CONNECTED
CLOUD&DC
EDGE

ADAPT

Inside the Project:
Migrating Blackwood's ERP to the Cloud

Alan Perkins
Profound Information Pty Ltd

The Background

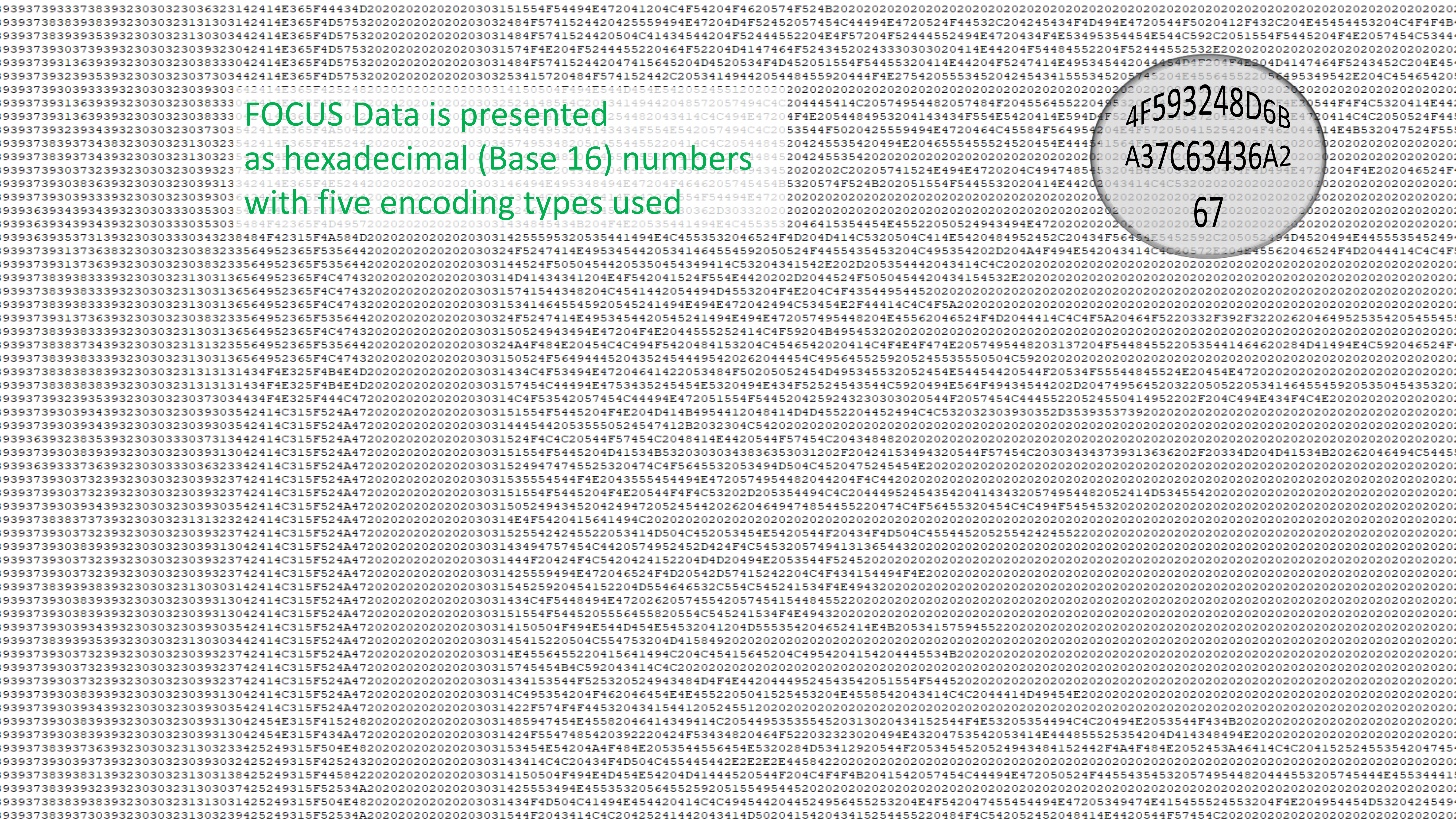
Blackwoods’ FOCUS ERP

FOCUS ERP

- 30 Years of COBOL Development
- COBOL
- Ancient, closed, idiosyncratic, proprietary
- Data is difficult and expensive to access

FOCUS Data

- Data for each object for each trading branch is stored in separate files.
- Currently 19 branches, with almost 100 historically
- Current year has 5737 files, over 200,000 over last seven years
- 750GB – 1TB raw data per year
- 750 Million records to process every day
- 5 Billion records to process over last seven years



FOCUS Data is presented as hexadecimal (Base 16) numbers with five encoding types used

