



# BANE NOR

bouvet



*Case Study: Using PLSQL and SQL features to migrate train data*  
Lasse Jenssen





# Bulk Processing Data with SQL and PL/SQL

Nordic OTN Tour - Norway  
12<sup>th</sup> October 2016

Martin Widlake  
Database Performance, Architecture & Training  
Ora600 Limited



[mwidlake@ORA600.org.uk](mailto:mwidlake@ORA600.org.uk)  
<http://mwidlake.wordpress.com/>  
Oh, and that twitter thing -  
[@MDwidlake](https://twitter.com/MDwidlake)

ORA600 Ltd



# SQL Processes Data Faster than PL/SQL

- SQL is a set-processing language, it is designed to take arrays of data and transform them.
- PL/SQL cannot alter data in the database. It has to use SQL to do so.
- The question is more, *is it more efficient to **control** SQL processing data via PL/SQL than not to.*
- If your processing is simple and the volumes are not massive, SQL will win over anything – (probably)
- However, SQL is a Single Command language. If you **cannot afford** to do the data processing in one command, you have a problem \*
- PL/SQL is the ideal language for controlling bulk data processing in the Oracle database.



## Issues with Straight SQL

- If the statement fails, all of it rolls back. All or Nothing.
- If the statement runs for too long you may get snapshot too old errors.
- Your data must be static or you have to capture any data since the command started.
- True RI cannot be enforced within a single SQL statement...
- The data volume may just be too large, or at least too large to process efficiently - especially if hashing or sorting spills over to disc. Sometimes it is simply too large.





FORALL

INSERT INTO ... SELECT ...

Bulk Collect

Analytical Functions

Bulk vs Row-by-row (slow-by-slow)



INSERT INTO ALL|FIRST WHEN ... SELECT ...

Finding you PROCESSING WINDOW

Things that slow down SQL:  
Indexes and referential integrity



<https://drive.google.com/drive/u/0/folders/0B10dftQmBjDLMW9nbHhpVXhhaFk>

## Agenda:

*Case study*

*See some features and techniques in real use case*

*PLSQL*

*"Advanced" SQL*







Lasse Jenssen

[www.jcon.no/oracle](http://www.jcon.no/oracle)

