unless circumstances require use of Transaction.

### CSASale Class

#### GetPageTitleByCSAType

public Task<string> GetPageTitleByCSAType(int p\_csa\_type)

IE\_GET\_FIELD\_CODE

CSA\_TYPE is relating to IE\_GET\_FIELD\_CODE.CD\_ID, do filter to receive result

Use LINQ to apply the extra filter, queryResult.Where(x => x.CD\_ID == CSA\_TYPE)

#### GetSaleInfoFull

public Task<List<SaleInfo>> GetFullSaleInfo(int p\_csa\_Id)

IE\_GET\_CSA\_SALE\_INFO\_FULL

#### UpdateSaleInfoWithCISaleAnalysis

public Task<int> UpdateWithCISaleAnalysis(UpdatedSaleInfoWithCIAnalysis p\_updated\_info)

//This is a transaction, make sure all below procedures can be fulfilled or rollback progress.

If PCT\_OWNER\_OCCUP.HasValue() in p\_updated\_info

{

IE\_DEL\_CSA\_LEASE

Update IE\_CSA\_SALE\_INFO using UpdatedSaleInfo entity

Update IE\_CSA\_AMOUNT table,

SET AMOUNT\_OWNERS = NULL,

    AMOUNT\_TENANT = NULL

WHERE CSA\_ID = **P\_CSA\_ID** AND AMOUNT\_TYPE = 'E';

Update IE\_CSA ANALYSIS table,

SET PCT\_PGI = NULL,

    ADDITION\_NET\_INCOM = NULL,

    TENANT\_REIMBURSE = NULL,

    ANTI\_PROPERTY\_TAX = NULL,

    OTHER\_TAXES\_DESC = NULL,

    OTHER\_TAXES\_AMOUNT = NULL,

    UPDATE\_TS = CURRENT TIMESTAMP, then use ModifiedBy entity

}

IE\_ MOD\_CSA \_SALE\_INFO

#### UpdateFullExpensesFl

public Task<int> UpdateFullExpensesFl( \_CSASaleExpenses.CSA\_Id, CSASaleExpenses.Full\_Expenses\_Fl, Worker\_Id)

UPDATE IE\_CSA\_SALE\_INFO,

SET FULL\_EXPENSES\_FL, UPDATE\_TS, UPDATE\_WORKER

WHERE CSA\_ID match

USING: Incoming parameters

#### UpdateExpenseByCSAIdAmountCd

public Task<int> UpdateExpenseByCSAIdAmountCd( int p\_CSA\_Id, int p\_Amount\_Cd, string p\_Update\_Desc\_Fl, string p\_Amount\_Description, Decimal? p\_Amount\_Owners, Decimal? p\_Amount\_Tenant, string p\_lastModifyPage, string p\_Worker\_Id)

UPDATE IEAOWN.IE\_CSA\_AMOUNT

SET

  AMOUNT\_CSA,

  UPDATE\_TS

  UPDATE\_WORKER

WHERE

CSA\_ID and AMOUNT\_CD matches

public List<CSAAdjustment> GetAdjustmentsForAllAdjustmentCodes(int p\_csa\_id)

#### CSA\_AMOUNT\_ModAdjustmentByCSAIdAmountCd

public int CSA\_AMOUNT\_ModAdjustmentByCSAIdAmountCd(

    int p\_CSA\_Id,

    int p\_Amount\_Cd,

    string p\_Update\_Desc\_Fl,

    string p\_Amount\_Description,

    decimal? p\_Amount\_CSA,

    string p\_LastModifyPage,

    string p\_Worker\_Id)

CSAAmount \_amount = IE\_CSA\_GET\_AMOUNT

// Here MAKE SURE the sp returned data row is exactly 1, 0 or more than 1 should return -1 directly.

Then following validations are required.

if (cmc\_amount\_type == null || cmc\_amount\_type != “A”)

Throw New ApplicationException("Invalid Amount Type for Adjustment.")

if (\_amount.CAM\_AMT\_CD == null)

return 0; //since no csa amount row is found, do nothing

Otherwise make update with following sp.

UPDATE IE\_CSA\_AMOUNT

SET AMOUNT\_CSA, UPDATE\_TS, UPDATE\_WORKER

WITH UpdatedCSAAmount entity

WHERE CSA\_ID and AMOUNT\_CD match

#### GetPropChar

public Task<PropChar>GetPropChar (int p\_csa\_id)

IE\_GET\_CSA\_PROP\_CHAR

#### GetIncomeExpenseAnalysis

public Task<IncomeExpenseAnalysis>GetIncomeExpenseAnalysis (int p\_csa\_id)

IE\_GET\_CSA\_INCOME\_EXPENSE\_ANALYSIS

#### GetExpensesForAllExpenseCodes

public Task<List<CSAExpense>> GetExpensesForAllExpenseCodes (int p\_csa\_id)

IE\_GET\_CSA\_EXPENSE\_CODE

Note use CSAExpenseCode entity to store the result, but make sure to transform them into a list of CSAExpense model to return to the BLL layer function. Also remember to call Validate function to make sure CSAExpense is ready to be used.

#### GetActualExpenses

public Task<List<CSAExpense>> GetActualExpenses(int p\_csa\_id)

IE\_GET\_CSA\_AMOUNT\_ACTUAL\_EXPENSE

Note use CSAExpenseCode entity to store the result, but make sure to transform them into a list of CSAExpense model to return to the BLL layer function. Also remember to call Validate function to make sure CSAExpense is ready to be used.

#### GetCSAExpense

public Task<List<CSAExpenseEntity>> GetCSAExpense (int p\_csa\_id)

IE\_GET\_CSA\_EXPENSE

#### GetCSARptSource

public Task<RptSource> GetCSARptSource (int p\_csa\_id)

IE\_GET\_CSA\_RPT\_SOURCE

#### GetCSARptAdjustments

public Task<CSARptAdjustments> GetCSARptAdjustments (int p\_csa\_id)

IE\_GET\_CSA\_RPT\_ADJ

#### GetAdjustmentsForAllAdjustmentCodes

Public Task< CSAAdjustment> GetAdjustmentsForAllAdjustmentCodes (int p\_csa\_id)

IE\_GET\_CSA\_ADJ\_FOR\_ALL\_ADJ\_CODES

Above entity will return a list of AdjustmentsForAllAdjustmentCodes entity. Iterate through the list and translate each entity into a CSAAdjustment model member variable. Once done return a list of CSAAdjustment model back.

### CSAProperty Class

#### RefreshPropChar

This function below is implemented within Improve that achieved the same result, the part used to initialize CSA is been taken out instead updating existing CSA.

public Task<bool> RefreshPropChar(string DOC\_PREFIX, int DOC\_SERIES, int PRIMARY\_PROP\_ID, short? CsaSelectType, ref int CsaId, string WorkerId, string UserId, DB2Transaction P\_transaction = null)

{

    bool retval;

    // This method creates the CSA rows for the key references passed

    string P\_ID\_string, ZoningCd;

    List<int> P\_ID\_ARRAY = new List<int>();

    DateTime? EVT\_TS;

    char? SepticFl, WellFl;

    short? TopoCd;

    short? AgencyCd;

    int BaseYear;

    DataTable DtTaxRate;

    // Get the certain CSA related Prop Char attributes as of the Event TS of the document passed

    DataTable DtPceByDoc = Pass in DOC\_PREFIX, DOC\_SERIES use existing SP: IE\_GET\_PCE\_AS\_OF\_COO\_EVT\_TS\_BY\_DOC

    // Loop thru the DtPceByDoc and populate the following variables =

    int? TotBldgArea = 0;

    int? TotParking = 0;

    decimal? MinLandImpRatio = 0;

    decimal? MaxLandImpRatio = 0;

    int? MinNumBldg = 0;

    int? MaxNumBldg = 0;

    decimal? TotLotSize = 0;

    decimal? MinFloorAreaRatio = 0;

    decimal? MaxFloorAreaRatio = 0;

    TotBldgArea = DtPceByDoc.AsEnumerable().Sum(row => row.Field<int?>("tot\_building\_area")); // -' legacy names as gross\_sq\_ft for some reason

    TotParking = DtPceByDoc.AsEnumerable().Sum(row => row.Field<int?>("tot\_parking")); // TotParking = Sum (tot\_parking)

    MinLandImpRatio = DtPceByDoc.AsEnumerable().Min(row => row.Field<decimal?>("land\_imp\_ratio")); // Min (land\_imp\_ratio)

    MaxLandImpRatio = DtPceByDoc.AsEnumerable().Min(row => row.Field<decimal?>("land\_imp\_ratio")); // Max (land\_imp\_ratio)

    MinNumBldg = DtPceByDoc.AsEnumerable().Min(row => row.Field<Int16?>("num\_buildings")); // Min (num\_buildings)

    MaxNumBldg = DtPceByDoc.AsEnumerable().Min(row => row.Field<Int16?>("num\_buildings")); // Max (num\_buildings)

    TotLotSize = DtPceByDoc.AsEnumerable().Sum(row => row.Field<decimal?>("lot\_size")); // sum (lot\_size\_sq\_feet)

    MinFloorAreaRatio = DtPceByDoc.AsEnumerable().Min(row => row.Field<decimal?>("floor\_area\_ratio")); // min (floor\_area\_ratio)

    MaxFloorAreaRatio = DtPceByDoc.AsEnumerable().Min(row => row.Field<decimal?>("floor\_area\_ratio")); // max (floor\_area\_ratio)

                                                                                                       //

                                                                                                       // P\_ID\_ARRAY = build a comma delimited string of all the DtPceByDoc.PROPERTY\_ID values

    P\_ID\_string = string.Join(",", DtPceByDoc.AsEnumerable().Select(x => x.Field<int>("PROPERTY\_ID")));

    P\_ID\_ARRAY = DtPceByDoc.AsEnumerable().Select(x => x.Field<int>("PROPERTY\_ID")).ToList;

    EVT\_TS = (DateTime)DtPceByDoc(0)("event\_ts");

    // Get the building table columns for all the property ID’s passed

    var entBuildingList = Pass in Property\_ID to use SP: IE\_GET\_PROP\_CHAR\_BUILDING

    // Filter DtBuilding to " EFFECTIVE\_DT <= DATE(EVT\_TS) AND (END\_DT >= DATE(EVT\_TS) OR END\_DT is NULL) AND (DEMOLISHED\_FL IS NULL OR DEMOLISHED\_FL <> ‘Y’)

    var entBuildingListFiltered = entBuildingList.Where(x => System.Convert.ToBoolean(x.EFFECTIVE\_DT <= EVT\_TS & (x.END\_DT >= (DateTime)EVT\_TS | x.END\_DT == null) & (x.DEMOLISHED\_FL == null | x.DEMOLISHED\_FL != "Y"))).ToList;

    // Sort DtBuilding by property\_id ASC, building\_num ASC, effective\_dt DESC

    entBuildingListFiltered.OrderBy(x => x.PROPERTY\_ID).OrderBy(y => y.BUILDING\_NUM).OrderByDescending(z => z.EFFECTIVE\_DT).ToList();

    short? MinSbeCode;

    short? MaxSbeCode;

    short? MinMvsCode;

    short? MaxMvsCode;

    // Loop thru the DtBuilding

    // Check the current row With the Next row, If the property\_id And building\_num column values on those 2 rows are the same then discard the next row.

    // Next

    List<ENT.IEGetPropCharBuilding> entList = new List<ENT.IEGetPropCharBuilding>(entBuildingListFiltered.Count);

    // Check If there are duplicate DtCsaLease.print\_parcel in the result set

    if (entBuildingListFiltered.Count > 0)

    {

        for (var i = 0; i <= entBuildingListFiltered.Count - 1; i++)

        {

            entList.Add(entBuildingListFiltered(i));

            for (var j = i + 1; j <= entBuildingListFiltered.Count - 1; j++)

            {

                if (entBuildingListFiltered(i).PROPERTY\_ID == entBuildingListFiltered(j).PROPERTY\_ID & entBuildingListFiltered(i).BUILDING\_NUM == entBuildingListFiltered(j).BUILDING\_NUM)

                {

                    entList.RemoveAt(i);

                    break;

                }

            }

        }

    }

    // Filter DtBuilding to sbe\_code <> “”

    if (entList.Where(x => x.SBE\_SHAPE\_CD\_NOT\_NULL > 0).ToList().Count > 0)

    {

        // If DtBuilding has rows Then

        // Sort DtBuilding to sbe\_code ASC

        entList = entList.OrderBy(x => x.SBE\_SHAPE\_CD).ToList;

        MinSbeCode = entList[0].SBE\_SHAPE\_CD; // from the 1St row

                                              // Sort DtBuilding to sbe\_code DESC

        entList = entList.OrderByDescending(x => x.SBE\_SHAPE\_CD).ToList;

        MaxSbeCode = entList[0].SBE\_SHAPE\_CD; // from the 1St row

    }

    // Clear all filters on DtBuilding

    // Filter DtBuilding to mvs\_code <> “”

    if (entList.Where(x => x.MVS\_SHAPE\_CD\_NOT\_NULL > 0).ToList().Count > 0)

    {

        // If DtBuilding has rows Then

        // Sort DtBuilding to mvs\_code ASC

        entList = entList.OrderBy(x => x.MVS\_SHAPE\_CD).ToList;

        MinMvsCode = entList[0].MVS\_SHAPE\_CD;

        // MinMvsCode = DtBuilding.mvs\_code from the 1St row

        // Sort DtBuilding to mvs\_code DESC

        entList = entList.OrderByDescending(x => x.MVS\_SHAPE\_CD).ToList;

        MaxMvsCode = entList[0].MVS\_SHAPE\_CD; // from the 1St row

    }

    // Perform the below calculations, details to this must be in the FR

    // Loop thru the DtBuilding and populate the following variables =

    var TotNetRentableArea = entBuildingListFiltered.Sum(x => x.NET\_RENTABLE\_AREA); // Sum (net\_rentable\_area)

    var MinBuiltYr = entBuildingListFiltered.Min(x => x.BUILT\_YR); // Min (built\_yr)

    var MaxBuiltYr = entBuildingListFiltered.Max(x => x.BUILT\_YR); // Max (built\_yr)

    var MinEffectiveYr = entBuildingListFiltered.Min(x => x.EFFECTIVE\_YR); // Min (effective\_yr)

    var MaxEffectiveYr = entBuildingListFiltered.Max(x => x.EFFECTIVE\_YR); // Max (effective\_yr)

    var MinPctBuiltOut = entBuildingListFiltered.Min(x => x.PCT\_BUILT\_OUT); // Min (pct\_built\_out)

    var MaxPctBuiltOut = entBuildingListFiltered.Max(x => x.PCT\_BUILT\_OUT); // Max (pct\_built\_out)

    var MinPctOffice = entBuildingListFiltered.Min(x => x.PCT\_OFFICE); // Min (pct\_office)

    var MaxPctOffice = entBuildingListFiltered.Max(x => x.PCT\_OFFICE); // Max (pct\_office)

    var MinNumStories = entBuildingListFiltered.Min(x => x.NUM\_STORIES); // Min (num\_stories)

    var MaxNumStories = entBuildingListFiltered.Max(x => x.NUM\_STORIES); // Max (num\_stories)

    var MinAvgActualHt = entBuildingListFiltered.Min(x => x.AVG\_ACTUAL\_HT); // Min (avg\_actual\_ht)

    var MaxAvgActualHt = entBuildingListFiltered.Max(x => x.AVG\_ACTUAL\_HT); // Max (avg\_actual\_ht)

    // Get the Land table columns for all the property ID’s passed

    List<PropCharLand> propList = DAL.CSAProperty.GetLand(0, PropertyIDArray, transaction);

    var templist = propList.Where(x => x.EFFECTIVE\_DT <= AsOfDate && (x.END\_DT == null || (x.END\_DT.HasValue && x.END\_DT >= AsOfDate)));

    List<PropCharLand> lstLand = templist;

    // Down filte already happening in the DAL layer

    // Filter DtLand to " EFFECTIVE\_DT <= DATE(EVT\_TS) AND (END\_DT >= DATE(EVT\_TS) OR END\_DT is NULL)”

    // Sort DtLand by EFFECTIVE\_DT DESC

    lstLand.OrderBy(x => x.EFFECTIVE\_DT).ToList();

    SepticFl = lstLand[0].SEPTIC\_FL; // DtLand.septic\_fl

    WellFl = lstLand[0].WELL\_FL; // DtLand.well\_fl

    // Get the topography table columns for all the property ID’s passed

    List<ENT.IEGetPropCharTopography> lstPropTopo = Use Property\_ID as Parameter for SP: IE\_GET\_PROP\_CHAR\_TOPOGRAPHY

    // If DtPropTopo has rows THEN

    if (lstPropTopo.Count() > 0)

        TopoCd = lstPropTopo[0].TOPO\_CD;// DtPropTopo.topo\_cd 'from the 1St row

    // Get the topography table columns for all the property ID's passed

    List<ENT.IEGetPropCharZoning> lstZoning = Use Property\_ID as Parameter for SP: IE\_GET\_PROP\_CHAR\_ZONING

    if (lstZoning.Count > 0)

    {

        ZoningCd = lstZoning[0].ZONING\_CD;  // ZoningCd = DtZoning.zoning\_cd from the 1St row

