



■ ■ ■ ■ **Practical Uses of Data
Services with Star Schemas**

**Scott Person, Senior Consultant,
Tiber Solutions**



■ ■ ■ ■ **Creating Great ETLs in Data Services**

**Scott Person, Senior Consultant,
Tiber Solutions**

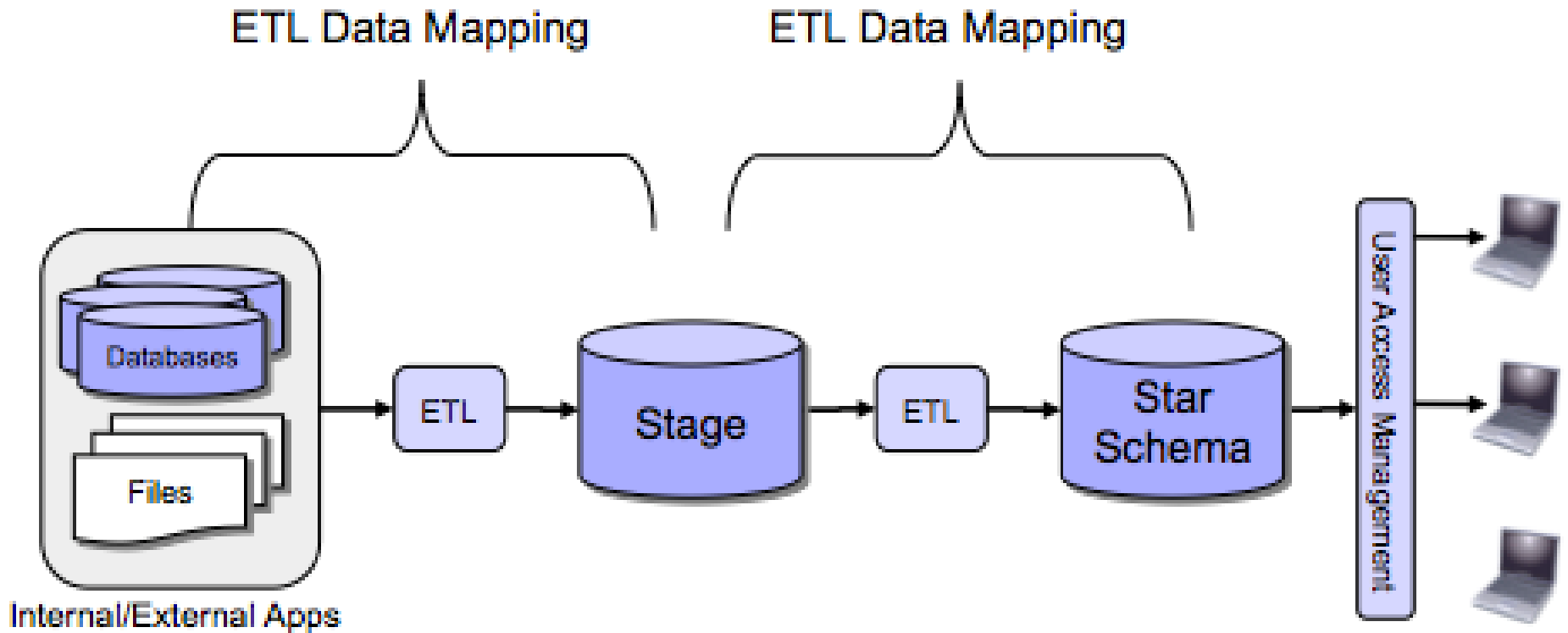
Goals

- Easy
- Low delta over non-great
- Maintainable
- Robust

Successful ETL Implementations

- Know what you're going to do
- Do it well
- Verify that it was done
- Prove that you did it well

