



Proactive Schema Design

HAMISH KINGSBURY
Principal Integration Specialist

Hamish.kingsbury@abley.com

Agenda

Introduction

The Problem

The Solution

The Result

INTRODUCTION

Introduction



Hamish Kingsbury

Principal Integration Specialist



- 8 years at Abley
- 7 years using FME
- Background of Geography and Computer Science
- Enjoy the flexibility of FME
- Integrating platforms and data
- Identify and solving business problems

The Problem

Data can come from anywhere



databricks



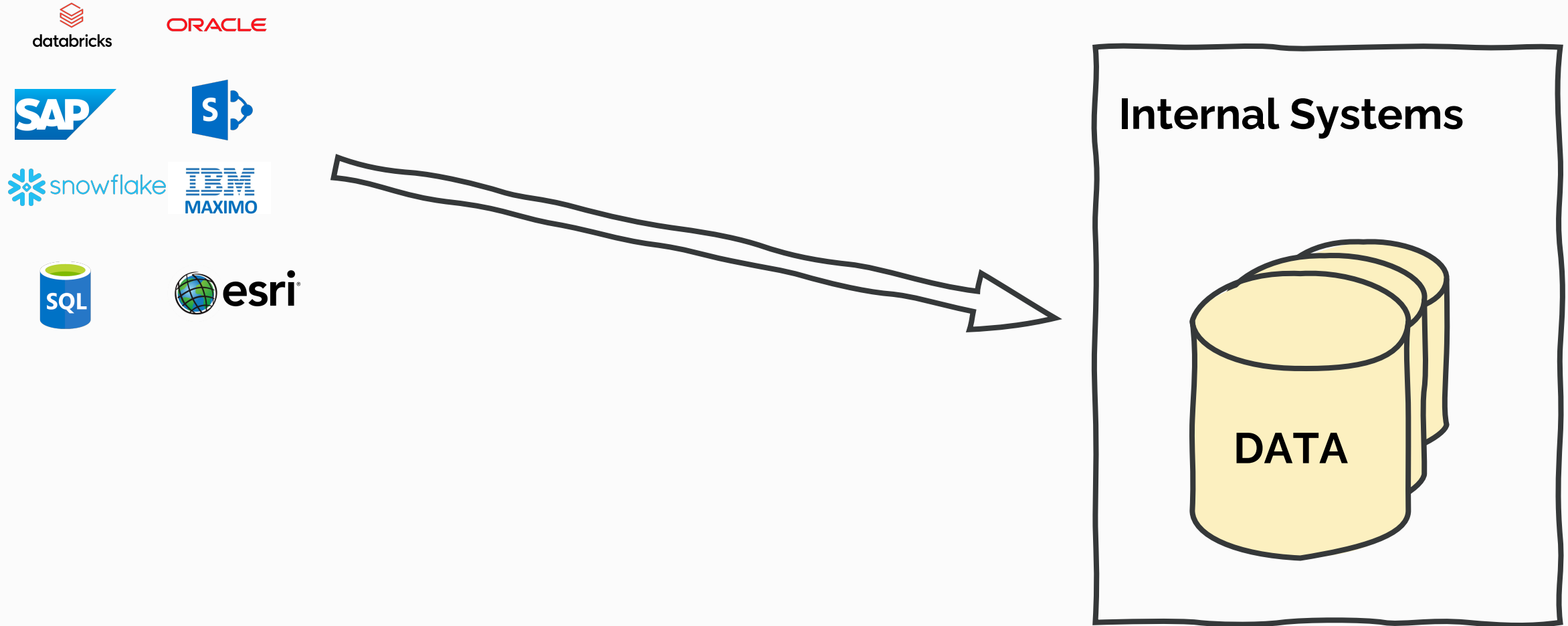
ORACLE



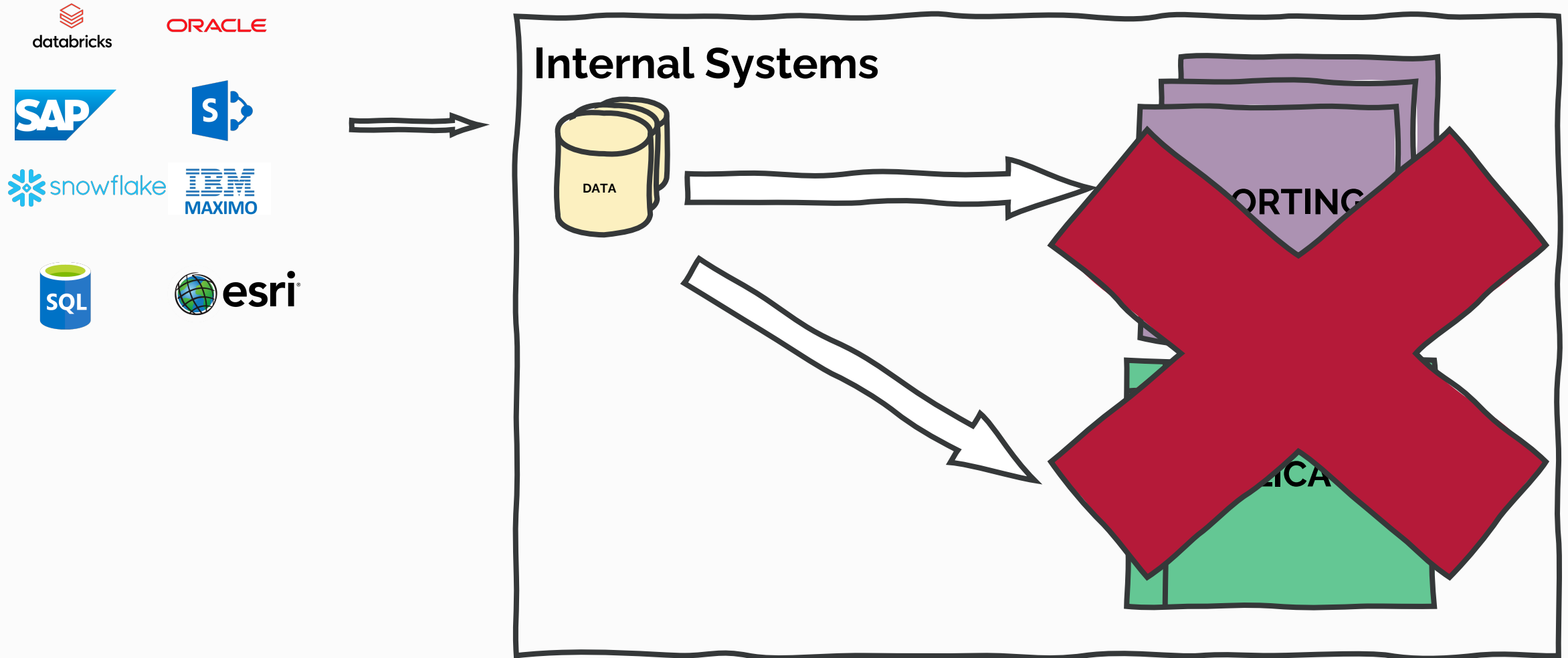
IBM
MAXIMO



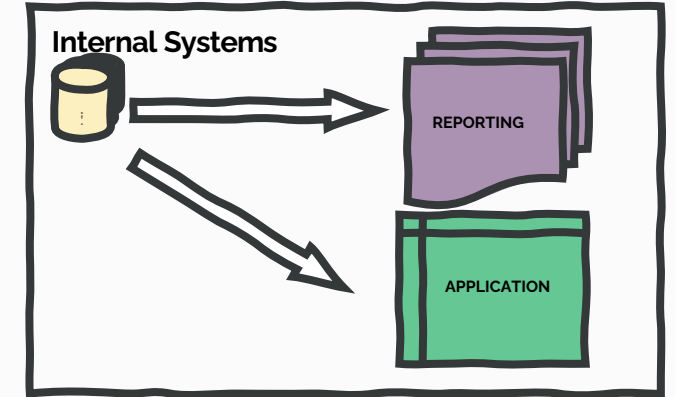
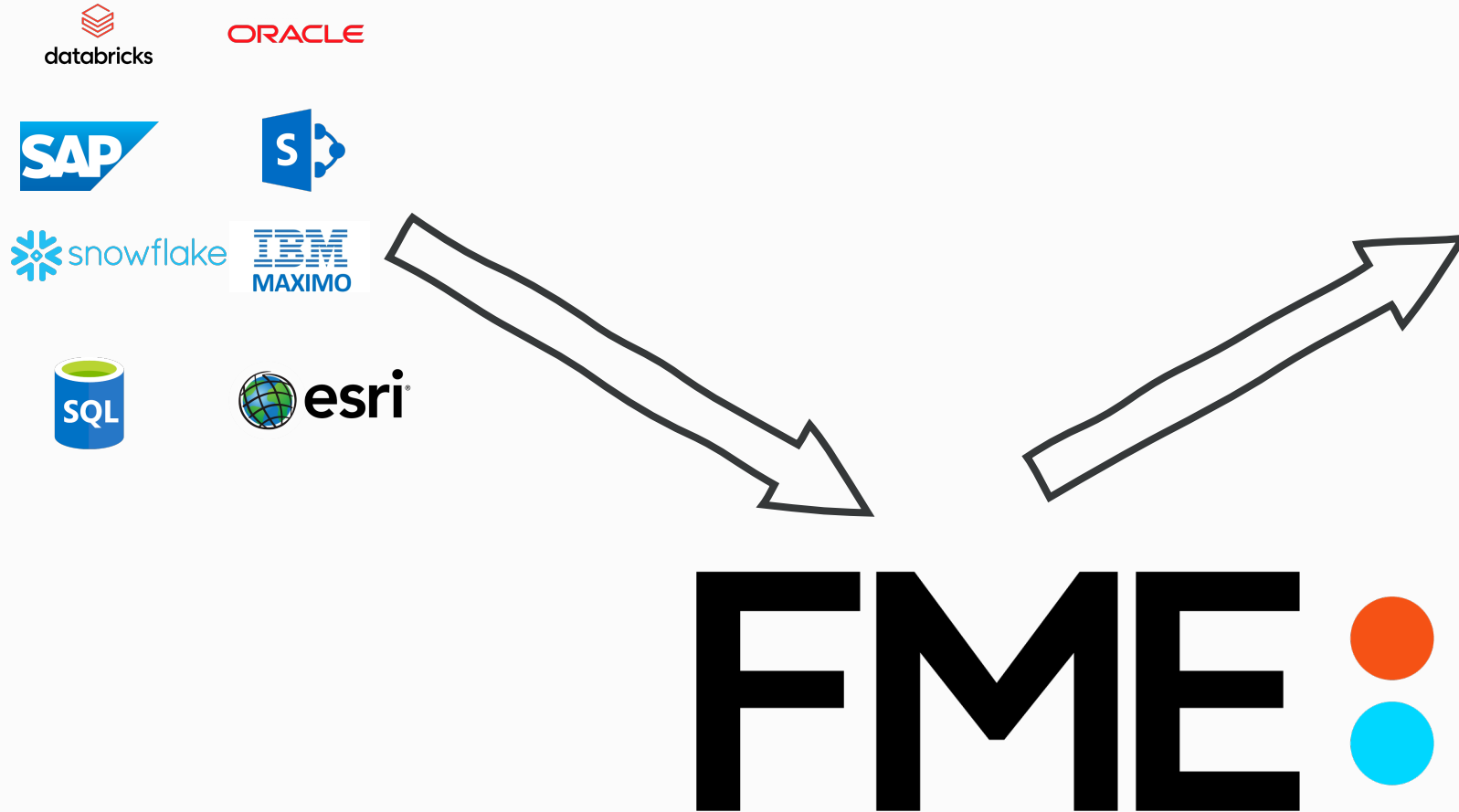
Data can come from anywhere



