



AAIR SIG - The Next Data Warehouse

“Unlock the Value of Information”

Robbie McGregor - July 2019

First - a little bit about me:

- Working in the data space since 1999 (1999-2009 N. America, 2009 onwards Hamilton)
- Domains: Healthcare, Academia, Construction, Military, Dairy, Telco
- Degrees in Computer Science and Information Systems
- <https://www.linkedin.com/in/rjmcgregor/>

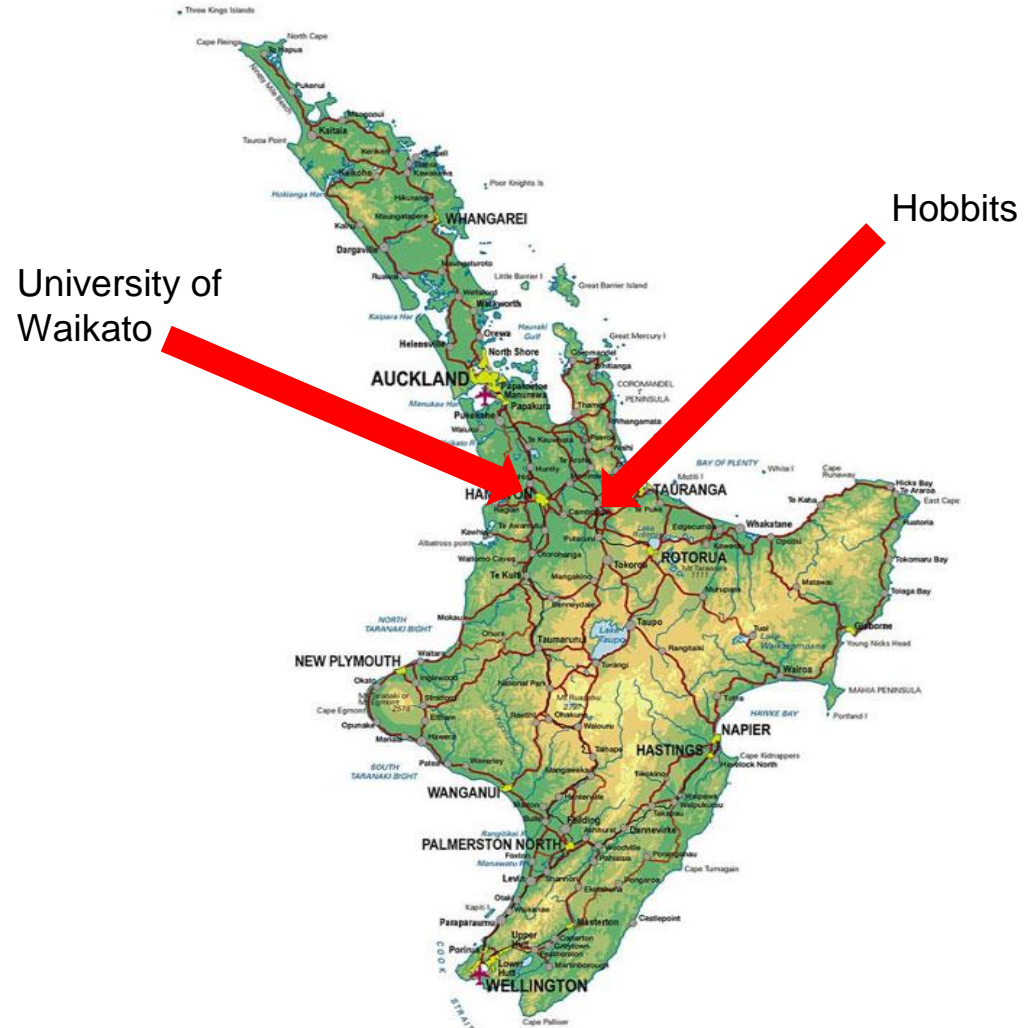
Why am I here?

- Share our story
- Our approach has been different
- Hopefully impart some wisdom
- Meet the wider community
- Darwin is warmer than Hamilton

How I got here:

- I'm here on behalf of the teams:
 - Data Services Group
 - Programme and Commercial
 - Business Intelligence and Analytics
- I wouldn't be here without them

University of Waikato







University of Waikato



Tauranga Campus



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



- By the numbers:
 - 10425 EFTS
 - 1612 Staff
 - Degrees completed (2018): 3,812

● What does our team do?

Data modelling

(Combining multiple source systems into single, consumable business objects)

Data transportation

(Integration and synchronization between systems)

Feed into Data Governance and Architecture

(Alignment with sanctioned UoW Data Framework)

Apply Data security policies

We do not create data

Why do we do it?

Ticket Management

File Edit View Options Tools

Summary of Non-transferred RM Transactions Entry Date: From 01/01/2002 To 20/01/2012

Filter

Co...	Source	Description - Source	Ticket Number	Ticket Date	Customer Order No.	Comment	RM Status	RM Type	RM File	RM Product
5	552 C100	West Quarry	2010072905	29/07/2010			0 - Entry	R	020002	0-208
6	553 C100	West Quarry	2010072906	29/07/2010			0 - Entry	R	020002	0-208
7	554 C100	West Quarry	2010072907	29/07/2010			0 - Entry	R	020002	0-208
8	555 C100	West Quarry	2010072908	29/07/2010			0 - Entry	R	020002	0-208
9	556 C100	West Quarry	2010072909	29/07/2010			0 - Entry	R	020002	0-208
10	557 C100	West Quarry	2010072910	29/07/2010			0 - Entry	R	020003	0-208

Detail Additional Fields

Counter 556 Source C100 Ticket No. 2010072909 Activity 44000 Group R
Date 29/07/2010 User No. 3 Date Entered 29/07/2010

Material
Status Entry Invoicing Type Receivable
Product 0-208 Quantity 50.00 UP 8.75 Amount \$437.50
Address Tr Qty 50.00 UP 5.00 Amount \$250.00
Customer Project JOB02 Activity
Contract Customer 020002 Town of Schomberg
Billing Date / / Date / / User No. 0 P. O. Inv. unit
Comment Customer P.O. RM Invoice

Extras	Quantity	Unit Price	Amount	Quantity	Unit Price	Amount
01	0.00	0.00	\$0.00	03	0.00	\$0.00
02	0.00	0.00	\$0.00	04	0.00	\$0.00

Transport
Status Entry Type Standard Region WEST Supplier 000033 Rennie Transport
Time 0 KM 0.00 UP 0.0000000 Truck # 105096
Tr Qty 50.00 Hours 0.00 Rate 0.00 Amount \$0.00
Invoice No. Project C100 Activity Group D
Billing Date / / Date 29/07/2010 User No. 0 TR P.O. TR Invoice

Northern Star Construction Edit

Why do we do it?

Play Data - Blue Link Elite - User: Darren - v.12.08.42 - [Sales Order Review]

File Edit Modules Security Utilities Window Help

Select item(s) to View
Apply Filter Set: [] Save / Update Filter Set Remove Filter Fill Run #: []

Orders To Update Refresh

Select	SO#	Location	Order Date	Invoice #	Invoice Date	Customer Name	SLSP	Ship Via	Order Status	Sub Total	Currency	OK			
+	30553	Ottawa	2009-08-04	50383	2009-08-04	Boston Gourmat Pizza	DB	BEST WAY	NEW	90.00	US	OK			
-	30551	Calgary	2009-06-18	0		Back Yard Burgers	RB	BEST WAY	NEW	47.39	CA	OK			
						Product Code	Product Description	UOM	Quantity	UBO	Price	Disc.	Extension	Ship Date	Ext
						0101-ARE	ARENA PLATE 9 3/8"	EACH	1	0	19.99	0.00%	19.99	2009-06-18	
						0101-ARE	ARENA PLATE 9 3/8"	EACH	1	0	13.50	0.00%	13.50	2009-06-18	
						0101-ARE	ARENA PLATE 9 3/8"	EACH	1	0	13.90	0.00%	13.90	2009-06-18	
+	30550	Calgary	2009-06-17	0		GEORGIA KITCHEN SUPPLIES		BEST WAY	NEW	30.00	US	OK			
+	30549	In-Bond	2009-06-15	0		Back Yard Burgers	RB	BEST WAY	NEW	0.00	CA	OK			
+	30548	Ottawa	2009-06-12	50378	2009-06-12	Black-Eyed Pea Restaurants	DB	BEST WAY	NEW	45.00	CA	OK			
+	30547	Ottawa	2009-06-10	0		Boston Gourmat Pizza	DB	BEST WAY	NEW	0.00	US	OK			
+	30544	Calgary	2009-05-26	0		Atlanta Bread Company		DHL	NEW	0.00	US	OK			
+	30543	Calgary	2009-05-25	0		Black-Eyed Pea Restaurants	DB	BEST WAY	NEW	50.05	CA	OK			
+	30537	Calgary	2009-05-25	0		Atlanta Bread Company		DHL	NEW	0.00	US	OK			
+	30536	Calgary	2009-05-25	0		Atlanta Bread Company		DHL	NEW	0.00	US	OK			
+	30535	Ottawa	2009-05-25	0		PIZZA OVENS R US		BEST WAY	NEW	45.00	US	OK			
+	30530	Ottawa	2009-05-22	0		Black-Eyed Pea Restaurants	DB	BEST WAY	NEW	0.00	CA	OK			
+	30529	Calgary	2009-05-22	0		DELTA CALGARY AIRPORT		BEST WAY	NEW	0.00	CA	OK			
+	30527	In-Bond	2009-05-13	0		Atlanta Bread Company		DHL	NEW	45.00	US	OK			
+	30526	Ottawa	2009-05-06	0		Black-Eyed Pea Restaurants	DB	BEST WAY	MGR Apprc	913.00	CA	OK			
+	30524	Calgary	2009-05-05	0		Back Yard Burgers	RB	BEST WAY	NEW	180.00	CA	OK			
+	30523	Calgary	2009-05-05	0		Black-Eyed Pea Restaurants	DB	BEST WAY	NEW	100.00	CA	OK			
+	30515	Calgary	2009-05-04	50419	2010-05-05	Atlanta Bread Company		DHL	NEW	45.00	US	OK			
+	30514	Calgary	2009-05-04	0		Atlanta Bread Company	CC	DHL	NEW	0.00	US	OK			
+	30513	Calgary	2009-05-04	50416	2010-04-19	Atlanta Bread Company	CC	DHL	NEW	49.00	US	OK			
+	30512	In-Bond	2009-04-30	0		Back Yard Burgers	RB	BEST WAY	NEW	0.00	CA	OK			

Record: 1 of 3

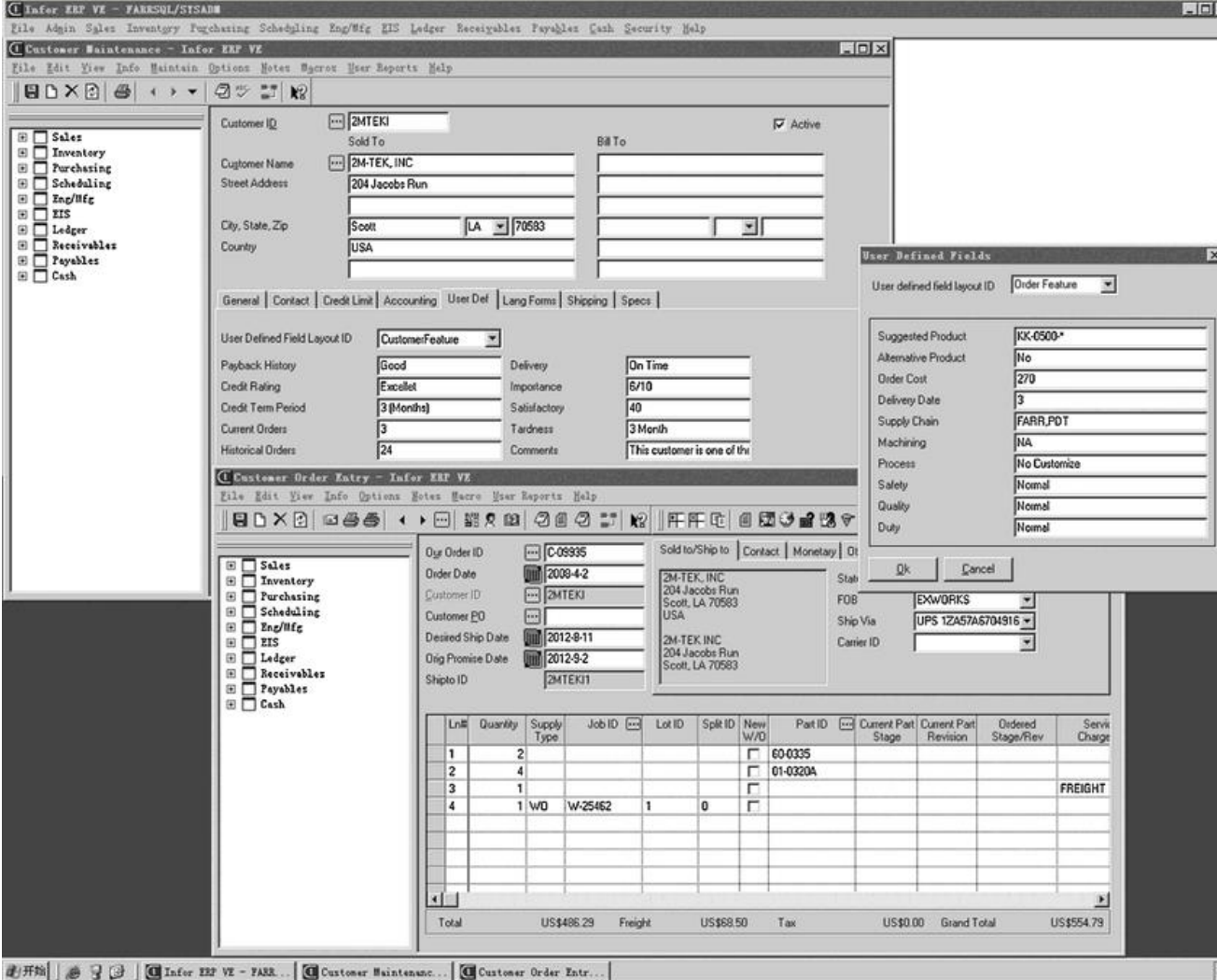
Take Actions Select All Select "OK" Only Select None Enter / View Notes

Tip: Click the + to view order details
Double-click to drill down.