Overall ETL Process



Stage Tables → Dimension Tables → Bridge Tables → Fact Tables

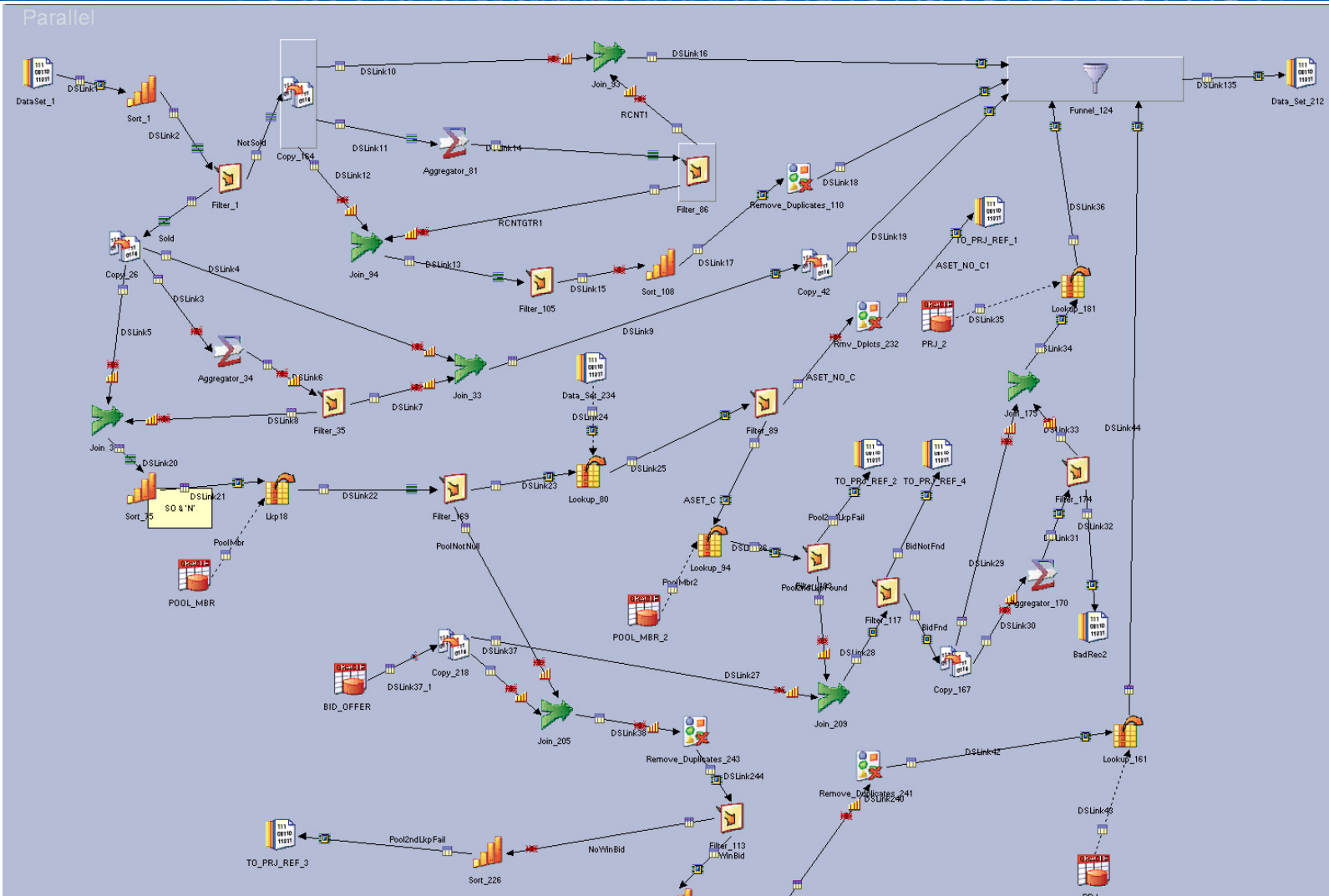
Know What You're Going to Do: ETL Mapping Sheets

Data Warehouse Customer																	
Table Name	Dimension																
View Name	Customer																
Display Name	Customer																
Description	The Customer dimension includes all corporate customers																
Used in schemas	Orders, Returns, CustomerCare, Shipping																
Target																	
Column Name	Display Name	Description	Attribute Group	Datatype	Size	Precision	Key?	FK To	NULL?	Default Value	SCD Type	Source System	Source Schema	Source Table	Source Field Name	Source Datatype	ETL Rules
customer_key	Customer Key	Surrogate primary key	Identifiers	int			PK		N								
account_number	Account Number	Account Number from the transaction system	Identifiers	varchar	10				N			POS	Sales	Customer	AccountNumber	varchar(10)	
customer_type	Customer Type	The type of the customer based on our relationship	Identifiers	char	10				N		1	Derived	Sales	Customer	CustomerType		Decode CustomerType from S/I to Reseller/Individual else Unknown
customer_id	Customer Id	Customer account number and full name (Last, First Middle)	Identifiers	varchar	100				N		1	Derived					AcctNum + ' ' + DW.FullName
customer_title	Customer Title	Courtesy title	Name and address	char	5				N		1	POS	Person	Contact	Title	nvarchar(8)	
customer_first_name	Customer First Name	Customer's first name	Name and address	varchar	30				N		1	POS	Person	Contact	FirstName	nvarchar(50)	
customer_middle_name	Customer Middle Name	Customer's middle name (often blank)	Name and address	varchar	30				N		1	POS	Person	Contact	MiddleName	nvarchar(50)	Map NULL to empty string
customer_last_name	Customer Last Name	Customer's last name	Name and address	varchar	30				N		1	POS	Person	Contact	LastName	nvarchar(50)	
customer_full_name	Customer Full Name	Customer's full name as Last, First Middle	Name and address	varchar	100				N		1	Derived					LastName + ', ' + FirstName + ' ' + MiddleName
birth_date	Customer Birth Date	Customer's date of birth	Demographics	datetime							1	POS	Sales	Individual	Demographics	xml	Shred Demographics: <BirthDate>
customer_email	Customer Email	Customer's email address	Name and address	varchar	50				N		1	POS	Person	Contact	EmailAddress	nvarchar(50)	
income_range	IncomeRange	Customer's annual income	Demographics	varchar	50				N		1	POS	Sales	Individual	Demographics	xml	Shred Demographics: <YearlyIncome>
total_children	TotalChildren	Customer's total number of children	Demographics	tinyint					N		1	POS	Sales	Individual	Demographics	xml	Shred Demographics: <TotalChildren>
education	Education	Customer's education level	Demographics	varchar	30				N		1	POS	Sales	Individual	Demographics	xml	Shred Demographics: <Education>
occupation	Occupation	Customer's general occupation (eg Managerial)	Demographics	varchar	30				N		1	POS	Sales	Individual	Demographics	xml	Shred Demographics: <Occupation>
phone	Phone	Customer's phone number	Name and address	varchar	20				N		1	POS	Person	Contact	Phone	nvarchar(25)	
address_line1	AddressLine1	First line of customer's address	Name and address	varchar	60				N		1	POS	Sales	CustomerA	AddressLine1	nvarchar(60)	Join from Person.Contact on CustomerID; pick up the most recent address
address_line2	AddressLine2	2nd line of customer's address (often blank)	Name and address	varchar	60				N		1	POS	Sales	CustomerA	AddressLine2	nvarchar(60)	see notes for AddrLine1; be sure to map null entries to empty string
postal_code	PostalCode	Postal code, eg zip code	Name and address	varchar	15				N		2	POS	Person	Address	PostalCode	nvarchar(15)	see notes for AddrLine1
city	City	City, cleaned up by way of postal code	Name and address	varchar	100				N		2	Derived				nvarchar(30)	Out of scope for Phase 1
state	State	State or Province	Name and address	varchar	50				N		2	POS	Person	StateProvin	Name	nvarchar(50)	Join from Person.Address.StateProvinceId
country	Country	Country	Name and address	varchar	50				N		2	POS	Person	CountryReg	Name	nvarchar(50)	Join from Person.StateProvince.CountryRegionCode
current_row_ind	Current Row Ind	Is this the current row for this member (Y/N)?	Housekeeping	char	1				N			Derived					Standard SCD-2
effective_date	Effective Date	When did this row become valid for this member?	Housekeeping	datetime					N			Derived					Standard SCD-2
expiration_date	Expiration Date	When did this row become invalid? (12/31/9999 if current row)	Housekeeping	datetime					N	12/31/9999		Derived					Standard SCD-2