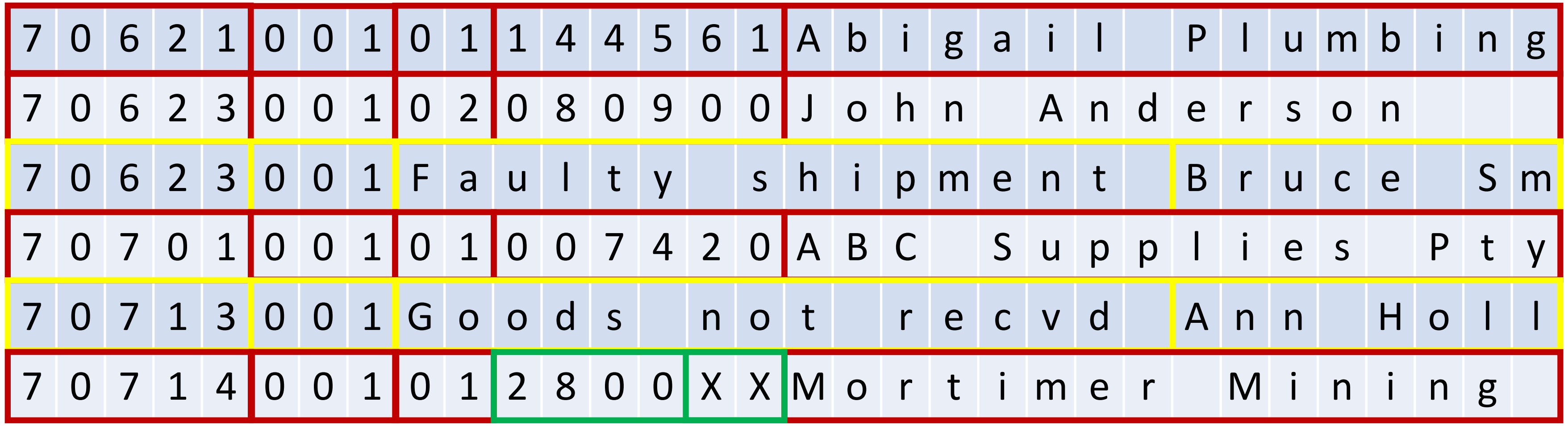
FOCUS Data

- Data for multiple record types can share the same space in the file.
  - For entire records, e.g. Invoices v Credit Notes
  - Within a record type where only one of several fields is used in certain circumstances

7	0	6	2	1	0	0	1	0	1	1	4	4	5	6	1	A	b	i	g	a	i	l		P	l	u	m	b	i	n	g
7	0	6	2	3	0	0	1	0	2	0	8	0	9	0	0	J	o	h	n		A	n	d	e	r	s	o	n			
7	0	6	2	3	0	0	1	F	a	u	l	t	y		s	h	i	p	m	e	n	t		B	r	u	c	e		S	m
7	0	7	0	1	0	0	1	0	1	0	0	7	4	2	0	A	B	C		S	u	p	p	l	i	e	s		P	t	y
7	0	7	1	3	0	0	1	G	o	o	d	s		n	o	t		r	e	c	v	d		A	n	n		H	o	l	l
7	0	7	1	4	0	0	1	0	1	2	8	8	3	0	0	M	o	r	t	i	m	e	r		M	i	n	i	n	g	

FOCUS Data

- Data for multiple record types can share the same space in the file.
  - For entire records, e.g. Invoices v Credit Notes
  - Within a record type where only one of several fields is used in certain circumstances



7	0	6	2	1	0	0	1	0	1	1	4	4	5	6	1	A	b	i	g	a	i	l		P	l	u	m	b	i	n	g
7	0	6	2	3	0	0	1	0	2	0	8	0	9	0	0	J	o	h	n		A	n	d	e	r	s	o	n			
7	0	6	2	3	0	0	1	F	a	u	l	t	y		s	h	i	p	m	e	n	t		B	r	u	c	e		S	m
7	0	7	0	1	0	0	1	0	1	0	0	7	4	2	0	A	B	C		S	u	p	p	l	i	e	s		P	t	y
7	0	7	1	3	0	0	1	G	o	o	d	s		n	o	t		r	e	c	v	d		A	n	n		H	o	l	l
7	0	7	1	4	0	0	1	0	1	2	8	0	0	X	X	M	o	r	t	i	m	e	r		M	i	n	i	n	g	

FOCUS Data – some challenges

- Lack of uniformity
- Five encodings used, often interchangeably
- Eight date formats used, even in same file
  - YYMMDD, YMMDD, DDMMYY, MMDDYY, YYYYMMDD, DD/MM/YY, MMYY, YYYYMMDDHHMMSSPP
- Lack of data integrity
  - Duplicate keys
  - Text stored in number fields
  - Lots of corrupt values
  - Data stored in multiple files with inconsistencies