        AgencyCd = lstZoning[0].AGENCY\_CD; // from the 1St row

    }

    CsaSaleInfoEnt = new ENT.CsaSaleInfo();

    // Populate the CSA Sales Info entity

    CsaSaleInfoEnt.CSA\_ID = CsaId;

    CsaSaleInfoEnt.CSA\_TYPE = CsaSelectType; // CsaSelectType passed

    CsaSaleInfoEnt.DOC\_PREFIX = DOC\_PREFIX; // passed

    CsaSaleInfoEnt.DOC\_SERIES = DOC\_SERIES; // passed

    CsaSaleInfoEnt.PROPERTY\_ID = PRIMARY\_PROP\_ID; // passed

    CsaSaleInfoEnt.USE\_CD = System.Convert.ToInt16(DtPceByDoc(0)("use\_cd"));

    CsaSaleInfoEnt.SITUS\_CITY\_NAME = Convert.ToString(DtPceByDoc(0)("situs\_city\_name"));

    CsaSaleInfoEnt.TOT\_BUILDING\_AREA = TotBldgArea;

    CsaSaleInfoEnt.TOT\_NET\_RENT\_AREA = TotNetRentableArea;

    CsaSaleInfoEnt.TOT\_LOT\_SIZE = System.Convert.ToInt32(TotLotSize);

    CsaSaleInfoEnt.ZONING\_CD = ZoningCd;

    CsaSaleInfoEnt.ZONING\_AGENCY\_CD = AgencyCd;

    if (CsaSelectType == 933)

    {

        // Populate the CSA Prop Char entity

        CsaPropCharEnt = new ENT.CsaPropChar();

        CsaPropCharEnt.CSA\_ID = CsaId;

        CsaPropCharEnt.TOT\_PARKING = TotParking;

        // If DtBuilding has row/s Then

        if (entBuildingList.Count > 0)

        {

            CsaPropCharEnt.SBE\_CLASS\_LOW = MinSbeCode.ToString();

            CsaPropCharEnt.SBE\_CLASS\_HI = MaxSbeCode.ToString();

            CsaPropCharEnt.MVS\_CLASS\_LOW = MinMvsCode.ToString();

            CsaPropCharEnt.MVS\_CLASS\_HI = MaxMvsCode.ToString();

            CsaPropCharEnt.BLDG\_BUILT\_YR\_LOW = MinBuiltYr;

            CsaPropCharEnt.BLDG\_BUILT\_YR\_HI = MaxBuiltYr;

            CsaPropCharEnt.BLDG\_EFF\_YR\_LOW = MinEffectiveYr;

            CsaPropCharEnt.BLDG\_EFF\_YR\_HI = MaxEffectiveYr;

            CsaPropCharEnt.PCT\_BUILT\_OUT\_LOW = MinPctBuiltOut;

            CsaPropCharEnt.PCT\_BUILT\_OUT\_HI = MaxPctBuiltOut;

            CsaPropCharEnt.PCT\_OFFICE\_LOW = MinPctOffice;

            CsaPropCharEnt.PCT\_OFFICE\_HI = MaxPctOffice;

            CsaPropCharEnt.NUM\_BUILDINGS\_LOW = MinNumBldg;

            CsaPropCharEnt.NUM\_BUILDINGS\_HI = MaxNumBldg;

            CsaPropCharEnt.NUM\_STORIES\_LOW = MinNumStories;

            CsaPropCharEnt.NUM\_STORIES\_HI = MaxNumStories;

            CsaPropCharEnt.AVG\_ACTUAL\_HT\_LOW = MinAvgActualHt;

            CsaPropCharEnt.AVG\_ACTUAL\_HT\_HI = MaxAvgActualHt;

            CsaPropCharEnt.LAND\_IMP\_RATIO\_LOW = MinLandImpRatio;

            CsaPropCharEnt.LAND\_IMP\_RATIO\_HI = MaxLandImpRatio;

            CsaPropCharEnt.FAR\_LOW = MinFloorAreaRatio;

            CsaPropCharEnt.FAR\_HI = MaxFloorAreaRatio;

        }

    }

    if (CsaSelectType == 934)

    {

        // Populate the CSA Vacant Land entity

        CsaVlCharEnt = Fetch current vacant land realting to the CSA\_ID;

        if (!(string.IsNullOrEmpty(DtPceByDoc(0)("slope\_cd").ToString)))

            CsaVlCharEnt.SLOPE\_CD = System.Convert.ToInt16(DtPceByDoc(0)("slope\_cd"));// DtPceByDoc.slope\_cd from the first row

        if (!(string.IsNullOrEmpty(DtPceByDoc(0)("view\_cd").ToString)))

            CsaVlCharEnt.VIEW\_CD = System.Convert.ToInt16(DtPceByDoc(0)("view\_cd"));// from the first row

        CsaVlCharEnt.TOPO\_CD = TopoCd;

        CsaVlCharEnt.SEPTIC\_TANK\_FL = SepticFl;

        CsaVlCharEnt.WELL\_FL = WellFl;

    }

    // DB Updates section

    DB2Transaction transaction;

    DB2Connection connection = null/\* TODO Change to default(\_) if this is not a reference type \*/;

    if (P\_transaction == null)

    {

        connection = DB2Helper.GetDBConnection();

        transaction = DB2Helper.BeginTransaction(connection);

    }

    else

        transaction = P\_transaction;

    try

    {

        if (CsaSaleInfoEnt != null)

        {

            CsaSaleInfoEnt.SetEntryFields(WorkerId, UserId);

            AddCsaSaleInfo(CsaSaleInfoEnt);

        }

        if (CsaSelectType == 933)

        {

            if (CsaPropCharEnt != null)

            {

                CsaPropCharEnt.SetEntryFields(WorkerId, UserId);

                AddCsaPropChar(CsaPropCharEnt);

            }

            if (CA\_LIST.Count > 0)

            {

                foreach (ENT.CsaAmount ent in CA\_LIST)

                {

                    // Loop thru And execute below for each entity found in the LIST

                    ent.SetEntryFields(WorkerId, UserId);

                    AddCsaAmount(ent);

                }

            }

        }

        if (CsaSelectType == 934)

        {

            if (CsaVlCharEnt != null)

            {

                CsaVlCharEnt.SetEntryFields(WorkerId, UserId);

                AddCsaVlChar(CsaVlCharEnt);

            }

        }

        retval = true;

        if (retval)

            transaction.Commit();

        else

            transaction.Rollback();

        return retval;

    }

    // Return CsaId

    catch (Exception ex)

    {

        if (P\_transaction == null)

            transaction.Rollback();

        throw new Exception("BLL.ManageSetCsaDetails: " + ex.Message, ex);

    }

    finally

    {

        if (P\_transaction == null)

            DAL.DB2Helper.EndTransaction(transaction, connection);

    }

}

#### GetLand

// PropCharLand entity will be required to be generated as entity

public Task<List<PropCharLand>> GetLand(int PropertyID = 0, List<int> PropertyIDArray = null, DB2Transaction transaction = null)

{

    // Dim propList As List(Of ENT.IEGetPropCharLand) = New List(Of ENT.IEGetPropCharLand)()

    string SPName = "IE\_GET\_PROP\_CHAR\_LAND\_MULTIPLE";

    List<ENT.IEGetPropCharLand> entityList = new List<ENT.IEGetPropCharLand>();

    DB2Connection db2Conn = null/\* TODO Change to default(\_) if this is not a reference type \*/;

    DB2DataReader db2Reader = null/\* TODO Change to default(\_) if this is not a reference type \*/;

    if (transaction != null)

        db2Conn = transaction.Connection;

    else

        db2Conn = DB2Helper.GetDBConnection();

    string PropertyIdString = "";

    if (PropertyID > 0)

        PropertyIdString = PropertyID.ToString();

    else if (PropertyIDArray != null)

        PropertyIdString = string.Join(",", PropertyIDArray);

    DB2Parameter[] parms = new DB2Parameter[1];

    parms[0] = new DB2Parameter("P\_PROPERTY\_ID", DB2Type.VarChar, 4000);

    parms[0].Value = PropertyIdString;

    // Dim dtList As DataTable = DB2Helper.ExecuteDataTable(SPName, parms, transaction)

    if (transaction != null)

        db2Reader = DB2Helper.ExecuteReader(db2Conn, transaction, CommandType.StoredProcedure, SPName, parms);

    else

        db2Reader = DB2Helper.ExecuteReader(db2Conn, CommandType.StoredProcedure, SPName, parms);

    // For Each dr As DataRow In dtList.Rows

    // Dim propcharLand As New ENT.IEGetPropCharLand

    // propcharLand.LoadFromDataRow(dr)

    // propList.Add(propcharLand)

    // Next

    entityList = Utility.GetEntitiesFromReader<ENT.IEGetPropCharLand>(db2Reader);

    // close the reader

    db2Reader.Close();

    // close connection if no transaction

    if (transaction == null)

        db2Conn.Close();

    return entityList;

}

#### GetPropertyInfo

public Task<PropertyInfo> GetPropertyInfo (int p\_csa\_id)

IE\_GET\_CSA\_PROP\_INFO

#### GetMailingNameAsOfDate

public Task<MailingName> GetMailingNameAsOfDate (int p\_CSA\_Id, DataTime p\_Event\_Ts)

IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE

#### ChangeAPN

public Task<int> ChangeAPN (int p\_csa\_id)

UPDATE ieaown.ie\_csa\_sale\_info

SET property\_id = @property\_id,

    update\_worker = @update\_worker,

    update\_ts = @update\_ts

#### GetAllAPNinDoc

public Task<List<APN>>GetAllAPNinDoc (int p\_csa\_id)

IE\_GET\_CSA\_ALL\_APN\_IN\_DOC

### CSALease Class

#### GetTotalNRA

public Task<string> GetTotalNRA (int p\_CSAId)

IE\_GET\_CSA\_TOTAL\_NRA

#### GetSqFtLeased

public Task<string> GetSqFtLeased (int p\_csa\_id)

IE\_GET\_CSA\_SQFT\_LEASED

#### UpdatePctOccupNRA

public Task<int> UpdatePctOccupNRA (int p\_csa\_id, decimal?\_pctOccupancy, int? \_propNRA, string p\_Worker\_id)

//Double check for existing SP

Update IE\_CSA\_ANALYSIS with Occupancy entity

#### GetLeaseInfo

public Task<List<LeaseInfo>> GetLeaseInfo (int p\_csa\_id)

IE\_GET\_CSA\_LEASE\_INFO

#### GetTotalRent

public Task<RentInfo> GetTotalRent (int p\_csa\_id)

IE\_GET\_CSA\_TOTAL\_RENT

#### GetTotalRent (Overloading)

public Task<RentInfo> GetTotalRent (int p\_csa\_id, int p\_excluded\_lease\_id)

IE\_GET\_CSA\_TOTAL\_RENT\_EXC\_LEASE

#### DeleteCSALease

public Task<int>DeleteCSALease (int p\_lease\_id)

IE\_DEL\_CSA\_LEASE

### CSASourceInput Class

#### UpdateCSASourceData

public Task<int> UpdateCSASourceData(CSASourceData source\_data )

//Make sure to complete two stored procedures in a transaction. Roll back if exception is thrown.

UPDATE IE\_CSA\_SALE\_INFO

WITH SaleInfoSuprvApprove

WHERE CSA\_ID match also SUPRV\_APPROVED\_FL in DB is DBNull or different from updated value

UPDATE IE\_CSA\_SOURCE table for all variable in CSASourceData

#### GetSource

public Task<CSASource> GetSource (int p\_csa\_id)

IE\_GET\_CSA\_SOURCE

#### GetWorker

public Task<List<WorkerInfo>> GetWorker()

IE\_GET\_CSA\_ALL\_WORKER

### CSAVacantLandInput Class

#### GetPropertyInfo

public Task<VLPropertyInfo> GetPropertyInfo (int p\_csa\_id)

IE\_GET\_CSA\_PROP\_INFO\_VACANT\_LAND

### CSAShared Class

#### GetTaxRate

public Task<Decimal?> GetTaxRate (int propertyId, DateTime eventDate, int baseYear)

IE\_GET\_CSA\_TAX\_RATE

#### GetNRAOccupancy

public Task<NRAOccupancy> GetNRAOccupancy (int p\_csa\_id)

IE\_GET\_CSA\_NRA\_OCCUPANCY

#### GetCSAComments

public Task<List<CSAComments>> GetCSAComments (int csa\_id)

IE\_GET\_CSA\_COMMENTS

#### GetFieldCode

public Task<List<FieldCode>> GetFieldCode (int p\_field\_id)

IE\_GET\_FIELD\_CODE

#### GetAnticipatedUseCodesNotForLandUse

public Task<AntiUseCodesNotForLandUse> GetAnticipatedUseCodesNotForLandUse()

IE\_GET\_CSA\_ANTICIPATED\_USE\_CODES\_NOT\_FOR\_LAND\_USE

#### GetLeaseUseCategories

public Task<List<LeaseUseCategory>> GetLeaseUseCategories ()

IE\_GET\_CSA\_LEASE\_USE\_CATEGORY

#### GetLeaseUseSubByUseCat

public Task<List<LeaseUseSubCat>> GetLeaseUseSubByUseCat (int p\_use\_cat)

IE\_GET\_CSA\_LEASE\_USE\_SUB\_BY\_USE\_CAT

#### UpdateCSAComments

public Task<int> UpdateCSAComments (CSAComment source\_data)

UPDATE IE\_CSA\_COMMENT

USE CSAComment entity

SKIP UPDATE\_TS and UPDATE\_WORKER

#### InsertCSAComments

public Task<int> InsertCSAComments (CSAComment source\_data)

INSERT INTO **IEAOWN.IE\_CSA\_COMMENT**

CSA\_ID, SEQ\_NUM, COMMENT\_TEXT, ENTRY\_TS, ENTRY\_WORKER

#### GetPCBuildingAsofDateByPropId

public Task<PCBuilding> GetPCBuildingAsofDateByPropId (int p\_prop\_id)

IE\_GET\_PC\_BUILDING\_AS\_OF\_DT

#### GetPCEAsofDate

public Task<PCEvent> GetPCEAsofDate (int p\_csa\_id)

IE\_GET\_PCE\_AS\_OF\_DT

#### GetUseCdAsofDateByPropId

public Task<string> GetUseCdAsofDateByPropId (int p\_prop\_id, Date p\_as\_of\_date)

IE\_GET\_USE\_CD\_AS\_OF\_DATE

#### GetSitusCityByPropertyId

public Task<string> GetSitusCityByPropertyId (int p\_prop\_id)

IE\_GET\_SITUS\_CITY

#### GetVLCharAsOfDateByPropId

public Task<VLPropChar> GetVLCharAsOfDateByPropId (int p\_prop\_id)

IE\_GET\_VL\_PROP\_CHAR\_AS\_OF\_DATE

### CSAIncAnalysisValInd Class

#### GetIncomeAnalysis

public Task<IncomeAnalysis> GetIncomeAnalysis(int p\_csa\_id)

IE\_GET\_CSA\_INCOME\_ANALYSIS

#### UpdateIncomeAnalysis

public Task<int> UpdateIncomeAnalysis(UpdatedIncomeAnalysis p\_income\_analysis)

UPDATE IE\_CSA\_ANALYSIS

WITH IncomeExpenseAnalysis entity

WHERE CSA\_ID match

#### GetAdjustCSAAmount

public Task<decimal> GetAdjustCSAAmount(int p\_csa\_id)

IE\_GET\_CSA\_ADJUST\_AMOUNT

#### GetPctOccupThreshld

public Task<decimal> GetPctOccupThreshld()

IE\_GET\_CSA\_PCT\_OCCUP\_THRESHLD

#### GetExpenseTotal

public Task<ExpenseTotal> GetExpenseTotal(int p\_csa\_id)

IE\_GET\_CSA\_EXPENSE\_TOTAL

#### GetPotentialGrossIncome

public Task<PotentialIncome> GetPotentialGrossIncome(int p\_csa\_id)

IE\_GET\_CSA\_POTENTIAL\_INCOME

#### GetSourceDesc

Note: Both GetSourceDesc and GetBenchmarkRating functions in BLL layer will be using this method. Only difference is GetSourceDesc function will use CD\_SHORT\_NAME while GetBenchmarkRating will use CD\_LONG\_NAME.

public Task<SourceDesc> GetSourceDesc(int p\_field\_id)

IE\_GET\_FIELD\_CODE

### CSALeaseInfo Class

Make sure to execute multiple stored procedure using transaction. Only proceed further when all succeed or rollback.