Do It Well: Why use design patterns?

- Consistency
- Define and maintain best practices
- Avoid the redevelopment spiral

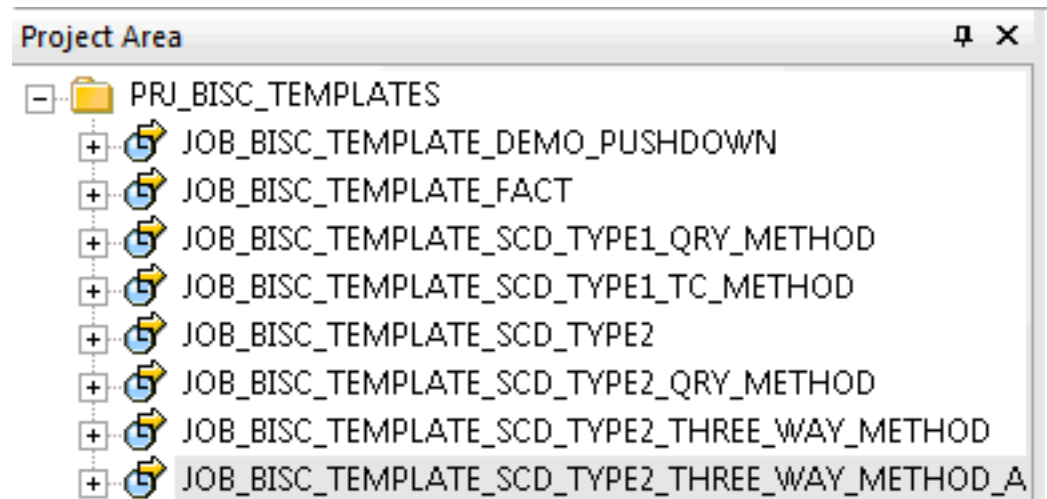
Do It Well: Design Patterns



Do It Well: Design Patterns

Create a templates project

- In central repository
- Fully functional (sample data)
- Properly documented
- Contains project standards for frequent patterns
 - Slowly Changing Dimensions
 - Fact Tables
 - Lookups

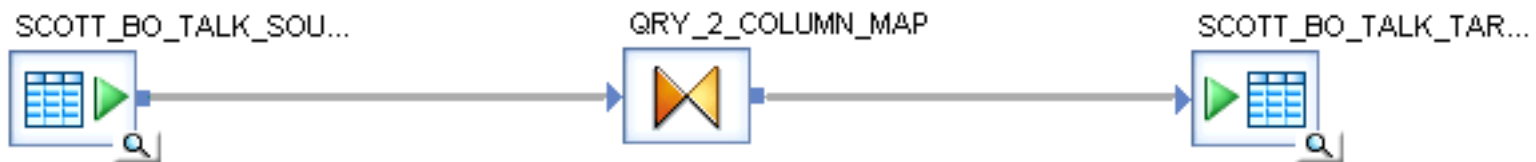


Do It Well: Staging

- “Always” Stage
 - Isolates dependency on the source systems
 - Provides a snapshot for testing and troubleshooting
- Keep it as simple as possible – no transformation logic
- CDC code if necessary – usually date based