@lasjen lasse.jenssen@rightgroup.no

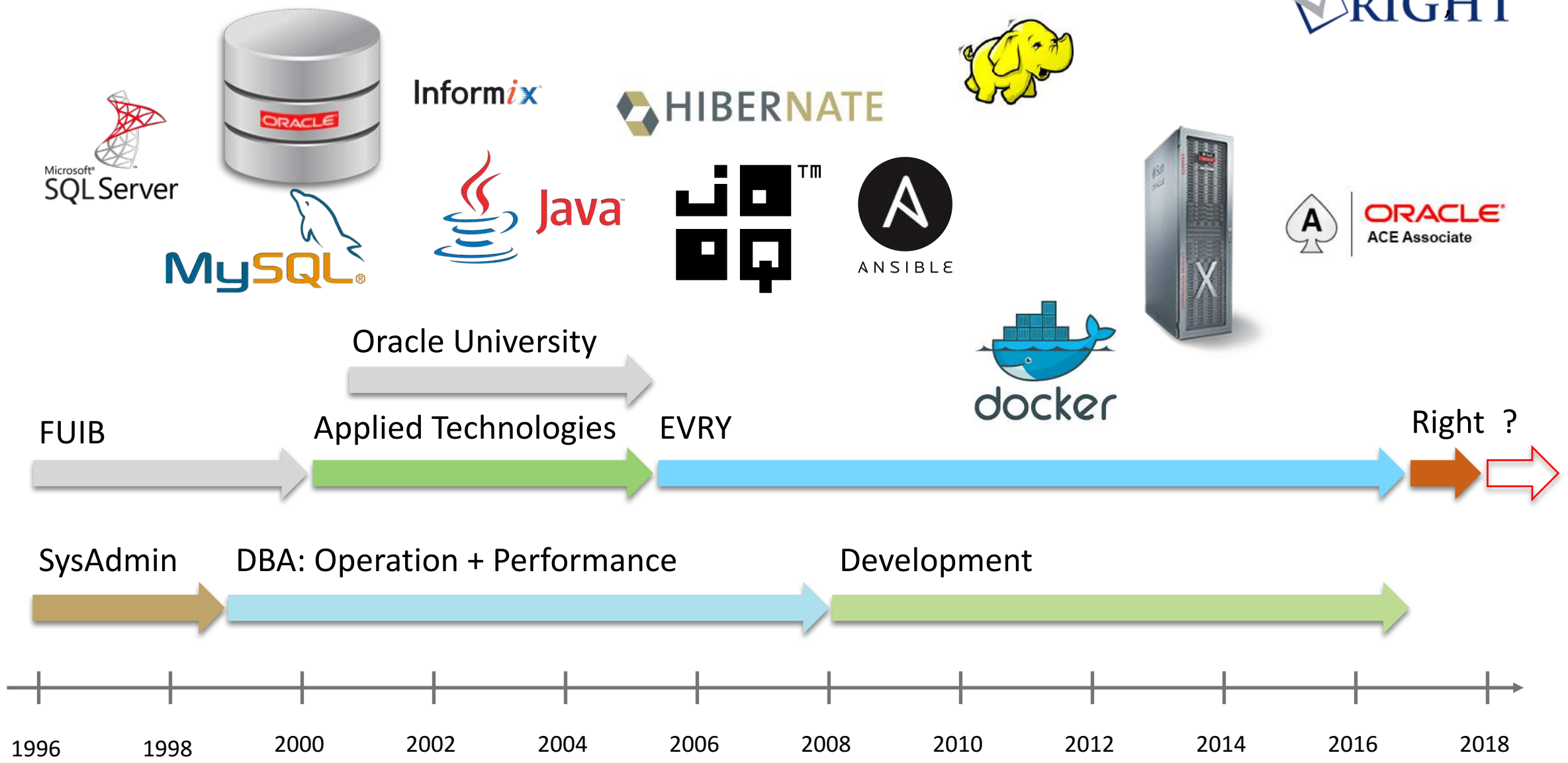




Lasse Jenssen  
Bergen











Statens vegvesen



Kollektivtransportproduksjon AS



bouvet



EVRY





Statens vegvesen



Kollektivtransportproduksjon AS



# bouvet



EVRY





# Bouvet

## ○ Norsk konsultentselskap

- 14 kontorer i Norge og Sverige
- Over 1 100 ansatte
- Notert på Oslo Børs

## ○ Bistår virksomheter med digitalisering og utforming av gode opplevelser for deres ansatte og kunder

## ○ IT – Kommunikasjon – Virksomhetsstyring





# Eress

## Europeisk samarbeid





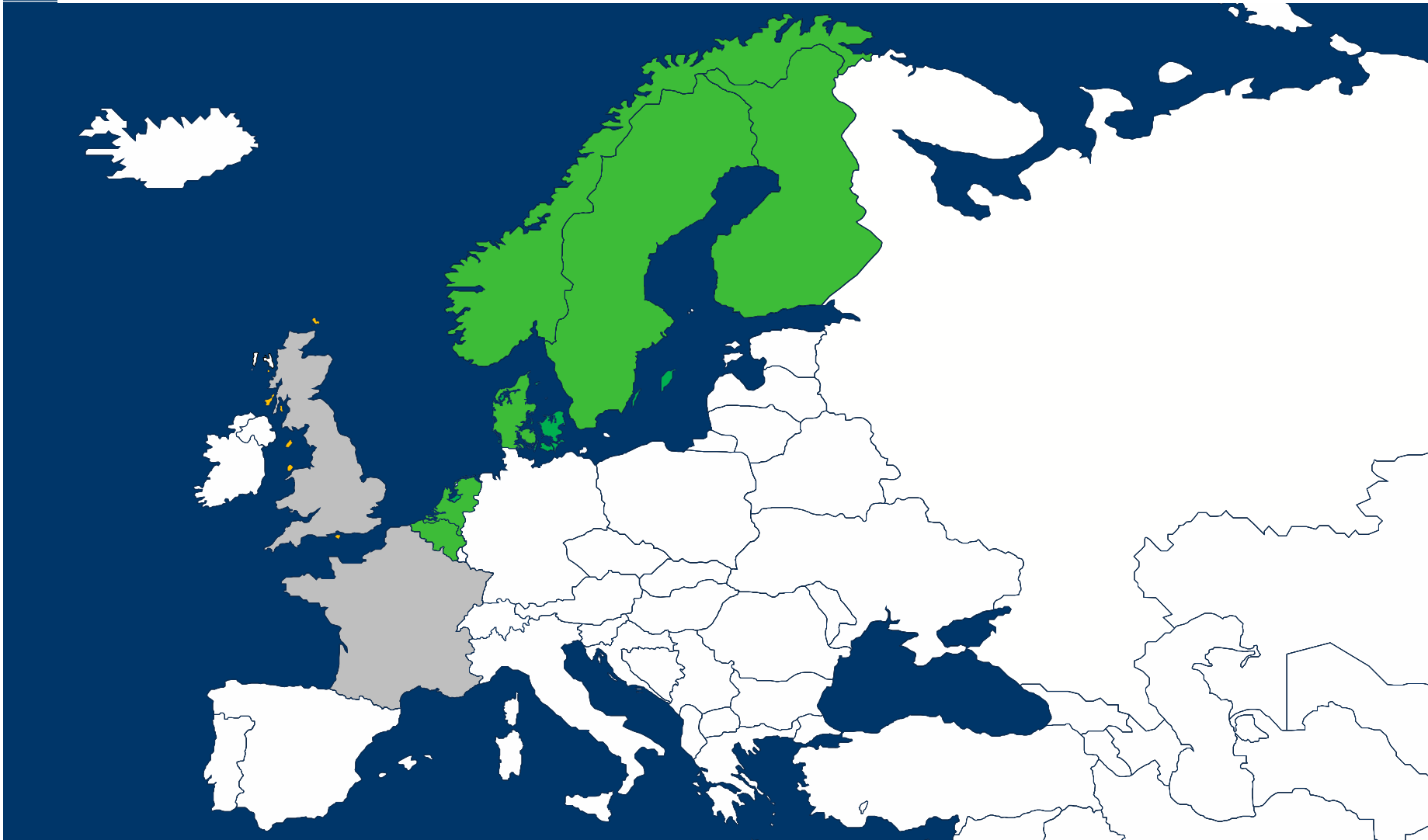
Europeisk partnerskap for avregning av jernbane energi. Eress er en non-profit organisasjon like eid av partnere.



Erex, er et europeisk system for avregning av energiforbruk på tog. Erex eies av partnerskapet Eress.

# Eress i Europa

- Partnerland
- Tester Erex systemet





Bane NOR  
(tidligere Jernbaneverket)  
er ideskaperen til Eress.

---

banedanmark



INFR/ABEL  
*Right On Track*

BANE NOR

Liikenne  
viro  
sto

Finnish Transport Agency

↔ SBB CFF FFS

 TRAFIKVERKET

VIVENS



# Calculating cost based on variable data

Settling the correct amount of kWh to the correct customer with the correct price and related data



Correct invoice  
Statistics  
Reports



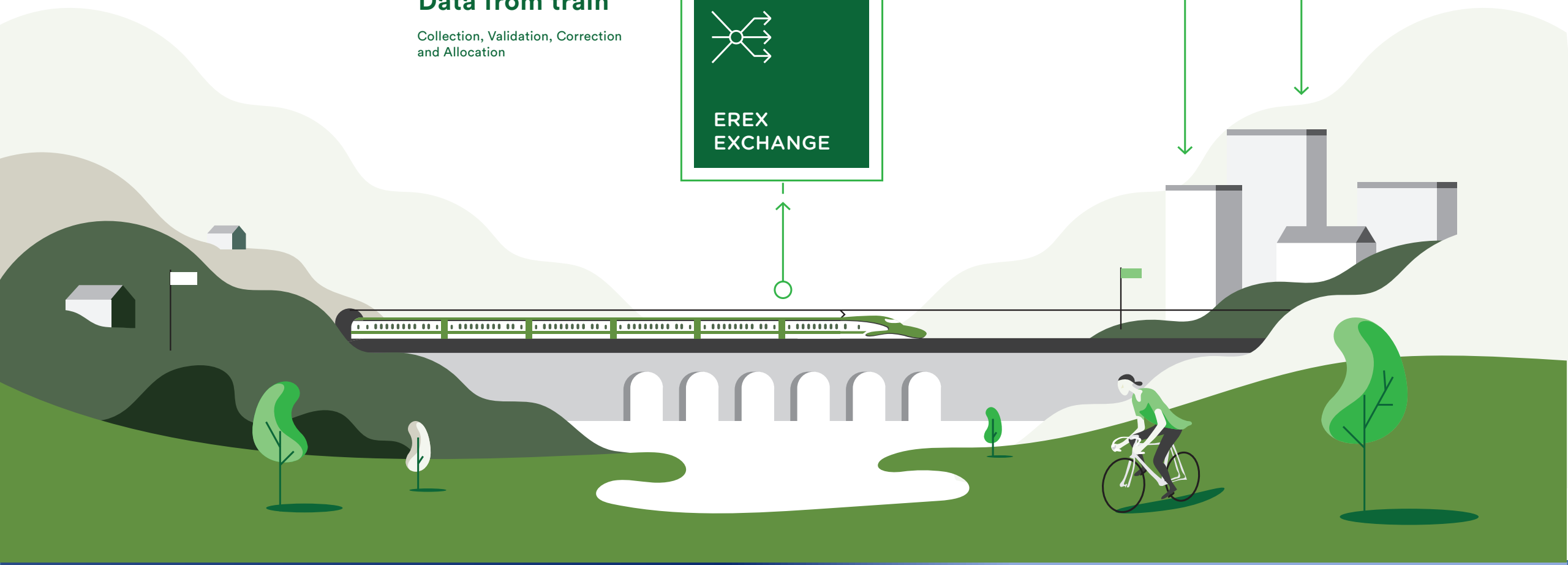
Infrastructure manager



Train operator

## Data from train

Collection, Validation, Correction and Allocation







30%  
Energiforbruk togene kan spare  
ved hjelp av Erex IT system

**33.000.000 EURO**  
har NSB spart i energiforbruk siden oppstarten i 2004

---

**NSB**



[www.eress.eu](http://www.eress.eu)

A large, two-stemmed green cactus with many sharp spines, sitting in a tan-colored pot. It is positioned in the foreground, partially obscuring the presentation screen.

Annou... 8c  
World's Fir... Driving  
**Autonom... base**



To Larry  
For trying to take away our jobs!





 JavaOne™  
JavaYourNext  
(Cloud)

To Java Developers  
For giving us the jobs back!



**Cary Millsap**

@CaryMillsap

Following



So the question now is, will you be able to  
autonomous yourself out of a problem you've  
hardwared or softwared yourself into?