CONNECTED
CLOUD&DC
EDGE

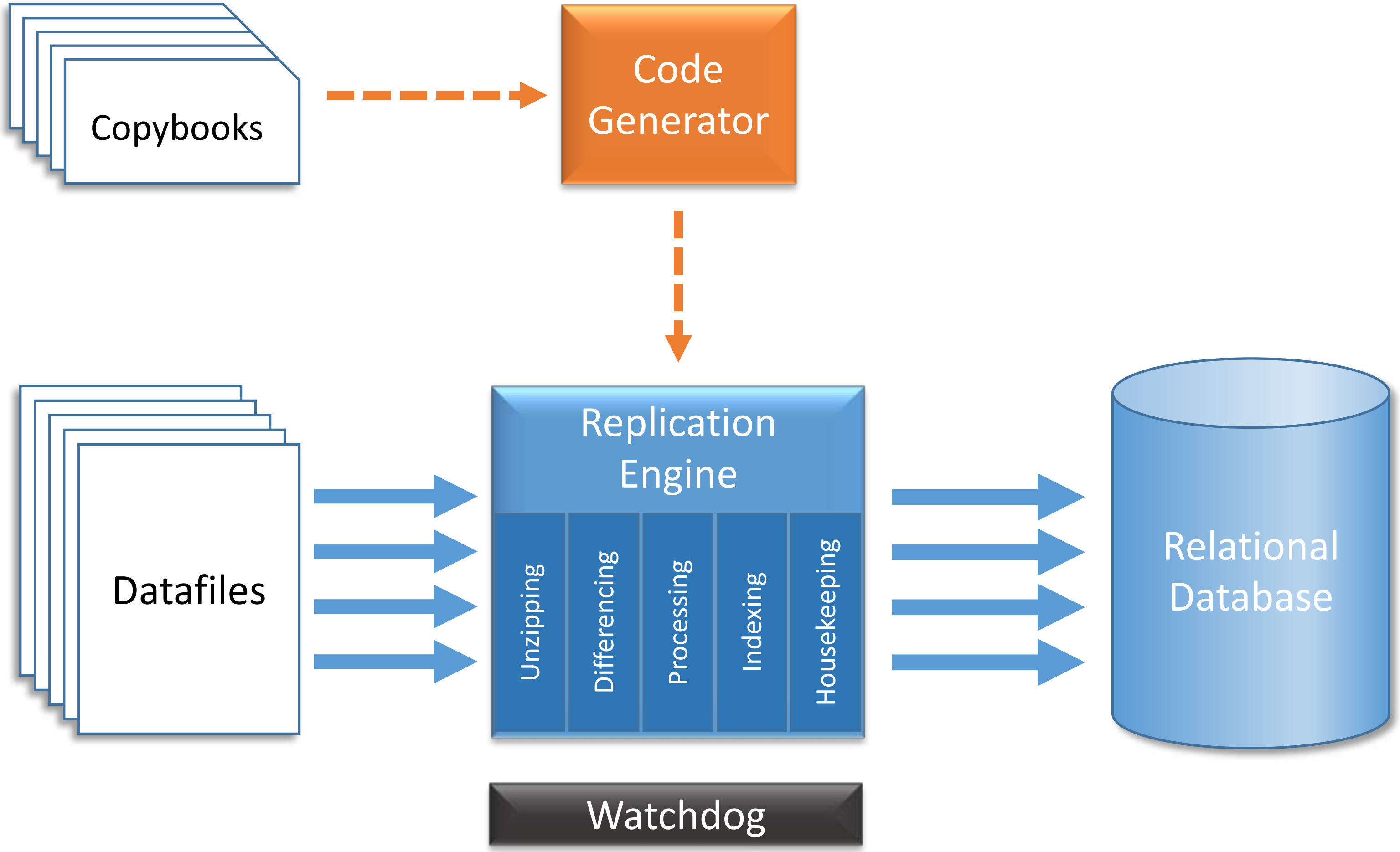
ADAPT

FOCUS Data Replica (FDR)