Do It Well: Staging Pushdown SQL Example

This:



```
INSERT /*+ APPEND */ INTO "BISC_ST"."SCOTT_BO_TALK_TARGET1" ( "NK" , "DTE" )  
SELECT  "SCOTT_BO_TALK_SOURCE1"."NK"  NK , "SCOTT_BO_TALK_SOURCE1"."DTE"  DTE  
FROM    "BISC_ST"."SCOTT_BO_TALK_SOURCE1" "SCOTT_BO_TALK_SOURCE1"  
WHERE   ( "SCOTT_BO_TALK_SOURCE1"."UPD_DTTM" > (SYSDATE - 30) )
```

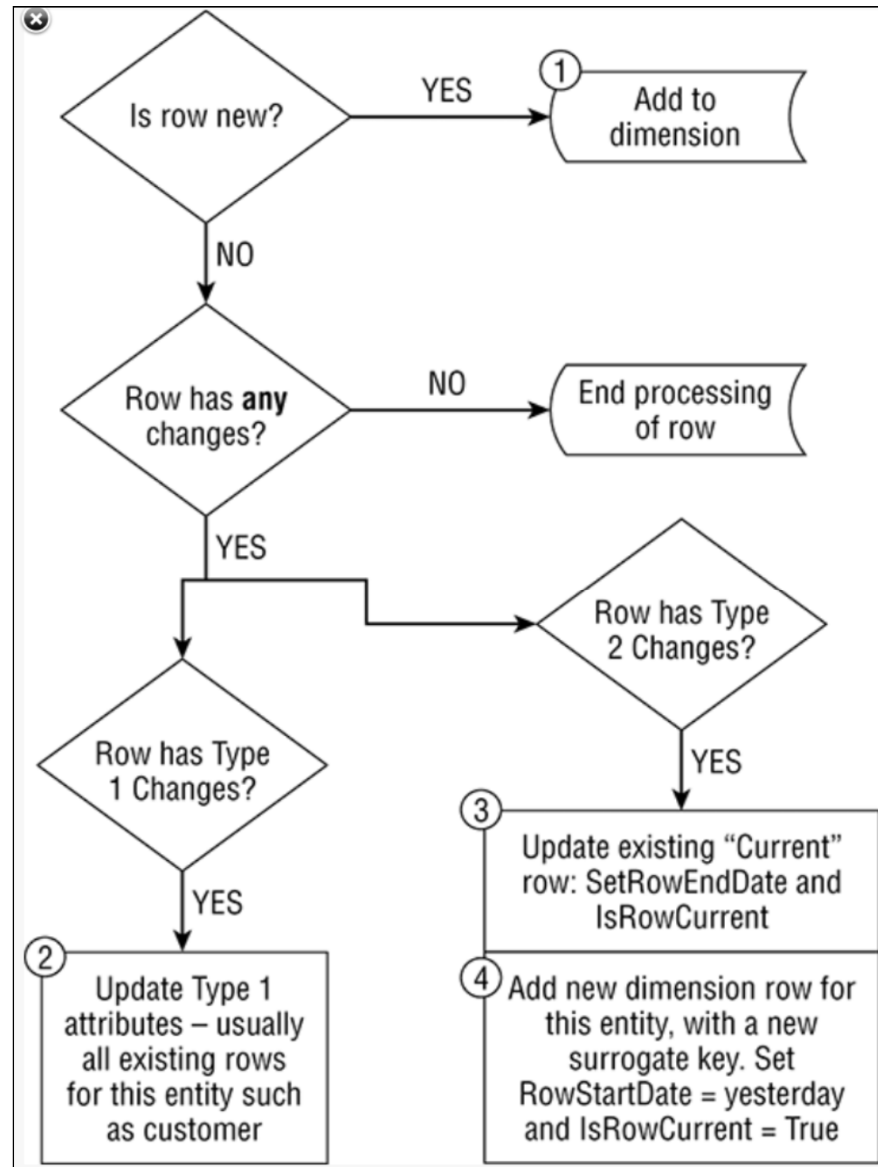
NOT This:



Do It Well: SCD Rules

- Type 1 – updates
- Type 2 – insert new record; track versions using dates
- Usually have both type 1 and 2 in the same table.
- Every record has a surrogate key