9:13 PM - 4 Oct 2017

4 Retweets 36 Likes



6

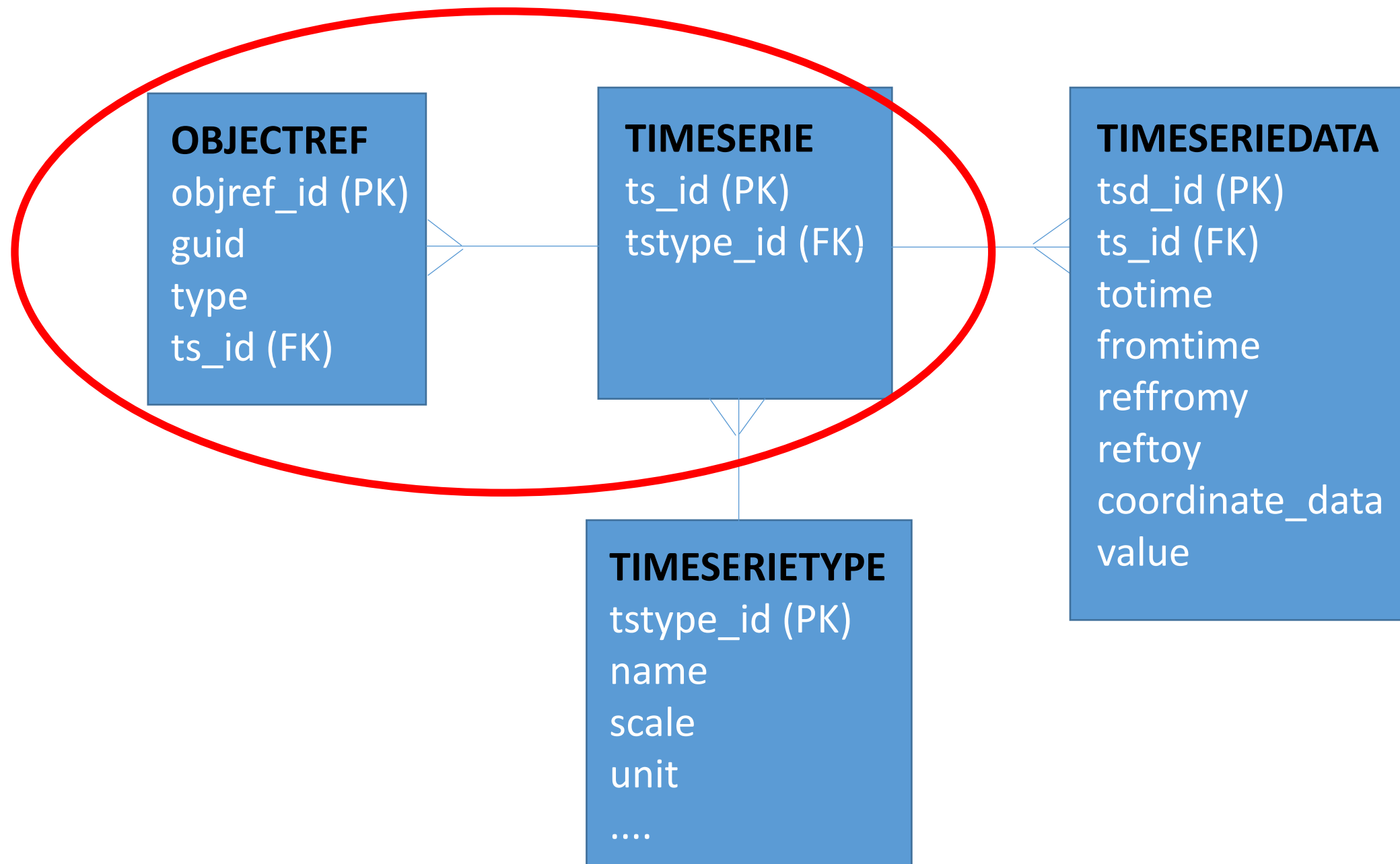


4



36





```
SELECT DISTINCT ts.ts_id, ts.tstype_id
FROM timeserie ts, objectref o1, objectref o2,
                objectref o3, objectref o4
WHERE ts.tstype_id = :1
      AND ts.ts_id = o1.ts_id
      AND o1.guid   = :2
      AND ts.ts_id = o2.ts_id
      AND o2.guid   = :3
      AND ts.ts_id = o3.ts_id
      AND o3.guid   = :4
      AND ts.ts_id = o4.ts_id
      AND o4.guid   = :5
ORDER BY ts.ts_id DESC;
```



Look at ENITITYREF.ENTITYID statistics (called GUID)

```
SELECT guid, count(*)  
FROM objref  
GROUP BY guid  
ORDER BY 2 desc;
```

```
SELECT DISTINCT ts.ts_id, ts.tstype_id
FROM timeserie ts, objectref o1, objectref o2,
                objectref o3, objectref o4
WHERE ts.tstype_id = :1
      AND ts.ts_id = o1.ts_id
      AND o1.guid   = :2
      AND ts.ts_id = o2.ts_id
      AND o2.guid   = :3
      AND ts.ts_id = o3.ts_id
      AND o3.guid   = :4
      AND ts.ts_id = o4.ts_id
      AND o4.guid   = :5
ORDER BY ts.ts_id DESC;
```

**So why is this a  
bad idea?**

```
SELECT DISTINCT ts.ts_id, ts.tstype_id
FROM timeserie ts, objectref o1, objectref o2,
                objectref o3, objectref o4
WHERE ts.tstype_id = :1
      AND ts.ts_id = o1.ts_id
      AND o1.guid   = :2 -> 5000
      AND ts.ts_id = o2.ts_id
      AND o2.guid   = :3 -> 1000
      AND ts.ts_id = o3.ts_id
      AND o3.guid   = :4 -> 5
      AND ts.ts_id = o4.ts_id
      AND o4.guid   = :5 -> 500
ORDER BY ts.ts_id DESC;
```

Example of  
JOIN order: o1 ts o2 o3 o4

So why is this a  
bad idea?



So the question now is, will you be able to  
autonomous yourself out of a problem you've  
hardwared or softwared yourself into?

9:13 PM - 4 Oct 2017

4 Retweets 36 Likes

6

4



36



Following

```
select
  sql_id, child_number,
  round(s.buffer_gets/ case s.executions
                        when 0 then 1
                        else s.executions
                        end, 2)          gets_per_exe,
  round(s.buffer_gets/ case s.rows_processed
                        when 0 then 1
                        else s.rows_processed
                        end, 2)          gets_per_row,
  executions, rows_processed, sql_text
from v$sql s
where 1=1
      -- and parsing_schema_name='<username>'
order by 3 desc;
```



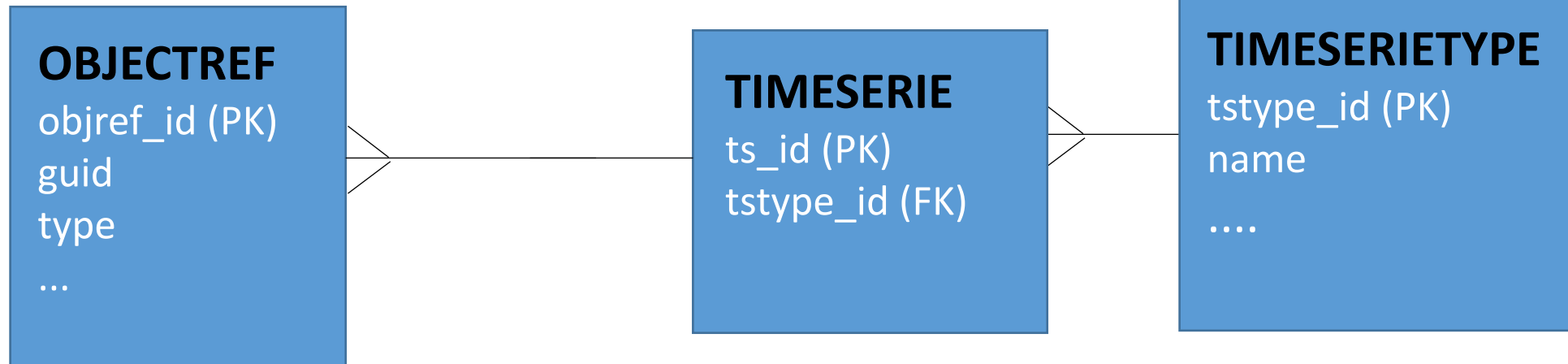
| SQL_ID        | CH... | GETS_PER_EXE | GETS_PER_ROW | EXECUTIONS | ROWS_P... | OUTL... | SQL_TEXT   |
|---------------|-------|--------------|--------------|------------|-----------|---------|------------|
| bns10qw5nhqcv | 0     | 279778,95    | 279778,95    | 5767       | 5767      | (null)  | /* select  |
| g3ykm0ypx8jpx | 5     | 23494,46     | 12187216     | 5705       | 10        | (null)  | /* schema= |
| c9wrdbn75w1mt | 0     | 13908,18     | 19436,06     | 40630      | 29074     | (null)  | /* schema= |
| g3ykm0ypx8jpx | 3     | 1484,99      | 1438,71      | 14142      | 14597     | (null)  | /* schema= |
| 6g9ffmsbwzx2n | 1     | 1175,82      | 146,57       | 44         | 360       | (null)  | /* load on |
| 73g0kvg9dvg97 | 0     | 1096,36      | 91,6         | 870        | 10424     | (null)  | /* load on |
| fmxd7uvdc497  | 0     | 1085,74      | 168,01       | 92         | 600       | (null)  | /* load on |
| g509nz08aq4qd | 0     | 823,18       | 1117,06      | 706        | 520       | (null)  | /* schema= |
| 4ttaufcara07n | 1     | 819,18       | 889,75       | 352        | 324       | (null)  | /* schema= |
| 7v5tpy0zkn0m4 | 1     | 624,01       | 30384,62     | 632        | 12        | (null)  | /* criteri |
| btyr3gryzgmfw | 0     | 360,91       | 360,91       | 2829       | 0         | (null)  | /* select  |
| 7qz15ftr29vy4 | 0     | 360,9        | 360,9        | 2883       | 0         | (null)  | /* select  |
| 44br57rs853dz | 0     | 52,88        | 6,87         | 735129     | 5658376   | (null)  | /* schema= |
| fxanr8k2a17xj | 0     | 32,85        | 32,85        | 322        | 322       | (null)  | /* insert  |
| fxanr8k2a17xj | 2     | 32,04        | 32,04        | 12843      | 12843     | (null)  | /* insert  |