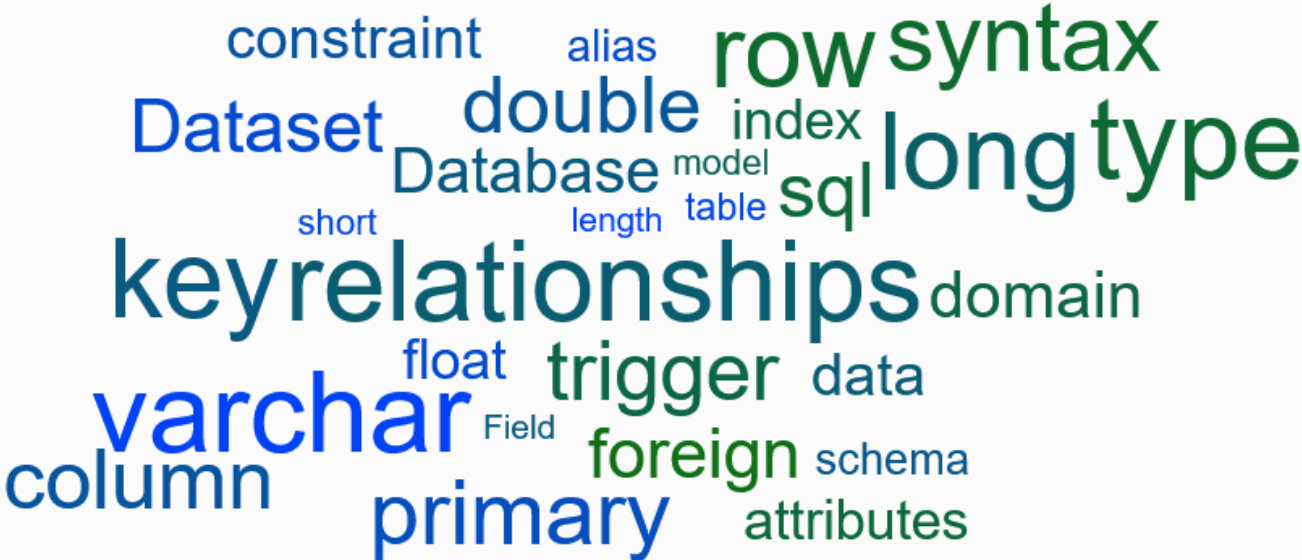
Data can come from anywhere



Data can come from anywhere



Schemas are confusing



Name	Alias	Domain	Data Type	Index
Asset ID	client_id		int	iClientId
Condition	condition	dAssetCon	int	
Asset Status	status	dAssetStatus	int	iStatus
	created_b			
Created By			varchar(32)	
...	...	...	...	...

**Schemas can have a very
specific use case**

**Not always the property of
GIS folk**

The Solution

FME:



Alias	Attribute	Domain	Data Type	Index
Asset ID	client_id		varchar(40)	idx_clientid
Condition	condition	CON	int	
Contractor Name	contractor		varchar(30)	
Created By	created_by		varchar(32)	
Criticality	criticality		int	
Current Type	current_type	TYPE	varchar(30)	
Commission Date	date_commission		datetime	

	A	B
1	Code	domainvalue
2	1	1 - Very Good
3	2	2 - Good
4	3	3 - Fair
5	4	4 - Poor
6	5	5 - Very Poor

**Everything feeds off it,
one source of truth**

Not all the same, but do the same thing

Oracle	
Data type	Description
NCHAR	Fixed length Unicode only
NVARCHAR2	Variable length Unicode only
VARCHAR2	Variable-length character string