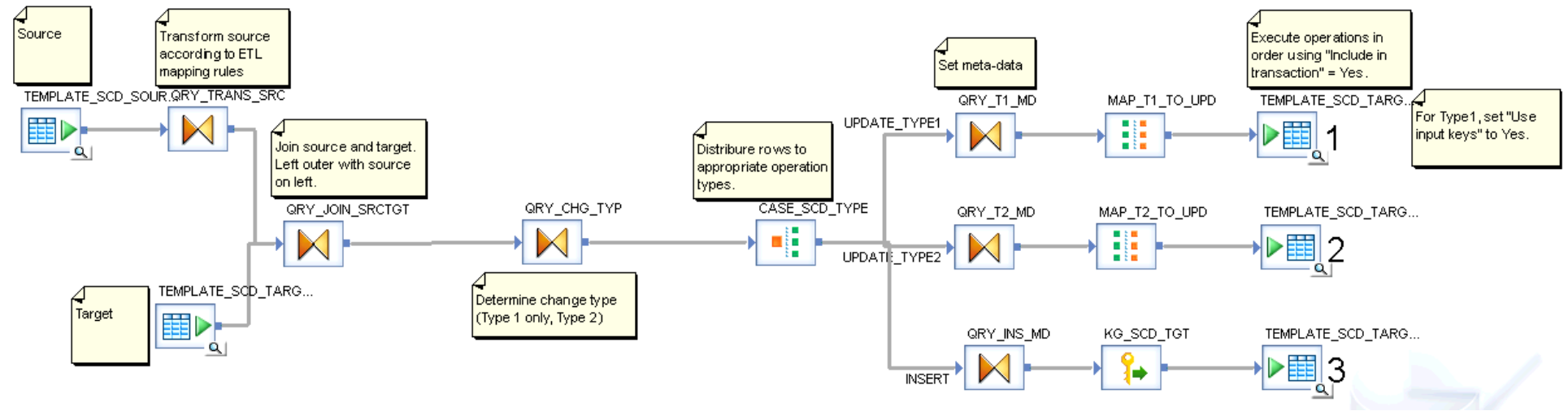
Do It Well: SCD Rules



Do It Well: Dimension Tables (SCD 1 & 2)

Data Services Slowly Changing Dimension Template

* Uses three way method rather than scripts or table compare.