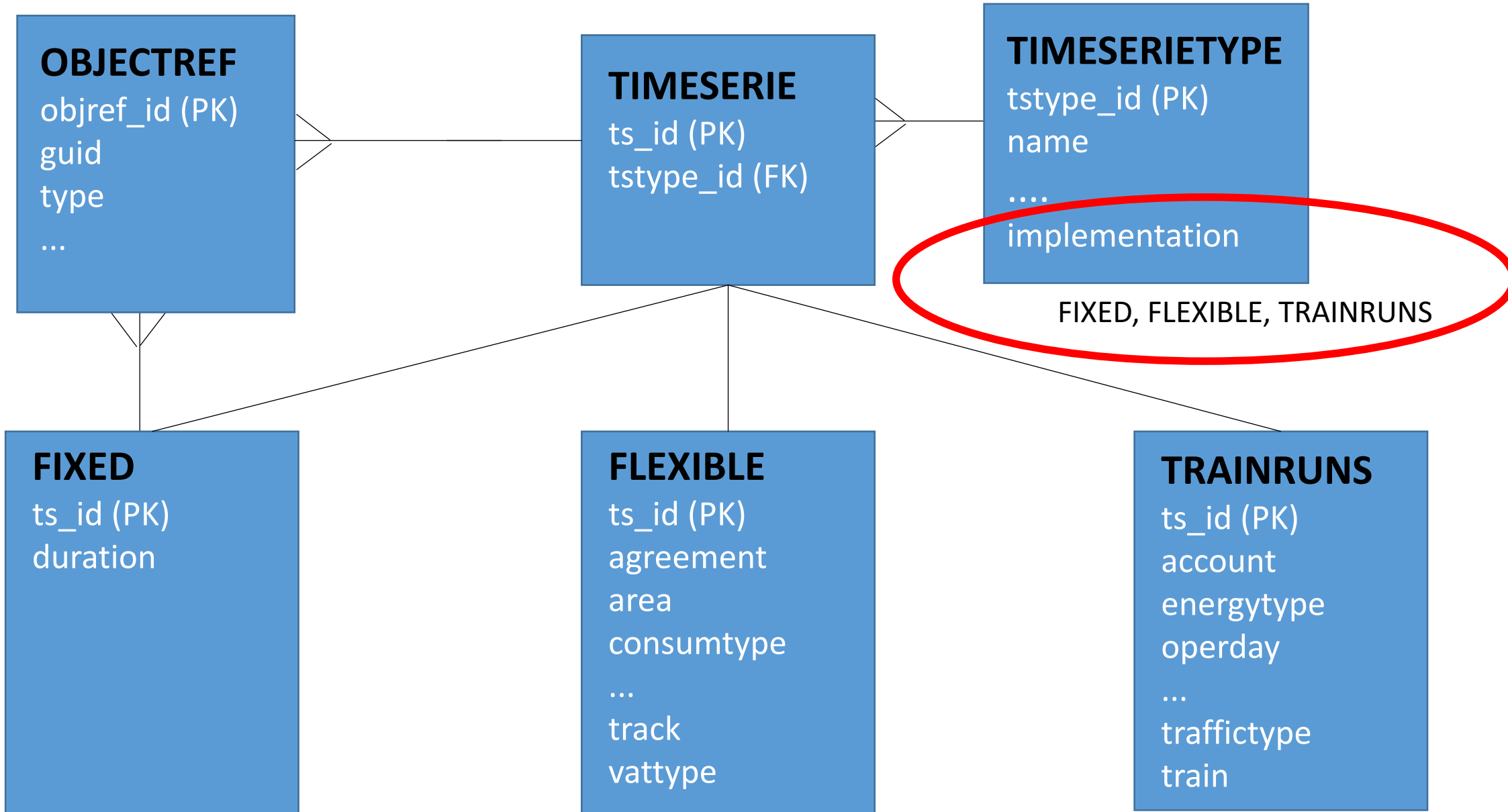
# How can we solve this?

SQL Plan Management (requires EE)

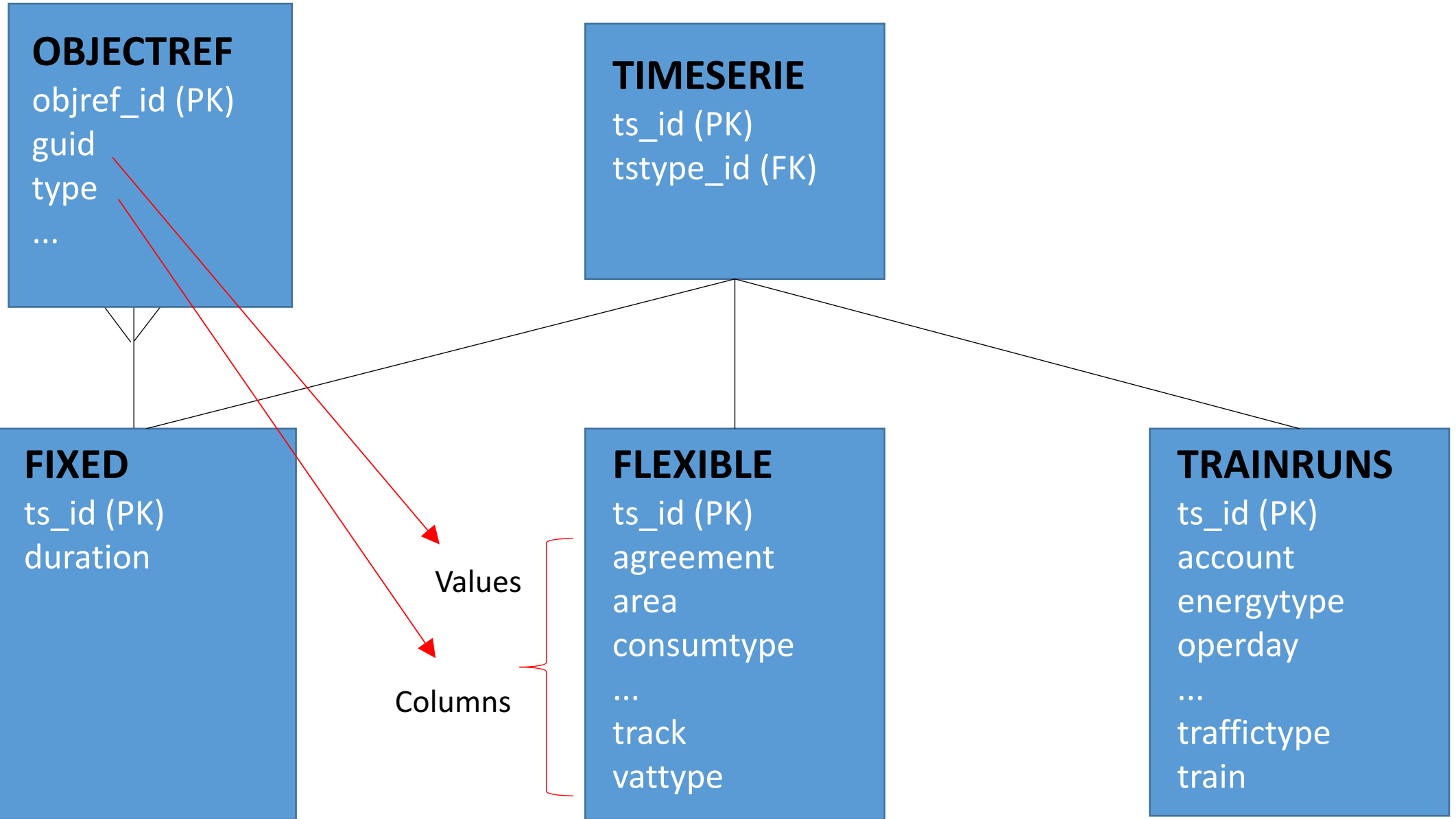
Stored Outlines (not supported in Oracle 12c)