Find Quote # Find SO # Find Individual: Find RMA # Find Customer:

Double-click to drill down to inventory screen NUM

Why do we do it?



The screenshot displays the Infor ERP V2 interface with two main windows open: 'Customer Maintenance' and 'Customer Order Entry'. The 'Customer Maintenance' window shows details for customer '2MTEKI', including name, address, and contact information. The 'Customer Order Entry' window shows order details for order 'C-09935', including dates, quantities, and a table of line items.

Customer Maintenance - Infor ERP V2

Customer ID: 2MTEKI (Active)
Customer Name: 2M-TEK, INC
Street Address: 204 Jacobs Run
City, State, Zip: Scott, LA, 70583
Country: USA

General | Contact | Credit Limit | Accounting | User Def | Lang Forms | Shipping | Specs

User Defined Field Layout ID: CustomerFeature

Payback History: Good | Delivery: On Time
Credit Rating: Excellent | Importance: 5/10
Credit Term Period: 3 (Months) | Satisfactory: 40
Current Orders: 3 | Tardness: 3 Month
Historical Orders: 24 | Comments: This customer is one of the

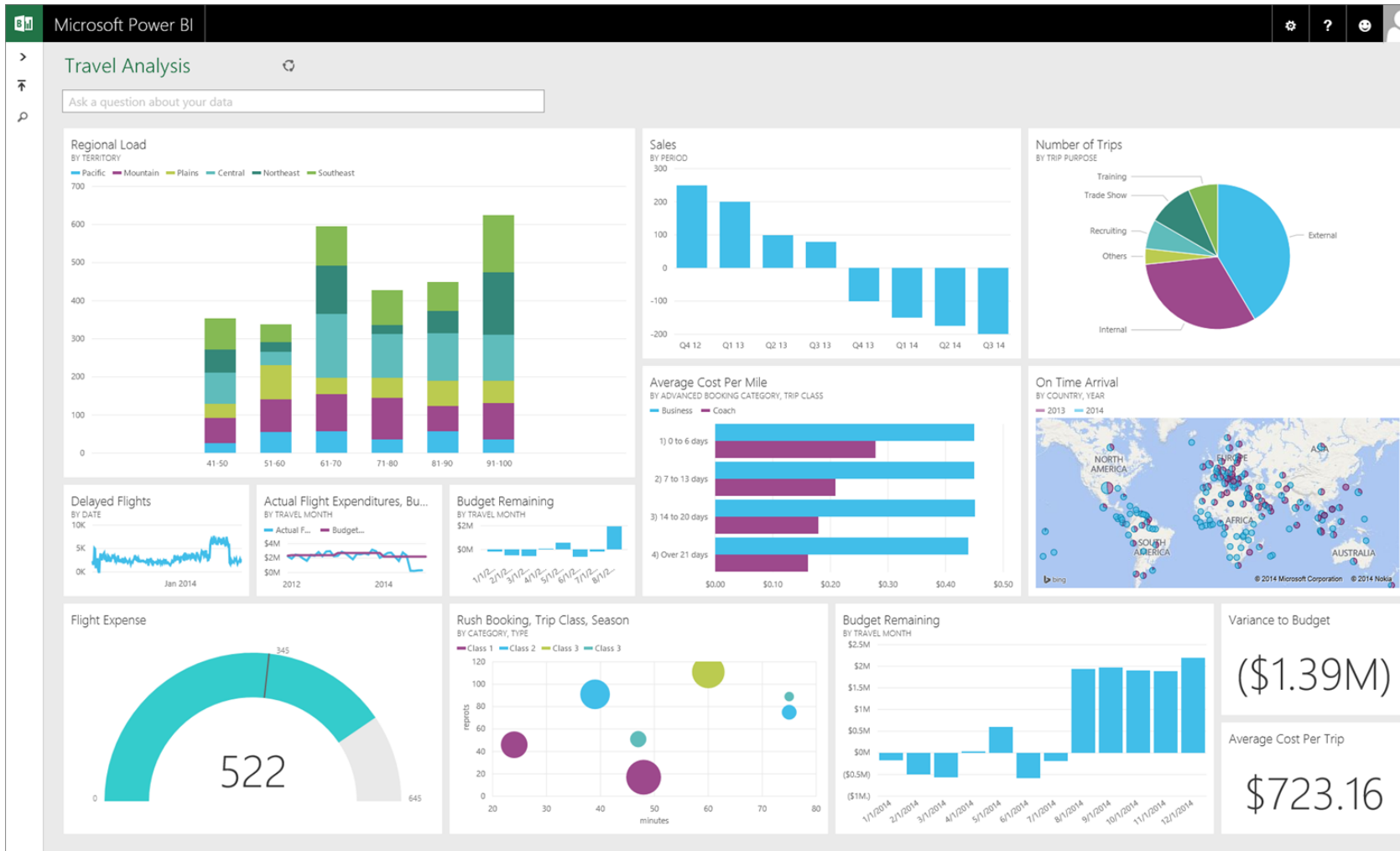
Customer Order Entry - Infor ERP V2

Order ID: C-09935 | Sold to/Ship to: Contact | Monetary: D
Order Date: 2008-4-2 | Customer ID: 2MTEKI
Customer EO: 2M-TEK, INC
Desired Ship Date: 2012-9-11 | Ship to: 204 Jacobs Run, Scott, LA 70583, USA
Orig Promise Date: 2012-9-2
Ship to ID: 2MTEKI1

Ship Via: UPS 1ZA57A6704916
Carrier ID: [blank]

Ln#	Quantity	Supply Type	Job ID	Lot ID	Split ID	New W/D	Part ID	Current Part Stage	Current Part Revision	Ordered Stage/Rev	Servic Charge
1	2					<input type="checkbox"/>	60-0335				
2	4					<input type="checkbox"/>	01-0320A				
3	1					<input type="checkbox"/>					FREIGHT
4	1	WO	W-25462	1	0	<input type="checkbox"/>					
Total											
		US\$486.29	Freight	US\$68.50	Tax	US\$0.00	Grand Total				US\$554.79

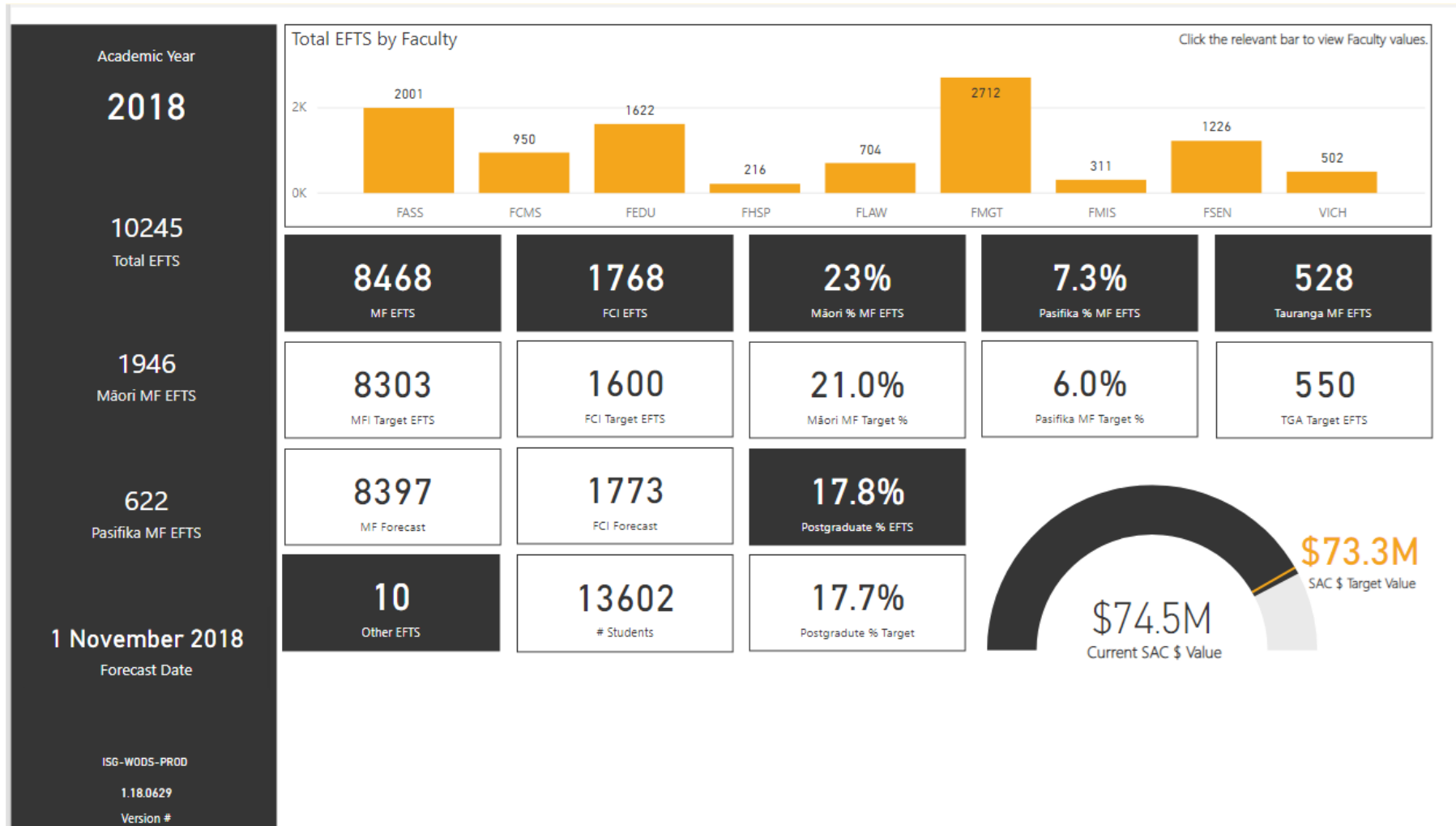
Why do we do it?



Why do we do it?



Why do we do it?



Let's do the time warp....