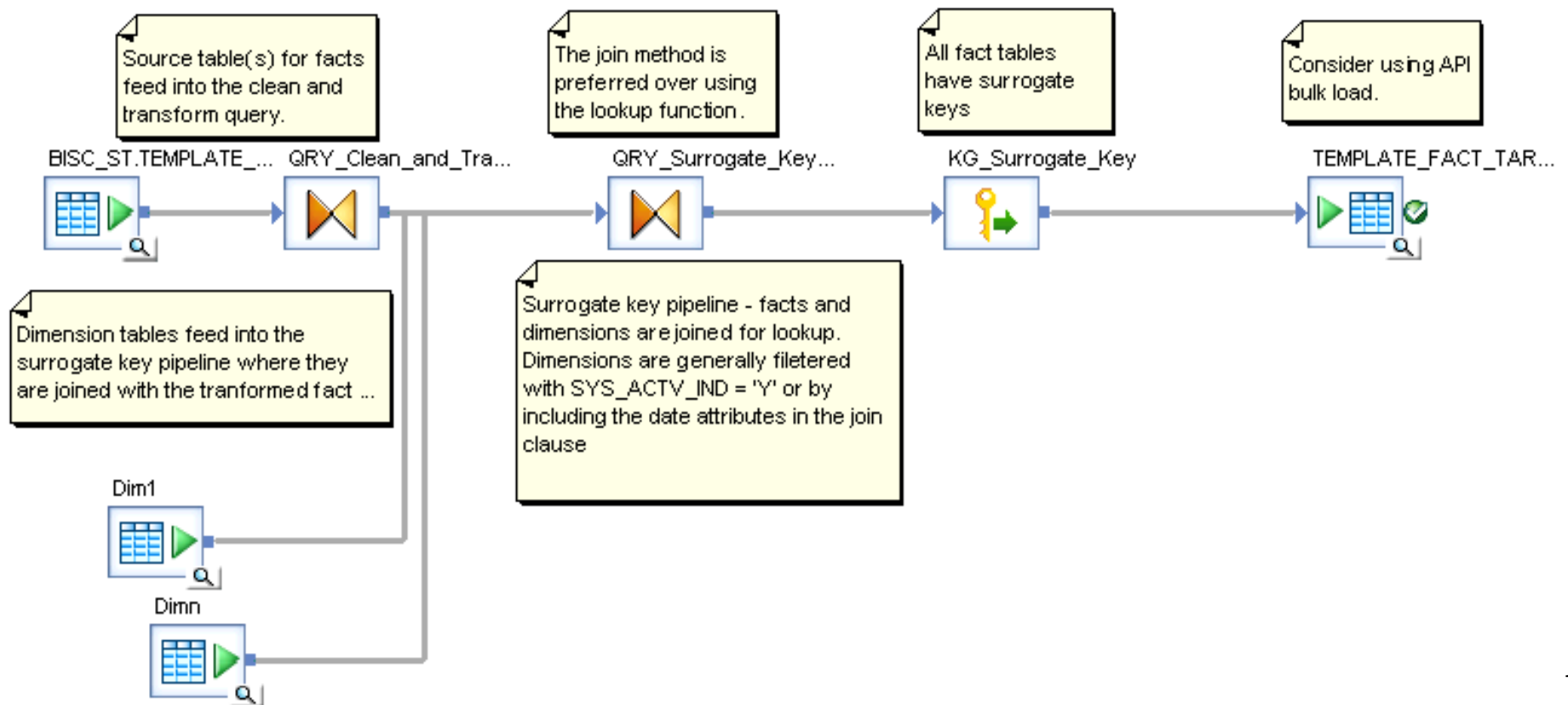
Do It Well: Fact Tables

JOB Name: DF_Template_Fact

Source Table Name: BISC_ST.TEMPLATE_FACT_SOURCE1

Target Table Name: BISC_ST.TEMPLATE_FACT_TARGET1

Description: This job is a template for transaction and periodic snapshot type fact tables.



Do It Well: Tips

- Know your business rules
- Think long and hard before using stored procedures
- Use SQL only as a last resort
- Make jobs rerunnable

Verify That It Was Done Cobbler's Children?

Old English Saying:

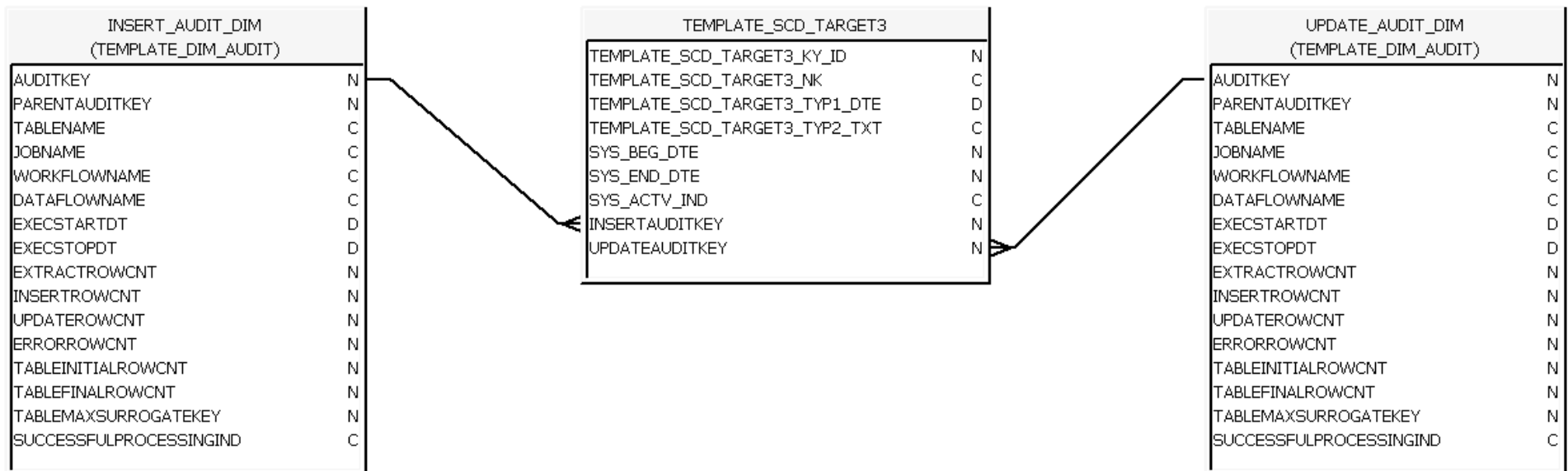
“The cobbler's children have no shoes.”

Verify That It Was Done: Auditing

- Ever ask yourself why there isn't more BI for BI systems?
 - Number of records loaded over time
 - Duration of loads over time
 - Number of new vs. changed records
- Solution: implement a Kimball Audit Dimension