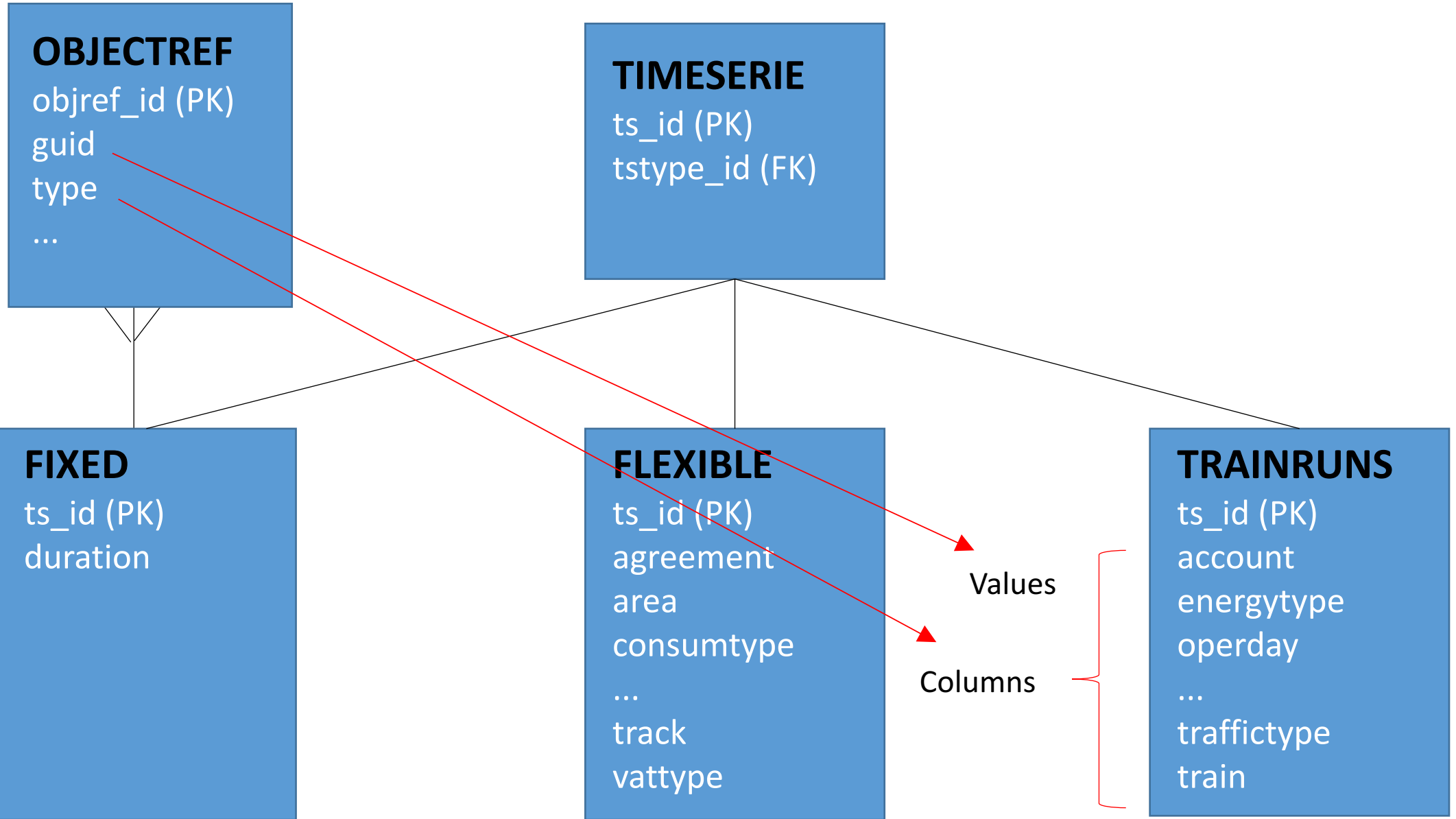
Datamodell redesign (requires code changes)

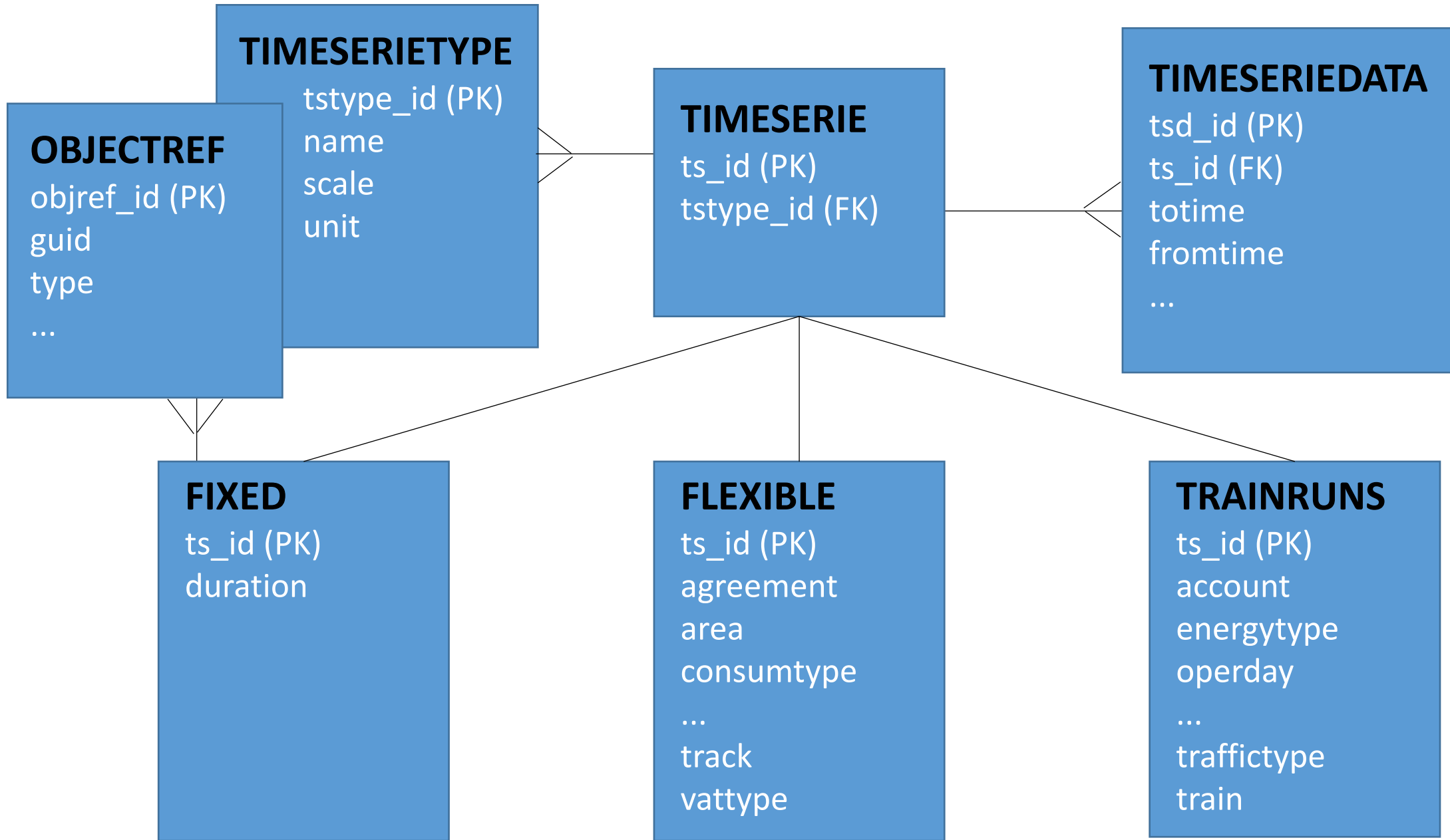












So how much better is this?

```
SELECT DISTINCT ts.ts_id, ts.tstype_id
FROM timeserie ts, trainrun tr
WHERE ts.ts_id = tr.ts_id
      AND ts.tstype_id = :1
      AND tr.train      = :2
      AND tr.unitset    = :3
      AND tr.account    = :4
      AND tr.operday    = :5
ORDER BY ts.ts_id DESC;
```

Joins with 1 row.

INDEX UNIQUE SCAN => 2-3 block reads  
TABLE ACCESS BY INDEX ROWID => 1 block read

Identifies 1 row.