#### GetLease

public Task<LeaseBO> GetLease (int p\_lease\_id)

Here make sure to combine results into LeaseBO and output back to BLL layer

IE\_GET\_CSA\_LEASE, return Lease entity

#### InsLeaseNoRentComp

public Task<int> InsLeaseNoRentComp (LeaseBO p\_lease\_bo)

INSERT INTO **IEAOWN.IE\_CSA\_LEASE**

lease\_id, property\_id, lease\_location, lessee\_name,  net\_rentable\_area, lease\_start\_dt, lease\_term, lease\_option, anticipated\_rent, anti\_rent\_psf, current\_rent, current\_rent\_psf, vacant\_fl, use\_for\_comp\_fl, lease\_type\_cd, expense\_type\_cd, lease\_created\_by,   lease\_create\_dt, entry\_worker, entry\_ts, suprv\_approved\_fl, suprv\_appr\_upd\_wkr, suprv\_appr\_upd\_ts, csa\_id

#### InsertLease

public Task<int> InsertLease (LeaseBO p\_lease\_bo)

INSERT INTO **IEAOWN.IE\_CSA\_LEASE**

lease\_id, property\_id, lease\_location, use\_category\_cd, lease\_subcategory,

 lessee\_name, net\_rentable\_area, lease\_start\_dt, lease\_term, lease\_option,

 ti\_type\_cd, ti\_allowance, info\_source\_text, rent\_adj\_text, lease\_parking\_2,

 initial\_rent, initial\_rent\_psf, stabil\_rent, stabil\_rent\_psf,

 rent\_adjustments, tenant\_imps\_psf, anticipated\_rent, anti\_rent\_psf, current\_rent, current\_rent\_psf, vacant\_fl, use\_for\_comp\_fl, lease\_type\_cd, expense\_type\_cd, expense\_base\_yr, tenancy\_cd, lease\_parking, sprinkler\_fl, info\_source\_cd, lease\_created\_by,

lease\_create\_dt, entry\_worker, entry\_ts, suprv\_approved\_fl, suprv\_appr\_upd\_wkr, suprv\_appr\_upd\_ts, csa\_id

INSERT INTO **IEAOWN.IE\_CSA\_LEASE\_DET**

lease\_id, ind\_clr\_hgt, ind\_offc\_area\_pct, ind\_lab\_pct, ind\_mfg\_pct, ind\_warehouse\_pct, ind\_retail\_pct, ind\_other\_pct, offc\_load\_factor, offc\_expense\_stop, retail\_dba, retail\_rent\_pct, rent\_breakpoint, retail\_store\_occ, entry\_worker, entry\_ts

#### UpdLeaseNoRentComp

public Task<int> UpdLeaseNoRentComp (LeaseBO p\_lease\_bo)

UPDATE IE\_CSA\_LEASE

WITH LeaseBO entity

WHERE LEASE\_ID match

#### UpdateLease

public Task<int> UpdateLease (LeaseBO p\_lease\_bo)

UPDATE IE\_CSA\_LEASE

WITH LeaseBO entity

WHERE LEASE\_ID match

UPDATE IE\_CSA\_LEASE\_DET

USING LeaseBO.LeaseDetailBO entity

WHERE LEASE\_ID match

### CSALease Class

#### GetComments

public Task<List<LeaseComment>> GetComments (int p\_lease\_id)

IE\_GET\_CSA\_LEASE\_COMMENTS

### wkSheetNewImps Class

#### GetAPNs

public Task<List<wkSheetNewImp>> GetAPN (int p\_csa\_wks\_num)

{

    List<wkSheetNewImp> \_apnList = new List<wkSheetNewImp>();

    DataTable dt = wkSheetNewImpsDAL.GetAPNList(csa\_wks\_num);

    string \_lotSize = null;

    string \_bldgArea = null;

    foreach (DataRow dr in dt.Rows())

    {

        if (PRIMARY\_APN\_FL == null)

            PRIMARY\_APN\_FL = "N";

        if (TOT\_LOT\_SIZE == null)

            \_lotSize = null;

        else

            \_lotSize = string.Format("{0:###,###,###}", TOT\_LOT\_SIZE);

        if (TOT\_BUILDING\_AREA == null)

            \_bldgArea = null;

        else

            \_bldgArea = string.Format("{0:###,###,###}", TOT\_BUILDING\_AREA);

        wkSheetNewImp \_wkSheet = new wkSheetNewImp(CSA\_SUBJECT, PRIMARY\_APN\_FL, PRINT\_PARCEL, PROPERTY\_ID, USE\_NAME\_SHORT, \_lotSize, \_bldgArea);

        \_apnList.Add(\_wkSheet);

    }

    return \_apnList;

}

#### GetAPNList

public Task<List<wksAPN>> GetAPNList (int p\_csa\_wks\_num)

IE\_GET\_CSA\_WKS\_APN\_LIST

#### UpdateCSAAPN

public Task<int> UpdateCSAAPN(int csa\_wks\_num, int prop\_id\_prime, string[] \_subjectPropID, string entry\_worker)

IE\_DEL\_CSA\_WKS\_APN

INSERT INTO IEAOWN.IE\_CSA\_WKS\_APN

(CSA\_WKS\_NUM, PROPERTY\_ID, PRIMARY\_APN\_FL, ENTRY\_TS, ENTRY\_WORKER)

UPDATE IE\_CSA\_WKS\_DET

SET PROPERTY\_ID

WHERE CSA\_WKS\_NUM matches and SEQ\_NUM is 0

Remember to add the apn that is been selected as primary to parameter and set its value to “Y” while th rest is set to “N”

If apn is selected Then

cmd.Parameters.Add("@primary\_apn\_fl", DB2Type.Char).Value = "Y"

Else

cmd.Parameters.Add("@primary\_apn\_fl", DB2Type.Char).Value = "N"

Add current entry worker as well as current time as Event TS to parameters

## BLL Layer

### Note:

**For any business layer method that simply call DAL layer function to retrieve data. Use below code as an example to apply try-catch block to catch any exception during the process:**

**First remember to inject \_CSALeaseDAL to class constructor in order to use it as service.**

Declare the object will be assigned and used with dependency injection.

private readonly CSALeaseDAL \_CSALeaseDAL

Constructor Dependency Injection

public CSALeaseBLL (CSALeaseDAL csaLeaseDAL)

{

\_CSALeaseDAL = csaLeaseDAL;

}

Use Asynchronous programming to improve performance when possible

public Task<LeaseInfo> GetLeaseInfo(int p\_csa\_id)

{

    LeaseInfo \_LeaseInfo;

Try catch block here in case any exception needs be thrown

try

{

leaseInfoTable = await \_CSALeaseDAL.GetLeaseInfo(p\_csa\_id);

}

catch

{

throw;

}

    return \_LeaseInfo;

}

### CSASale Class

#### GetSaleInfoFull

Function aims to retrieve sale information, enhance it with property details, and include buyer and seller names based on mailing information. These data will be combined as output: FullSaleInfo, PropertyInfo and MailingName based on CSA ID given.

public Task<List<SaleInfo>> GetFullSaleInfo (int p\_csa\_id)

{

    int \_csa\_Id, \_property\_Id;

    DateTime \_event\_ts;

Use to store output result

    List<CSASaleInfo> outputList = new List<CSASaleInfo>();

Use to store query result from stored procedure, use this to retrieve SaleInfoFull entity

List<SaleInfoFull> \_ListOfSaleInfo = new List<SaleInfoFull>();

try

{

\_ListOfSaleInfo = await \_CSASaleDAL.GetFullSaleInfo (p\_CSA\_Id);

}

catch

{

throw;

}

Go through all data from entity and store them within CSASaleInfo model to output

    foreach (SaleInfoFull drSaleInfo in \_ListOfSaleInfo)

    {

CSASaleInfo \_CSASaleInfo = new CSASaleInfo();

Below values can skip check and directly assign them:

CSA\_ID, CSA\_TYPE, DOC\_PREFIX, DOC\_SERIES, ENTRY\_TS, ENTRY\_WORKER, PROPERTY\_ID,

For the rest of the data exist in the SaleInfoFull entity, check whether if it’s null and assign the value to the model CSASaleInfo when it’s not null.

**PropertyInfo** \_drProperty = null;

try

{

Retrieve data to assign to CSASaleInfo model

\_drProperty = \_CSAPropertyBLL.GetPropertyInfo (p\_csa\_id);

}

catch

{

throw;

}

Assign the retrieved Property Info information to corresponding field of the CSASaleInfo

        if (\_drProperty != null)

        {

            if (drProperty.SITUS\_CITY\_NAME != null)

            {

                \_CSASaleInfo.SITUS\_CITY = drProperty.SITUS\_CITY\_NAME;

            }

            if (drProperty.SITUS\_STATE\_NAME != null)

            {

                \_CSASaleInfo.SITUS\_STATE = drProperty.SITUS\_STATE\_NAME;

            }

        }

**MailingName** \_dtMailingName;

try

{

This is used to return buyer and seller name, using exist same function but event\_ts minute will need to bew modified.

\_dtMailingName = \_CSAPropertyBLL.GetMailingNameAsOfDate(\_property\_Id, \_event\_ts);

}

catch

{

throw;

}

        if (\_dtMailingName)

        {

            \_CSASaleInfo.BUYER\_NAME = \_dtMailingName.MAILING\_NAME;

        }

try

{

This is used to return seller’s name, when event\_ts minus 1 minute, without will retrieve buyer’s name

\_dtMailingName = \_CSAPropertyBLL.GetMailingNameAsOfDate(\_property\_Id, \_event\_ts.AddMinutes(-1));

}

catch

{

throw;

}

        if (\_dtMailingName != null && \_dtMailingName.MAILING\_NAME != null)

        {

            \_CSASaleInfo.SELLER\_NAME = \_dtMailingName.MAILING\_NAME;

        }

        outputList.Add(\_CSASaleInfo);

    }

    return outputList;

}

#### UpdateSaleInfoWithCIAnalysis

Since validations are performed in Angular already here just use try catch block to run the DAL function UpdateSaleInfoWithCISaleAnalysis would be sufficient

public Task<int> UpdateSaleInfoWithCIAnalysis(UpdatedSaleInfoWithCIAnalysis SaleInfo)

#### GetSaleExpenses

This function will use CSA ID to do a simple extraction of all Sale Expenses related to this particular CSA ID. The return result is a list of the CSASaleExpense entity.

public Task<List<CSASaleExpenses>> GetSaleExpenses(int p\_csa\_id, bool p\_expense\_bool)

{

    List<CSASaleExpenses> \_ListOfSaleExpenses = new List<CSASaleExpenses>();

    List<SaleInfoFull> \_ListOfSaleInfo = \_CSADal.GetFullSaleInfo(p\_csa\_id);

    if (\_ListOfSaleInfo.Count > 0)

    {

**CSASaleExpenses** \_SaleExpenses = new CSASaleExpenses(\_ListOfSaleInfo[0].CSA\_Id)

        {

Here only the top result retrieved will be add, and used to init this CSASaleExpenses

            full\_expenses\_fl = \_ListOfSaleInfo[0].FULL\_EXPENSES\_FL,

            tot\_building\_area = \_ListOfSaleInfo[0].TOT\_BUILDING\_AREA,

            tot\_net\_rent\_area = \_ListOfSaleInfo[0].TOT\_NET\_RENT\_AREA,

            csa\_net\_rent\_area = \_ListOfSaleInfo[0].CSA\_NET\_RENT\_AREA

        };

        \_ListOfSaleExpenses.Add(\_SaleExpenses);

    }

    return \_ListOfSaleExpenses;

}

#### GetExpenses

This function is served to quick retrieve a list of expenses corresponding to a given CSA ID and expensesforallexpensecodesbool value to extract list of expenses in a CSASaleExpense model, treat this more as a function to be used in case there are loading issue with the CSASaleExpenses model property get method.

public Task<List<CSASExpense>> GetExpenses(int p\_csa\_id, bool expense\_bool)

{

The expense\_bool will determine which function below to call,

    List<CSAExpense> \_ListOfExpenses = new List<CSAExpenses>

    if (expense\_bool)

    {

        Get expense amounts for all expense codes

        \_ListOfExpenses = \_CSASaleBll.GetExpensesForAllExpenseCodes (p\_csa\_id);

    }

    else

    {

        Get expense amounts for actual CSA amount rows only for the given CSA ID

        \_ListOfExpenses = \_CSASaleBll.GetActualExpenses(p\_csa\_id);

    }

    return \_ListOfExpenses;

}

#### UpdateExpensesInSaleExpenses

In summary, the code updates the full expenses flag in the CSA sale information and iterates through a list of expenses, updating or inserting each expense's details. The method returns 0 for invalid input, -1 for errors during the update, and 1 for a successful update.

public Task<int> UpdateExpensesInSaleExpenses(CSASaleExpenses csasaleExpense, string \_LastModifyPage, string workerId)

{

    string \_Full\_Expenses\_Fl;

    int \_CSA\_Id, rc;

    if (csasaleExpenses == null) return 0;

    \_CSA\_Id = csasaleExpenses.CSA\_Id;

    \_Full\_Expenses\_Fl = csasaleExpenses.Full\_Expenses\_Fl;

    rc = \_CSASaleDAL.UpdateFullExpensesFl(\_CSA\_Id, \_Full\_Expenses\_Fl, workerId);

    if (rc < 0) return -1;

    // Loop through the list of expenses

    foreach (CSAExpense \_Exp in csasaleExpenses.ListOfExpenses)

    {

        if (\_Exp.CMC\_Amount\_Cd <= 0) continue;

        // Insert or update the expense row

        rc = \_CSASaleDAL.CSA\_AMOUNT\_ModExpenseByCSAIdAmountCd(

                \_CSA\_Id, \_Exp.CMC\_Amount\_Cd,

                \_Exp.CMC\_Update\_Desc\_Fl, \_Exp.Expense\_Description,

                \_Exp.Amount\_Owners, \_Exp.Amount\_Tenant,

                \_LastModifyPage, workerId);

        if (rc < 0) return -1;

    }

    return 1;

}

#### GetSaleAdjustments

Here we will use GetPropertyInfo to obtained a list of property info relating to the given csa id. Code will translate each property info into CSASaleAdjustments of the CSA ID. The final output will be a list of CSASaleAdjustments objects.

public Task< CSASaleAdjustments> GetSaleAdjustments(int p\_csa\_id)

{

// This list of CSASaleAdjustments object will be the final output

    CSASaleAdjustments \_CSASaleAdjustments = new CSASaleAdjustments(p\_csa\_id)

   PropertyInfo \_PropertyInfo = \_CSASaleDAL.GetPropertyInfo(p\_csa\_id);

// Retrieved Property Info entity will be used to assign CSASaleAdjustments object

if (\_PropertyInfo.ADJ\_SALES\_PRICE != null)

\_SaleAdjustments.adj\_sales\_price = (double)\_Property.Adj\_Sales\_PRICE;

if (\_PropertyInfo.IND\_PUR\_PRICE != null)

\_SaleAdjustments.ind\_pur\_price = (double)\_Property.IND\_PUR\_PRICE;

   return \_CSASaleAdjustments;

}

#### UpdateAdjustmentsInSaleAdjustments

Go through each adjustment in ListOfAdjustments property within CSAAdjusment and updating them using CSA\_AMOUNT\_ModAdjustmentByCSAIdAmountCd in DAL to update

public Task<int> UpdateAdjustmentsInSaleAdjustments (CSASaleAdjustments p\_csa\_sale\_adjustments, string p\_last\_modify\_page, string p\_worker\_id)

First check if p\_csa\_sale\_adjustments is not null. Then go through each CSAAdjustment the ListOfAdjustments in CSASaleAdjustments object.

Use CSA\_AMOUNT\_ModAdjustmentByCSAIdAmountCd(p\_csa\_id, Adj.CMC\_Amount\_Cd, Adj.CAM\_Update\_Desc\_Fl, Adj.CAM\_Amount\_Desc, Adj.Amount\_CSA, p\_last\_modify\_page, p\_worker\_id)

Adj is p\_csa\_sale\_adjustments

To update, if row change is less than 0, return -1.

#### GetTaxRate

This will only return the tax rate as a decimal from DataBase using DAL function GetTaxRate

public Task<Decimal?> GetTaxRate (int p\_property\_id, DateTime p\_event\_date)

{

// GetBaseYear is a helper function, check above for its implementation

    int baseYear = DateHelper.GetBaseYear(p\_event\_date);

    decimal? taxRate = await \_CSASharedDAL.GetTaxRate(propertyID, eventDate, baseYear);

    return taxRate;

}

#### GetPropChar

Await and return **PropChar** entity with GetPropChar function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<PropChar>GetPropChar (int p\_csa\_id)

#### GetIncomeExpenseAnalysis

Await and return **IncomeExpenseAnalysis** entity with GetIncomeExpenseAnalysis function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<GetIncomeExpenseAnalysis>GetIncomeExpenseAnalysis (int p\_csa\_id)

#### GetCSAExpense

Await and return **CSAExpenseEntity** entity withGetCSAExpense function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<CSAExpenseEntity> GetCSAExpense (int p\_csa\_id)

#### GetExpensesForAllExpenseCodes

Await and return list of **CSAExpense** model withGetExpensesForAllExpenseCodes function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<List<CSAExpense>> GetExpensesForAllExpenseCodes(int p\_csa\_id)

Iterate over each CSAExpenseCode entity and assign it to corresponding variable in the CSAExpense model.

#### GetActualExpenses

Await and return list of **CSAExpense** model withGetActualExpenses function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<List<CSAExpense>> GetActualExpenses(int p\_csa\_id)

Iterate over each CSAExpenseCode entity and assign it to corresponding variable in the CSAExpense model.