Verify That It Was Done: Auditing Audit Dimension

- One Audit Dimension table
- Two meta-data fields per target



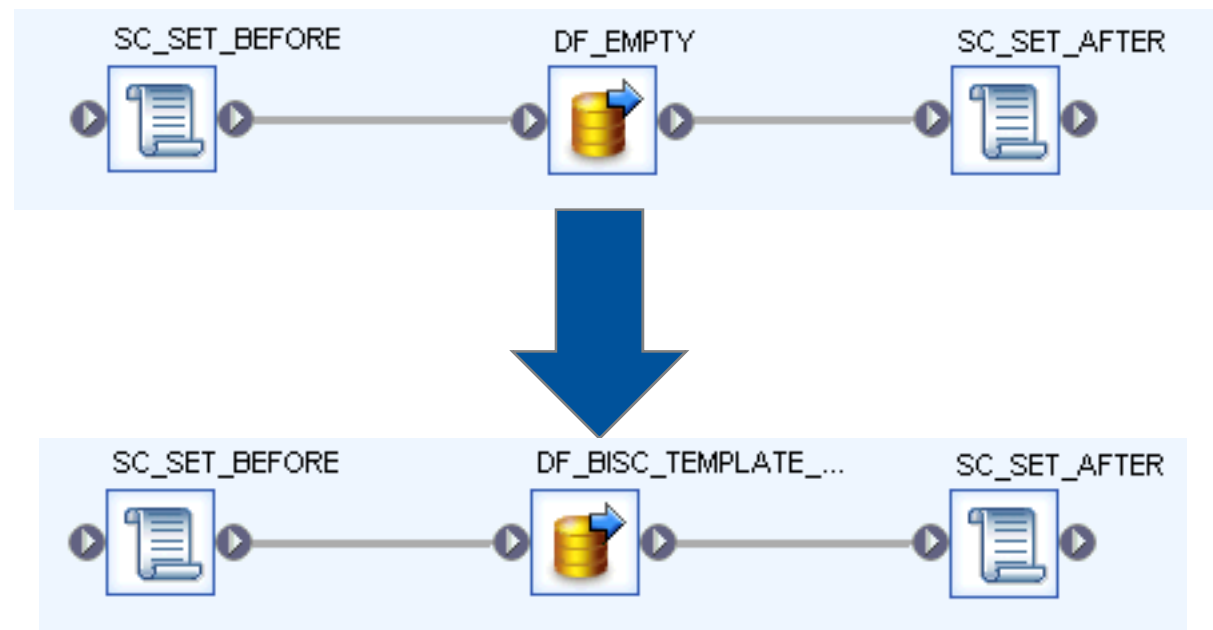
Verify That It Was Done: Auditing

15 min retrofit

5 min to basic implementation

1. Replicate template workflow (to get the variables)
2. Change source, target, and dataflow name; edit SQL
3. Copy in and connect the dataflow

10 min to add columns and set meta-data in dataflow



Prove That You Did It Well: Testing

- Testing takes 20% of development time so one tester can keep up with five developers
- Tester uses the same business rules to develop test script
- Properly handle NULLs

Prove That You Did It Well: Test Outline

```
SELECT COUNT(*) FROM TARGET
FULL OUTER JOIN (
    TRANSFORM SELECT FROM SOURCE
)
ON T.NK = S.NK
WHERE T.C1 != S.C1
...
T.Cn != S.Cn
```

Questions

TIBER SOLUTIONS

Scott Person, Senior Consultant

Email: sperson@tibersolutions.com

Phone: 703.867.3475

Backup Slides

Audit Table DDL

```
-----  
-- DDL for Table TEMPLATE_DIM_AUDIT  
-----
```

```
CREATE TABLE "BISC_ST"."TEMPLATE_DIM_AUDIT"  
(  
    "AUDITKEY" NUMBER(18,0),  
    "PARENTAUDITKEY" NUMBER(18,0),  
    "TABLENAME" NVARCHAR2(256),  
    "JOBNAME" NVARCHAR2(256),  
    "WORKFLOWNAME" NVARCHAR2(256),  
    "DATAFLOWNAME" NVARCHAR2(256),  
    "EXECSTARTDT" DATE,  
    "EXECSTOPDT" DATE,  
    "EXTRACTROWCNT" NUMBER(18,0),  
    "INSERTROWCNT" NUMBER(18,0),  
    "UPDATEROWCNT" NUMBER(18,0),  
    "ERRORROWCNT" NUMBER(18,0),  
    "TABLEINITIALROWCNT" NUMBER(18,0),  
    "TABLEFINALROWCNT" NUMBER(18,0),  
    "TABLEMAXSURROGATEKEY" NUMBER(18,0),  
    "SUCCESSFULPROCESSINGIND" NCHAR(1)  
  
    ) SEGMENT CREATION DEFERRED  
    PCTFREE 10 PCTUSED 40 INITRANS 1 MAXTRANS 255  
    NOCOMPRESS LOGGING  
    TABLESPACE "BISC_ST" ;
```

```
-----  
-- DDL for Index TEMPLATE_DIM_AUDIT_PK  
-----
```

```
CREATE UNIQUE INDEX "BISC_ST"."TEMPLATE_DIM_AUDIT_PK" ON "BISC_ST"."TEMPLATE_DIM_AUDIT" ("AUDITKEY")  
    PCTFREE 10 INITRANS 2 MAXTRANS 255 COMPUTE STATISTICS  
    TABLESPACE "BISC_ST" ;
```

Audit Start Script

```
$vTargetTableName = 'BISC_ST.TEMPLATE_SCD_TARGET3';
$vSourceTableName = 'BISC_ST.TEMPLATE_SCD_SOURCE2';
$vDataFlowName = 'DF_BISC_TEMPLATE_SCD_TYPE2_THREE_WAY_METHOD_AUDITING';

$vAuditKey = sql('ODS_DS','SELECT BISC_ST.AUDIT_SEQ.NEXTVAL FROM DUAL');
$vTableRowCount = sql('ODS_DS','SELECT COUNT(*) FROM [$vTargetTableName]');
$vExtractRowCount = sql('ODS_DS','SELECT COUNT(*) FROM [$vSourceTableName]');
$vParentAuditKey = '0';
$vJobName = job_name();
$vWorkFlowName = workflow_name();

sql('ODS_DS','INSERT INTO BISC_ST.TEMPLATE_DIM_AUDIT
(AUDITKEY,PARENTAUDITKEY,TABLENAME,JOBNAME,WORKFLOWNAME,DATAFLOWNAME,EXECSTARTDT,EXTRACTROWCNT, TABLEINITIALROWCNT,SUCCESSFULPROCESSINGIND) VALUES
([$vAuditKey],[vParentAuditKey],\'[$vTargetTableName]\',\'[$vJobName]\',\'[$vWorkFlowName]\',\'[$vDataFlowName]\',SYSDATE,[$vExtractRowCount],[vTableRowCount],\'N\')');
```