INDEX RANGE SCAN => 2-3 blocks read

(If we add TS\_ID to index – don't need to read table)

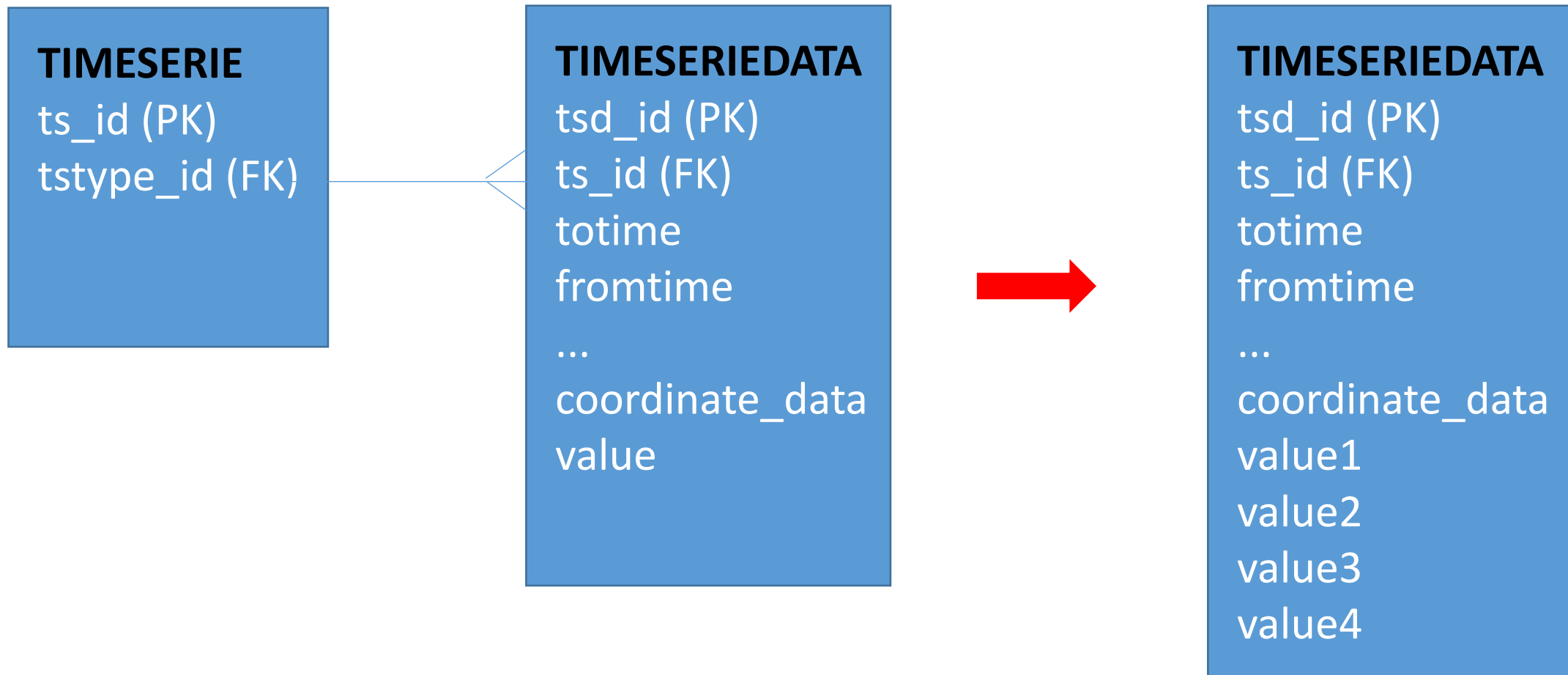
5-7 blocks - instead of 279779 blocks

Look at method of moving data to new tables  
TRAINRUN, FIXED, FLEXIBLE



## Task: EnergyType refactoring

Refactor data model storing timeseries data related to EnergyTypes.



## Task: EnergyType refactoring

Refactor data model storing timeseries data related to EnergyTypes.

### **TIMESERIEDATA**

tsd\_id (PK)  
ts\_id (FK)  
totime  
fromtime  
...  
coordinate\_data  
value



### **TIMESERIEDATA**

tsd\_id (PK)  
ts\_id (FK)  
totime  
fromtime  
...  
coordinate\_data  
value1  
value2  
value3  
value4

**Table: FLEXIBLE**

No data for the country in question.

**Table: TRAINRUNS**

| TypeName  | TSTYPE_ID | ENAME              | Into             | New Energy Type |
|-----------|-----------|--------------------|------------------|-----------------|
| Energy 21 | 21        | act gen<br>act con | value1<br>value2 | Energy New 5    |
| Energy 22 | 22        |                    | value1           | Energy New 7    |
| Energy 23 | 23        |                    | value1           | Energy New 8    |

**Table: FIXED**

| TypeName  | TSTYPE_ID | Into   | New Energy Type |
|-----------|-----------|--------|-----------------|
| -----     | -----     | -----  | -----           |
| Energy 11 | 11        | value1 | Energy New 1    |
| Energy 12 | 12        | value2 |                 |
| Energy 13 | 13        | value3 |                 |
| Energy 14 | 14        | value4 |                 |
| Energy 15 | 15        | value1 | Energy New 2    |
| Energy 16 | 16        | value2 |                 |
| Energy 17 | 17        | value1 | Energy New 3    |
| Energy 18 | 18        | value2 |                 |
| Energy 19 | 19        | value1 | Energy New 4    |
| Energy 20 | 20        | value2 |                 |



```
SELECT tstype_id, count(*)  
FROM timeseries  
GROUP BY tstype_id;
```

| TSTYPE_ID | COUNT(*) |       |
|-----------|----------|-------|
| 11        | 100      | FIXED |
| 12        | 100      |       |
| 13        | 100      |       |
| 14        | 100      |       |
| 15        | 1        |       |
| 17        | 100      |       |
| 18        | 100      |       |
| 19        | 100      |       |
| 20        | 100      |       |
| 21        | 213160   |       |
| 22        | 8093835  |       |
| 23        | 9260631  |       |

### **TIMESERIEDATA**

tsd\_id (PK)

ts\_id (FK)

totime

fromtime

...

coordinate\_data

value



### **TIMESERIEDATA**

tsd\_id (PK)

ts\_id (FK)

totime

fromtime

...

coordinate\_data

value1

value2

value3

value4

## Alternative 1:

- Table TIMESERIEDATA:
  - Rename column VALUE to VALUE1
  - Add columns VALUE2, VALUE3, VALUE4
- Migration:
  - UPDATE value2, value3 and value4
  - DELETE records for old value2-4 (about 40% of the rows)

### Pros:

- Relatively simple logic

### Cons:

- More complex rollback
- Hard to pre-process data
- Existing indexes slowing down
- DELETE relatively slow

## Alternative 2:

- Migrate into new table TIMESERIEDATA\_NEW (nologging):
- Migration:
  - INSERT /\*+ APPEND \*/ value1
  - UPDATE value2, value3 and value4

### Pros:

- Easy rollback
- Possible to preprocess data

### Cons:

- Relatively slow UPDATE (heavy redo)