#### GetCSARptSource

Await and return **RptSource** entity withGetCSARptSource function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<RptSource> GetCSARptSource(int p\_csa\_id)

#### GetCSARptAdjustments

Await and return **CSARptAdjustments** entity withGetCSARptAdjustments function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<CSARptAdjustments> GetCSARptAdjustments(int p\_csa\_id)

#### GetAdjustmentsForAllAdjustmentCodes

Await and return a list of **CSAAdjustment** model withGetAdjustmentsForAllAdjustmentCodes function in DAL layer. Use Try-Catch block to catch any exceptions.

Public Task<AdjustmentsForAllAdjustmentCodes> GetAdjustmentsForAllAdjustmentCodes (int p\_csa\_id)

### CSAProperty class

#### GetPropertyInfo

Await and return **PropertyInfo** entity withGetPropertyInfo function in DAL layer. Use Try-Catch block to catch any exceptions.

Task<CSAProperty> GetPropertyInfo (int p\_csa\_id)

This is the model to return, CSAProperty

CSAProperty \_CSAProperty = new CSAProperty();

PropertyInfo \_PropertyInfo = null;

Property info will be retrieved and assigned its value to CSAProperty model.

try

{

\_PropertyInfo = \_CSAPropertyDAL.GetPropertyInfo();

}

catch

{

throw;

}

if (\_PropertyInfo != null)

{

\_CSAProperty.doc\_prefix = \_PropertyInfo.DOC\_PREFIX;

\_CSAProperty.print\_parcel = \_PropertyInfo.PRINT\_PARCEL;

For CSAProperty entity returned, make sure to check, IND\_PUR\_PRICE, ADDRESS, SITUS\_CITY\_NAME, SITUS\_STATE, ZIP\_CD of PropertyInfo entity are not null. Then assign them to corresponding fields of the CSAProperty model.

}

#### GetMailingNameAsOfDate

Await and return **MailingName** entity withGetMailingNameAsOfDate function in DAL layer. Use Try-Catch block to catch any exceptions.

Task<MailingName> GetMailingNameAsOfDate (int p\_csa\_id, DateTime p\_event\_ts)

\_CSAPropertyDAL.GetMailingNameAsOfDate(p\_csa\_id, p\_event\_ts)

#### RefreshPropChar

Call below function in DAL layer, in CSAProperty class, add exception block

public Task<bool> RefreshPropChar(string DOC\_PREFIX, int DOC\_SERIES, int PRIMARY\_PROP\_ID, short? CsaSelectType, char? CsaAction = null, ref int CsaId, string WorkerId, string UserId, DB2Transaction P\_transaction = null)

#### GetAllAPNinDoc

Await and return a list of **APN** entity withGetAllAPNinDoc function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<List<APN>> GetAllAPNinDoc(int p\_csa\_id)

//Call GetAllAPNinDoc in DAL layer and return its result, remember to use try-catch block, take structure above as a reference

### CSALease class

#### SetCSAPctOccupancy

This function is used to update pct occupancy, property NRA associated with given CSA ID, together with worker ID to specify who made the update. Below are necessary steps:

**Update total NRA**: First using DAL function GetTotalNRA with CSA ID, remember to convert the string to decimal as shown below then store it in proper variable.

**Update pct occupancy**: Include some necessary calculation. First retrieve and convert Square Feet Leased to a decimal (Rememer to trim the string pulled from database first before convert it to decimal). If property NRA we previously got is greater than 0, and square feet leased is greater than property NRA, set pct occupancy to 100 directly. Otherwise, calculating it using: \_pctOccupancy = (\_sqFtLeased / \_propNRA) \* 100, remember to round result to integer. Finally use UpdatePctOccupNRA to update database and return row changes.

public Task<int> SetCSAPctOccupancy(int p\_csa\_id, string p\_worker\_id)

{

    string strTotalNRA;

    int \_rc;

    decimal \_propNRA, tryDecimal;

    decimal \_sqFtLeased;

    string \_sqFtLeasedStr;

Occupancy \_Occupancy = new Occupancy

{

CSA\_ID = p\_csa\_id,

WORKER\_ID = p\_worker\_id

};

Calculation for the value of PCT\_OCCUPANCY

try

{

    strTotalNRA = \_CSALeaseDAL.GetTotalNRA(p\_csa\_id);

}

Catch

{

Throw;

}

    if (decimal.TryParse(strTotalNRA, out tryDecimal))

    {

        \_propNRA = tryDecimal;

    }

    \_sqFtLeasedStr = \_CSALeaseDAL.GetSqFtLeasedByCSAId(p\_csa\_id);

    if (\_sqFtLeasedStr.Trim().Length > 0)

    {

        \_sqFtLeased = Convert.ToDecimal(\_sqFtLeasedStr);

    }

    if (\_propNRA > 0)

    {

        if (\_sqFtLeased > \_propNRA)

        {

            \_Occupancy .PCT\_OCCUPANCY = 100;

        }

        else

        {

            \_Occupancy .PCT\_OCCUPANCY = (\_sqFtLeased / \_propNRA) \* 100;

            \_Occupancy .PCT\_OCCUPANCY = Math.Round(\_Occupancy .PCT\_OCCUPANCY.Value, 0);

        }

    }

Prevent the update of the CSA\_NET\_RENT\_AREA by assigning it to null

\_Occupancy.CSA\_NET\_RENT\_AREA = null;

If PCT\_OCCUPANCY never is assigned a value, assign DBNull to it.

if (!\_Occupancy.PCT\_OCCUPANCY)

\_Occupancy.PCT\_OCCUPANCY = DBNull;

Use injected CSALeaseDAL service to complete the update

    \_rc = \_CSALeaseDAL.UpdatePctOccupNRA(\_Occupancy);

    return \_rc;

}

#### GetLeaseInfo

Await and return **LeaseInfo** entity with GetLeaseInfo function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<List<LeaseInfo>> GetLeaseInfo(int p\_csa\_id)

#### DeleteCSALease

Used to call DAL layer function to delete lease entry, remember to use try-catch block to catch any exceptions.

public Task<int>DeleteCSALease (int p\_lease\_id)

*GetComments*

This will return a list of comments relating to given lease id using GetComments(int p\_lease\_id) function in LeaseDAL.

public Task<List<LeaseComment>> GetComments(int p\_lease\_id)

### CSASource Class

#### UpdateCSASourceData

This is used to update CSA Source Data. Update is performed with UpdateCSASourceData in the DAL layer. Use this as example to how update method should be called within try catch block

public Task<int> UpdateCSASourceData(CSAUpdatedSourceData updated\_source\_data)

{

int rc = 0;

try

{

await rc = \_sourceInputDAL.UpdateCSASourceData(updated\_source\_data);

}

catch

{

throw;

}

    return rc;

}

#### GetVacantLandComments

Await and return a list of **VacantLandComment** entity withGetVacantLandComments function in DAL layer. Use Try-Catch block to catch any exceptions.

public Task<List<VacantLandComment>>GetVacantLandComments(int p\_csa\_id)

List<VacantLandComment> comments = null;

try

{

comments = await \_CSALeaseDAL.GetVacantLandComments (p\_csa\_id);

}

catch

{

throw;

}

return comments;

#### GetTotalRent

Await and return **RentInfo** entity withGetTotalRent function in DAL layer. RentInfo entity contains but not only contains total rent value. Use Try-Catch block to catch any exceptions.

public Task<RentInfo> GetTotalRent (int p\_csa\_id)

#### GetTotalRent

Await and return **RentInfo** entity withGetTotalRent function in DAL layer. Different from original is its getting ther result WITHOUT the input that has matching p\_excluded\_lease\_id. RentInfo entity contains but not only contains total rent value. Use Try-Catch block to catch any exceptions.

public Task<RentInfo> GetTotalRent (int p\_csa\_id, int p\_excluded\_lease\_id)

#### GetSource

Await and return **CSASourceData** entity withGetSource function in DAL layer, SourceInputDAL Class. Use Try-Catch block to catch any exceptions.

public Task<CSASource> GetSource(int p\_csa\_id)

#### GetWorker

Await and return **WorkerInfo** entity withGetWorker function in DAL layer, SourceInputDAL Class. Use Try-Catch block to catch any exceptions.

public Task<List<WorkerInfo>> GetWorker()

### CSAShared Class

#### GetPageTitleByCSAType

Await and return **CD\_LONG\_NAME in IE\_FIELD\_CODE** withGetPageTitle function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<string> GetPageTitleByCSAType(int p\_csa\_type)

#### GetNRAOccupancy

Await and return **NRAOccupancy** entity withGetNRAOccupancy function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<NRAOccupancy> GetNRAOccupancy(int p\_csa\_id)

#### GetCSAComments

Await and return **CSAComments** entity withGetCSAComments function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<List<CSAComments>> GetCSAComments(int p\_csa\_id)

// await and return \_CSASharedDAL.GetCSAComments (int p\_csa\_id). Use Try-Catch block to catch any exceptions.

#### GetFieldCode

Await and return **FieldCode** entity withGetFieldCode function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<List<FieldCode>> GetFieldCode(int p\_field\_id)

#### GetAnticipatedUseCodesNotForLandUse

Await and return **AntiUseCodesNotForLandUse** entity withGetAnticipatedUseCodesNotForLandUse function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<AntiUseCodesNotForLandUse> GetAnticipatedUseCodesNotForLandUse()

#### GetLeaseUseCategories

Await and return a list of **LeaseUseCategory** entities withGetLeaseUseCategories function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<List<LeaseUseCategory>> GetLeaseUseCategories ()

#### GetLeaseUseSubByUseCat

Await and return a list of **LeaseUseSubCat** entities withGetLeaseUseCategories function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<List<LeaseUseSubCat>> GetLeaseUseSubByUseCat (int p\_use\_cat)

#### UpdateCSAComments

Await and return a integer to show row affected withUpdateCSAComments function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> UpdateCSAComments (CSAComment source\_data)

#### InsertCSAComments

Await and return a integer to show insert successful status withInsertCSAComments function in DAL layer, CSASharedDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> InsertCSAComments (CSAComment source\_data)

### IncAnalysisValIndBLL Class

#### GetAdjustCSAAmount

Await and return **decimal value** withGetAdjustCSAAmount function in DAL layer, IncAnalysisValIndDAL Class. Use Try-Catch block to catch any exceptions.

public Task<decimal> GetAdjustCSAAmount(int p\_csa\_id)

#### GetIncomeAnalysis

UseGetIncomeAnalysis, GetTaxRate, GetAdjustCSAAmount, GetPctOccupThreshId, GetSqFtLeased, GetPotentialGrossIncome, CalcVacCollLoss, GetExpenseTotal functions in DAL layer to get all necessary data for either directly assignment or calculation to fill in all data of the IncAnalysisValInd model. Use Try-Catch block to catch any exceptions. The goal of this function is to making all necessary calculations and varifications to output a completed **IncAnalysisValInd** model

public Task<IncAnalysisValInd> GetIncomeAnalysis(int p\_csa\_id)

In this layer, do all necessary calculations using IncomeAnalysis entity returned from data base for IncAnalysisValInd. Finally return IncAnaylsisValInd as final result.

Incomeanalysis is used to stored data pulled from DB, IncAnalysisValInd is used to store data that will be returned as a response from API

IncAnalysisValInd \_IncAnalysisValInd = new IncAnalysisValInd();

IncomeAnalysis \_IncomeAnalysis;

try

{

\_IncomeAnalysis = await \_IncAnalysisValIndDAL.GetIncomeAnalysis(p\_csa\_id);

}

catch

{

throw;

}

Make sure to also check if IncomeAnalysis is not null. Otherwise don’t proceed with any of the following step

Make variables nullable before assigning them values from IncomeAnalysis entity. As below example shows

if(IncomeAnalysis.CSA\_ID != null)

\_IncAnalysisValInd.CSA\_ID = IncomeAnalysis.CSA\_ID

deciaml? \_pct\_pgi = IncomeAnalysis.PCT\_PGI;

TODO: ADD THE REST OF VARIABLES IN SIMILAR FASHION: addition\_net\_incom, tenant\_reimburse, anti\_property\_tax, other\_tax\_desc, other\_tax\_amount, \_pct\_occupancy, \_full\_expenses\_fl, \_tot\_net\_rent\_area, \_tot\_building\_area, \_benchmark\_rate\_cd, PropertyID, EventTs, TaxRate, \_ind\_pur\_price, \_adj\_sales\_price As decimal?, \_csa\_net\_rent\_area, \_bm\_desc, \_OAR\_desc type should all be nullable, type details can be checked in Entity section’s second column for detailed specification

Additional initializations:

decimal? \_expenses\_owner = null;

decimal? \_expenses\_tenant = null;

decimal? \_PGI = null;

decimal \_vac\_coll\_loss = 0;

decimal \_EGI;

decimal \_NOI;

decimal? \_rent\_NRA = null;

decimal? \_NOI\_NRA = null;

decimal? \_owner\_expenses\_NRA = null;

int \_expenses\_total;

decimal? \_owners\_expenses\_total = null;

decimal? \_total\_expenses\_NRA = null;

decimal? \_min\_occupancy = null;

int? \_tot\_Lease\_NRA = null;

try

{

\_IncAnalysisValInd.tax\_rate = await \_CSASaleBLL.GetTaxRate(\_IncAnalysis.PropertyID, \_IncAnalysis.Event);

}

catch

{

throw;

}

Get Adjustment Amount CSA

decimal? \_AdjustCSAAmount = null;

try

{

\_AdjustCSAAmount = await GetAdjustCSAAmount(p\_csa\_id);

}

catch

{

throw;

}

if (\_AdjustCSAAmount != null)

{

\_IncAnalysisValInd.adj\_csa\_amt = \_AdjustCSAAmont;

}

Compute the Adj Sale Price

decimal? \_adj\_sales\_price = null;

decimal \_min\_occupancy = 0.0;

if (\_AdjustCSAAmount.ind\_pur\_price.HasValue && \_AdjustCSAAmount .adj\_csa\_amt.HasValue)

{

    \_adj\_sales\_price = \_ind\_pur\_price + \_adj\_csa\_amt;

}

else if (\_ind\_pur\_price.HasValue && !\_adj\_csa\_amt.HasValue)

{

    \_adj\_sales\_price = \_ind\_pur\_price.Value;

}

else if (!\_ind\_pur\_price.HasValue && \_adj\_csa\_amt.HasValue)

{

    \_adj\_sales\_price = \_adj\_csa\_amt.Value;

}

\_IncAnalysisValInd.cmp\_adj\_sale\_price = \_adj\_sales\_price;

Get percent occupancy threshold from Global Control Table

try

{

\_IncAnalysisValInd.cmp\_pct\_occup\_threshld = await IncAnalysisValIndDAL.GetPctOccupThreshld();

}

catch

{

throw;

}

If IncomeAnalysis.pct\_occup\_minimum has value, then assign its value to \_IncAnalysisValInd.pct\_occup\_minimun directly

Otherwise, if pct\_occup\_threshld has value, then assign that value to \_IncAnalysisValInd.pct\_occup\_minimun instead

If \_IncAnalysisValInd.pct\_occup\_minimun has value, assign this value to \_min\_occupancy

string \_tmp\_tot\_lease\_nra

try

{

\_tmp\_tot\_lease\_nra = await GetSqFtLeased(p\_csa\_id);

}

catch

{

throw;

}

Validate \_tmp\_tot\_lease\_nra has value and parse it to int and assign it to \_IncAnalysisValInd.cmp\_tot\_lease\_nra

Decimal? \_PGI = null

PotentialGrossIncome \_PotentialGrossIncome = null;

try

{

\_PotentialGrossIncome = await GetPotentialGrossIncome(p\_csa\_id);

}

catch

{

throw;

}

decimal \_tot\_current\_rent = 0.0;

decimal \_tot\_anti\_rent = 0.0;

for tot\_current\_rent and tot\_anti\_rent, assign their value to \_tot\_current\_rent and \_tot\_anti\_rent, otherwise leave them be as 0

decimal \_tot\_rent = \_tot\_current\_rent + \_tot\_anti\_rent;

if (\_tot\_rent == 0)

\_PGI = null;

else

\_PGI = Round((\_tot\_rent \* 12), 0)

if (\_PGI != null)

\_IncAnalysisValInd.cmp\_pot\_gross\_inc = \_PGI;

decimal \_vac\_call\_loss = 0.0;

decimal \_vac\_call\_loss = 0.0;

decimal? \_pct\_pgi = null;

\_vac\_coll\_loss = CalcVacCollLoss(\_PGI, \_IncomeAnalysis.pct\_pgi);

\_IncAnalysisValInd.Cmp\_Vac\_Coll\_Loss = \_vac\_coll\_loss;

PGI %

\_IncAnalysisValInd.Pct\_PGI = \_pct\_pgi

Additional Net Income

\_IncAnalysisValInd.addition\_net\_incom = \_addition\_net\_incom

Tenant Expense Reimbursement

\_IncAnalysisValInd.tenant\_reimburse = \_tenant\_reimburse

Anticipated Property Tax

\_IncAnalysisValInd.anti\_property\_tax = \_anti\_property\_tax

if(!IsNullOrEmpty(IncomeAnalysis.other\_taxes\_desc))

\_IncAnalysisValInd.Other\_Taxes\_Desc = \_IncomeAnalysis.other\_taxes\_desc.Trim();

Other Tax Amount

\_IncAnalysisValInd.Other\_Taxes\_Amount = \_other\_taxes\_amount;

Benchmark Rating code

\_IncAnalysisValInd.Benchmark\_Rate\_Cd = \_benchmark\_rate\_cd;

Decimal \_PGIwork = 0.0;

if (\_PGI != null)

\_PGIwork = \_PGI;

if (\_addition\_net\_income == null)

\_addition\_net\_incom = 0;

if (\_tenant\_reimburse == null)

\_tenant\_reimburse = 0;

decimal \_EGI = 0.0;

Effective Gross Income is calculated and stored.