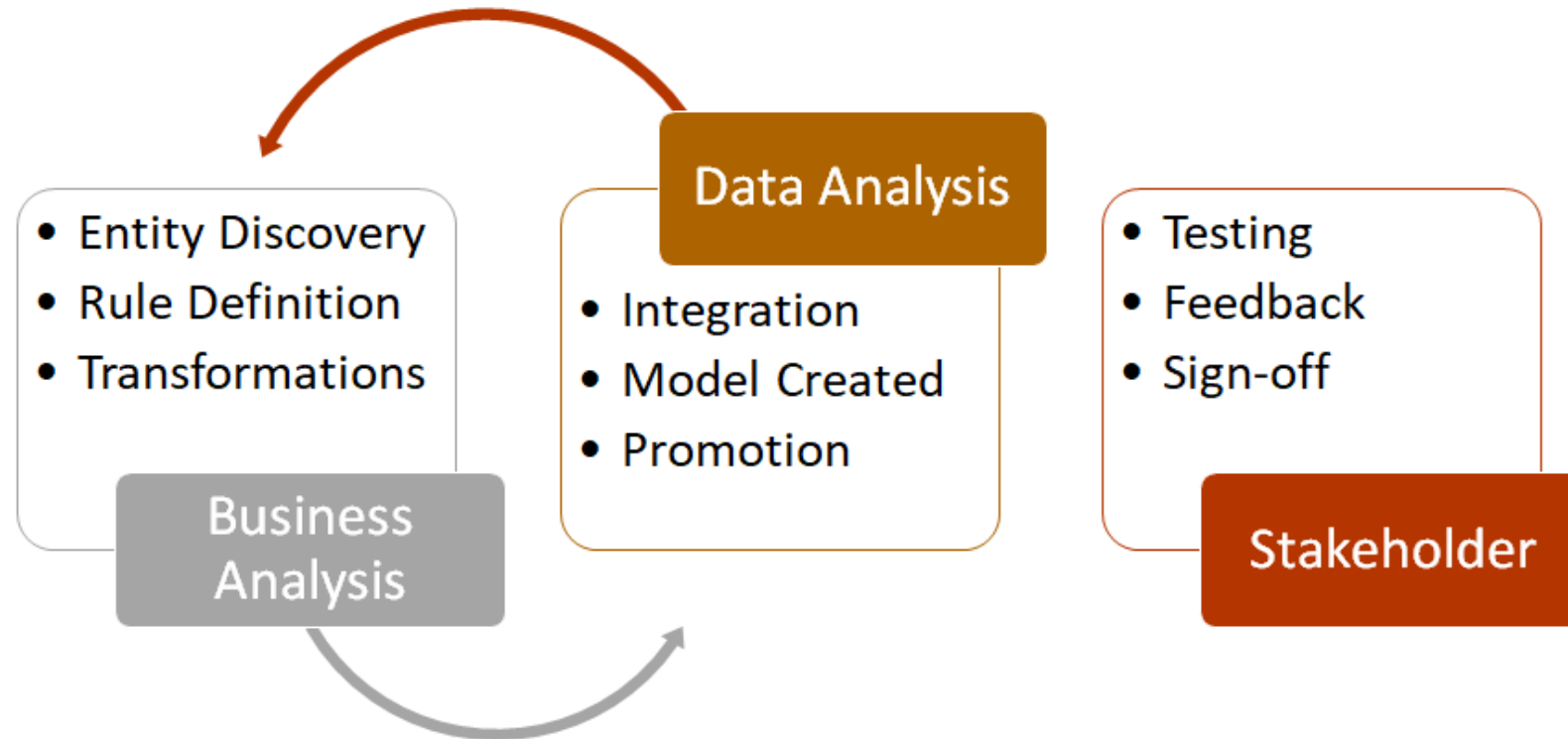
... all the way back to Dec 2017

- Student management System to be replaced July 2018
- Data requirements were vague, changing or unknown
- *Some* reporting was required on go-live, Ministry of Education plus operational reporting
- Data Warehouse, including time-bound data required
- Reporting Portal to be created – One high level:
- **Self Service** Reporting needed

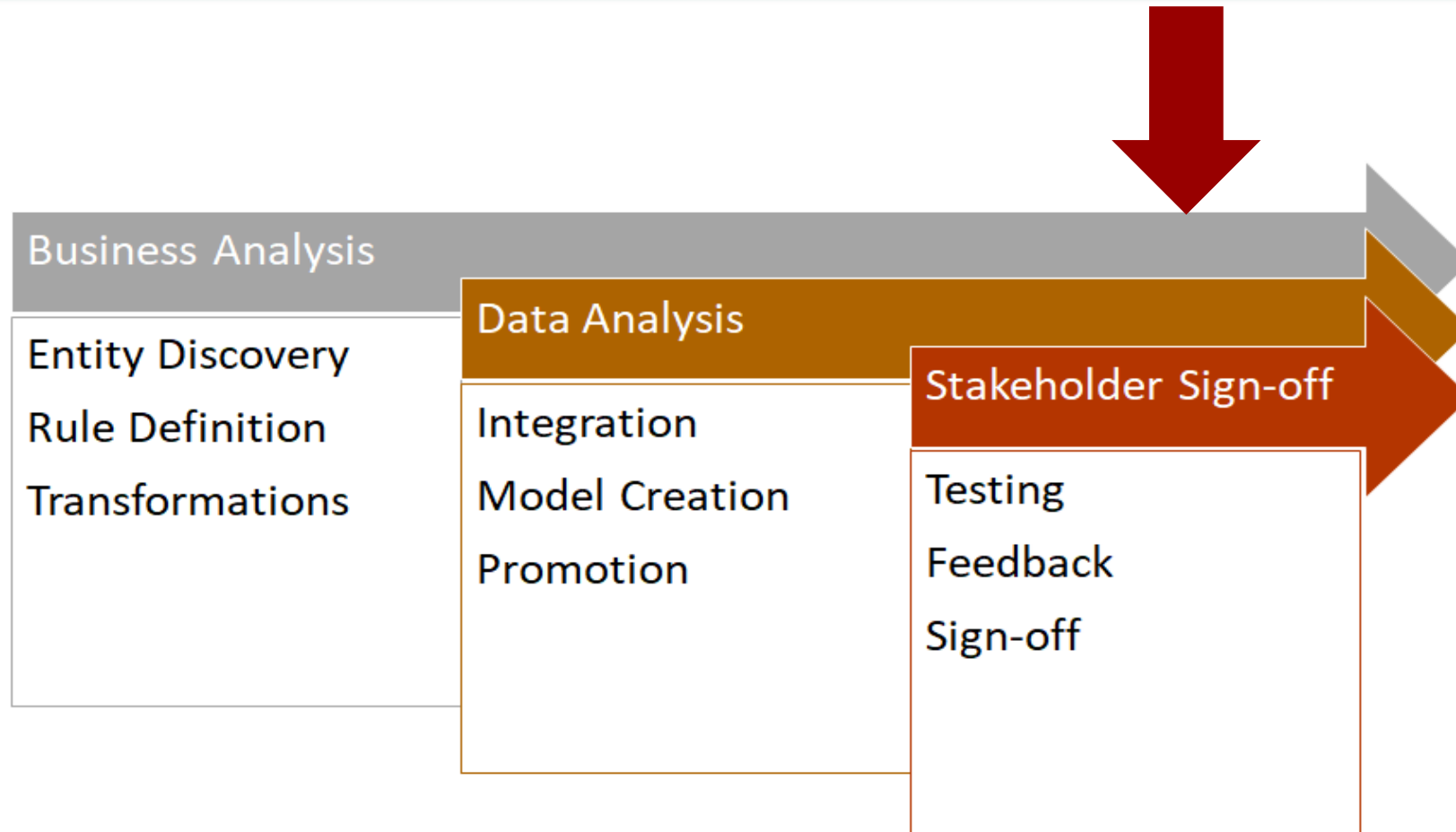
The UOW Solution

- Beginning Dec 2017
- Assembled the brain trust: four Groups - Data Services, Architecture, Business Analysis, SMS Project Team reps
- A “war room” was established
- Went to work

The UOW Solution



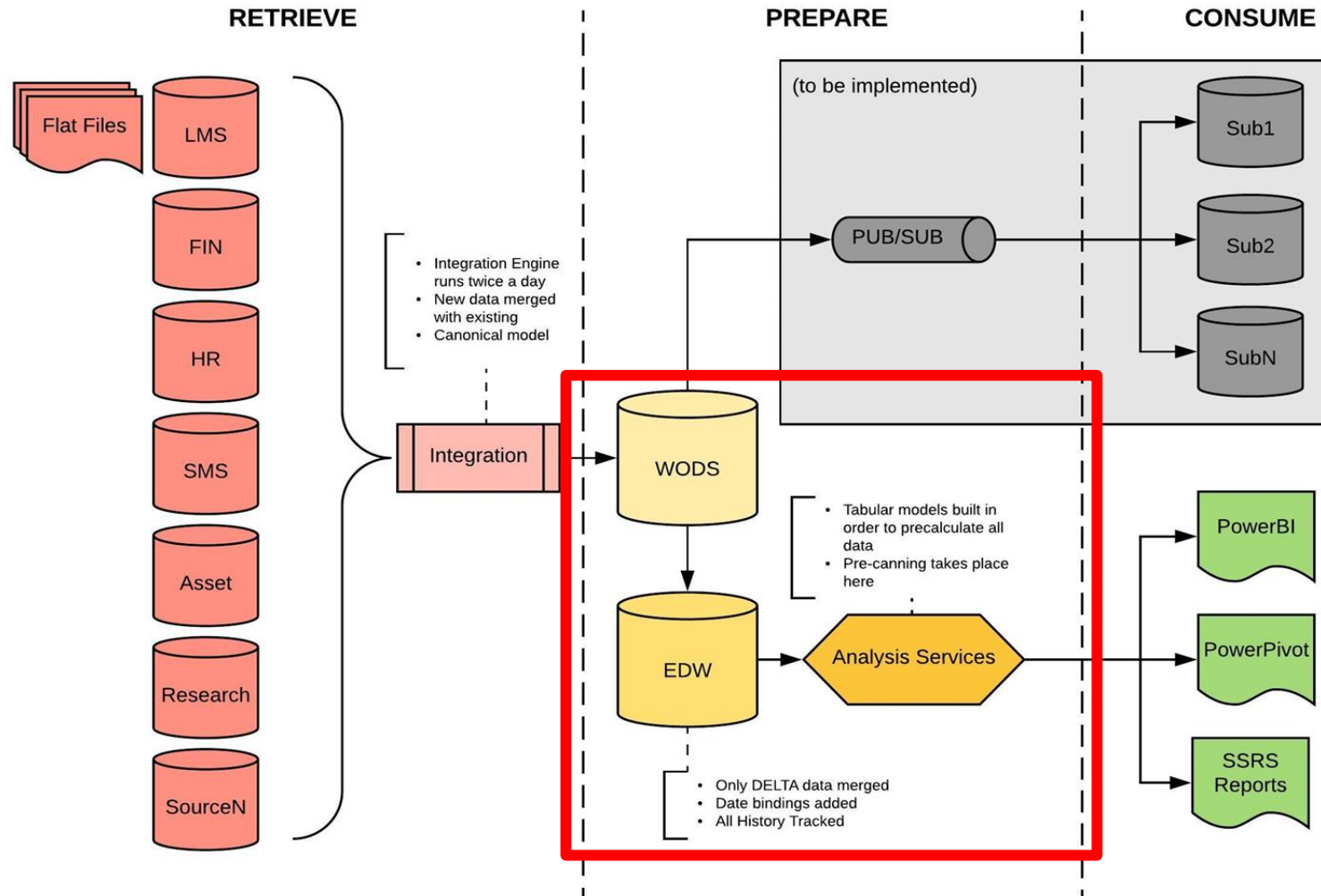
The UOW Solution



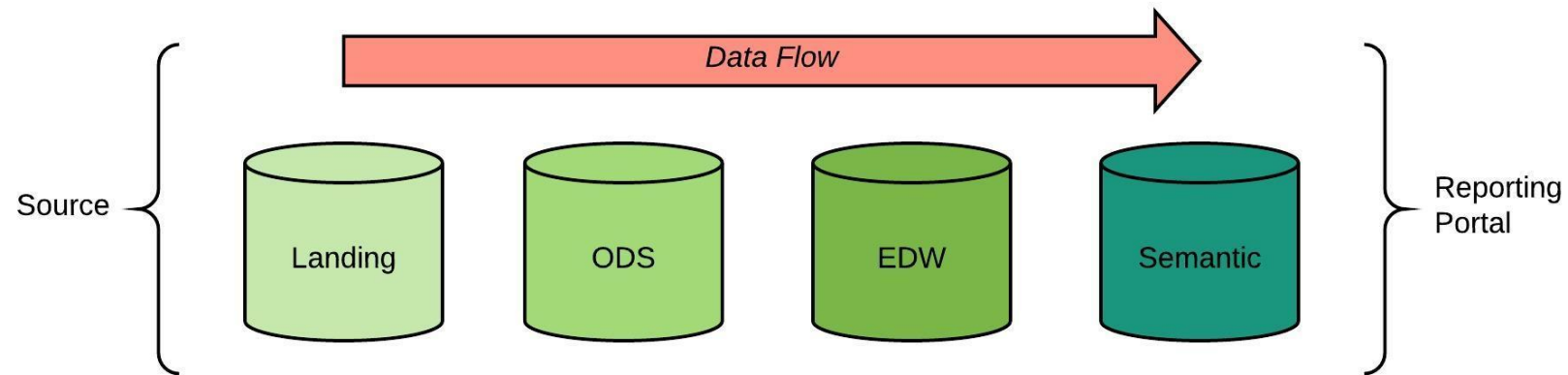
A Data Platform is Born

- This all led to...