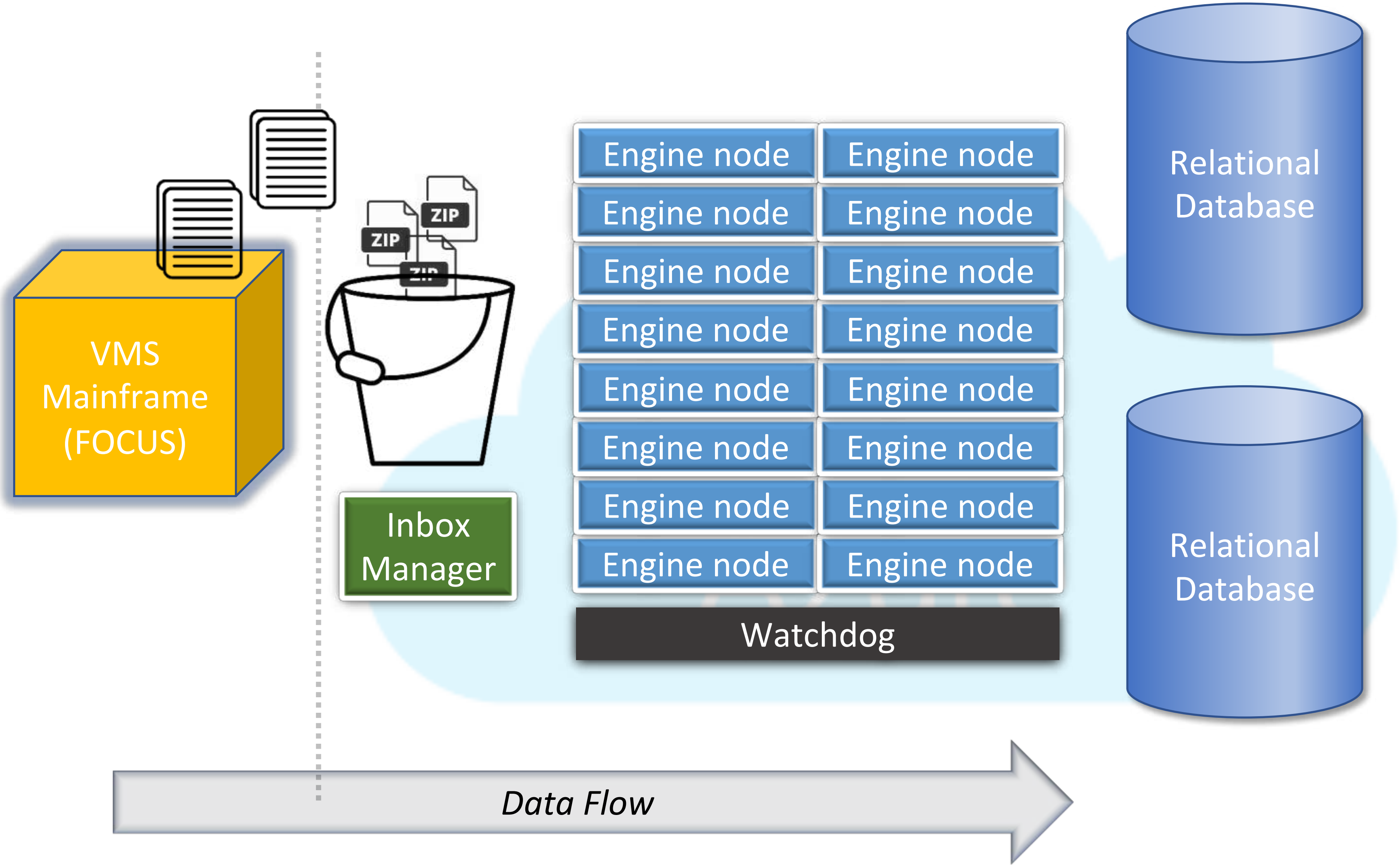
What it is and how it works

FDR Logical Overview

ADAPT



FDR Physical Overview



Cloud Infrastructure

- Azure services
- Azure PostgreSQL PaaS
- VMs
- VMSS – Virtual Machine Scale Set
- File Storage
- Terraform for infrastructure setup

- Separate environments for Dev/QA and Production
- Initial development done on local machine