## What if - Alternative 3:

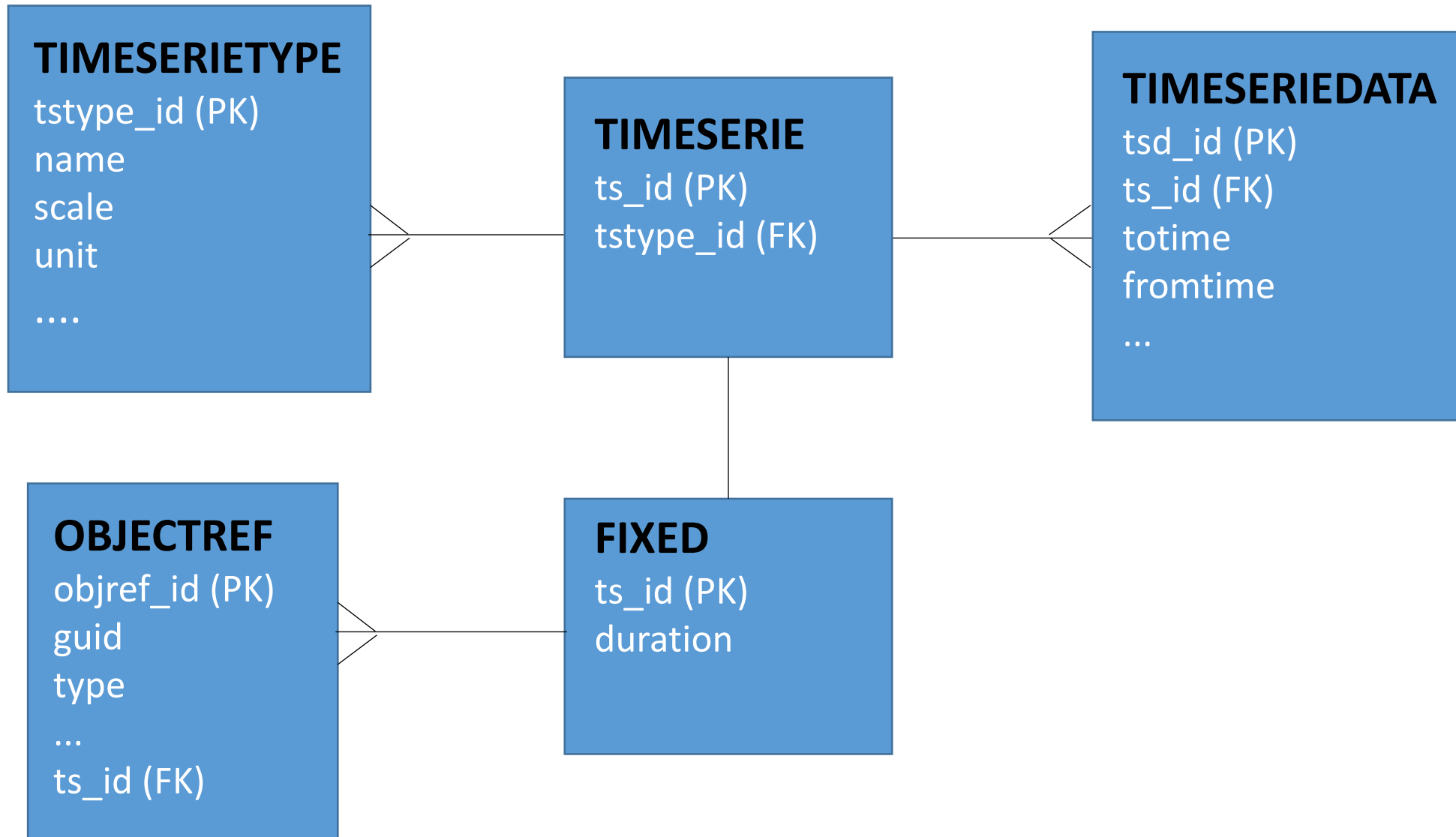
- Migrate into new table TIMESERIEDATA\_NEW (nologging):
- Migration:
  - INSERT /\*+ APPEND \*/ value1, value2, value3, value4

### Pros:

- Easy rollback
- Possible to preprocess data
- No heavy update (less REDO)

### Cons:

- Complex logic







# DIVIDE AND CONQUER



"Really? — my people always say *multiply* and conquer."

```
CREATE OR REPLACE PACKAGE MIG_ERR AS
```

```
    table_not_exist          EXCEPTION;
```

```
    table_not_exist_errnum    CONSTANT NUMBER := -24001;
```

```
    table_not_exist_errmsg    CONSTANT VARCHAR2(64) := 'Table referenced do not exist';
```

```
    PRAGMA EXCEPTION_INIT (table_not_exist, -24001);
```

```
    wrong_user               EXCEPTION;
```

```
    wrong_user_errnum        CONSTANT NUMBER := -24010;
```

```
    wrong_user_errmsg        CONSTANT VARCHAR2(64) := 'The code is ran with wrong user';
```

```
    PRAGMA EXCEPTION_INIT (wrong_user, -24010);
```

```
...
```

```
END;
```

```
/
```

```
CREATE OR REPLACE PACKAGE MIG_CONS AS
```

```
-- New Energy Type Names
```

```
C_TSFIX_NEW_ENERGY1      CONSTANT VARCHAR2(50) := 'NEW ENERGY TYPE 1'; -- 11,12,13,14
```

```
C_TSFIX_NEW_ENERGY2      CONSTANT VARCHAR2(50) := 'NEW ENERGY TYPE 2'; -- 15,16
```

```
C_TSFIX_NEW_ENERGY3      CONSTANT VARCHAR2(50) := 'NEW ENERGY TYPE 3'; -- 17,18
```

```
C_TSFIX_NEW_ENERGY4      CONSTANT VARCHAR2(50) := 'NEW ENERGY TYPE 4'; -- 19,20
```

```
...
```

```
-- Table Names
```

```
C_TBL_NAME_TSD_TMP      CONSTANT VARCHAR2(50) := 'TIMESERIEDATA_NEW';
```

```
C_TBL_NAME_TSD_OLD      CONSTANT VARCHAR2(50) := 'TIMESERIEDATA';
```

```
C_TBL_NAME_TSD_NEW      CONSTANT VARCHAR2(50) := 'TIMESERIESDATA';
```

```
-- Log statuses
```

```
LOG_STATUS_SUCCESS      CONSTANT et_mig_log.status%TYPE      := 'SUCCESS';
```

```
LOG_STATUS_ERROR        CONSTANT et_mig_log.status%TYPE      := 'ERROR';
```

```
LOG_STATUS_MAIN         CONSTANT et_mig_log.status%TYPE      := 'MAIN';
```

```
LOG_STATUS_END          CONSTANT et_mig_log.status%TYPE      := 'END';
```

```
...
```

```
END;
```

```
/
```

```
CREATE OR REPLACE PACKAGE MIG_UTIL AS
```

```
    PROCEDURE log(...);          -- Writes to ET_MIG_LOG (what, timing, number of rows, etc)
```

```
    FUNCTION get_runnr RETURN number;          -- Run nr (if needs to be ran more than once)
```

```
    FUNCTION valid_user(user_i varchar2) RETURN boolean;
```

```
    FUNCTION get_date_string(format_i IN varchar2) RETURN varchar2;
```

```
    FUNCTION diff_seconds (start_io IN OUT timestamp, end_i timestamp) RETURN number;
```

```
    FUNCTION set_tbs_for_user RETURN varchar2;
```

```
    PROCEDURE upd_counters(seconds_ins_io IN OUT number,    rowcount_agr_io IN OUT number,  
                           loopnr_io IN OUT number,        seconds_tot_io IN OUT number,  
                           time_io IN OUT timestamp,        rowcount_i IN number);
```

```
END;
```

```
/
```

```
CREATE OR REPLACE PACKAGE MIG_RUN AS
```

```
-----
```

```
-- Functions and procedures
```

```
-----
```

```
FUNCTION is_processed(v1_i number, v2_i number, v3_i number, v4_i number) RETURN boolean;
```

```
PROCEDURE open_fixed_energy_cursor (et_refcur_io IN OUT SYS_REFCURSOR);
```

```
PROCEDURE insert_energy_series(energy_type_i varchar2,  
                                v1_i number, v2_i number, v3_i number, v4_i number,  
                                rowcount_io IN OUT number);
```

```
PROCEDURE insert_error_series(v1_i IN number, v2_i IN number,  
                                v3_i IN number, v4_i IN number,  
                                first_not_null_i IN number, rowcount_io IN OUT number,  
                                energytype_i IN varchar2);
```

```
END;
```

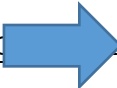
```
/
```

```
FUNCTION is_processed(v1_i number, v2_i number, v3_i number, v4_i number) RETURN boolean IS
    l_cnt number;
BEGIN
    SELECT sum(cnt) INTO l_cnt
    FROM (
        SELECT count(*) cnt
        FROM TS_MIG_LOG
        WHERE nvl(value1,0)=nvl(v1_i,0) AND nvl(value2,0)=nvl(v2_i,0)
            AND nvl(value3,0)=nvl(v3_i,0) AND nvl(value4,0)=nvl(v4_i,0)
        UNION ALL
        SELECT count(*)
        FROM TS_MIG_ERR
        WHERE nvl(value1,0)=nvl(v1_i,0) AND nvl(value2,0)=nvl(v2_i,0)
            AND nvl(value3,0)=nvl(v3_i,0) AND nvl(value4,0)=nvl(v4_i,0)
    );

    RETURN CASE WHEN l_cnt = 0 THEN false ELSE true END;
END;
```



```
PROCEDURE open_fixed_energy_cursor (et_refcur_io IN OUT SYS_REFCURSOR) IS
BEGIN
    OPEN et_refcur_io FOR
        SELECT energy_type, value1, value2, value3, value4, is_ok
        FROM ...
END;
```