The UOW Solution



Getting down to brass tacks:



Let's break it down.. Landing

- Standalone database
- Decoupled entry point
- Exact replica of source
- Schema defines source system - example:

```
moodle.mdl_grade_items  
oraclefin.AP_SUPPLIERS  
peoplesoft.PS_EMPLOYEES  
sits.ins_stu
```

- Data can be injected via a number of means, BCP, Windows Load, OPENDATASOURCE, API GET
- Data is not altered - no transformations, one for one
- Some filtering may be applied
- Exception being foreign data types (Oracle, PostgreSQL, GreenTree, etc)

The UOW Solution - ODS

- Primary system
- Canonical Data Model
- Amalgamate all disparate sources into one model
- Agnostic of source - underlying system(s) can be replaced
- Data is sanitized and constrained
- Anomalies or erroneous records that fail referential integrity checks are moved sideways during processing to a logging table for exception reporting - they are not exposed in the consumption layer
- Clean data is then added, or merged with existing if updating (no truncates)
- Orphans (source record deleted) are removed in **reverse constraint order**
- API layer across the top
- Tables are ranked by criticality - a rollback is created before each load and if a critical table fails the rollback state is applied otherwise the run continues

-Replica of the ODS
- Incrementally loaded using a delta queue from ODS changed/created data - we can do this because we have last updated dates in the ODS. Logic as follows:

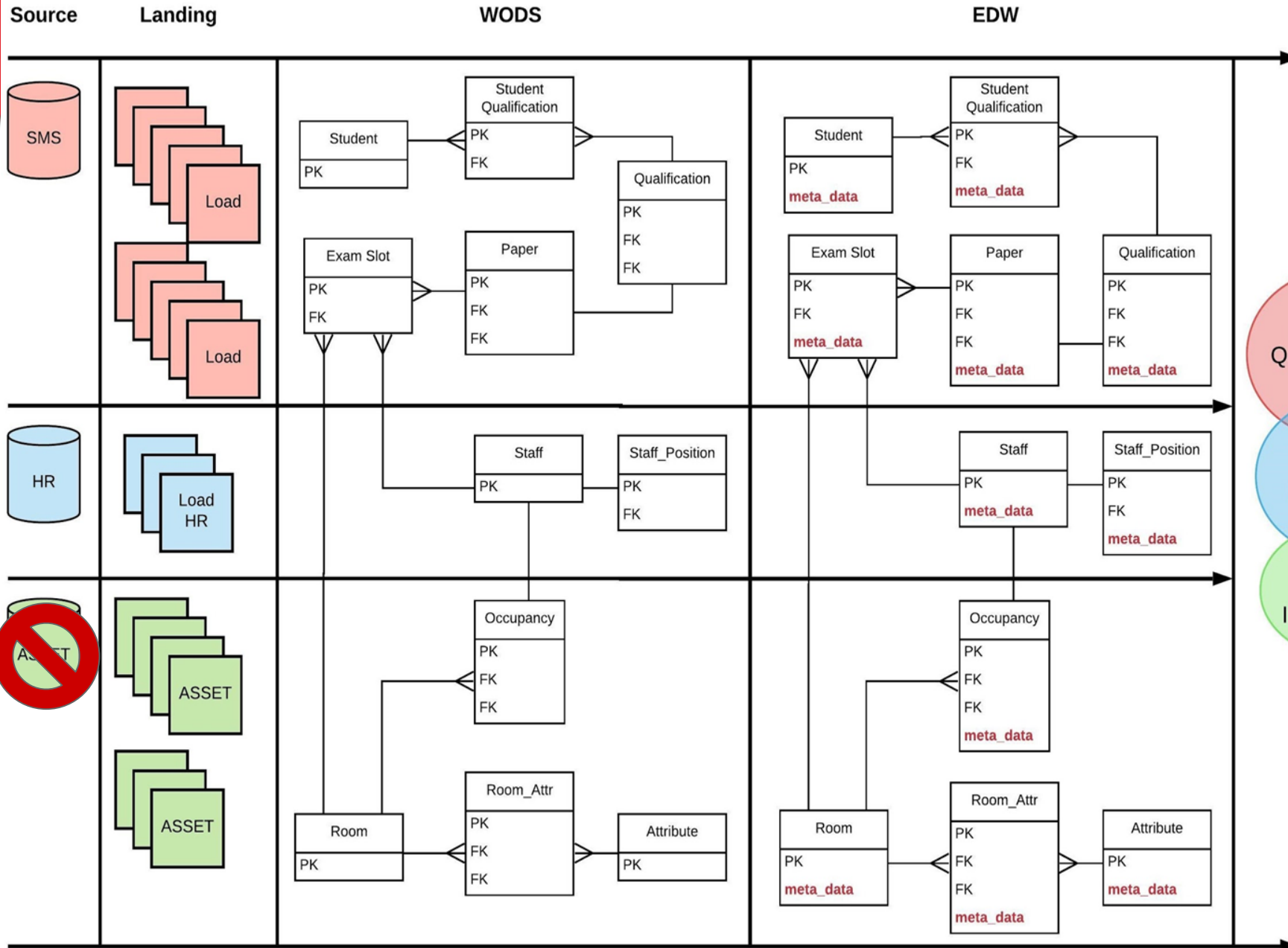
Find the high-water mark (max start time from last ETL run, for that object)

Collect all data from the ODS for that object with an updated datetime > the high water

- **Metadata is added:** date created, date updated, start date, end date, active flag
- Each table is treated as “Type 2” slowly changing
- Number of columns listened to per object is variable - high churn, lower number of columns
- Pulls data once every 24 hours
- Foundational structure for all reporting

The UOW Solution - Semantic

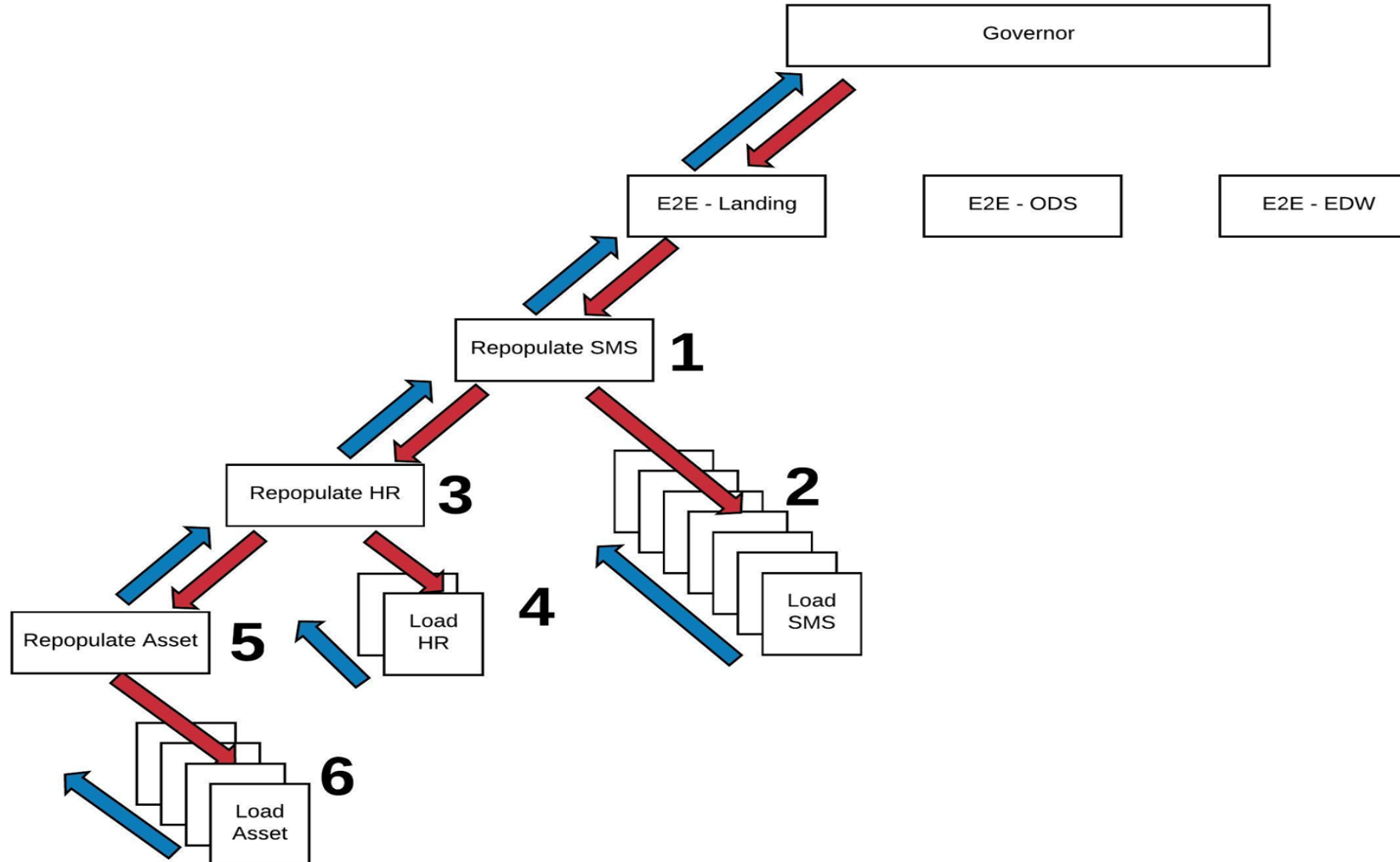
- Liked entities grouped into large, flat objects
- All transformational calculations completed
- Aggregations (roll ups) pre-calculated
- Tabular Model Analysis Cubes are built
- Any number of variations can be built - if more than one predicate is required a new view is built. Otherwise the single predicate is applied in the reporting layer



Optimisations during load

- Cascading Stored Procedures:
 - 1) A single Governor calls one stored procedure for each Tranche (Landing, ODS, EDW, Semantic Layer):
 - 2) The parent stored procedure in Tranche 1 iterates through all children before passing control flow back to the Gov'
 - 3) The Governor then passes control flow to the parent stored procedure in Tranche 2, etc
 - 4) The loop continues until all tranches have finished running, or a critical error is encountered
- How this looks visually:

Landing Cascade



Landing SQL 1/2



```
--Generate cursor of tables and their fields
if object_id('tempdb..#temp_TableCursor') is not null
drop table #temp_TableCursor
create table #temp_TableCursor
(
  SchemaName varchar(50)
  , TableName varchar(50)
  , FieldList varchar(max)
)
set @SQL =
'
select distinct
  sch.name as SchemaName
  , obj.name as TableName
  , substring( --strip first comma
    (
      select ', ' + col.name
      from ' + @WODSServer + '.WaikatoODS_Landing.sys.columns col
      where obj.object_id = col.object_id
      order by col.column_id
      for XML path ('''')
    )
    , 3, 99999) as FieldList
from ' + @WODSServer + '.WaikatoODS_Landing.sys.objects obj
join ' + @WODSServer + '.WaikatoODS_Landing.sys.schemas sch on (obj.schema_id = sch.schema_id)
where sch.name = ''sits''
and obj.Type = ''U'' --Table
and obj.name like ''' + @TablePattern + '''
'

insert #temp_TableCursor
exec (@SQL)

declare TableCursor cursor
for select *
from #temp_TableCursor

--Loop through tables and their fields
open TableCursor
fetch next from TableCursor into @SchemaName, @TableName, @FieldList
while @@fetch_status = 0 and @error_status = 0
begin
set @run_etl_start = sysdatetime()
set @run_etl_end = null
set @run_rows_inserted = null
set @run_error_log = 'Successful'
```