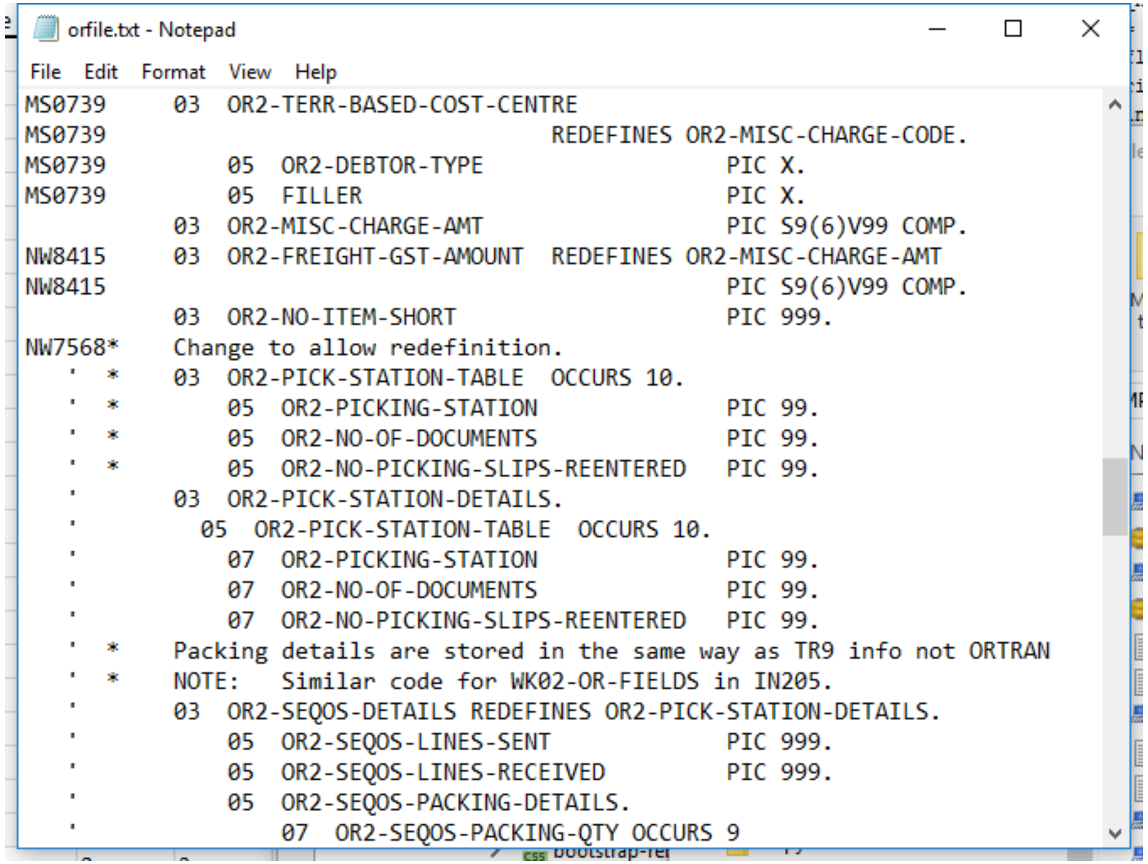
Security Considerations

ADAPT

- Cloud VMs and storage behind VPN requiring 2FA
- PostgreSQL database access required white labelled IP address
- Azure accounts limited exposure to specific resources required
- All data remained in Australia
- Logs of all access maintained
- Network subnet limited to relevant compute resources

Steps to produce results

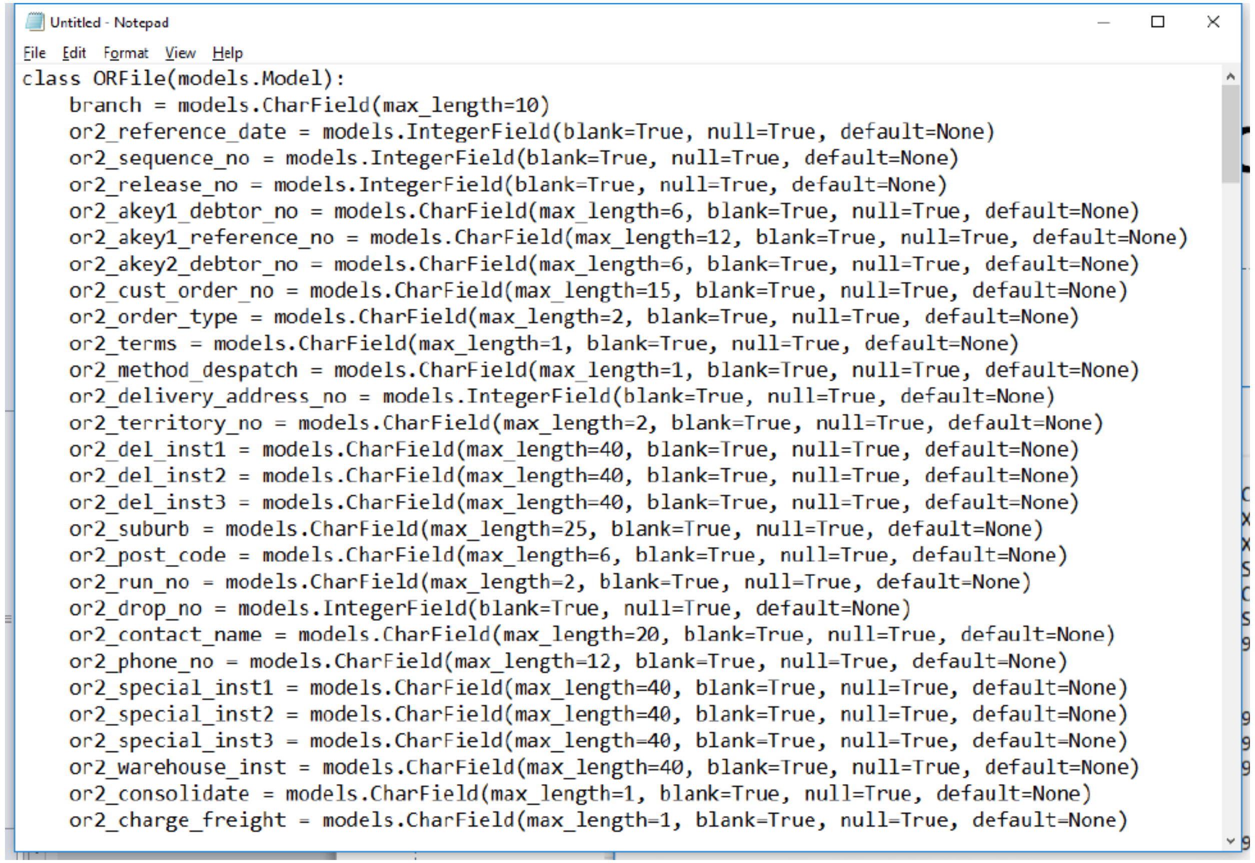
1. Read Copybooks



```
MS0739     03  OR2-TERR-BASED-COST-CENTRE
MS0739                                 REDEFINES OR2-MISC-CHARGE-CODE.
MS0739         05  OR2-DEBTOR-TYPE                  PIC X.
MS0739         05  FILLER                           PIC X.
           03  OR2-MISC-CHARGE-AMT                  PIC S9(6)V99 COMP.
NW8415     03  OR2-FREIGHT-GST-AMOUNT  REDEFINES OR2-MISC-CHARGE-AMT
NW8415                                              PIC S9(6)V99 COMP.
           03  OR2-NO-ITEM-SHORT                    PIC 999.