\_EGI = (\_PGIwork - \_IncAnalysisValInd.Cmp\_Vac\_Coll\_Loss + \_addition\_net\_incom.Value + \_tenant\_reimburse)

\_IncAnalysisValInd.Cmp\_Eff\_Gross\_Inc = \_EGI

Decimal \_expenses\_owner = 0;

Decimal \_expenses\_tenant = 0;

ExpenseTotal \_ExpenseTotal = GetExpenseTotal(p\_csa\_id);

if (\_ExpenseTotal != null)

{

if (\_ExpenseTotal.TOT\_EXPENSE\_OWNER != null)

\_expenses\_owner = \_ExpenseTotal.TOT\_EXPENSE\_OWNER;

if (\_ExpenseTotal.TOT\_EXPENSE\_OWNER != null)

\_expenses\_tenant = \_ExpenseTotal.TOT\_EXPENSE\_TENANT;

}

decimal \_NOI = 0;

//Calculation for Net\_Op\_Inc

\_NOI = (\_EGI - \_owners\_expenses\_total.Value - \_anti\_property\_tax.Value - \_other\_taxes\_amount.Value);

\_IncAnalysisValInd.cmp\_net\_op\_inc = \_NOI;

decimal? \_EGIpct = null;

if (\_EGI > 0)

{

    \_EGIpct = Math.Round((\_owners\_expenses\_total.Value / \_EGI) \* 100, 2);

    \_EGIpct = Math.Round((\_expenses\_owner.Value / \_EGI) \* 100, 2);

}

\_IncAnalysisValInd.Cmp\_EGI\_Pct = \_EGIpct;

Rent/NRA (Net Rentable Area) - if tot\_net\_rent\_area > 0 (PGI / NRA tot\_net\_rent\_area)

elseif tot\_building\_area(gross sq ft) > 0 (PGI / tot\_building\_area)

int? \_csa\_net\_rent\_area = \_IncomeAnalysis.csa\_net\_rent\_area;

for rent NRA calculation

if (\_PGI.HasValue)

{

    if (\_csa\_net\_rent\_area.HasValue)

    {

        if (\_csa\_net\_rent\_area.Value > 0)

        {

            \_rent\_NRA = Math.Round((\_PGI.Value / \_csa\_net\_rent\_area.Value), 2);

        }

        else if (\_tot\_building\_area.HasValue)

        {

            if (\_tot\_building\_area.Value > 0)

            {

                \_rent\_NRA = Math.Round((\_PGI.Value / \_tot\_building\_area.Value), 2);

            }

        }

    }

    else if (\_tot\_building\_area.HasValue)

    {

        if (\_tot\_building\_area.Value > 0)

        {

            \_rent\_NRA = Math.Round((\_PGI.Value / \_tot\_building\_area.Value), 2);

        }

    }

}

\_IncAnalysisValInd.Cmp\_Rent\_NRA = \_rent\_NRA;

//This include NOI\_NRA

if (\_csa\_net\_rent\_area.HasValue)

{

    if (\_csa\_net\_rent\_area.Value > 0)

    {

        \_NOI\_NRA = Math.Round((\_NOI / \_csa\_net\_rent\_area.Value), 2);

        \_owner\_expenses\_NRA = Math.Round((\_owners\_expenses\_total.Value / \_csa\_net\_rent\_area.Value), 2);

        \_total\_expenses\_NRA = Math.Round((\_expenses\_total / \_csa\_net\_rent\_area.Value), 2);

    }

    else if (\_tot\_building\_area.HasValue)

    {

        if (\_tot\_building\_area.Value > 0)

        {

            \_NOI\_NRA = Math.Round((\_NOI / \_tot\_building\_area.Value), 2);

            \_owner\_expenses\_NRA = Math.Round((\_owners\_expenses\_total.Value / \_tot\_building\_area.Value), 2);

            \_total\_expenses\_NRA = Math.Round((\_expenses\_total / \_tot\_building\_area.Value), 2);

        }

    }

}

else if (\_tot\_building\_area.HasValue)

{

    if (\_tot\_building\_area.Value > 0)

    {

        \_NOI\_NRA = Math.Round((\_NOI / \_tot\_building\_area.Value), 2);

        \_owner\_expenses\_NRA = Math.Round((\_owners\_expenses\_total.Value / \_tot\_building\_area.Value), 2);

        \_total\_expenses\_NRA = Math.Round((\_expenses\_total / \_tot\_building\_area.Value), 2);

    }

}

\_IncAnalysisValInd.Cmp\_NOI\_NRA = \_NOI\_NRA;

\_IncAnalysisValInd.Cmp\_Expense\_NRA = \_owner\_expenses\_NRA;

\_IncAnalysisValInd.Cmp\_Tot\_Xpens\_NRA = \_total\_expenses\_NRA;

NOI / Adj Sale Price

decimal? \_noi\_AdjSalePrice = null;

if (\_adj\_sales\_price.HasValue)

{

    if (\_adj\_sales\_price.Value > 0)

    {

        \_noi\_AdjSalePrice = Math.Round((\_NOI / \_adj\_sales\_price.Value) \* 100, 2);

    }

}

\_IncAnalysisValInd.Cmp\_NOI\_Adj\_Sale = \_noi\_AdjSalePrice;

Adj Sale Price / PGI

decimal? \_ASP\_PGI = null;

if (\_adj\_sales\_price.HasValue && \_PGI.HasValue)

{

    if (\_PGI.Value > 0)

    {

        \_ASP\_PGI = Math.Round((\_adj\_sales\_price.Value / \_PGI.Value), 1);

    }

}

\_IncAnalysisValInd.Cmp\_Adj\_Sale\_PGI = \_ASP\_PGI;

Adj Sale Price / NRA

decimal? \_ASP\_NRA = null; // Nullable Decimal

if (\_adj\_sales\_price.HasValue)

{

    if (\_csa\_net\_rent\_area.HasValue)

    {

        if (\_csa\_net\_rent\_area.Value > 0)

        {

            \_ASP\_NRA = (int)(\_adj\_sales\_price.Value / \_csa\_net\_rent\_area.Value);

        }

        else if (\_tot\_building\_area.HasValue && \_tot\_building\_area.Value > 0)

        {

            \_ASP\_NRA = (int)(\_adj\_sales\_price.Value / \_tot\_building\_area.Value);

        }

    }

    else if (\_tot\_building\_area.HasValue && \_tot\_building\_area.Value > 0)

    {

        \_ASP\_NRA = (int)(\_adj\_sales\_price.Value / \_tot\_building\_area.Value);

    }

}

\_IncAnalysisValInd.Cmp\_Adj\_Sale\_NRA = \_ASP\_NRA;

OAR - default is NO, YES if full\_expenses\_fl is 'Y' and \_PGI => 0

string \_OAR\_desc = "No";

if (!\_min\_occupancy.HasValue)

{

    \_min\_occupancy = 0;

}

if (\_PGI.HasValue && \_full\_expenses\_fl != null && \_pct\_occupancy.HasValue)

{

    if (\_full\_expenses\_fl == "Y" && \_PGI.Value >= 0 && \_pct\_occupancy.Value >= \_min\_occupancy.Value)

    {

        \_OAR\_desc = "Yes";

    }

}

\_IncAnalysisValInd.cmp\_OAR\_desc = \_OAR\_desc;

return \_IncAnalysisValInd;

#### CalcVacCollLoss

Vacancy and Coll Loss calculation method.

public decimal CalcVacCollLoss(decimal? PGI, decimal? Pct\_PGI)

{

decimal \_Vac\_Coll\_Loss = 0;

if (PGI.HasValue && Pct\_PGI.HasValue)

{

\_Vac\_Coll\_Loss = Math.Round((PGI.Value \* (Pct\_PGI.Value / 100)), 0);

}

return \_Vac\_Coll\_Loss;

}

#### GetExpenseTotal

Await and return **ExpenseTotal** entity withGetExpenseTotal function in DAL layer, IncAnalysisValIndDAL Class. Use Try-Catch block to catch any exceptions.

public Task<ExpenseTotal> GetExpenseTotal(int p\_csa\_id)

Use try catch block to call GetExpenseTotal from DAL layer.

#### GetPotentialGrossIncome

Await and return **PotentialIncome** entity withGetPotentialGrossIncome function in DAL layer, IncAnalysisValIndDAL Class. Use Try-Catch block to catch any exceptions.

public Task<PotentialIncome> GetPotentialGrossIncome(int p\_csa\_id)

Use try catch block to call GetPotentialIncome from DAL layer.

#### GetSourceDesc

Await and return **SourceDesc** entity withGetSourceDesc function in DAL layer, IncAnalysisValIndDAL Class. Use Try-Catch block to catch any exceptions.

Note both normal source will just use cd\_short\_name, with exception for BenchmarkRatings, in this case it will require cd\_long\_name.

public Task<SourceDesc> GetSourceDesc(int p\_field\_id)

Use try catch block to call GetSourceDesc from DAL layer.

#### UpdateIncomeAnalysis

Await and return **row change** with UpdateIncomeAnalysisfunction in DAL layer, IncAnalysisValIndDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> UpdateIncomeAnalysis(UpdatedIncomeAnalysis p\_income\_analysis, string p\_last\_modified\_page, string p\_worker\_id)

update using two DAL functions

try

{

UpdateIncomeAnalysis(p\_income\_analysis)

}

catch

{

throw;

}

try

{

UpdateCSASaleInfoData(p\_income\_analysis.csa\_id, p\_income\_analysis.benchmark\_rate\_cd, p\_income\_analysis.worker\_id)

}

catch

{

throw;

}

### CSAVacantLandInput Class

#### GetVacantLandComments

Use GetCSAComments, they suppose to do exactly same thing

#### UpdateVacantLandComments

This is used to update existing comment inside the database

public Task<int> UpdateVacantLandComments(VacantLandComments comments, string p\_last\_modify\_page, string p\_entry\_worker)

#### InsertVacantLandComments

This is used to add new comments entry into the database

public Task<int> InsertVacantLandComments(VacantLandComments comments, string p\_last\_modify\_page, string p\_entry\_worker)

### CSALeaseInfo Class

#### GetLease

Await and return **Lease** with GetLeasefunction in DAL layer, LeaseInfoDAL Class. Use Try-Catch block to catch any exceptions.

public Task<LeaseBO> GetLease (int p\_lease\_id)

#### InsLeaseNoRentComp

Await and return **row change** with InsLeaseNoRentCompfunction in DAL layer, LeaseInfoDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> InsLeaseNoRentComp (LeaseBO p\_lease\_bo)

If p\_lease\_bo.csa\_id.HasValue()

required to update CSA Percentage occupancy too if csa id is present

\_CSALeaseBLL.SetCSAPctOccupancy(p\_lease\_bo.csa\_id, p\_lease\_bo.entry\_worker)

#### InsertLease

Await and return **row change** with InsertLeasefunction in DAL layer, LeaseInfoDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> InsertLease (LeaseBO p\_lease\_bo)

If p\_lease\_bo.csa\_id.HasValue()

required to update CSA Percentage occupancy too if csa id is present

\_CSALeaseBLL.SetCSAPctOccupancy(p\_lease\_bo.csa\_id, p\_lease\_bo.entry\_worker)

#### UpdLeaseNoRentComp

Await and return **row change** with UpdLeaseNoRentCompfunction in DAL layer, LeaseInfoDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> UpdLeaseNoRentComp (LeaseBO p\_lease\_bo)

If p\_lease\_bo.csa\_id.HasValue()

required to update CSA Percentage occupancy too if csa id is present

\_CSALeaseBLL.SetCSAPctOccupancy(p\_lease\_bo.csa\_id, p\_lease\_bo.entry\_worker)

#### UpdateLease

Await and return **row change** with UpdateLeasefunction in DAL layer, LeaseInfoDAL Class. Use Try-Catch block to catch any exceptions.

public Task<int> UpdateLease (LeaseBO p\_lease\_bo)

If p\_lease\_bo.csa\_id.HasValue()

required to update CSA Percentage occupancy too if csa id is present

\_CSALeaseBLL.SetCSAPctOccupancy(p\_lease\_bo.csa\_id, p\_lease\_bo.entry\_worker)

### wkSheetNewImps Class

#### GetAPNs

public Task<List<wkSheetNewImp>> GetAPN (int p\_csa\_wks\_num)

#### GetAPNList

public Task<List<wksAPN>> GetAPNList (int p\_csa\_wks\_num)

#### UpdateCSAAPN

Call below function in DAL layer, in CSAProperty Class, add exception block

public Task<int> UpdateCSAAPN(int csa\_wks\_num, int prop\_id\_prime, string[] \_subjectPropID, string entry\_worker)

## API Controllers

### For CI Sales Analysis (Controllers => CISalesAnalysisController.cs)

|  |  |  |
| --- | --- | --- |
| Call Type | Call Name | SP to Use |
| *GET* | *GetPageTitleByCSAType* | IE\_GET\_CSA\_PAGE\_TITLE |
| *GET* | *GetSaleInfoFull* | IE\_GET\_CSA\_SALE\_INFO\_FULL,  IE\_GET\_CSA\_PROP\_INFO, IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE |
| *PATCH* | *UpdateSaleInfoWithCIAnalysis* | IE\_DEL\_CSA\_LEASE, |
| *POST* | *SetCSAPctOccupancy* | IE\_GET\_CSA\_TOTAL\_NRA, IE\_GET\_CSA\_SQFT\_LEASED, |
| *GET* | *GetSaleExpenses* | IE\_GET\_CSA\_SALE\_INFO\_FULL,  IE\_GET\_CSA\_EXPENSE\_CODE |
| *GET* | *GetExpenses (ADDED)* | IE\_GET\_CSA\_EXPENSE\_CODE ||  IE\_GET\_CSA\_AMOUNT\_ACTUAL\_EXPENSE |
| *PATCH* | *UpdateExpensesInSaleExpenses* |  |
| *GET* | *GetSaleAdjustments* | IE\_GET\_CSA\_PROP\_INFO |
| *PATCH* | *UpdateAdjustmentsInSaleAdjustments* | IE\_GET\_CSA\_AMOUNT, |
| *PUT* | *UpdateCSASourceData* |  |
| *GET* | *GetMailingNameAsOfDate* | IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE |
| *GET* | *GetTaxRate* | IE\_GET\_CSA\_TAX\_RATE |
| *GET* | *GetPageTitle* | IE\_GET\_CSA\_FEILD\_CODE |
| *GET* | *GetLeaseInfo* | IE\_GET\_CSA\_LEASE\_INFO |
| *GET* | *GetTotalRent* | IE\_GET\_CSA\_TOTAL\_RENT |
| *GET* | *GetNRAOccupancy* | IE\_GET\_CSA\_NRA\_OCCUPANCY |
| *DELETE* | *DeleteCSALease* | IE\_DEL\_CSA\_LEASE |
| *GET* | *GetPropertyInfo* | IE\_GET\_CSA\_PROP\_INFO |
| *GET* | *GetAllAPNinDoc* | IE\_GET\_CSA\_ALL\_APN\_IN\_DOC |
| *GET* | *GetPropChar* | IE\_GET\_CSA\_PROP\_CHAR |
| *GET* | *GetIncomeExpenseAnalysis* | IE\_GET\_CSA\_INCOME\_EXPENSE\_ANALYSIS |
| *GET* | *GetCSAExpense* | IE\_GET\_CSA\_EXPENSE |
| *GET* | *GetCSARptSource* | IE\_GET\_CSA\_RPT\_SOURCE |
| *GET* | *GetCSARptAdjustments* | IE\_GET\_CSA\_RPT\_ADJ |
| *GET* | *GetComments* | IE\_GET\_CSA\_COMMENTS *OR*  IE\_GET\_CSA\_LEASE\_COMMENTS |
| *PUT* | *UpdateCSAComments* |  |
| *POST* | *InsertCSAComments* |  |
| *GET* | *GetAdjustmentsForAllAdjustmentCodes* | IE\_GET\_CSA\_ADJ\_FOR\_ALL\_ADJ\_CODES |
| *GET* | *GetFieldCodeCd* | IE\_GET\_FIELD\_CODE |
| *GET* | *GetPageTitle* | IE\_GET\_CSA\_PAGE\_TITLE |
| *GET* | *GetAnticipatedUseCodesNotForLandUse* | IE\_GET\_CSA\_ANTICIPATED\_USE\_CODES\_NOT\_FOR\_LAND\_USE |
| *GET* | *GetAllAPNinDoc* | IE\_GET\_CSA\_ALL\_APN\_IN\_DOC |
| *GET* | *GetSource* | IE\_GET\_CSA\_SOURCE |
| *GET* | *GetIncomeAnalysis* | IE\_GET\_CSA\_INCOME\_ANALYSIS,  IE\_GET\_CSA\_TAX\_RATE,  IE\_GET\_CSA\_PCT\_OCCUP\_THRESHLD,  IE\_GET\_CSA\_EXPENSE\_TOTAL,  IE\_GET\_CSA\_BENCHMARK\_RATING\_DESC |
| *PATCH* | *UpdateIncomeAnalysis* |  |
| *GET* | *GetExpenseTotal* | IE\_GET\_CSA\_EXPENSE\_TOTAL |
| *GET* | *GetPotentialGrossIncome* | IE\_GET\_CSA\_POTENTIAL\_INCOME |
| *GET* | *GetBenchmarkRating* | IE\_GET\_FIELD\_CODE |
| *GET* | *GetSourceDesc* | IE\_GET\_FIELD\_CODE |
| *GET* | *GetWorker* | IE\_GET\_CSA\_WORKER |
| *PATCH* | *ChangeAPN* |  |

### For CSALeaseInfo

|  |  |  |
| --- | --- | --- |
| *GET* | *GetLease* | IE\_GET\_CSA\_LEASE |
| *POST* | *InsLeaseNoRentComp* | IE\_ADD\_CSA\_LEASE\_NO\_RENT\_COMP |
| *POST* | *InsertLease* |  |
| *PUT* | *UpdLeaseNoRentComp* |  |
| *PUT* | *UpdateLease* |  |
| *GET* | *GetLeaseUseCategories* | IE\_GET\_CSA\_LEASE\_USE\_CATEGORY |
| *GET* | *GetLeaseUseSubByUseCat* | IE\_GET\_CSA\_LEASE\_USE\_SUB\_BY\_USE\_CAT |

### For CSAPropertyInfo

|  |  |  |
| --- | --- | --- |
| *GET* | *GetPropertyInfo* | IE\_GET\_CSA\_PROP\_INFO\_VACANT\_LAND |

### For wkSheet

|  |  |  |
| --- | --- | --- |
| *GET* | *GetAPNs* | IE\_GET\_CSA\_WKS\_APN\_LIST |
| *POST* | *UpdateCSAAPN* | IE\_DEL\_CSA\_WKS\_APN |

CSA APIs will be setup under Improve, thus API call route should be like:

*http://<improveurl>/csa/api/[controller]*

For coding example please refer to HTTP Request Life Cycle -> API Controller layer example. Different controllers will be set up as different web components, as CISalesAnalysis page should have its own controller as CISalesAnalysisController.cs. Remember to return the proper HTTP status code in response. As well as applying Asynchronous programming style.