Landing SQL 2/2



```
--Loop through tables and their fields
open TableCursor
fetch next from TableCursor into @SchemaName, @TableName, @FieldList
while @@fetch_status = 0 and @error_status = 0
begin
    set @run_etl_start = sysdatetime()
    set @run_etl_end = null
    set @run_rows_inserted = null
    set @run_error_log = 'Successful'

    --Generate and execute statements to truncate and insert
    set @SQL =
    truncate table ' + @WODSDatabase + '.' + @SchemaName + '.' + @TableName + ';

    insert ' + @SchemaName + '.' + @TableName + '(' + @FieldList + ')'
    select ' + @FieldList + '
    from ' + @SITSServer + '.' + @SITSDatabase + '.dbo.' + @TableName + ' with (nolock);

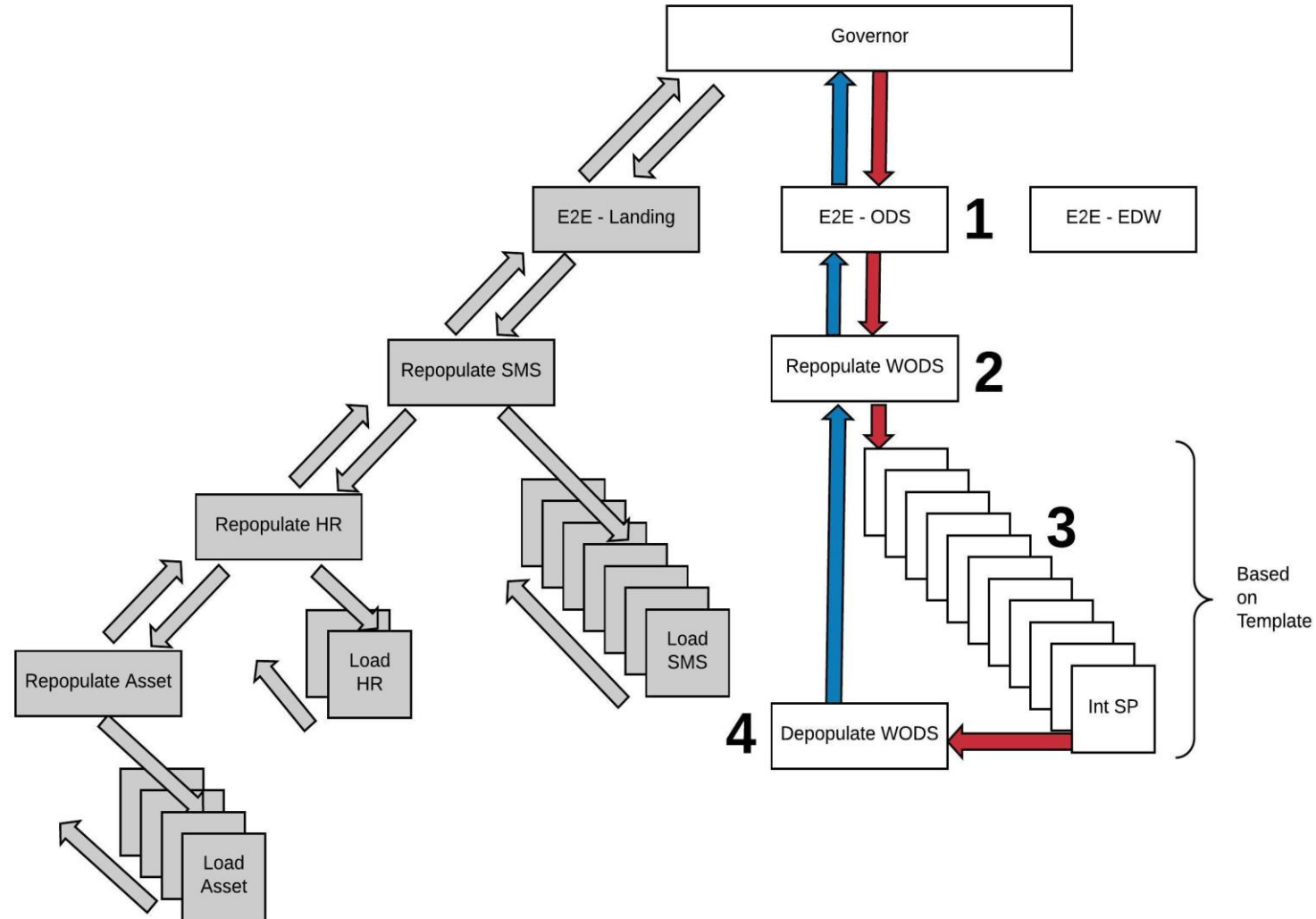
    set @Message = '--' + @TableName
    raiserror (@Message, 0, 1) with nowait
    raiserror (@SQL, 0, 1) with nowait

    if @mode <> 'debug'
    begin
        --Start Run Log
        insert meta.landing_run_log(source_system, table_name, etl_start)
        values (@SITSServer, @SchemaName + '.' + @TableName, @run_etl_start)
        set @run_update_key =
        (
            select max(run_log_id)
            from meta.landing_run_log
            where etl_start = @run_etl_start
            and etl_end is null
        )
        BEGIN TRY
            --Try to execute the SQL
            exec (@SQL)
            set @run_rows_inserted = @@rowcount
        END TRY
        BEGIN CATCH
            --Exit the loop if an error occurs executing the SQL
            set @error_status = 1
            set @run_error_log = error_message()
            set @error_log = 'Error occurred in ' + @procedure_name + ': ' + error_message()
        END CATCH

        --End Run Log
        set @run_etl_end = sysdatetime()
        update meta.landing_run_log
        set etl_end = @run_etl_end
        , rows_inserted = @run_rows_inserted
        , error_log = @run_error_log
        where run_log_id = @run_update_key
    end

    fetch next from TableCursor into @SchemaName, @TableName, @FieldList
end
close TableCursor
deallocate TableCursor
```

ODS Cascade



```
-----  
--Create hierarchy based on FK relationships  
-----  
;with Fkeys as  
(  
  select distinct  
    OnTable = OnTable.name  
    , OnSchema = onSchema.name  
    , AgainstTable = AgainstTable.name  
  from sys.foreignkeys fk  
  join sys.objects onTable on (fk.fkeyid = onTable.object_id)  
  join sys.objects againstTable on (fk.rkeyid = againstTable.object_id)  
  join sys.schemas onSchema on (onTable.schema_id = onSchema.schema_id)  
  where AgainstTable.type = 'U'  
  and OnTable.type = 'U'  
  and OnTable.name <> AgainstTable.name --ignore self joins  
)  
--Tables which have a load  
, MyData as  
(  
  select OnTable = o.name  
    , OnSchema = sch.name  
    , AgainstTable = FKeys.againstTable  
  from sys.objects o  
  join sys.schemas sch on (o.schema_id = sch.schema_id)  
  left join FKeys on (o.name = FKeys.onTable)  
  where o.type = 'U'  
  and 'load_' + o.name in  
  (  
    select name  
    from sys.objects  
    where type = 'P'  
    and name like 'load[_]%'  
  )  
  and o.name not in ('etl_monitor_log') --postload  
)
```

1

```
--Table hierarchy  
, MyRecursion as  
(  
  --Base CTE  
  select TableName = OnTable  
    , SchemaName = OnSchema  
    , Lvl = 1  
  from MyData  
  where AgainstTable is null  
  union all  
  --Recursive CTE  
  select TableName = OnTable  
    , SchemaName = OnSchema  
    , Lvl = r.Lvl + 1  
  from MyData d  
  join MyRecursion r on (d.AgainstTable = r.TableName)  
)  
--Main  
select cast(SchemaName as varchar(255)) as SchemaName  
  , cast(TableName as varchar(255)) as TableName  
  , max(Lvl) as NodeLevel  
into #temp_hierarchy  
from MyRecursion  
group by TableName  
  , SchemaName  
order by max(Lvl) desc  
  , TableName desc
```

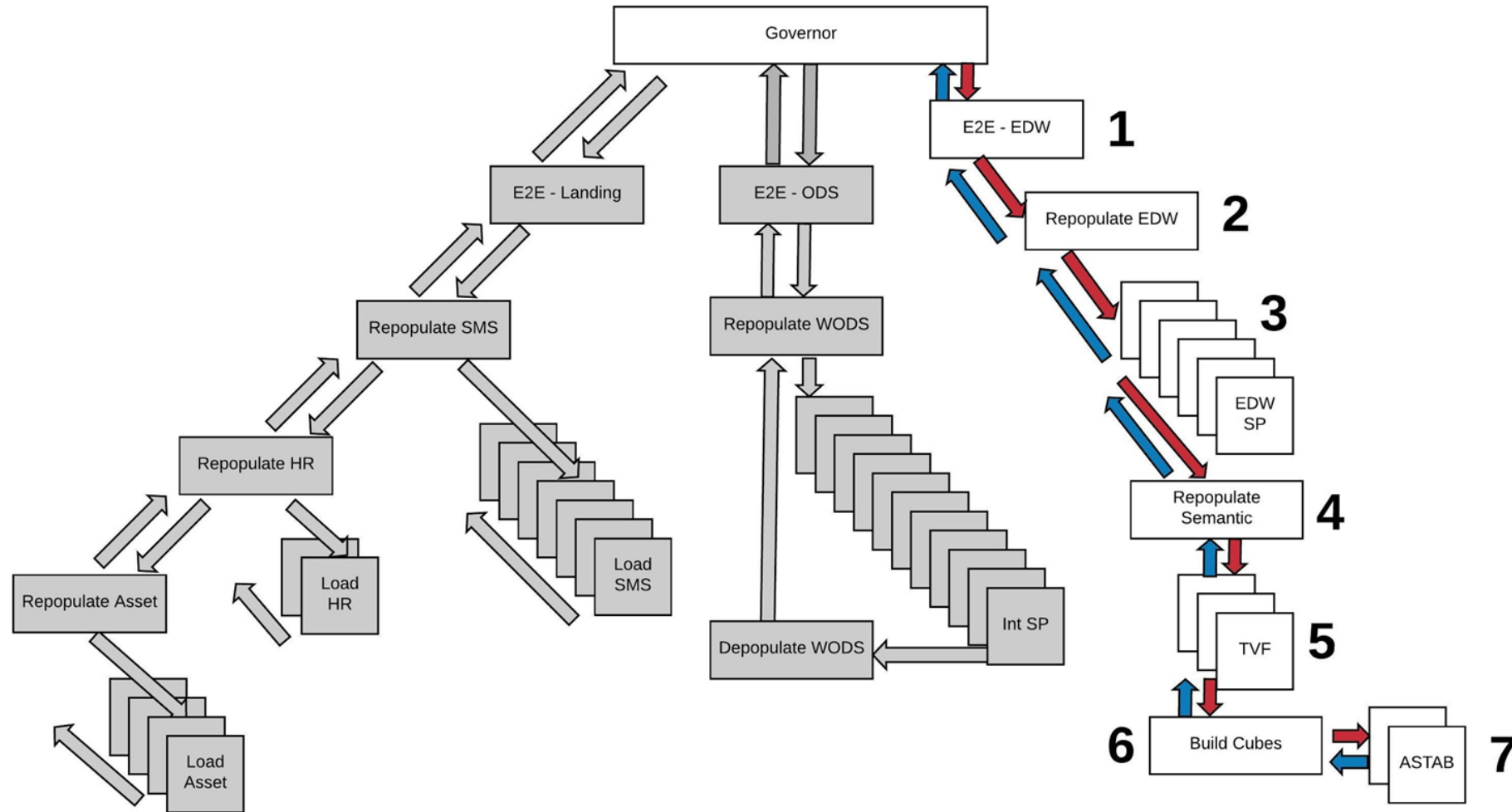
2

ODS SQL 2/2



```
-----  
--Repopulate in Hierarchy Sequence  
-----  
declare HierarchyCursor cursor forward_only read_only  
for select NodeLevel, SchemaName, TableName  
from #temp_hierarchy  
order by NodeLevel, TableName  
  
open HierarchyCursor  
fetch next from HierarchyCursor into @NodeLevel, @SchemaName, @TableName  
while @@fetch_status = 0 and @error_status = 0  
begin  
begin  
BEGIN TRY  
set @subprocedure_name = 'integration.load_' + @TableName  
if object_id(@subprocedure_name) is not null  
begin  
if @NodeLevel <> @PreviousNodeLevel  
begin  
set @message = '--Level ' + cast(@NodeLevel as varchar)  
raiserror (@message, 0, 1) with nowait  
set @PreviousNodeLevel = @NodeLevel  
end  
  
set @SQL = 'exec ' + @subprocedure_name  
raiserror (@SQL, 0, 1) with nowait  
if @mode <> 'debug'  
begin  
--Try to execute the subprocedure  
exec (@SQL)  
set @error_log = isnull(  
(  
select top 1 error_log  
from meta.etl_run_log  
where etl_start >= @etl_start  
and table_name = @SchemaName + '.' + @TableName  
order by etl_start desc, run_log_id desc  
)  
, 'Unsuccessful')  
--Exit the loop if an error occurs inside the subprocedure  
end  
  
end  
END TRY  
END TRY
```