NW7568*    Change to allow redefinition.
   '  *    03  OR2-PICK-STATION-TABLE  OCCURS 10.
   '  *        05  OR2-PICKING-STATION              PIC 99.
   '  *        05  OR2-NO-OF-DOCUMENTS              PIC 99.
   '  *        05  OR2-NO-PICKING-SLIPS-REENTERED   PIC 99.
   '       03  OR2-PICK-STATION-DETAILS.
   '         05  OR2-PICK-STATION-TABLE  OCCURS 10.
   '           07  OR2-PICKING-STATION              PIC 99.
   '           07  OR2-NO-OF-DOCUMENTS              PIC 99.
   '           07  OR2-NO-PICKING-SLIPS-REENTERED   PIC 99.
   '  *    Packing details are stored in the same way as TR9 info not ORTRAN
   '  *    NOTE:   Similar code for WK02-OR-FIELDS in IN205.
   '       03  OR2-SEQOS-DETAILS REDEFINES OR2-PICK-STATION-DETAILS.
   '           05  OR2-SEQOS-LINES-SENT             PIC 999.
   '           05  OR2-SEQOS-LINES-RECEIVED         PIC 999.
   '           05  OR2-SEQOS-PACKING-DETAILS.
   '               07  OR2-SEQOS-PACKING-QTY OCCURS 9
```
]