Audit End Script

```
$vTableRowCount = sql('ODS_DS','SELECT COUNT(*) FROM [$vTargetTableName]');  
$vTableInsertRowCount = sql('ODS_DS','SELECT COUNT(*) FROM [$vTargetTableName] WHERE INSERTAUDITKEY =  
[$vAuditKey]');  
$vTableUpdateRowCount = sql('ODS_DS','SELECT COUNT(*) FROM [$vTargetTableName] WHERE UPDATEAUDITKEY =  
[$vAuditKey]');  
$vTableMaxSurKey = sql('ODS_DS','SELECT NVL(MAX(TEMPLATE_SCD_TARGET3_KY_ID),0) FROM [$vTargetTableName]');  
  
sql('ODS_DS','UPDATE BISC_ST.TEMPLATE_DIM_AUDIT SET EXECSTOPDT = SYSDATE, INSERTROWCNT =  
[$vTableInsertRowCount], UPDATEROWCNT = [$vTableUpdateRowCount], ERRORROWCNT = 0, TABLEFINALROWCNT =  
[$vTableRowCount], TABLEMAXSURROGATEKEY = [$vTableMaxSurKey], SUCCESSFULPROCESSINGIND = \'Y\' WHERE  
AUDITKEY = [$vAuditKey]');
```

Sample Test Case

```
select s.AGMT_KY_ID, t.AGMT_KY_ID
,s.AGMT_PRD_END_DTE_KY_ID, t.AGMT_PRD_END_DTE_KY_ID
,s.FINC_INST_ACQ_KY_ID, t.FINC_INST_ACQ_KY_ID
,s.FINC_INST_RCVR_KY_ID, t.FINC_INST_RCVR_KY_ID
,s.AGMT_ASET_TYP_KY_ID, t.AGMT_ASET_TYP_KY_ID
,s.AGMT_CERT_TYP_KY_ID, t.AGMT_CERT_TYP_KY_ID
,s.AGMT_SBMT_STAT_KY_ID, t.AGMT_SBMT_STAT_KY_ID
,s.AGMT_CERT_SBMT_TRK_KY_ID, t.AGMT_CERT_SBMT_TRK_KY_ID
,s.AGMT_LN_PERF_CNT, t.AGMT_LN_PERF_CNT
,s.AGMT_LN_PERF_BAL_AMT, t.AGMT_LN_PERF_BAL_AMT
,s.LN_DELO_THRTY_LT_SXTY_DY_CNT, t.LN_DELO_THRTY_LT_SXTY_DY_CNT
,s.LN_DELO_THRTY_LT_SXTY_DY_AMT, t.LN_DELO_THRTY_LT_SXTY_DY_AMT
,s.LN_DELO_SXTY_LT_NNTY_DY_CNT, t.LN_DELO_SXTY_LT_NNTY_DY_CNT
,s.LN_DELO_SXTY_LT_NNTY_DY_AMT, t.LN_DELO_SXTY_LT_NNTY_DY_AMT
,s.LN_DELO_GT_NNTY_DY_CNT, t.LN_DELO_GT_NNTY_DY_CNT
,s.LN_DELO_GT_NNTY_DY_BAL_AMT, t.LN_DELO_GT_NNTY_DY_BAL_AMT
,s.LN_FORCL_CNT, t.LN_FORCL_CNT
,s.LN_FORCL_BAL_AMT, t.LN_FORCL_BAL_AMT
,s.AGMT_ASET_REPO_CNT, t.AGMT_ASET_REPO_CNT
,s.AGMT_ASET_REPO_BAL_AMT, t.AGMT_ASET_REPO_BAL_AMT
,s.AGMT_LN_PROP_TOT_CNT, t.AGMT_LN_PROP_TOT_CNT
,s.AGMT_LN_PROP_TOT_BAL_AMT, t.AGMT_LN_PROP_TOT_BAL_AMT
,s.AGMT_CERT_UPDT_CNTCT_USER_ID, t.AGMT_CERT_UPDT_CNTCT_USER_ID
,s.AGMT_CERT_CRID_DTE, t.AGMT_CERT_CRID_DTE
,s.AGMT_CERT_UPDT_DTE, t.AGMT_CERT_UPDT_DTE
from (
  select
    AGMT_KY_ID
    , AGMT_PRD_END_DTE_KY_ID
    , FINC_INST_ACQ_KY_ID
    , FINC_INST_RCVR_KY_ID
    , AGMT_ASET_TYP_KY_ID
    , AGMT_CERT_TYP_KY_ID
```