EDW, Semantic Layer, Cubes



EDW - Order of Stored Procs



- + [icon] integration.load_StudentDemographic
- + [icon] integration.load_StudentDisability
- + [icon] integration.load_StudentEducation
- + [icon] integration.load_StudentEthnicity
- + [icon] integration.load_StudentFeesFree
- + [icon] integration.load_StudentFeeTransaction
- + [icon] integration.load_StudentHall
- + [icon] integration.load_StudentIwi
- + [icon] integration.load_StudentPaperCreditTransfer
- + [icon] integration.load_StudentPaperOccurrence
- + [icon] integration.load_StudentPersonalDetail
- + [icon] integration.load_StudentQualification
- + [icon] integration.load_StudentQualificationYear
- + [icon] integration.load_StudentScholarship
- + [icon] integration.load_StudentSecondaryEducation
- + [icon] integration.load_StudentSubcategory
- + [icon] integration.load_StudentTertiaryEducation
- + [icon] integration.load_Subcategory
- + [icon] integration.load_SubcategoryType
- + [icon] integration.load_Subject
- + [icon] integration.load_SubjectPaperOccurrence
- + [icon] integration.load_Supervisor
- + [icon] integration.load_SupervisorType
- + [icon] integration.load_Suspension
- + [icon] integration.load_TeachingActivity
- + [icon] integration.load_TeachingActivityRoom
- + [icon] integration.load_TeachingActivityType
- + [icon] integration.load_ThesisSubmission
- + [icon] integration.load_VisaLog
- + [icon] integration.repopulate_analysis_cube
- + [icon] integration.repopulate_edw
- + [icon] integration.repopulate_load_tables
- + [icon] integration.repopulate_semantic_layer
- + [icon] integration.run_e2e_edw
- + [icon] reporting._DynamicCreateMVScript
- + [icon] reporting._EFTSSummaryForEFTSActualYear
- + [icon] reporting.ETL_1000CreateLoadTableTVFs
- + [icon] reporting.ETL_2000dimReportingData
- + [icon] reporting.ETL_2010dimReportingDataEFTS
- + [icon] reporting.ETL_4010mvPaperEnrolmentSearch
- + [icon] reporting.ETL_4020mvAwardCompletions
- + [icon] reporting.ETL_4030mtvfEFTS_Core
- + [icon] reporting.ETL_4040mvGradesAnalysis
- + [icon] reporting.ETL_4050mtvfApplicationsForAdmission
- + [icon] reporting.ETL_4060mvStudentDisabilityEnrolmentsList
- + [icon] reporting.ETL_4070mvEFTS_Summary_V2
- + [icon] reporting.ETL_4080mvTRM_Allocation
- + [icon] reporting.ETL_4084mvTRM_3_EFTS_Current_2019
- + [icon] reporting.ETL_4085mvTRM_4_Composite_Key
- + [icon] reporting.ETL_5000mvFinanceMonthEndCY
- + [icon] reporting.ETL_5001mvFinanceMonthEndNY
- + [icon] reporting.ETL_6000EFTSHistory_2019
- + [icon] reporting.ETL_6010mvUsageLogging
- + [icon] reporting.ETL_9000ReprocessAllSSASTabularModels
- + [icon] reporting.ETL_9999SendEmail_LastRunETLReport

SQL Demo cont..



```
-----  
--Populate #temp_hierarchy  
-----  
select s.name + '.' + o.name as SchemaProcedureName  
into #temp_hierarchy  
from sys.objects o  
join sys.schemas s on (o.schema_id = s.schema_id)  
where o.type = 'P'  
and s.name = 'integration'  
and o.name like 'load%'  
  
-----  
--Repopulate in Integration Hierarchy Sequence  
-----  
declare HierarchyCursor cursor forward_only read_only  
for select SchemaProcedureName  
from #temp_hierarchy  
order by SchemaProcedureName  
  
open HierarchyCursor  
fetch next from HierarchyCursor into @SchemaProcedureName  
while @@fetch_status = 0 and @error_status= 0  
begin  
BEGIN TRY  
set @sql = 'exec ' + @SchemaProcedureName  
raiserror (@sql, 0, 1) with nowait  
if @mode <> 'debug' exec (@sql)  
END TRY  
BEGIN CATCH  
set @error_log = 'Error occurred in ' + @SchemaProcedureName + ': ' + error_message()  
set @error_status = 1  
END CATCH  
fetch next from HierarchyCursor into @SchemaProcedureName  
end  
close HierarchyCursor  
deallocate HierarchyCursor
```

EDW - Order of Stored Procs

- + integration.load_StudentDemographic
- + integration.load_StudentDisability
- + integration.load_StudentEducation
- + integration.load_StudentEthnicity
- + integration.load_StudentFeesFree
- + integration.load_StudentFeeTransaction
- + integration.load_StudentHall
- + integration.load_StudentIwi
- + integration.load_StudentPaperCreditTransfer
- + integration.load_StudentPaperOccurrence
- + integration.load_StudentPersonalDetail
- + integration.load_StudentQualification
- + integration.load_StudentQualificationYear
- + integration.load_StudentScholarship
- + integration.load_StudentSecondaryEducation
- + integration.load_StudentSubcategory
- + integration.load_StudentTertiaryEducation
- + integration.load_Subcategory
- + integration.load_SubcategoryType
- + integration.load_Subject
- + integration.load_SubjectPaperOccurrence
- + integration.load_Supervisor
- + integration.load_SupervisorType
- + integration.load_Suspension
- + integration.load_TeachingActivity
- + integration.load_TeachingActivityRoom
- + integration.load_TeachingActivityType
- + integration.load_ThesisSubmission
- + integration.load_VisaLog
- + integration.repopulate_analysis_cube
- + integration.repopulate_edw
- + integration.repopulate_load_tables
- + integration.repopulate_semantic_layer
- + integration.run_e2e_edw
- + reporting._DynamicCreateMVScript
- + reporting._EFTSSummaryForEFTSActualYear
- + reporting.ETL_1000CreateLoadTableTVFs
- + reporting.ETL_2000dimReportingData
- + reporting.ETL_2010dimReportingDataEFTS
- + reporting.ETL_4010mvPaperEnrolmentSearch
- + reporting.ETL_4020mvAwardCompletions
- + reporting.ETL_4030mtvfEFTS_Core
- + reporting.ETL_4040mvGradesAnalysis
- + reporting.ETL_4050mtvfApplicationsForAdmission
- + reporting.ETL_4060mvStudentDisabilityEnrolmentsList
- + reporting.ETL_4070mvEFTS_Summary_V2
- + reporting.ETL_4080mvTRM_Allocation
- + reporting.ETL_4084mvTRM_3_EFTS_Current_2019
- + reporting.ETL_4085mvTRM_4_Composite_Key
- + reporting.ETL_5000mvFinanceMonthEndCY
- + reporting.ETL_5001mvFinanceMonthEndNY
- + reporting.ETL_6000EFTSHistory_2019
- + reporting.ETL_6010mvUsageLogging
- + reporting.ETL_9000ReprocessAllSSASTabularModels
- + reporting.ETL_9999SendEmail_LastRunETLReport

Dynamic Processing ASTAB 1/2

```
-----  
--Populate #temp_hierarchy  
-----  
select s.name + '.' + o.name as SchemaProcedureName  
into #temp_hierarchy  
from sys.objects o  
join sys.schemas s on (o.schema_id = s.schema_id)  
where o.type = 'P'  
and s.name = 'reporting'  
and o.name like 'ETL_%'  
and o.name not like 'ETL_9%' --exclude 9000+  
  
-----  
--Repopulate in Reporting Hierarchy Sequence  
-----  
declare HierarchyCursor cursor forward_only read_only  
for select SchemaProcedureName  
from #temp_hierarchy  
order by SchemaProcedureName  
  
open HierarchyCursor  
fetch next from HierarchyCursor into @SchemaProcedureName  
while @@fetch_status = 0 and @error_status= 0  
begin  
BEGIN TRY  
set @sql = 'exec ' + @SchemaProcedureName  
raiserror (@sql, 0, 1) with nowait  
if @mode <> 'debug' exec (@sql)  
END TRY  
BEGIN CATCH  
set @error_log = 'Error occurred in ' + @SchemaProcedureName + ': ' + error_message()  
set @error_status = 1  
END CATCH  
fetch next from HierarchyCursor into @SchemaProcedureName  
end  
close HierarchyCursor  
deallocate HierarchyCursor
```

EDW - Order of Stored Procs

- + [icon] integration.load_StudentDemographic
- + [icon] integration.load_StudentDisability
- + [icon] integration.load_StudentEducation
- + [icon] integration.load_StudentEthnicity
- + [icon] integration.load_StudentFeesFree
- + [icon] integration.load_StudentFeeTransaction
- + [icon] integration.load_StudentHall
- + [icon] integration.load_StudentIwi
- + [icon] integration.load_StudentPaperCreditTransfer
- + [icon] integration.load_StudentPaperOccurrence
- + [icon] integration.load_StudentPersonalDetail
- + [icon] integration.load_StudentQualification
- + [icon] integration.load_StudentQualificationYear
- + [icon] integration.load_StudentScholarship
- + [icon] integration.load_StudentSecondaryEducation
- + [icon] integration.load_StudentSubcategory
- + [icon] integration.load_StudentTertiaryEducation
- + [icon] integration.load_Subcategory
- + [icon] integration.load_SubcategoryType
- + [icon] integration.load_Subject
- + [icon] integration.load_SubjectPaperOccurrence
- + [icon] integration.load_Supervisor
- + [icon] integration.load_SupervisorType
- + [icon] integration.load_Suspension
- + [icon] integration.load_TeachingActivity
- + [icon] integration.load_TeachingActivityRoom
- + [icon] integration.load_TeachingActivityType
- + [icon] integration.load_ThesisSubmission
- + [icon] integration.load_VisaLog
- + [icon] integration.repopulate_analysis_cube
- + [icon] integration.repopulate_edw
- + [icon] integration.repopulate_load_tables
- + [icon] integration.repopulate_semantic_layer
- + [icon] integration.run_e2e_edw
- + [icon] reporting._DynamicCreateMVScript
- + [icon] reporting._EFTSSummaryForEFTSActualYear
- + [icon] reporting.ETL_1000CreateLoadTableTVFs
- + [icon] reporting.ETL_2000dimReportingData
- + [icon] reporting.ETL_2010dimReportingDataEFTS
- + [icon] reporting.ETL_4010mvPaperEnrolmentSearch
- + [icon] reporting.ETL_4020mvAwardCompletions
- + [icon] reporting.ETL_4030mtvfEFTS_Core
- + [icon] reporting.ETL_4040mvGradesAnalysis
- + [icon] reporting.ETL_4050mtvfApplicationsForAdmission
- + [icon] reporting.ETL_4060mvStudentDisabilityEnrolmentsList
- + [icon] reporting.ETL_4070mvEFTS_Summary_V2
- + [icon] reporting.ETL_4080mvTRM_Allocation
- + [icon] reporting.ETL_4084mvTRM_3_EFTS_Current_2019
- + [icon] reporting.ETL_4085mvTRM_4_Composite_Key
- + [icon] reporting.ETL_5000mvFinanceMonthEndCY
- + [icon] reporting.ETL_5001mvFinanceMonthEndNY
- + [icon] reporting.ETL_6000EFTSHistory_2019
- + [icon] reporting.ETL_6010mvUsageLogging
- + [icon] reporting.ETL_9000ReprocessAllSSASTabularModels
- + [icon] reporting.ETL_9999SendEmail_LastRunETLReport



Dynamic Processing ASTAB 1/2

```
declare @RowCounter INT
declare @RowCount INT
declare @cube_name VARCHAR(50)
declare @run_id INT;

-- Temp table to hold the names of the Tabular Models
declare @db_table TABLE (
    i int not null identity(1,1) PRIMARY KEY CLUSTERED,
    DATABASE_ID SYSNAME NOT NULL);
insert into @db_table (DATABASE_ID)

select CATALOG_NAME
    from openquery([BI-ASTAB-ALL], 'SELECT * FROM $SYSTEM.DBSHEMA_CATALOGS')
set @RowCount = @@rowcount
set @RowCounter = 1

-- Template of XMLA SSAS Command to do full rebuild of a Model
DECLARE @SSAS_Command_Template AS NVARCHAR(MAX) = '
{
  "refresh": {
    "type": "full",
    "objects": [
      {
        "database": "MODEL_NAME"
      }
    ]
  }
}';
```