Steps to produce results

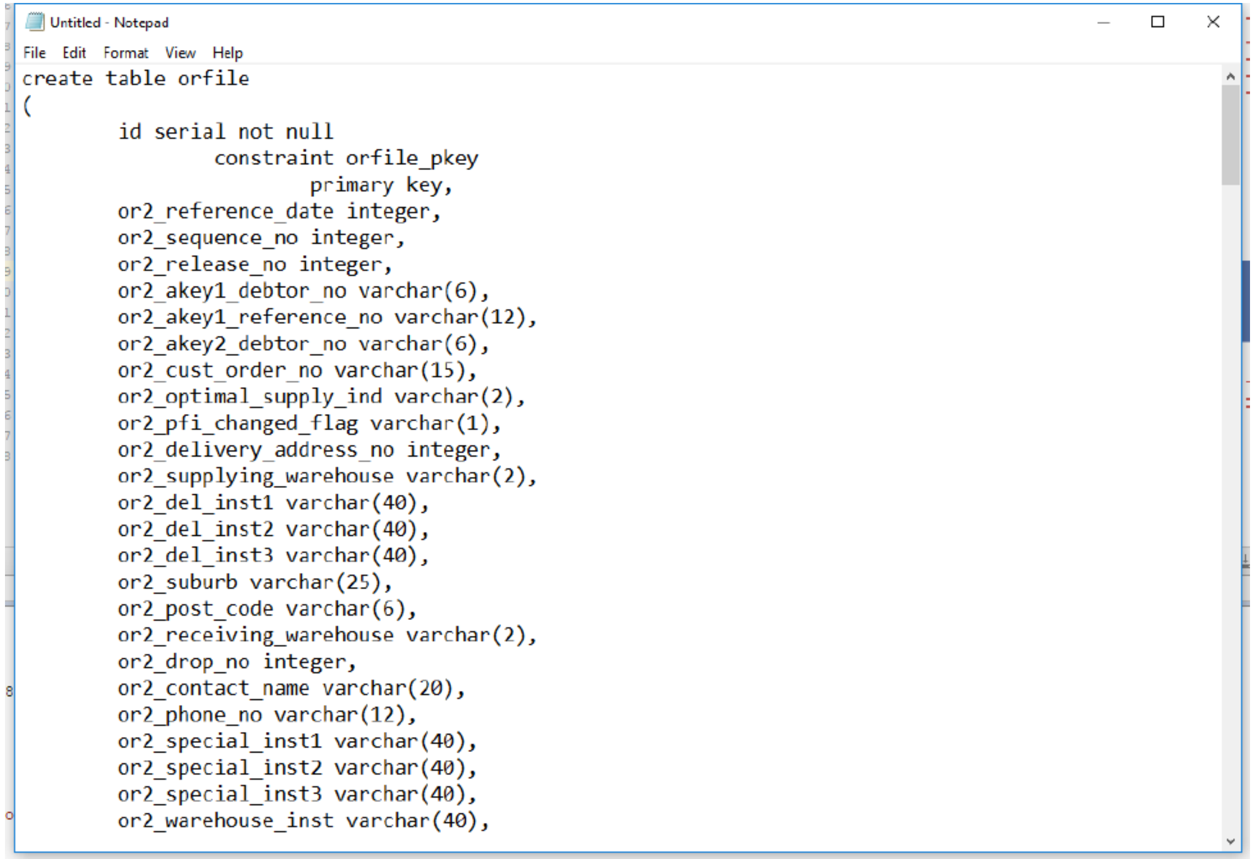
2. Define Model Class programmatically



```
class ORFile(models.Model):
    branch = models.CharField(max_length=10)
    or2_reference_date = models.IntegerField(blank=True, null=True, default=None)
    or2_sequence_no = models.IntegerField(blank=True, null=True, default=None)
    or2_release_no = models.IntegerField(blank=True, null=True, default=None)
    or2_akey1_debtor_no = models.CharField(max_length=6, blank=True, null=True, default=None)
    or2_akey1_reference_no = models.CharField(max_length=12, blank=True, null=True, default=None)
    or2_akey2_debtor_no = models.CharField(max_length=6, blank=True, null=True, default=None)
    or2_cust_order_no = models.CharField(max_length=15, blank=True, null=True, default=None)
    or2_order_type = models.CharField(max_length=2, blank=True, null=True, default=None)
    or2_terms = models.CharField(max_length=1, blank=True, null=True, default=None)
    or2_method_despatch = models.CharField(max_length=1, blank=True, null=True, default=None)
    or2_delivery_address_no = models.IntegerField(blank=True, null=True, default=None)
    or2_territory_no = models.CharField(max_length=2, blank=True, null=True, default=None)
    or2_del_inst1 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_del_inst2 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_del_inst3 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_suburb = models.CharField(max_length=25, blank=True, null=True, default=None)
    or2_post_code = models.CharField(max_length=6, blank=True, null=True, default=None)
    or2_run_no = models.CharField(max_length=2, blank=True, null=True, default=None)
    or2_drop_no = models.IntegerField(blank=True, null=True, default=None)
    or2_contact_name = models.CharField(max_length=20, blank=True, null=True, default=None)
    or2_phone_no = models.CharField(max_length=12, blank=True, null=True, default=None)
    or2_special_inst1 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_special_inst2 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_special_inst3 = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_warehouse_inst = models.CharField(max_length=40, blank=True, null=True, default=None)
    or2_consolidate = models.CharField(max_length=1, blank=True, null=True, default=None)
    or2_charge_freight = models.CharField(max_length=1, blank=True, null=True, default=None)
```