HTTP status code scenarios:

1. 200 No error occurred: Ok(response). *This can be used with a Get request, as well as delete, put and patch. The later should return rc (row changes) as a response.*
2. 201 Created: Created("/api/resource/123", new { /\* data \*/ }). If adding new entry with Post request and record is empty, use no content.
3. 400 Bad request: BadRequest(ModelState), *Use this when model state is not valid (double check incoming data, especially Json can be correctly map to existing model).*
4. 401 Require authentication: StatusCode(401, ex.Message) *Use this if windows authentication fails.*
5. 404 Unauthorized: NotFound(), *Use this when cannot get item with request parameters.*
6. 500 Server error: StatusCode(500, ex.Message), *Use this when server error occurred as under maintenance or is currently down.*
7. 555 General generic exception: StatusCode(500, ex.Message), *Use this when other exceptions are encountered, either from catch block or from middleware pipeline.*

## Report Design and Generation

For report CSA will use previous CSA report design, here there will be two parts to create.

### Server Side:

Here’s how to install report service for Visual Studio 2019: [IMPROVE.NET - SSRS Reports Visual Studio 2019 (sharepoint.com)](https://acgovt.sharepoint.com/sites/IMPROVENET/Wiki/SSRS%20Reports%20Visual%20Studio%202019.aspx)

Graphical user interface, text, application, email

Description automatically generated

Previously created CSA reports after redesign will be placed inside the ImproveReports subfolder. Make sure to import this project from the folder below to save time instead of recreating the whole report again:

Graphical user interface, text, application

Description automatically generated

Once report is imported using existing rdl file. Make sure header is correct well as the rest of the report. Create stored procedures and call them in report to get report required data. Referring to the report section in the CSA Client Side design to locate the stored procedure details. Once every fields in report is confirmed correct. This concludes server side report building. Make sure to upload the updated rdl report using Power BI Report Server’s upload feature, select the rdl file and upload it to the correct environment folder as shown below (Dev env):

Graphical user interface, application

Description automatically generated

### Client Side:

Locate the “Report Handling” section from the [Angular Frontend Design document](https://acgovt.sharepoint.com/:w:/r/sites/IMPROVENET/_layouts/15/Doc.aspx?sourcedoc=%7B007CB72F-F81B-42D5-8E08-71136E12416C%7D&file=CSA%20Frontend%20Framework%20Design.docx&wdLOR=c7BDB4754-4433-41B7-A13A-63C4D523A7A2&action=default&mobileredirect=true).

## Exception handling and error logging

### Exception handling

Try catch block will be set up in the business layer, each time upon calling a function from the DAL layer, make sure any exception will be caught and throw. Also additionally, Improve has already have exception middleware implemented to handle exception from a global scale.

### Error logging

Since CSA APIs will be nested within Improve, thus CSA will directly make use of improve’s exception logging api. Follow the below code example to call upon the API and log errors.

API url: [*https://improveapit.acgov.org/api/ErrorLogging/LogException*](https://improveapit.acgov.org/api/ErrorLogging/LogException)

Use LogExceptionAsync function below with proper parameters to log exceptions in Improve ms sql server database.

public async Task<string> LogExceptionAsync(int applicationId, int logTypeId, string customMessage, string bearerToken)

{

    // Your API endpoint URL

    string apiUrl = "https://improveapinewt.acgov.org/api/ErrorLogging/LogException";

    // Don't forget to inject configuration in constructor

    string bearerToken = \_configuration["BearerTokenSettings:BearerToken"];

    // Prepare the JSON payload

    string jsonPayload = $@"{{

        ""applicationId"": {applicationId},

        ""logTypeId"": {logTypeId},

        ""customMessage"": ""{customMessage}""

    }}";

    // Call the API with bearer token

    HttpResponseMessage response = await PostJsonWithBearerTokenAsync(apiUrl, jsonPayload, bearerToken);

return response.Content.ReadAsStringAsync().Result;

}

public async Task<HttpResponseMessage> PostJsonWithBearerTokenAsync(string apiUrl, string jsonPayload, string bearerToken)

{

HttpResponseMessage response = new HttpResponseMessage();

    using (HttpClient client = new HttpClient())

    {

        try

        {

            // Set the Authorization header with bearer token

            client.DefaultRequestHeaders.Authorization = new AuthenticationHeaderValue("Bearer", bearerToken);

            // Prepare the HTTP content with JSON payload

            StringContent content = new StringContent(jsonPayload, Encoding.UTF8, "application/json");

            // Make the POST request

            response = await client.PostAsync(apiUrl, content);

        }

        catch (Exception ex)

        {

            Console.WriteLine($"Exception: {ex.Message}");

        }

    }

return response;

}

### New Exceptions

#### Database concurrency error code -999

For mod operation, if retMsg is -999 meaning database writing concurrency has happened. Make sure to add and throw this new exception to Improve API.

# CSA Client Side Design

## Core Technologies

Angular Framework 16.0.6

Kendo Telerik UI

Note: For Kendo Telerik, remember to keep the theme consistent with current Improve UI design.

## Angular Components

\*\*\*\*\*\*\*\*\*

**In the CSA tab design, when clicking another tab save function for the current tab will automatically triggered. If there’s any validation error, show error messages together in a pop up menu and prevent page from being swapped to another tab.**

### PBPage (Testing page)

This page is specifically made to generate query string to be used in the Default to page in order to link to the correct landing page. This page is mainly used for testing and debugging.

PBPage is already setup in the code repo.

### Default page (Loading page)

This page is made to take input query string as it will include all necessary information passing from Improve, then the query string will be used to land user in correct CSA page based on CSAID that has been passed in.

Default page is already setup in the code repo.

Remember to store data from query string into local storage to be recalled conveniently in the future.

### ToolBar

#### Edit Button

For this button, it will call function showModifyReason(), as this function will open a pop up window popModifyReason component.

#### Save Button

Save button functionality should be changing based on which tab is currently under selection and perform different api calls as save call of each tab specified.

#### Refresh Button

Refresh button should be perform rebinding based on which tab is under selection. As bind for each tab specified.

### popSubjectAPN page

Bind

GetAllAPNinDoc

Save

Make sure only one row can be highlighted as it should also be the current primary APN

ChangeAPN

### popWksSubjectAPN page

Bind

GetAPN

Save

Make sure only one row can be highlighted as it should also be the current primary APN

UpdateCSAAPN

### popModifyReason page

### CISalesAnalysis page

As CISalesAnaysis page core components will be the RadTags as well as all contents it contains. For most tabs (except for prop char and income) there will be two necessary functions:

1. Bind: This is meant to call upon the Get methods in the APIs in order to fetch corresponding data from the database as well as bind them to variables of web application and display them to client. This should be run automatically upon page load (ngOnInit). As well as a separate button should be available to manually rebind “Refresh” button. In order to manually load updated data to the web application. Note, do not use two-way binding nor display updated calculations directly on the web page. ONLY update when refresh button is clicked or client refresh webpage through browser (bind function is called again).
2. Save: This is meant to call upon the POST, PUT or PATCH method in order to update database info based on current input. Based on each tab, save button in the toolbar should be calling different update api depending on the current tab location. All value to be updated (text box field value) should be initialize as nullable type as well as set their value to null. In the update all values should be updated as is (null or with value using giving stored procedure).

#### ngOnInit(Page\_Load)

Data will be fetched using current session. Properties in session are: CSAID, CSAStartDt, CSAType, CurrTabValue, IEMode, IsNewSession, LseID, LeaseID, LsePID, SessionExist, WorkerID. These properties will either return corresponding session data, or they will return null.

If get to this page the **first time (not isPostBack)** without previous page interaction (like submit a form or click a button in previous page and redirect to current CISalesAnalysis page)

Check whether user coming from an authorized page, see if Session’s N\_CSASTARTDT has value

If it’s not, lead to Not Authorized Error page (code 403)

if coming from authorized page: Assign variable values from current session. Page title is fetched using GetPageTitle which take CSA Type as its parameter.

Page check whether mode is editing mode or not depending on \_IeMode is “E”

If yes, will enter editing mode

if not, will enter read only mode

Checking in current session there’s a selected tab value (CurrTabValue)

If no, then Sale Info tab will be loaded and bind data from data base

If yes, then check the tab value.

If Tab value cannot be find in current tab collection or given index is less than 0, display SaleInfo tab.

Otherwise, bind and display selected tab.

if Get to this page with page refresh in the browser (one of the PostBack situation)

Assign all session data including CurrTabValue, this is why when you refresh the page, the page will **display current tab selection**

Load **selected property information**, by calling CSA property information control (This is should be made into a shared component). Giving CSAId, IEMode, WorkerId, CSAType as its parameter. This is the current top session display all property info like Use Cd, Adj Sales Price etc. Detail please check CSAPropertyInfo.ascx component.

#### Api Calls and Business Logic By Tab

PctOwnerOccup\_ServerValidate, this function needs double checking. It’s basically for a information pops up when percentage is 100

**For *GetFieldCode* method, all code been used are listed here:**

|  |  |  |
| --- | --- | --- |
| Field ID | Items | Tab |
| 120 | Condition at Sale | SaleInfo |
| 118 | Property Use Code | SaleInfo |
| 121 | Benchmark Rating | Income & Value |
| 114 | Primary & Secondary | Sources |
| 130 | CSA Modify Reason | Sources |

##### Sale Info

Graphical user interface, application, Word

Description automatically generated

Bind

GetSaleInfo

Mapping on page:

Benchmark Rating <= benchmard\_rate\_cd\_name

Details <= csa\_prop\_use\_detl

Supervisor Approved <= suprv\_approved\_fl\_text

% Owner Occupied <= pct\_owner\_occup

Broker Involved <= broker\_involved\_fl

Buyer/Seller Relationship <= buy\_sell\_rel\_fl

If so, specify <= buy\_sell\_rel\_desc

Purchase Predated By:

Option <= pur\_predate\_by\_opt\_bool

Contract of Sale <= pur\_pred\_cont\_sale

Contract Date <= predate\_cont\_date, (Format: {0:MM/dd/yyyy})

Condition At Sale <= cond\_at\_sale\_cd

GetAnticipatedUseCodesNotForLandUse

Anticipated Use <= ANTICIPATED\_USE\_CD\_NAME

List<FieldCode> GetFieldCode (int p\_field\_id)

For drop down menu on this tab

List<LeaseInfo> GetLeaseInfo(int p\_csa\_id)

This is used to assign SaleInfoHasIncomesExpenses to true if there’s lease info available.

List<CSAExpense> GetExpenses(int p\_csa\_id, bool expense\_bool)

This is used to check for IncomesExpenses after GetLeaseInfo return an empty list. If any expense available, tenant or owner, assign SaleInfoHasIncomesExpenses to true.

Save

Validation:

If % Owner Occupied == 100

“Income” tab and “Expense” tab and their content should be disabled

else

“Income” tab and “Expense” tab and their content should be enabled.

If “% Owner Occupied” less than 0 or greater than 100

Show error "% Owner Occupied must be between 0 and 100."

If worker ID is nothing **or** just trailing white space

Show error "Worker Id is required."

If “Buyer/Seller Relatioship” is selected “Yes” **then check**

if “if so, specify” text box is empty or just trailing white space

show error "Buyer/Seller Relationship description is required."

else (Buyer/Seller Relationship is empty or “No”)

If “if so, specify” text box is filled with not just empty space

show error "Do not specify Buyer/Seller Relationship if none is indicated."

SaleInfoHasIncomesExpenses: bool = true

If GetLeaseInfo(CSA\_ID) returned list is not empty (have exist lease)

SaleInfoHasIncomesExpenses = true

Else

Call GetExpenses(CSA\_ID, false) api and get a new list of CSAExpense model using current CSA ID, and set its value of ExpensesBool as false. Once received the model, iterate through list of expenses for amount\_owner and amount\_tenant. If a none value is located, SaleInfoHasIncomesExpenses value will be set to true

If SaleInfoHasIncomesExpenses == true && PCT\_OWNER\_OCCUP == 100

error message should be thrown: '% Owner Occupied cannot be 100 if income or expense exists.<BR>Do you want to delete incomes and expenses?',

Update the value as is, if value is not available, assign null to current variable. Also remember to check variables UpdatedSaleInfoWithCIAnalysis entity contains. If there’s value update with value in DB2, other wise assign value in DB2 as DBNull.

int UpdateSaleInfoWithCIAnalysis(UpdatedSaleInfoWithCIAnalysis update)

##### Prop Chraracteristics

Bind

Simply Display all the data with correct name on the web page, no specific logic is included

GetPropChar

Mapping on this tab:

Zoning <= Zoning\_Name

SBE <= Low: SBE\_CLASS\_LOW, High <= SBE\_CLASS\_HI

No of Bldgs <= low: NUM\_BUILDINGS\_LOW, high: NUM\_BUILDINGS\_HI

Gross Sq Ft <= TOT\_BUILDING\_AREA, (Fomat: {0:N0})

MVS <= low: MVS\_CLASS\_LOW, hugh: MVS\_CLASS\_HIGH

Stories <= low: NUM\_STORIES\_LOW, high: NUM\_STORIES\_HI (Format: {0:N1})

Net Rentable Area <= TOT\_NET\_RENT\_AREA, (Format: {0:N0})

Year Build <= low: BLDG\_BUILT\_YR\_LOW, high: BLDG\_BUILT\_YR\_HI

Wall Height <= low: AVG\_ACTUAL\_HT\_LOW, high: AVG\_ACTUAL\_HT\_HI, (Format: {0:N1})

Land Area <= TOT\_LOT\_SIZE, (Format: {0:N0}), unit is Square Ft, make sure to add it after number

Effective Year <= low: BLDG\_EFF\_YR\_LOW, high: BLDG\_EFF\_YR\_HI

Land Imp Ratio <= low: LAND\_IMP\_RATIO\_LOW, high: LAND\_IMP\_RATIO\_HI

Parking <= TOT\_PARKING, (Format: {0:N0})

Percentage Built Out <= low: PCT\_BUILT\_OUT\_LOW, high PCT\_BUILT\_OUT\_HI

FAR <= low: FAR\_LOW, high: FAR\_HI, (Format: {0:N2})

Percentage Office <= low: PCT\_OFFICE\_LOW, high: PCT\_OFFICE\_HI

##### Income

When click on +Add new lease,CSALeaseEdit component should show for client to insert new LeaseInfo

When click on pencil button on the left of existing list, CSALeaseEdit component should appear and showing existing information within current lease id.

When click on trash can button, the lease with current lease id should be deleted using DeleteCSALease(int p\_lease\_id) api.

Bind

GetAllAPNinDoc

GetLeaseInfo,

Check WKS\_EXIST\_FL, if it’s true then the trash can can’t delete lease instead upon click warning message should show:

Delete not allowed.