GDB, SHP	
Data type	Description
Text	Specify length, default 255

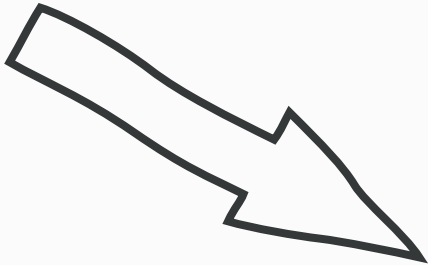
PostgreSQL	
Data type	Description
character varying(<i>n</i>), varchar(<i>n</i>)	variable-length with limit
character(<i>n</i>), char(<i>n</i>)	fixed-length, blank padded
text	variable unlimited length

SQL Server	
Data type	Description
char(<i>n</i>)	Fixed width character string
varchar(<i>n</i>)	Variable width character string
varchar(max)	Variable width character string
text	Variable width character string
nchar	Fixed width Unicode string
nvarchar	Variable width Unicode string
nvarchar(max)	Variable width Unicode string
ntext	Variable width Unicode string
binary(<i>n</i>)	Fixed width binary string
varbinary	Variable width binary string
varbinary(max)	Variable width binary string
image	Variable width binary string

	A	B	C	D	E
1	Alias	Attribute	Domain	Data Type	Index
2	Asset ID	client_id		varchar(40)	idx_clientid
3	Condition	condition	MAX_ASSCON	int	
4	Contractor Name	contractor		varchar(30)	
5	Created By	created_by		varchar(32)	
6	Criticality	criticality		int	
7	Current Type	current_type	MAX_CURRENTTYPE	varchar(30)	
8	Commission Date	date_commission		datetime	

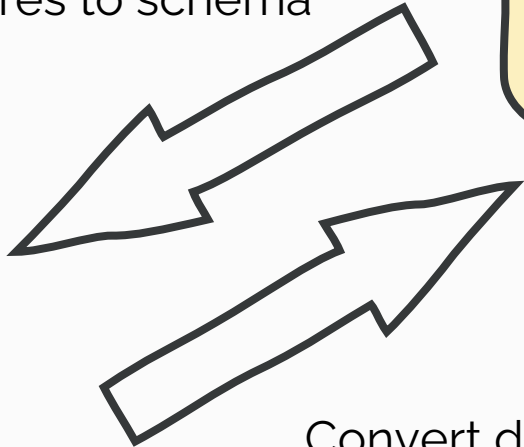
	A	B
1	Code	domainvalue
2		1 1 - Very Good
3		2 2 - Good
4		3 3 - Fair
5		4 4 - Poor
6		5 5 - Very Poor

Read schema
Read Domains



FME

Compares to schema



Convert data types
Create Domains
Create Indexes

Oracle	
Data type	Description
NCHAR	Fixed length Unicode only
NVARCHAR2	Variable length Unicode only
VARCHAR2	

GDB, SHP	
Data type	Description
Text	Specify length, default 255

PostgreSQL	
Data type	Description
character varying(n), varchar(n)	variable-length with limit
character(n), char(n)	fixed-length, blank padded
text	variable unlimited length

SQL Server	
Data type	Description
char(n)	Fixed width character string
varchar(n)	Variable width character string
varchar(max)	Variable width character string
text	Variable width character string
nchar	Fixed width Unicode string
nvarchar	Variable width Unicode string
nvarchar(max)	Variable width Unicode string
ntext	Variable width Unicode string
binary(n)	Fixed width binary string
varbinary	Variable width binary string
varbinary(max)	Variable width binary string
image	Variable width binary string

The Results

**The end user is included
in the conversation**



**Technical and non
technical users can work
at a common level**



**Documentation is your
implementation**



**GIS and FME
administrators have
control**



Automation for source changes



Streamlining for IT and Database Managers



In Summary

- Data can come from anywhere
- Information requires data
- FME provides the interface layer
- Simplifying the complexities
- Benefits the entire company



Thank You!

HAMISH KINGSBURY

Principal Integration Specialist

Hamish.kingsbury@abley.com

Questions