Dynamic Processing ASTAB 2/2



```
-- Loop over all Tabular Models
while @RowCount <= @RowCount
begin

    -- Get name of Model as Text
    select @cube_name = CAST(DATABASE_ID AS NVARCHAR(50)) from @db_table where i = @RowCount

    -- Logging (start)
    insert into [dbo].[edw_run_log_reportingETLTabular] (cube_name, build_start) values (@cube_name, GETDATE())
    SELECT @run_id = SCOPE_IDENTITY();

    DECLARE @LoggingMessage NVARCHAR(MAX)

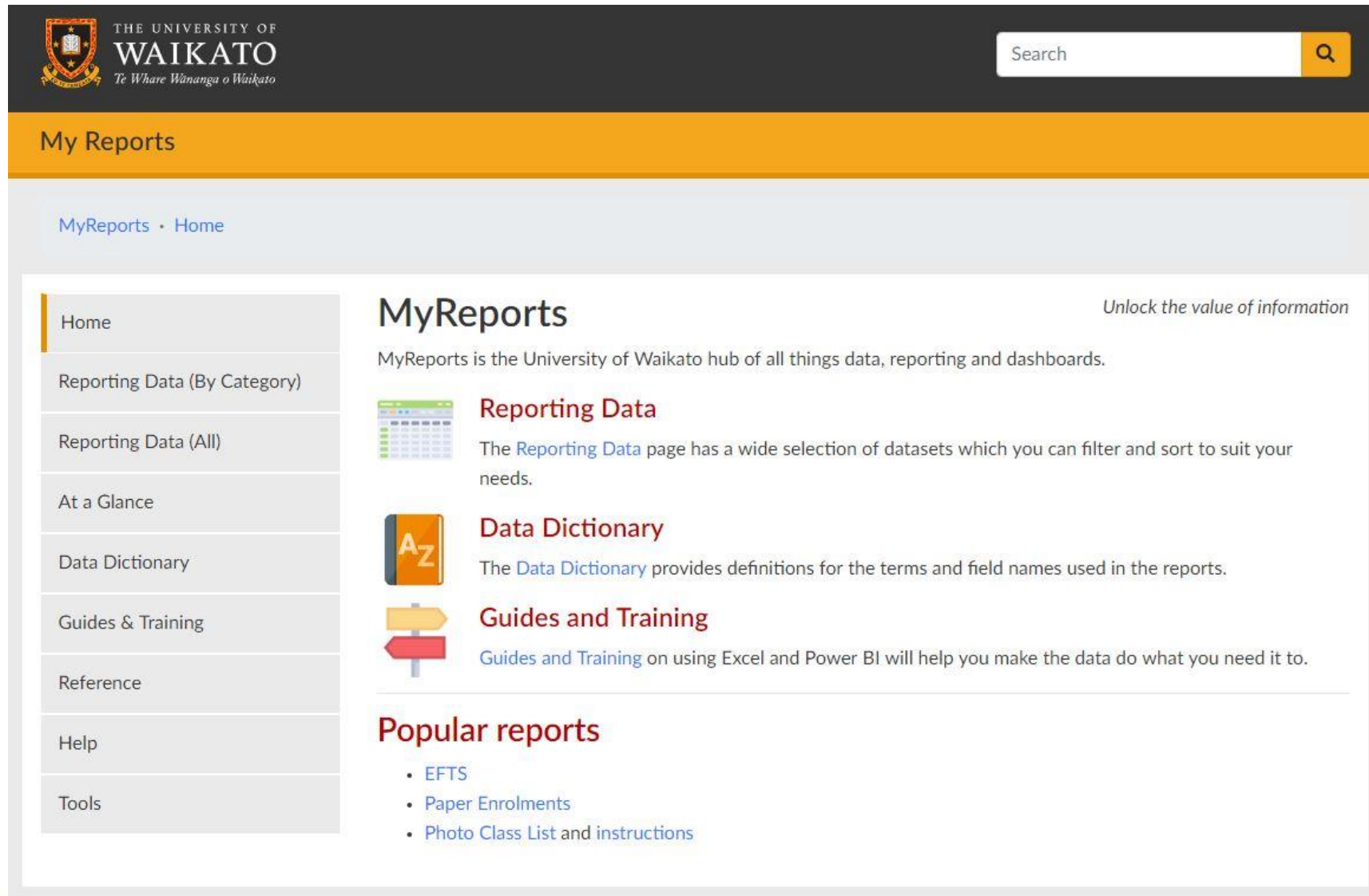
    IF LEFT(@cube_name,1)='_'
        SET @LoggingMessage = 'Skip - Manual Process Only'
    ELSE
    BEGIN
        -- Build command for current model from Template
        DECLARE @SSAS_Command NVARCHAR(MAX) = REPLACE(@SSAS_Command_Template, 'MODEL_NAME', @cube_name);
        print @SSAS_Command;

        BEGIN TRY
            EXEC (@SSAS_Command) AT [BI-ASTAB-ALL];
            SET @LoggingMessage = 'Success'
        END TRY
        BEGIN CATCH
            SET @LoggingMessage = ERROR_MESSAGE();
            SET @LoggingMessage2 = 'Model Processing Failure(s)';
        END CATCH
    END

    --Logging (end)
    update [dbo].[edw_run_log_reportingETLTabular]
    set build_end = GETDATE(), message=@LoggingMessage
    where run_log_id = @run_id

    set @RowCount = @RowCount+ 1

end;
```



The screenshot shows the 'My Reports' section of the University of Waikato website. At the top left is the university logo and name. A search bar is located at the top right. Below the header is a yellow banner with the text 'My Reports'. The main content area has a light blue background and includes a breadcrumb trail 'MyReports > Home'. On the left is a vertical navigation menu with items: Home, Reporting Data (By Category), Reporting Data (All), At a Glance, Data Dictionary, Guides & Training, Reference, Help, and Tools. The main content area features a 'MyReports' heading with the tagline 'Unlock the value of information' and a descriptive paragraph. Below this are three featured sections: 'Reporting Data' with a spreadsheet icon, 'Data Dictionary' with an 'AZ' icon, and 'Guides and Training' with a directional sign icon. A 'Popular reports' section at the bottom lists 'EFTS', 'Paper Enrolments', and 'Photo Class List and instructions'.

THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Search

My Reports


MyReports > Home


- Home
- Reporting Data (By Category)
- Reporting Data (All)
- At a Glance
- Data Dictionary
- Guides & Training
- Reference
- Help
- Tools


MyReports

Unlock the value of information

MyReports is the University of Waikato hub of all things data, reporting and dashboards.

 **Reporting Data**
The [Reporting Data](#) page has a wide selection of datasets which you can filter and sort to suit your needs.

 **Data Dictionary**
The [Data Dictionary](#) provides definitions for the terms and field names used in the reports.

 **Guides and Training**
[Guides and Training](#) on using Excel and Power BI will help you make the data do what you need it to.

Popular reports

- [EFTS](#)
- [Paper Enrolments](#)
- [Photo Class List and instructions](#)

Reporting Data

This page has a wide selection of dashboards, reports and Excel datasets, which you can filter and sort to suit your needs.

To get up and running working with Excel quickly and independently, make sure you check out the [Guides and Training](#).

01 Self Service Excel Data

[Admissions](#)

Summarises by counting an applicant in their most active application - Student Admissions (Applicants, Offers, Acceptance, Conversion rates, ...) with custom grouping and filtering by many common attributes (Faculty, Campus, Qualification, Semester, New or Returning, ...)

[Award Completions](#)

Qualification completions (Awards) with custom grouping and filtering by many common attributes (Faculty of Study, Qualification Type, Student Age, ...)

[EFTS](#)

EFTS (Total, % of Target) with custom grouping and filtering by many common attributes (Faculty, Funding Category, Source of Funding, ...)

[Paper Enrolments](#)

Students in each paper occurrence with custom grouping and filtering by many common attributes (Faculty, Location, Academic Year, Paper Enrolment Status, ...)

[Retention](#)

Students retained (and/or completed) from one year to the next (Enrolled, Retained, Retention %) with custom grouping and filtering by many common attributes (Faculty of Study, Qualification Type, Residency, ...)

02 Popular

Paper Search

List of papers with related attributes, such as department and location.

Photo Class List

Class lists including student details and photos.

Student Search

All students who applied or enrolled with either Jade or MyWaikato. Includes contact details.

03 Management

EFTS

EFTS (Total, % of Target) with custom grouping and filtering by many common attributes (Faculty, Funding Category, Source of Funding, ...)

04 Educational Performance

Award Completions

Qualification completions (Awards) with custom grouping and filtering by many common attributes (Faculty of Study, Qualification Type, Student Age, ...)

Board of Examiners

Provisional paper results and completions for approval by the Board of Examiners.

Failed Students

Students who have failed assessments, including assessment details.

Grade Analysis

Students marks and grades by paper occurrence.

Retention

Students retained (and/or completed) from one year to the next (Enrolled, Retained, Retention %) with custom grouping and filtering by many common attributes (Faculty of Study, Qualification Type, Residency, ...)

Tertiary Data Warehouse (TDW)

TEC SDR comparative data from all NZ Universities from past 5 years.

05 Applications, Admissions and Enrolments

Admission Turnaround

Admissions turnaround times for both Current and Completed Admissions grouped by their current location and duration in the admissions process

Admissions

Summarises by counting an applicant in their most active application - Student Admissions (Applicants, Offers, Acceptance, Conversion rates, ...) with custom grouping and filtering by many common attributes (Faculty, Campus, Qualification, Semester, New or Returning, ...)

Applications for Admission

List of everyone who has applied for admission to the University in the current or following year, including application status.

Change of Enrolment

Students with a relevant Change of Enrolment status (Change pending submission, Change pending approval, Change approved, Change approved - conditional).

Conversion Made Offer

List of international students with admission offers and their response statuses, so the University can send this to Hobsons agency for follow up.

Current Enrolments

List of students who are enrolled as of today, with contact details.

Incomplete Applications

Student applications that were last updated more than 6 days ago and are still incomplete.

Paper Enrolments

Students in each paper occurrence with custom grouping and filtering by many common attributes (Faculty, Location, Academic Year, Paper Enrolment Status, ...)

SGR Batch Admissions

All School of Graduate Research Batch Admissions, grouped by batch and status.

U18 International

Under 18 International Students

UNISStart

UNISStart Applications and Students

Grouped Entities

06 Students

- [Exam Attendance](#)
Provides printable lists of students scheduled into specific examinations in order to record their attendance.
- [Failed Students](#)
Students who have failed assessments, including assessment details.
- [Paper Enrolments](#)
Students in each paper occurrence with custom grouping and filtering by many common attributes (Faculty, Location, Academic Year, Paper Enrolment Status, ...)
- [Photo Class List](#)
Class lists including student details and photos.
- [Student Search](#)
All students who applied or enrolled with either Jade or MyWaikato. Includes contact details.
- [StudentSafe Insurance](#)
Insurance-related transactions for monthly reporting to StudentSafe provider, Marsh.
- [UNISStart](#)
UNISStart Applications and Students

07 Papers

- [Bennets Students by Paper](#)
Student numbers by occurrence to help Bennetts Bookstore plan their stock.
- [Exam Attendance](#)
Provides printable lists of students scheduled into specific examinations in order to record their attendance.
- [Library Copyright](#)
A count of all enrolled students per paper in a teaching period, to calculate associated royalties (Copyright Licensing NZ).
- [Paper Enrolments](#)
Students in each paper occurrence with custom grouping and filtering by many common attributes (Faculty, Location, Academic Year, Paper Enrolment Status, ...)
- [Paper Search](#)
List of papers with related attributes, such as department and location.
- [Staff Paper Occurrence Association](#)
List of staff associated with each paper occurrence.
- [Timetable Enrolment Monitoring](#)
Comparison of room capacity and enrolment numbers.

Home
Reporting Data (By Category)
Reporting Data (All)
At a Glance
Data Dictionary
Guides & Training
Reference
Help
Tools

Reporting Data

This page has a wide selection of dashboards, reports and Excel datasets, which you can filter and sort to suit your needs.

To get up and running working with Excel quickly and independently, make sure you check out the [Guides and Training](#).

A

[Admission Turnaround](#)

Admissions turnaround times for both Current and Completed Admissions grouped by their current location and duration in the admissions process

[Admissions](#)

Summarises by counting an applicant in their most active application - Student Admissions (Applicants, Offers, Acceptance, Conversion rates, ...) with custom grouping and filtering by many common attributes (Faculty, Campus, Qualification, Semester, New or Returning, ...)

[Applications for Admission](#)

List of everyone who has applied for admission to the University in the current or following year, including application status.

[Award Completions](#)

Qualification completions (Awards) with custom grouping and filtering by many common attributes (Faculty of Study, Qualification Type, Student Age, ...)

B

[Bennets Students by Paper](#)

Student numbers by occurrence to help Bennetts Bookstore plan their stock.

[Board of Examiners](#)

Provisional paper results and completions for approval by the Board of Examiners.

C

[Change of Enrolment](#)

Students with a relevant Change of Enrolment status (Change pending submission, Change pending approval, Change approved, Change approved - conditional).