Lease is referenced in Income Approach worksheet.

Otherwise confirm with this message: Delete this lease?

GetTotalRent

GetNRAOccupancy

Within RentInfo, All bottom row info (total rent, occupancy, total sq ft … ) should be able to find.

Table

Description automatically generated with low confidence

Here’s a break down:

Total Sq Ft: NRA\_SUM

% Occupancy: PCT\_OCCUPANCY

Occupied Sq Ft: OCCUPIED\_SQ\_FT

Total Rent: TOTAL\_RENT

Total Current Rent: CURRENT\_RENT\_SUM

Total Anticipated Rent: ANTICIPATED\_RENT\_SUM

Validation for above income:

When inserting a new lease get current total with:

GetTotalRentByCSAId(p\_csa\_id)

When updating an existing lease (pencil icon), showing total excluding current lease using:

GetTotalRentByCSAId(p\_csa\_id, p\_lease\_id)

When Net Rentable Area is greater than Integer max value or less than Integer Minimum value,

Note this page will have multiple drop down menus, check detailed number requirement on FR

Memo:

GetListOfNoYes

GetListOfYesNo

GetCSALeaseType

GetCSAExpenseType

\*\*\* Issue: when click save twice, two same record will be saved

##### Expenses

Few things to note here, in expense not just expense amount is saved in DB2, but also for expense description. Thus both will need to be displayed on the page while percentage and total is calculated with giving data. The calculation should not refresh itself unless a refresh button is clicked or browser does a page refresh.

Bind

GetSaleExpenses

Note in this page there’s user customized expense. Only show this expense name when can\_update\_desc\_fl is Y, meaning this expense description is provided by user. However only ONE customer defined expense can be shown here, the column name in entity should be expense\_desc

If CSAExpense object at one point is more than page can displayed, break the loop

Loop through all CSAExpense in first item of list of CSASaleExpense retrieved by GetSaleExpenses api:

Use for OAR <= full\_expenses\_fl\_bool

Management, Insurance etc, these are expense\_description

Owner’s Amount <= amount\_owners

Tenant’s Amount <= amount\_tenant

For calculation of $/NRA SF

expense\_pct\_of\_area\_denominator: number = 0.0;

if (csa\_net\_rent\_area.HasValue && csa\_net\_rent\_area > 0)

expense\_pct\_of\_area\_denominator = csa\_net\_rent\_area;

else if (tot\_building\_area.HasValue && tot\_building\_area > 0)

expense\_pct\_of\_area\_ denominator = tot\_building\_area;

if (expense\_pct\_of\_area\_denominator > 0) {

For owner: $/NRA SF <= amount\_owners / expense\_pct\_of\_area\_denominator

For tenant: $/NRA SF <= amount\_tenant / expense\_pct\_of\_area\_denominator

}

Subtotals <= if the given expense !istax add it to subtotal for owner and tenant column.

Totals <= include all expenses, tax included

$/NRA SF for subtotal and grand total, use subtoral or total divide by expense\_pct\_of\_area\_denominator if its value is positive.

Total Expense = Totals of Owner’s Amount + Totals of Tenant’s Amount

Divide Total Expense by expense\_pct\_of\_area\_denominator to get its $/NRA SF

Note $/NRA SF display text will change to $/Gross SF based on CSA\_NET\_RENT\_AREA, TOT\_BUILDING\_AREA

If CSA\_NET\_RENT\_AREA has value and it’s greater than 0

Use $/NRA SF for column text of both owner and tenant

ElseIf TOT\_BUILDING\_AREA has value and it’s greater than 0

Use $/Gross Sf for column display text instead.

Save

Required service models in DAL: CSASaleExpenses, CSAExpense

Validation:

All expense must be a value in interval [0, 999,999,999,999] inclusive. Other wise show error message: “An expense must be a whole number between 0 and 999,999,999,999.”

If owner expense description text box is empty user expense amount not empty

show error message: “Expense description is required if Owner's Amount is entered.”

else if tenant expense description text box is empty user expense amount not empty

show error message: “Expense description is required if Tenant's Amount is entered.”

Note the validation function should be included in the service model CSAExpense, below is possible implementation:

validate(): void {

  this.errors = [];

  let errorMsg: string | null = null;

  if (this.CAM\_Update\_Desc\_Fl === 'Y') {

    if (this.Amount\_Owners && this.Amount\_Owners !== 0) {

      // Expense amount is entered

      if (!this.Expense\_Description || this.Expense\_Description.trim().length === 0) {

        errorMsg = 'Expense description is required if Owner\'s Amount is entered.';

        this.errors.push(errorMsg);

        this.is\_valid = false;

      }

    }

    if (this.Amount\_Tenant && this.Amount\_Tenant !== 0) {

      // Expense amount is entered

      if (!this.expense\_description || this.expense\_description.trim().length === 0) {

        errorMsg = 'Expense description is required if Tenant\'s Amount is entered.';

        this.errors.push(errorMsg);

        this.is\_valid = false;

      }

    }

  }

}

For is\_valid property, add following to call Validate() function above:

get is\_valid(): boolean {

  if (this.\_isValid === null) {

    this.\_isValid = this.validate();

  }

  return this.\_isValid;

}

Make sure is\_valid returned by each CSAExpense model service is true before assign all of them to list\_of\_expense member variable of CSASaleExpenses object.

Create a CSAExpense (Check details in Model section of this document) object to store each expense item. Make sure expense is **greater than 0** to save. Update the expense amount to both cmc\_amount\_cd and cam\_amount\_cd. As for cmc\_update\_desc\_fl, cam\_update\_desc\_fl, cmc\_amount\_desc, cam\_amount\_desc will be updated based on the value of the GetMethod called by bind function above unchanged.

\*0 in expense should be stored as DBNull and finally all related CSAExpense should be included in the CSASaleExpenses model, check model session for more details about what member variables should be included in the service model

int UpdateExpensesInSaleExpenses(CSASaleExpenses, "CI\_Expenses", WorkerId)

##### Adjustments

Very much similar to the setup with the expense page. Ind Pur Price calculation. It’s a simple summary from a to d and then Adj Sales Price is the total from e to h.

Bind

GetSaleAdjustments

Thing to note, check if ind\_pur\_price already has existing value. This would mean the total will no longer require to be sum up from a to d but instead should directly use the ind\_pur\_price value stored.

Also only display data if (cmc\_update\_desc\_fl != null && cmc\_update\_desc\_fl == “Y”).

Make sure to count data see if it’s surpassed the data supposed to be displayed, which is 4 in total for user defined adjustments.

cam\_amount\_desc does not need to be displayed, but cmc\_amount\_desc should be displayed as adjustment description text. amount\_csa is used to display balue in the text box if there’s any value. Also remember to check this value with adjustment value user typed in. If adjustment total sum is different than the already existing Ind\_Pur\_Price or Adj\_Sales\_Price. When save is clicked make sure to mention one of the following error message:

Display when input value can’t match both ind\_pur\_price and adj\_sales\_price (is\_mismatched\_ind\_pur\_price and is\_mismatched\_adj\_sales\_price are true

"Save successful. (Ind Pur Price not equal to Total (a thru d) and Adj Sales Price not equal to Total (e thru h))."

Display when is\_mismatched\_ind\_pur\_price is true

"Save successful. (Ind Pur Price not equal to Total (a thru d))."

Display when is\_mismatched\_adj\_sales\_price is true

"Save successful. (Adj Sales Price not equal to Total (e thru h))."

For the CSASaleAdjustments model service in DAL:

total\_not\_user\_defined\_not\_bonds\_adjustments getter:

private total\_not\_user\_defined\_not\_bonds\_adjustments: number = 0;

get total\_not\_user\_defined\_not\_bonds\_adjustments(): number {

  this. total\_not\_user\_defined\_not\_bonds\_adjustments =

    this.indPurPrice !== null && this.indPurPrice !== undefined

      ? this.indPurPrice

      : 0;

  this.list\_of\_adjustments.forEach((adj: CSAAdjustment) => {

    if (adj.camUpdateDescFl === null || adj.camUpdateDescFl === 'N') {

      // this adjustment is not user-defined

      if (

        adj.camAmountCd !== null &&

        adj.camAmountCd !== undefined &&

        adj.camAmountCd !== 21

      ) {

        // not Bonds

        if (adj.amountCSA !== null && adj.amountCSA !== undefined) {

          this.total\_not\_user\_defined\_not\_bonds\_adjustments += adj.amountCSA;

        }

      }

    }

  });

  return this.total\_not\_user\_defined\_not\_bonds\_adjustments;

}

total\_user\_defined\_adjustments getter:

private total\_user\_defined\_adjustments: number = 0;

get total\_user\_defined\_adjustments(): number {

  this.total\_user\_defined\_adjustments = 0;

  this.listOfAdjustments.forEach((adj: CSAAdjustment) => {

    if (

      adj.camUpdateDescFl !== null &&

      adj.camUpdateDescFl === 'Y' &&

      adj.amountCSA !== null &&

      adj.amountCSA !== undefined

    ) {

      this.total\_user\_defined\_adjustments += adj.amountCSA;

    }

  });

  return this.total\_user\_defined\_adjustments;

}

is\_mismatched\_ind\_pur\_price getter:

private is\_mismatched\_ind\_pur\_price: boolean = false;

get is\_mismatched\_ind\_pur\_price (): boolean {

  this.is\_mismatched\_ind\_pur\_price = false;

  if (

    this.ind\_pur\_price !== null &&

    this.ind\_pur\_price !== undefined &&

    this.ind\_pur\_price !== this.total\_user\_defined\_adjustments

  ) {

    this.is\_mismatched\_ind\_pur\_price = true;

  } else if (

    (this.ind\_pur\_price === null || this.ind\_pur\_price === undefined) &&

    this.total\_user\_defined\_adjustments !== 0

  ) {

    this.is\_mismatched\_ind\_pur\_price = true;

  }

  return this.\_isMismatchedIndPurPrice;

}

is\_mismatched\_adj\_sales\_price getter:

private is\_mismatched\_adj\_sales\_price: boolean = true;

get is\_mismatched\_adj\_sales\_price(): boolean {

  this.is\_mismatched\_adj\_sales\_price = true;

  if (this.adjSalesPrice !== null && this.adjSalesPrice !== undefined) {

    if (this.adjSalesPrice === this.total\_not\_user\_defined\_not\_bonds\_adjustments) {

      this.is\_mismatched\_adj\_sales\_price = false;

    }

  } else {

    if (this.indPurPrice !== null && this.indPurPrice === this.total\_not\_user\_defined\_not\_bonds\_adjustments) {

      this.\_isMismatchedAdjSalesPrice = false;

    } else {

      if (this.is\_mismatched\_adj\_sales\_price) {

        if (this.total\_not\_user\_defined\_not\_bonds\_adjustments === 0) {

          this.is\_mismatched\_adj\_sales\_price = false;

        }

      }

    }

  }

  return this.is\_mismatched\_adj\_sales\_price;

}

Save

Validation:

If “Down” payment is negative

show error message: "The down payment cannot be negative."

If “1st” payment is negative

show error message: "1st mortgage cannot be negative."

If “end” payment is negative

show error message: "2nd mortgage cannot be negative."

Declare a List of CSAAdjustment instance to store each adjustment, save rule is same as expense page, only positive value will be stored in CSAAdjustment. Both cmc\_amount\_cd and cam\_amount\_cd will be updated. cmc\_update\_desc\_fl and cam\_amount\_desc\_fl should be keep the way how it got binded in.

After added all CSAAdjustment into the list, use it as a parameter to call update method.

Build the following Validate function in the model service:

private \_errors: string[] = [];

private \_isValid: boolean | null = null;

validate(): boolean {

  this.errors = [];

  this.is\_valid = true;

  if (this.cmc\_amount\_cd === 16) {

    // down payment

    if (this.amount\_csa !== null && this.amount\_csa !== undefined) {

      if (this.amount\_csa < 0) {

        const errorMsg = 'The down payment cannot be negative.';

        this.errors.push(errorMsg);

        this.is\_valid = false;

      }

    }

  }

  if (this.cmc\_amount\_cd === 17) {

    // 1st mortgage

    if (this.amount\_csa !== null && this.amount\_csa !== undefined) {

      if (this.amount\_csa < 0) {

        const errorMsg = '1st mortgage cannot be negative.';

        this.errors.push(errorMsg);

        this.is\_valid= false;

      }

    }

  }

  if (this.cmc\_amount\_cd === 18) {

    // 2nd mortgage

    if (this.amount\_csa !== null && this.amount\_csa !== undefined) {

      if (this.amount\_csa < 0) {

        const errorMsg = '2nd mortgage cannot be negative.';

        this.\_errors.push(errorMsg);

        this.\_is\_valid = false;

      }

    }

  }

  return this.is\_valid!;

}

get errors(): string[] {

  return this.errors;

}

All input fields should be used to update, a bit extra is Ind\_Pur\_Price and Adj\_Sales\_Price, if there are value existing and no above validation error. They should be stored in ind\_pur\_price and adj\_sales\_price respectively and updating database. Original has bug for this feature.

To update use following function:

UpdateAdjustmentsInSaleAdjustments(\_SaleAdjustments, "CI\_Adjustments", \_WorkerId)

##### Income Analysis & Value Ind

Graphical user interface, application

Description automatically generated

Bind

IncAnalysisValInd GetIncomeAnalysis(int p\_csa\_id)

For the mapping of different elements:

Potential Gross Income <= cmp\_pot\_gross\_inc, *take integer value, the number format should have ‘,’ to separate every three digits.*

Adj Sale Price <= cmp\_adj\_sale\_price, *the number should be integer*

NOI/Adj. Sale Price (%) <= cmp\_noi\_adj\_sale, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Vacancy and Coll Loss <= cmp\_vac\_coll\_loss, *take integer value, the number format should have ‘,’ to separate every three digits.*

PGI % <= pct\_pgi, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Adj. Sale Price/PGI <= cmp\_adj\_sale\_pgi, *one digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Additional Net Income <= addition\_net\_incom, *take integer value, the number format should have ‘,’ to separate every three digits.*

Adj. Sale Price/NRA <= cmp\_adj\_sale\_nra, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Tenant Expense Reimbursement <= tenant\_reimburse, *take integer value, the number format should have ‘,’ to separate every three digits.*

Rent(PGI)/NRA/Yr <= cmp\_rent\_nra, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Effective Gross Income <= cmp\_eff\_gross\_inc, *take integer value, the number format should have ‘,’ to separate every three digits.*

NOI/NRA/Yr <= cmp\_noi\_nra, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros. It should be formatted as currency*

Owner’s Expenses <= cmp\_tot\_expenses, *take integer value, the number format should have ‘,’ to separate every three digits.*

EGI % <= cmp\_egi\_pct, *two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.*

Owners Expenses/NRA/Yr <= cmp\_expense\_nra, two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.

Anticipated Property Taxes <= anti\_property\_tax, take integer value, the number format should have ‘,’ to separate every three digits.

Tax Rate <= tax\_rate, at least two characters reserved for the integer part of the number. If the number has fewer than two digits before the decimal point, leading zeros will be added. two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros.

Total Expenses/NRA/Yr <= cmp\_tot\_xpens\_nra, two digits will be displayed to the right of the decimal point. If there are fewer than two digits, it pads with zeros. It should be formatted as currency

The blank description under anticipated property tax <= (other\_taxes\_desc = “”) ? “ “ : (other\_taxes\_desc + :)

For the bland description, its tax box (value) <= other\_taxes\_amount, take integer value, the number format should have ‘,’ to separate every three digits.

Min % Occupancy <= pct\_occup\_minimum, take whole number

Net Operating Income <= cmp\_net\_op\_inc, take integer value, the number format should have ‘,’ to separate every three digits

% Occupancy <= pct\_occupancy, take whole number

OAR <= cmp\_oar\_desc

Property NRA <= csa\_net\_rent\_area, take integer value, the number format should have ‘,’ to separate every three digits

Benchmark Rating <= bm\_desc

Notice there’s a display change on the page depending on CSA Net Rent Area has a positive value or not

if (this.\_CSA\_Net\_Rent\_Area != null && this.\_CSA\_Net\_Rent\_Area > 0) {

  this.\_lblAdjSalePrice = "Adj. Sale Price/NRA:";

  this.\_lblRentPGI = "Rent(PGI)/NRA/Yr:";

  this.\_lblNOI = "NOI/NRA/Yr:";

  this.\_lblOwnersExpenses = "Owners Expenses/NRA/Yr:";

  this.\_lblTotalExpenses = "Total Expenses/NRA/Yr:";

} else {

  this.\_lblAdjSalePrice = "Adj. Sale Price/Gross SF:";

  this.\_lblRentPGI = "Rent(PGI)/Gross SF/Yr:";

  this.\_lblNOI = "NOI/Gross SF/Yr:";

  this.\_lblOwnersExpenses = "Owners Expenses/Gross SF/Yr:";

  this.\_lblTotalExpenses = "Total Expenses/Gross SF/Yr:";

}

Save

Validation:

PGI %: "PGI % must be a number between 1.00 and 100.00"

Additional Net Income:

"Additional Net Income must be a whole number between 1 and 999,999,999,999."

Tenant Expense Reimbursement:

"Tenant Expense Reimbursement must be a whole number between 1 and 999,999,999,999."