Steps to produce results

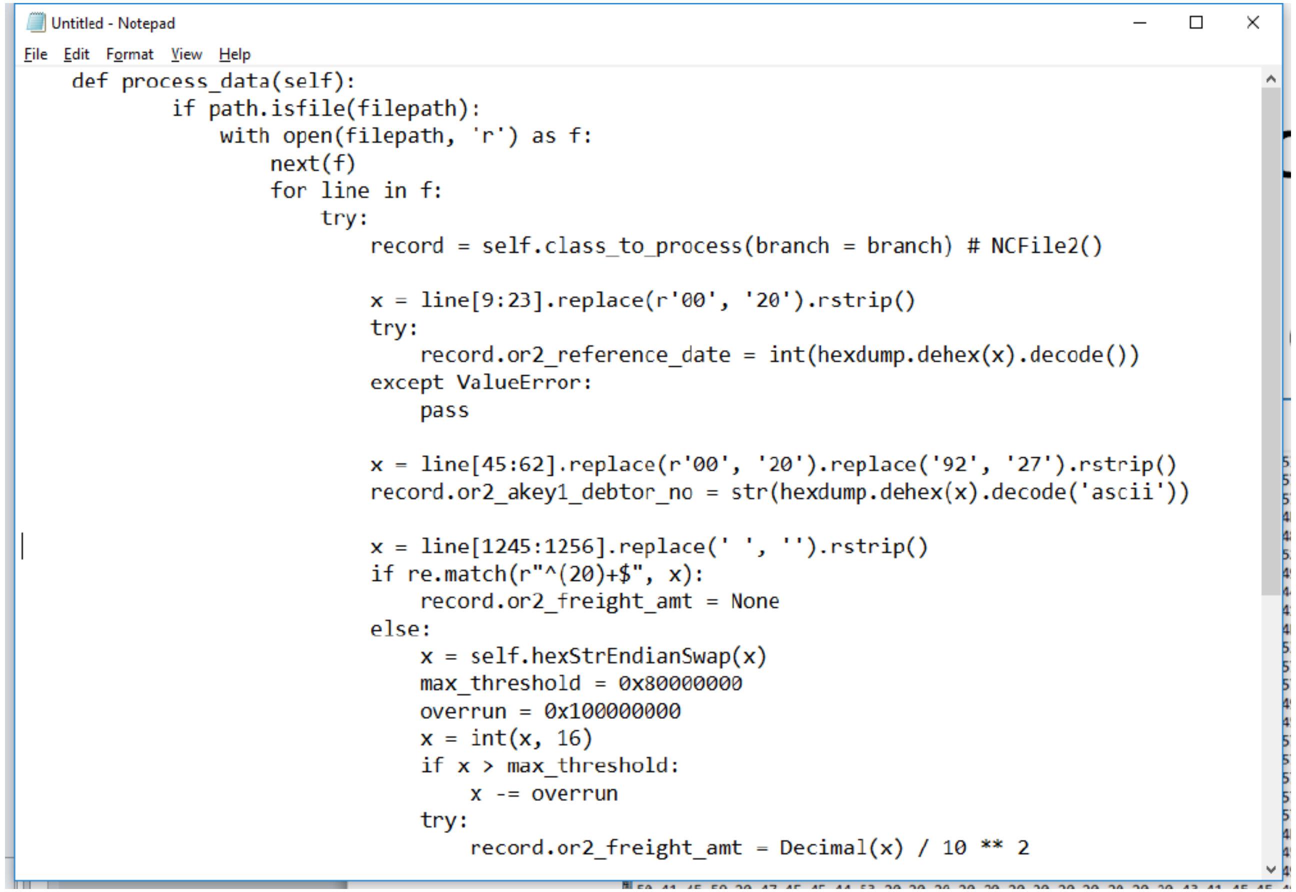
3. Automatically generate DB Tables from model



```
create table orfile
(
        id serial not null
                constraint orfile_pkey
                        primary key,
        or2_reference_date integer,
        or2_sequence_no integer,
        or2_release_no integer,
        or2_akey1_debtor_no varchar(6),
        or2_akey1_reference_no varchar(12),
        or2_akey2_debtor_no varchar(6),
        or2_cust_order_no varchar(15),
        or2_optimal_supply_ind varchar(2),
        or2_pfi_changed_flag varchar(1),
        or2_delivery_address_no integer,
        or2_supplying_warehouse varchar(2),
        or2_del_inst1 varchar(40),
        or2_del_inst2 varchar(40),
        or2_del_inst3 varchar(40),
        or2_suburb varchar(25),
        or2_post_code varchar(6),
        or2_receiving_warehouse varchar(2),
        or2_drop_no integer,
        or2_contact_name varchar(20),
        or2_phone_no varchar(12),
        or2_special_inst1 varchar(40),
        or2_special_inst2 varchar(40),
        or2_special_inst3 varchar(40),
        or2_warehouse_inst varchar(40),
```

Steps to produce results

4. Use automatically generate code to read file
   and write to database



```
  def process_data(self):
      if path.isfile(filepath):
        with open(filepath, 'r') as f:
          next(f)
          for line in f:
            try:
              record = self.class_to_process(branch = branch) # NCFile2()

              x = line[9:23].replace(r'00', '20').rstrip()
              try:
                record.or2_reference_date = int(hexdump.dehex(x).decode())
              except ValueError:
                pass

              x = line[45:62].replace(r'00', '20').replace('92', '27').rstrip()
              record.or2_akey1_debtor_no = str(hexdump.dehex(x).decode('ascii'))

              x = line[1245:1256].replace(' ', '').rstrip()
              if re.match(r"^(20)+$", x):
                record.or2_freight_amt = None
              else:
                x = self.hexStrEndianSwap(x)
                max_threshold = 0x80000000
                overrun = 0x100000000
                x = int(x, 16)
                if x > max_threshold:
                  x -= overrun
                try:
                  record.or2_freight_amt = Decimal(x) / 10 ** 2
```

Boring Technical Details

Data Types:

- ASCII encoded as hex values (00 – FF) for strings and some numbers,
  - e.g. ‘50 68 69 6C’ = ‘Phil’, ‘31 34 30 37’ = 1407
- Multibyte numbers for computed values
  - e.g., ‘50 68 69 6C’ is reversed and read as hex converted binary sets,
    - ‘6C 69 68 50’, = 1,818,847,312 decimal, with the PIC clause defining decimal point placement, or
      0110 1100 0110 1001 0110 1000 0101 0000 binary
  - If the number is signed and over half the value of the bytes allowed, it is counted back as negative, e.g. FF FF FF FD = -3

More Boring Technical Details

More Data Types:

- Some numbers are stored with trailing overpunches – a code that stores the hex ASCII equivalent of A – R or { or }, with each character representing +(0-9) or –(0-9).
- For example
  - ‘31 30 30 7B’ = ‘100{’, which equals 1000,
  - ‘31 30 30 7D’ = ‘100}’, which equals -1000

Program Structure

- Three Programs
  - Data Processor application
    - Decompresses the data files
    - Produces differential from previous run
    - Performs the processing
    - Writes to the database
  - Code Generator application
    - Parses the COBOL copybooks
    - Writes the data processor application code
  - Watchdog application
    - Manages the inbox,
    - Orchestrates the cloud computers