[Conversion Made Offer](#)

List of international students with admission offers and their response statuses, so the University can send this to Hobsons agency for follow up.

[Current Enrolments](#)

List of students who are enrolled as of today, with contact details.

Home

Reporting Data (By Category)

Reporting Data (All)

At a Glance

Data Dictionary

Guides & Training

Reference

Help

Tools

Admissions

Access Granted

Summarises by counting an applicant in their most active application - Student Admissions (Applicants, Offers, Acceptance, Conversion rates, ...) with custom grouping and filtering by many common attributes (Faculty, Campus, Qualification, Semester, New or Returning, ...)

Report Links

1.  [Applicant Admission Summary - Current Academic Year \(Excel Pivot Table\)](#)

Related Report Links

Additional Notes

This self-service dataset summarises the number of applicants who have submitted an admission for a qualification within a particular academic year.

Default View

When opening this dataset the information is filtered to include 'New to University of Waikato' applicants only. This can be changed by altering the slicers in the left hand column.

An applicant is counted once per year regardless of the number of admission applications they have submitted. The most active application is determined by checking if the student has enrolled in the qualification related to that application. If not, a check is made to see if the applicant has accepted an offer, followed by an (unconditional or conditional) offer being given, followed by application being submitted (and it has not been declined, withdrawn or lapsed). The most active record is then used to determine next level information, such as Qualification or Faculty.

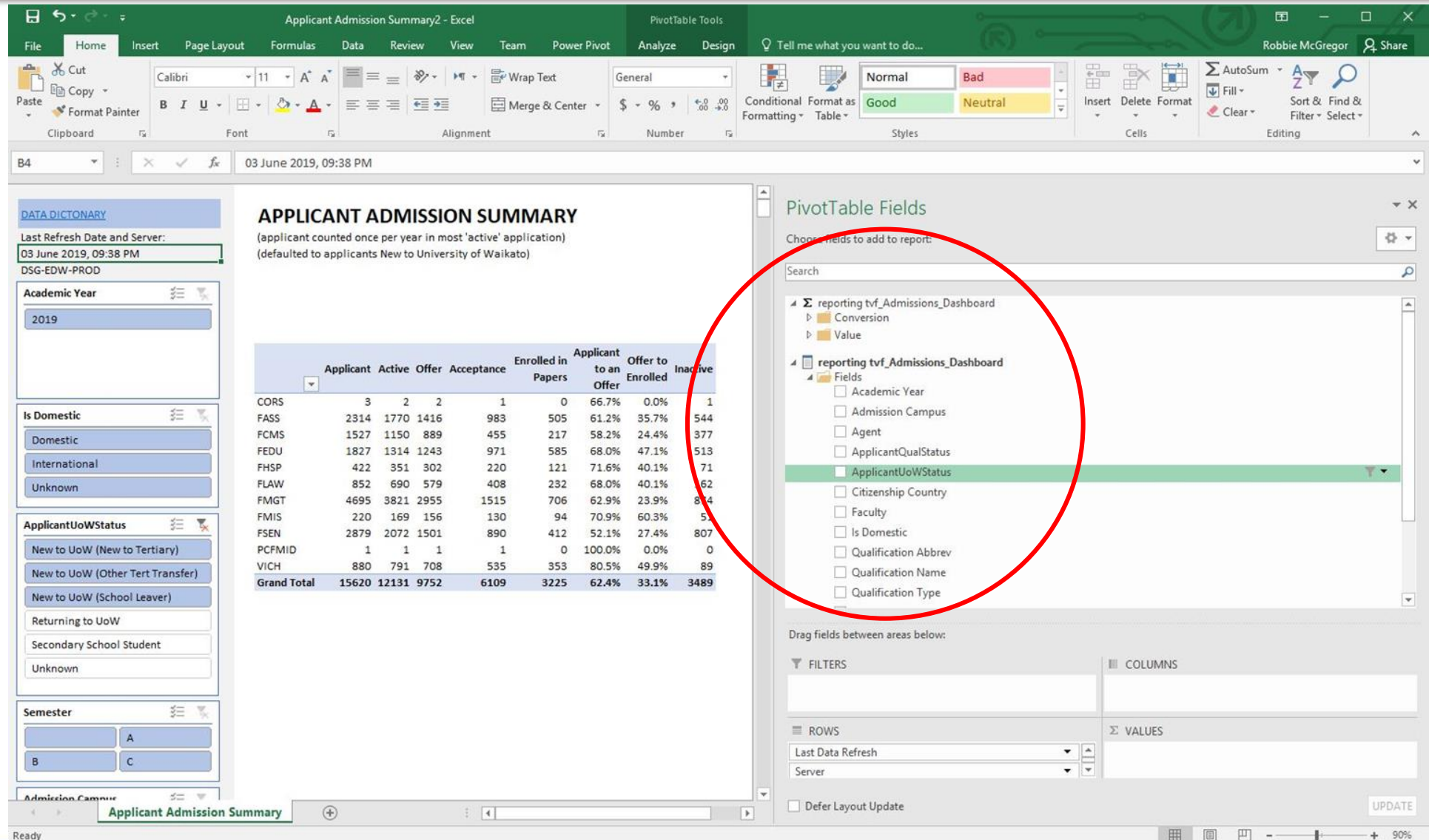
Values

- Active Indicators to show if conversion to enrolment is possible (Active, Inactive)
- Progress Indicators to show how far an applicant has got through the application process (Applicant, Conditional, Unconditional, Offer, Acceptance, Enrolled in Papers, Enrolled in Qual) and
- Conversion Ratios (Applicant to Acceptance, Applicant to an Offer, Applicant to Unconditional Offer, Applicant to Conditional Offer, Offer to Acceptance, Offer to Enrolment)

Terms used in this reporting data

 [Academic Year](#)

Power Pivot Example 1/2



APPLICANT ADMISSION SUMMARY
(applicant counted once per year in most 'active' application)
(defaulted to applicants New to University of Waikato)

	Applicant	Active	Offer	Acceptance	Enrolled in Papers	Applicant to an Offer	Offer to Enrolled	Inactive
CORS	3	2	2	1	0	66.7%	0.0%	1
FASS	2314	1770	1416	983	505	61.2%	35.7%	544
FCMS	1527	1150	889	455	217	58.2%	24.4%	377
FEDU	1827	1314	1243	971	585	68.0%	47.1%	513
FHSP	422	351	302	220	121	71.6%	40.1%	71
FLAW	852	690	579	408	232	68.0%	40.1%	62
FMGT	4695	3821	2955	1515	706	62.9%	23.9%	844
FMIS	220	169	156	130	94	70.9%	60.3%	51
FSEN	2879	2072	1501	890	412	52.1%	27.4%	807
PCFMID	1	1	1	1	0	100.0%	0.0%	0
VICH	880	791	708	535	353	80.5%	49.9%	89
Grand Total	15620	12131	9752	6109	3225	62.4%	33.1%	3489

PivotTable Fields

reporting tvf_Admissions_Dashboard

- Conversion
- Value

reporting tvf_Admissions_Dashboard

- Fields
 - Academic Year
 - Admission Campus
 - Agent
 - ApplicantQualStatus
 - ApplicantUoWStatus**
 - Citizenship Country
 - Faculty
 - Is Domestic
 - Qualification Abbrev
 - Qualification Name
 - Qualification Type

Drag fields between areas below:

FILTERS

COLUMNS

ROWS

- Last Data Refresh
- Server

VALUES

Defer Layout Update

UPDATE

Power Pivot Example 2/2

PivotTable Fields

Choose fields to add to report:

Search

- reporting tvf_Admissions_Dashboard
 - Conversion
 - Applicant to Acceptance
 - Applicant to an Offer
 - Applicant to Conditional Offer
 - Applicant to Enrolled
 - Applicant to Unconditional Offer
 - Offer to Acceptance
 - Offer to Enrolled
 - Value
 - Acceptance
 - Active
 - Applicant
 - Conditional
 - Enrolled in Papers
 - Enrolled in Qual
 - Inactive
 - Offer
 - Unconditional
- reporting tvf_Admissions_Dashboard
 - Fields

Grouped Entities



With E as

```
(Select distinct spo.StudentId, spo.StudentQualificationCode, spo.QualificationCode, po.AcademicYear, CAST(Min (SPO.StartDate) AS DATE) [Min Start Date]
from [dbo].[tvf_StudentPaperOccurrence] (NULL) spo INNER JOIN [dbo].[tvf_PaperOccurrence] (NULL) po ON po.PaperOccurrenceCode = spo.PaperOccurrenceCode
where po.AcademicYear = @Year and spo.EnrolmentStatusCode in ('ENR','WDN', 'WDIP')
group by spo.StudentId, spo.StudentQualificationCode, spo.QualificationCode, po.AcademicYear),
```

S AS (

```
Select A.*,
Case when A.[UoW Decision 1] = 'Conditional Offer' OR A.[UoW Decision 2] = 'Conditional Offer' then 1 else 0 End AS [Has Conditional],
Case when A.[UoW Decision 1] = 'Unconditional Offer' OR A.[UoW Decision 2] = 'Unconditional Offer' then 1 else 0 End AS [Has Unconditional],
Case when A.[UoW Decision 1] like '%Offer%' OR A.[UoW Decision 2] like '%Offer%' then 1 else 0 End AS [Has an offer],
Case when A.[Student Reponse 1] = 'Accept Offer' OR A.[Student Reponse 2] = 'Accept Offer' then 1 else 0 End AS [Has Accepted Offer],
Case when A.Enrolled = 'Enrolled' then 1 else 0 End AS [Has Enrolled], COALESCE(E.[Min Start Date], E1.[Min Start Date], '19000101') [Min Start Date],
[reporting].[fn_GetIsDomestic] (A.ResidencyStatusCode) [Is Domestic],
SUM(A.[Negative Status]) OVER (PARTITION By A.[Student ID]) [Student Has Negative Status],
Count(A.[Student ID]) OVER (PARTITION By A.[Student ID]) [Student Has Applications]
```

```
from [reporting].[tvf_ApplicationsForAdmissions](@date, @Year) A
left join E on E.AcademicYear = A.[Academic Year] and E.StudentId = A.[Student ID] and E.StudentQualificationCode = A.StudentQualificationCode
left join E E1 on E1.AcademicYear = A.[Academic Year] and E1.StudentId = A.[Student ID] and E1.QualificationCode = A.QualificationCode
where A.[Academic Year] = @Year),
```

S1 as

```
(Select S.*,
Max(S.[Has Conditional]) OVER (PARTITION By S.[Student ID]) [Student Has Conditional Offer],
Max(S.[Has Unconditional]) OVER (PARTITION By S.[Student ID]) [Student Has Unconditional Offer],
Max(S.[Has an offer]) OVER (PARTITION By S.[Student ID]) [Student Has An Offer],
Max(S.[Has Accepted Offer]) OVER (PARTITION By S.[Student ID]) [Student Has Accepted Offer],
Max(S.[Has Enrolled]) OVER (PARTITION By S.[Student ID]) [Student Has Enrolled],
RANK ( ) OVER ( partition by S.[Student ID] Order by S.[Min Start Date] DESC, S.[Has Enrolled], S.[Has Accepted Offer], [Has an offer],
S.[StudentQualificationCode], S.[Semester], S.[ApplicationSequence], S.[QualificationCode], S.[ApplicantUoWStatus], S.[StudentUoWStatus],
S.[Student ID]) [Include if 1],
```

```
CASE when S.[Student Has Applications] - S.[Student Has Negative Status] = 0 then 1 else 0 END [Student is Inactive],
CASE when S.[Student Has Applications] - S.[Student Has Negative Status] > 0 then 1 else 0 END [Student is Active],
CASE when DATEPART(Year, S.[Min Start Date]) = @Year then 1 else 0 END [Has Enrolled Papers]
```

from S

)

```
Select distinct
S1.Server [Server],
S1.[Last Data Refresh],
S1.[Student ID],
S1.[Academic Year],
S1.Semester,
S1.Residency,
S1.[Qualification Abbrev],
S1.[QualificationCode],
S1.[Qualification Type],
```

EFTS by Day Example

- Home
- Reporting Data (By Category)
- Reporting Data (All)
- At a Glance**
- Data Dictionary
- Guides & Training
- Reference
- Help
- Tools



Related Report Links

- [EFTS](#)

- Why no star schema and why it's not needed:
 - Queries are required under reports regardless
 - Academic data model is simple enough
 - Low churn
 - Canonical Relational Model \neq Source Relational Model
- The Wedge – what it is and why it's needed
- Consumer confidence - transparent logs, low failure rate, alerts, monitoring

Thank you

“May all of your problems be technical ones”