Anticipated Property Taxes: "Anticipated Property Taxes must be a whole number between 1 and 999,999,999,999."

Other Taxes Amount (amount for blank text description): "Other taxes must be a whole number between 1 and 999,999,999,999."

Min % Occupancy: "Min % Occupancy must be a whole number between 0 and 100."

int UpdateIncomeAnalysis(UpdatedIncomeAnalysis p\_income\_analysis, string p\_last\_modified\_page, string p\_worker\_id)

Only updating the ones with text box, as for the rest of the data just submit the value get method obtained along with new updates made to text box data. Same number fomat should be applied to the outbound data.

##### Sources

Bind

CSASource GetSource(int p\_csa\_id)

SourceDesc GetSourceDesc(int p\_field\_id)

Here use 114

List<WorkerInfo> GetWorker()

This will get all workers to be showed on the dropdown

List<FieldCode> GetFieldCode (int p\_field\_id)

Graphical user interface, application, Word

Description automatically generated

Mapping of elements (from CSASource): 3530 continue

Source

Primary <= CD\_SHORT\_NAME

For other drop down options, use GetFieldCode(XXX) result CD\_LONG\_NAME

Name <= SOURCE\_NAME\_PRI

Title <= SOURCE\_TITLE\_PRI

Company <= SOURCE\_COMPANY\_PRI

Email <= SRCE\_EMAIL\_PRI

Phone <= SOURCE\_PHONE\_PRI (Format: {0:(###) ###-####})

Ext <= SRCE\_PH\_EX\_PRI

Alternate Phone <= SRCE\_ALT\_PH\_PRI (Format: {0:(###) ###-####})

Ext <= SRCE\_ALT\_PH\_EX\_PRI

Secondary <= SOURCE\_DESC\_SEC

Name <= SOURCE\_NAME\_SEC

Title <= SOURCE\_TITLE\_SEC

Company <= SOURCE\_COMPANY\_SEC

Email <= SRCE\_EMAIL\_SEC

Phone <= SOURCE\_PHONE\_SEC (Format: {0:(###) ###-####})

Ext <= SRCE\_PH\_EX\_SEC

Alternate Phone <= SRCE\_ALT\_PH\_SEC (Format: {0:(###) ###-####})

Ext <= SRCE\_ALT\_PH\_EX\_SEC

Additional Sources of Information

IQ <= SRCE\_IQ\_FL

AAB <= SRCE\_AAB\_FL

PCOR/COS <= SRCE\_PCOR\_COS\_FL

RFR <= SRCE\_RFR\_FL

Other <= SOURCE\_OTHER\_DESC

Source History

Created By <= GetWorker() result

Created Date <= SOURCE\_CREATE\_DT, (Format: {0:MM/dd/yyyy})

Modified By <= CMP\_MODIFIED\_BY

Modified Date <= SOURCE\_MODIFIED\_DT, (Format: {0:MM/dd/yyyy})

Modify Reason <= SRCE\_MODIFY\_RSN\_CD\_LONG\_NAME

Supervisor Approved <= SUPRV\_APPROVED\_FL\_TEXT

Suprv Update Date <= Suprv\_Appr\_Upd\_Ts, (Format: {0:MM/dd/yyyy})

Suprv Approval Update By <= SUPRV\_APPR\_UPD\_WKR\_NAME

Save

Validation:

If Unformat(any phone number text box) is not a 10 digit number

show error message: "The Primary Phone must be a valid 10 digit US phone number."

If “Ext” number less than 0 or greater than 99999

show error message: "Primary Extension is out of range."

If “Ext” has non blank value but “Phone” does not have value or only white space

show error message: "Primary Phone is required if Primary Extension is entered."

Similar setup should be also applied to Alternate phone and its Ext below are their error messages:

"The Primary Alternate Phone must be a valid 10 digit US phone number."

"Primary Alternate Extension is out of range."

"Primary Alternate Phone is required if Primary Alternate Extension is entered."

If “Primary” source is selected, Either “Name” or “Company” is required otherwise

show error message: "The Name or Company is required if Primary Source is entered."

If “Name”, “Title”, “Company”, “Email”, “Phone”, “Alternate Phone” any is filled while “Primary” source is not given

show error message: "The Primary Source is required if primary details are entered."

SAME validation rules should be applied to the secondary session with “primary” keyword swap out for “secondary”.

If “Created By” input is none of the drop down value

show error message: "Invalid Created By worker."

If “Created Date” > “Modified Date”

show error message: "Created Date cannot be greater than today."

Data should be stored in CSAUpdatedSourceData model. Check its content in the model session of this document to map its member name to corresponding block (most should have exact same name as in data mapping)

UpdateCSASourceData

This is used to make update. Note that validation should be applied before calling the patch function. Please refer to FR for more details about validation requirements.

### VacantLandInput page

VacantLandInput page will be reusing Title Bar, PropChar Bar, but will build its own tab strip as well as all tab components that nested within this page.

#### ngOnInit

#### Tabstrip Components

Sale Info

Prop Char

TODO

### Title Bar

This will be a shared component throughout every CSA pages, display Improve logo as well as current worker. Worker ID should come from the query string provided to default page, then default page should pass in the QueryParameterService that contains worker ID into this component for it to display.

### Control Bar

This will be a shared component. This information bar is designed to interact with user to perform multiple operations on the tabs.

*Save Button*

This button’s functionality should vary depends on current tab. Please refer to function requirement document and check under which tab which CSA api should be called to update input information.

*Refresh Characteristic*

This is used to do manual refresh. This will run bind function again to rebind all data from database. If there’s any update made with Save Button. This should reflect that change in the database.

For detailed information of how control bar is implemented, check frontend design for more information.

#### Report Button(Sales Analysis Report)

Once clicked, PopReportViewer Angular component should be displayed. Modify PopReportViewer component to display the right report under [report session](#_Reports).

### CSAPropertyInfo

This will be a shared component. This information bar is designed to display current selected Improved Property’s property characteristics information.

This page should contain several properties: CSAID, IEMode, WorkerID, CSAType. These should be loaded by the NgOnInit function of the page that contains this property. As data pass in standard. Please check out Angular Frame Work design document. Use query string to get CSAID, IEMode, WorkerID, CSAType.

**ngOnInit**

For this page it should be using HttpClient service to call API VacantLandInputDAL.GetPropertyInfo(CSA\_ID). This should be able to return all required information as data source for displaying.

Added: This page should include Document information, Event Date, Buyer and Seller information too that could be obtained through GetSaleInfoFull(CSA\_ID) api call

**Buttons**

If we are not in the edit mode (IEMode != “E”) OR if there’s only one APN available relating to the current CSA ID. Edit APN button should not be visible. Otherwise with button click it should show pop up page, which is popAPNList with client id as eventts and csa id, 350, 400

If IEMode !=

***For ImageButton***, app\_hone.PNG on click should pop up ShowAPNListView page. With parameter csaid, 200 and 300.

***For Edit button***

If IE Mode is I, then use CSASaleBLL.GetSupApprFlag(p\_csa\_id) to check:

If return is “Y” show button,

Else, hide button

Else (If IE mode is not I), do not show button

First do GetLeaseInfo, if retrieved, Then SaleInfoHasIncomesExpenses is true, if not then create CSASaleExpenses and set ExpensesForAllExpenseCodesBool as false, then iterate through all return result, if it’s not 0 update SaleInfoHasIncomesExpenses to true.

***Data Mapping***

APN <= PRINT\_PARCEL

APN Count <= DOC\_PARCEL\_CNT

Ind Pur Price <= IND\_PUR\_PRICE {0:N0}

Gross SF <= TOT\_BUILDING\_AREA {0:N0}

Address <= ADDRESS

Use Cd <= USE\_CD {0:0000}

Use <= USE\_NAME

Adj Sales Price <= ADJ\_SALES\_PRICE {0:N0}

PC NRA <= TOT\_NET\_RENT\_AREA {0:#,###,###,###}

City <= SITUS\_CITY\_NAME

State <= SITUS\_STATE

Zip <= ZIP\_CD

Trans Tax Price <= TRAN\_TAX\_PRICE {0:N0}

Land Area <= TOT\_LOT\_SIZE {0:#,###,###,###}

***Doc Image***

Feature TBD

### Comments

Also make sure to give the option to disable add “+” button and edit button. As in lease pop up window, it should be readonly and no interaction should be available to user.

Bind

First function is used to display comments relating to the current CSA document, second function is used to display comments relating to certain lease id which should be used in the pop up lease modification window:

List<CSAComments> GetCSAComments (int p\_csa\_id)

Overloading

List<CSAComments> GetCSAComments (int p\_lease\_id)

Save

Add: int InsertCSAComments (CSAComments p\_comments)

Update: int UpdateCSAComments (CSAComments p\_comments)

### Loading Overlay page

This is a mask page to be placed on top of current page during loading in order to temporarily prevent interaction with current page during page loading, page is already built, check assr-csa-web\src\app\Shared\app-loading-overlay. To use, simply add component within current component markup page as: <app-loading-overlay></app-loading-overlay>, Also inject loading service to toggle the loading overlay on and off. Example is set in app component.

### CSALeaseEdit page

There are two modes available to this page:

I: Insert mode, in this mode lease will be created a new and add to database, thus HttpClient should use POST request.

E: Edit mode, in this mode existing lease will be edited and updated information will be sent to server side to update database. In this mode HttpClient should use PUT request.

#### ngOnInit(Page\_Load)

Use GetPropertyInfo to retrieve PROPERTY\_ID and PRINT\_PARCEL, PRINT\_PARCEL, APN will be used to display.

Use GetLeaseUseCategories to retrieve lease uses.

Use Category <= USE\_NAME

Once Use is selected, use GetLeaseUseSubByUseCatto retrieve drop down for sub category

Subcat <= SUBCATEGORY\_NAME

GetFieldCode

Use CD\_LONG\_NAME to assign to each drop down

|  |  |
| --- | --- |
| Field ID (Parameter) | Item Field |
| 125 | Tenancy |
| 133 | TI Allowance Type |
| 124 | Expense Type |
| 123 | Lease Type |

Use GetLease(int p\_lease\_id) to get lease information including comments of the lease

For yellow marks, feel free to use cd value for display, such as use\_category\_cd instead of lease\_use\_name, but remember when save only the current cd value needs to be saved.

**Mapping from LeaseBO**

APN <= apn

Use for Rent Comp <= use\_for\_comp\_fl

Location <= lease\_location

Sq Ft Leased/Available <= net\_rentable\_area

Vacant <= vacant\_fl

Lessee <= lessee\_name

Start Date <= lease\_start\_dt

Term(Months) <= lease\_term

Use Category <= lease\_use\_name

Subcat <= subcategory\_name

Tenancy <= tenancy\_text

Current Rent <= current\_rent

Per Sq Ft <= current\_rent\_psf {0:C2}

Lease Type <= lease\_type\_text

Anticipated Rent <= anticipated\_rent

Per Sq Ft <= anti\_rent\_psf {0:C2}

Expense Type <= expense\_type\_text

Expense Base Year <= expense\_base\_yr

Start Rent <= initial\_rent

Per Sq Ft <= initial\_rent\_psf {0:C2}

Option <= lease\_option

TI Allowance <= ti\_allowance

TI Allowance Type <= ti\_type\_text

Info Source <= info\_source\_text

Rent Adjustment <= rent\_adj\_text

Parking <= lease\_parking

Parking 2 (the empty block below Parking) <= lease\_parking\_2

**Mapping from LeaseBO.lease\_detail**

Clear Height <= ind\_clr\_hgt

% Office Area <= ind\_offc\_area\_pct

% Lab <= ind\_lab\_pct

% Manufacturing <= ind\_mfg\_pct

% Total <= Sum above three percentages up (if no value available, assign 0)

Load Factor <= offc\_load\_factor

Annual Expense Stop <= offc\_expense\_stop

Percentage Rent <= retail\_rent\_pct

Breakpoint <= rent\_breakpoint

**Display Comments component here under all other fields**

#### Save Lease

Validation:

If Use for Rent Comp = no

apn, use\_for\_comp\_fl, lease\_location, net\_rentable\_area, vacant\_fl **cannot be empty**

Else

Not only all above variables, lease\_start\_dt, lease\_term, use\_category\_cd, lease\_subcategory, expense\_type\_cd, initial\_rent, info\_source\_cd **cannot be empty**

net\_rentable\_area: "Sq Ft Leased/Available must be > 0 and <= 999,999,999."

vacant\_fl: check use\_for\_comp\_fl, "Vacant cannot be Yes when Use For comp is Yes."

current\_rent: "Current Rent must be a whole number."

anticipated\_rent: "Anticipated Rent must be a whole number."

expense\_base\_yr: “Expense Base Year is invalid”

initial\_rent: "Start Rent must be a whole number."

stabil\_rent: "Stabilized Rent must be a whole number."

ti\_allowance: "TI Allowance must have only 2 decimals."

IND\_OFFC\_AREA\_PCT: "% Office Area must be a whole number and <=100."

IND\_LAB\_PCT: "% Lab must be a whole number and <= 100."

IND\_MFG\_PCT: "% Manufacturing must be a whole number and <= 100."

offc\_load\_factor: "Load Factor must have only 2 decimals, between 1.00 and 9.99."

offc\_expense\_stop: "Annual Expense Stop must be a whole number."

retail\_rent\_pct: "Percentage Rent must have only 2 decimals, between 0.00 and 99.99."

rent\_breakpoint: "Breakpoint must be a whole number."

First use model LeaseDetailBO to store below information:

lease\_id <= leaseID

entry\_ts <= dataTime

entry\_worker <= workerID

update\_ts <= dateTime

update\_worker <= workerID

Use model LeaseBO to store all relating data fields in the form.

Depending on current mode and \_LeaseBO.use\_for\_comp\_fl value “Save” behavior varies:

If mode = “I”

\_LeaseBO.entry\_ts = dateTime

\_LeaseBO.entry\_worker = workerID

\_LeaseBO.lease\_created\_by = workerID

\_LeaseBO.lease\_create\_dt = date

If \_LeaseBO.use\_for\_comp\_fl = “N”

call \_LeaseBLL.InsLeaseNoRentComp(\_LeaseBO)

else

call \_LeaseBLL.InsertLease(\_LeaseBO)

Else //Model is “E”

\_LeaseBO.update\_ts = \_dateTime

\_LeaseBO.update\_worker = \_workerID

If there’s change in the form

\_LeaseBO.lease\_modify\_by = workerID

\_LeaseBO.lease\_modify\_dt = \_date

//Do this because this info not in form

Else \_LeaseBO.lease\_modify\_by = workerID\_original

If\_LeaseBO.use\_for\_comp\_fl = “N”

call \_LeaseBLL.UpdLeaseNoRentComp(\_LeaseBO)

Else

call \_LeaseBLL.UpdateLease(\_LeaseBO)

Finally if there’s a CSA ID,call updateModifyReason controller action

If \_csaID > 0

let modifyReasonCD: number | null

call updateModifyReason(\_csaID, modifyReasonCD, \_workerID)

If LeaseMode == “I”, meaning app is inserting new lease

Save button should give the following setup to prevent same lease being insert multiple times, disable the save button till insert api call is completed. At the same time show a spinner on screen indicating progress is ongoing.

HTML:

<app-loading-spinner \*ngIf="isSaveInProgress"></app-loading-spinner>

<button (click)="performSave()" [disabled]="isSaveInProgress">Save</button>

TypeScript:

import { Component } from '@angular/core';

@Component({

  selector: 'app-your-component',

  templateUrl: './your-component.component.html',

  styleUrls: ['./your-component.component.css']

})

export class YourComponent {

  isSaveInProgress = false;

  performSave() {

this.isSaveInProgress = true;

    // Call InsertLease API

    // Other logic

this.isSaveInProgress = false;

  }

}

## Reports

below are each report’s specification, including which **stored procedure** they will use for each **dataset**. Both SP and Dataset should be created as following speculation using Visual Studio SSRS Report Designer. For anything unclear about reporting structure, [click here](#_Report_Design_and)

### SalesAnalysis Report

Dataset and SP

SaleInfo

IE\_GET\_CSA\_SALE\_INFO\_FULL

GetPropertyInfo

IE\_GET\_CSA\_PROP\_INFO

GetIncomeExpenseAnalysis

IE\_GET\_CSA\_INCOME\_EXPENSE\_ANALYSIS

GetLeaseInfo

IE\_GET\_CSA\_LEASE\_INFO

GetCSAExpense

IE\_GET\_CSA\_EXPENSE

GetCSARptSource

IE\_GET\_CSA\_RPT\_SOURCE

GetCSAComments

IE\_GET\_CSA\_COMMENTS

GetMailingNameAsOfDate, for report data table sale buyer,

add -1 minute to \_event\_ts parameter to get report data table sale seller.

IE\_GET\_CSA\_MAILING\_NAME\_AS\_DATE

GetAllAPNinDoc, for a full APN list

IE\_GET\_CSA\_ALL\_APN\_IN\_DOC

Add on data:

GetTaxRate, for tax rate, if this value is available, add it to the GetIncomeExpenseAnalysis result.

IE\_GET\_CSA\_TAX\_RATE

Add all above info as report data source to create a report use SSRS.