mid  **.open\_fixed\_energy\_cursor**(l\_cur);

```
LOOP
  FETCH l_cur INTO l_energy_type, l_value1, l_value2, l_value3, l_value4, l_is_ok;
  EXIT WHEN l_cur%NOTFOUND;

  IF NOT mig_run.is_processed(l_value1, l_value2, l_value3, l_value4) THEN
    IF l_is_ok = 1 THEN
      mig_run.insert_energy_type(l_energy_type, l_value1, l_value2, l_value3, l_value4, ...);
      l_loop_status:= 'SUCCESS';
    ELSE
      mig_run.insert_error_series(l_value1, l_value2, l_value3, l_value4,...);
      l_loop_status:= 'ERROR';
    END IF;

    mig_util.log(l_runnr, l_log_type, 'SUB', l_loop_status, l_energy_type, l_start_dt,
                l_value1, l_value2, l_value3, l_value4, l_rowcount, l_seconds_ins);

    commit;
  END IF;
END LOOP;
CLOSE l_cur;
```

```

SELECT ts.timeserie_id,
       case when ts.tstype_id in (11,12,13,14) then 'NEW ENERGYTYPE 1'
            when ts.tstype_id in (15,16)         then 'NEW ENERGYTYPE 2'
            when ts.tstype_id in (17,18)         then 'NEW ENERGYTYPE 3'
            when ts.tstype_id in (19,20)         then 'NEW ENERGYTYPE 4' end energy_type,

       case when ts.tstype_id = 11                then 'VALUE1'
            when ts.tstype_id = 12                then 'VALUE2'
            when ts.tstype_id = 13                then 'VALUE3'
            when ts.tstype_id = 14                then 'VALUE4'
            when ts.tstype_id = 15                then 'VALUE1'
            when ts.tstype_id = 16                then 'VALUE2'
            when ts.tstype_id = 17                then 'VALUE1'
            when ts.tstype_id = 18                then 'VALUE2'
            when ts.tstype_id = 19                then 'VALUE1'
            when ts.tstype_id = 20                then 'VALUE2' end          valueX,
       tst.tstype_id typ_id, tst.IMPLEMENTATION, tst.name, o.type, o.objref_id
FROM timeseriestype tst, timeseries ts, objectref o
WHERE ts.tstype_id in (11,12,13,14,15,16,17,18,19,20)
      AND tst.tstype_id=ts.tstype_id
      AND ts.ts_id=o.ts_id;

```

| TS_ID | ENERGY_TYPE |            |   | VALUEX | TS_ID | IMPL  | NAME     | TYPE               | OBJREF_ID                        |
|-------|-------------|------------|---|--------|-------|-------|----------|--------------------|----------------------------------|
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy 2 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy 4 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy 2 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy 4 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy 1 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy 3 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |

| TS_ID | ENERGY_TYPE |            |   | VALUEX | TYP_ID | IMPL  | NAME   |   | TYPE               | OBJREF_ID                        |
|-------|-------------|------------|---|--------|--------|-------|--------|---|--------------------|----------------------------------|
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |

| TS_ID | ENERGY_TYPE |            |   | VALUEX | TS_ID | IMPL  | NAME   |   | TYPE               | OBJREF_ID                        |
|-------|-------------|------------|---|--------|-------|-------|--------|---|--------------------|----------------------------------|
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy | 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy | 2 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy | 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy | 4 | TrainMeteringPoint | 8B7403130000014D7AD8C0EA94C41965 |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy | 2 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12    | FIXED | Energy | 2 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy | 4 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968 |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14    | FIXED | Energy | 4 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11    | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13    | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB |

| TS_ID | ENERGY_TYPE |            |   | VALUEX | TYP_ID | IMPL  | NAME   |   | TYPE               | OBJREF_ID                               |
|-------|-------------|------------|---|--------|--------|-------|--------|---|--------------------|-----------------------------------------|
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | <b>8B7403130000014D7AD7967494C4016C</b> |
| 70310 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | <b>8B7403130000014D7AD8C0EA94C41965</b> |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | <b>8B7403130000014D7AD7967494C4016C</b> |
| 70311 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | <b>8B7403130000014D7AD8C0EA94C41965</b> |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | MessageType        | <b>8B7403130000014D7AD7967494C4016C</b> |
| 70312 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | TrainMeteringPoint | <b>8B7403130000014D7AD8C0EA94C41965</b> |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | MessageType        | <b>8B7403130000014D7AD7967494C4016C</b> |
| 70313 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | TrainMeteringPoint | <b>8B7403130000014D7AD8C0EA94C41965</b> |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70314 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968        |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70315 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968        |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70316 | NEW         | ENERGYTYPE | 1 | VALUE2 | 12     | FIXED | Energy | 2 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968        |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | TrainMeteringPoint | 8B7403130000014D7AD8C11D94C41968        |
| 70317 | NEW         | ENERGYTYPE | 1 | VALUE4 | 14     | FIXED | Energy | 4 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70318 | NEW         | ENERGYTYPE | 1 | VALUE1 | 11     | FIXED | Energy | 1 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB        |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | MessageType        | 8B7403130000014D7AD7967494C4016C        |
| 70319 | NEW         | ENERGYTYPE | 1 | VALUE3 | 13     | FIXED | Energy | 3 | TrainMeteringPoint | 8B7403130000014D7AD8B60494C418CB        |



```

SELECT *
FROM (SELECT ts.timeserie_id, case ... end energy_type, case ... end valueX,
          tst.timeserietype_id, tst.IMPLEMENTATION, tst.name,
          o.type, o.objref_id
      FROM timeserietype tst, timeseries ts, objectref o
      WHERE ts.tstype_id in (11,12,13,14,15,16,17,18,19,20)
          AND tst.tstype_id=ts.tstype_id
          AND ts.ts_id=o.ts_id)
PIVOT (
    max(o.objref_id)
    for o.type in (
        'TrainMeteringPoint'          as "TRAINMETERINGPOINT",
        'MessageType'                 as "MESSAGE TYPE",
        'TractionUnit'                as "TRACTIONUNIT"
    )
)

```

```

SELECT *
FROM (SELECT ts.timeserie_id, case ... end energy_type, case ... end valueX,
            tst.timeserietype_id, tst.IMPLEMENTATION, tst.name,
            o.type, o.objref_id
      FROM timeseriestype tst, timeseries ts, objectref o
     WHERE ts.tstype_id in (11,12,13,14,15,16,17,18,19,20)
           AND tst.tstype_id=ts.tstype_id
           AND ts.ts_id=o.ts_id)
PIVOT (
  max(o.objref_id)
  for o.type in (
    'TrainMeteringPoint'      as "TRAINMETERINGPOINT",
    'MessageType'             as "MESSAGE TYPE",
    'TractionUnit'            as "TRACTIONUNIT"
  )
)

```

| TS_ID | .. | VALUE_ | TYP_ID | ... | NAME   | MESSAGE | TYPE                             | TRAIN                            | METERING | POINT | TRACTI... |
|-------|----|--------|--------|-----|--------|---------|----------------------------------|----------------------------------|----------|-------|-----------|
| ----- | .. | -----  | -----  | ... | -----  | -----   | -----                            | -----                            | -----    | ----- | -----     |
| 70310 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70311 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70312 | .. | VALUE2 | 12     | ... | Energy | 2       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70313 | .. | VALUE4 | 14     | ... | Energy | 4       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70314 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70315 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70316 | .. | VALUE2 | 12     | ... | Energy | 2       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70317 | .. | VALUE4 | 14     | ... | Energy | 4       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70318 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8B60494C418CB | (null)   |       |           |
| 70319 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8B60494C418CB | (null)   |       |           |

...

| TS_ID | .. | VALUE_ | TYP_ID | ... | NAME   | MESSAGE | TYPE                             | TRAIN                            | METERING | POINT | TRACTI... |
|-------|----|--------|--------|-----|--------|---------|----------------------------------|----------------------------------|----------|-------|-----------|
| ----- | .. | -----  | -----  | ... | -----  | -----   | -----                            | -----                            | -----    | ----- | -----     |
| 70310 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70311 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70312 | .. | VALUE2 | 12     | ... | Energy | 2       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70313 | .. | VALUE4 | 14     | ... | Energy | 4       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C0EA94C41965 | (null)   |       |           |
| 70314 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70315 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70316 | .. | VALUE2 | 12     | ... | Energy | 2       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70317 | .. | VALUE4 | 14     | ... | Energy | 4       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8C11D94C41968 | (null)   |       |           |
| 70318 | .. | VALUE1 | 11     | ... | Energy | 1       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8B60494C418CB | (null)   |       |           |
| 70319 | .. | VALUE3 | 13     | ... | Energy | 3       | 8B7403130000014D7AD7967494C4016C | 8B7403130000014D7AD8B60494C418CB | (null)   |       |           |

...

```

WITH tsn AS (
  SELECT *
  FROM (SELECT ts.timeserie_id, case ... end energy_type, case ... end valueX,
              tst.timeserietype_id, tst.IMPLEMENTATION, tst.name,
              o.type, o.objref_id
        FROM timeseriestype tst, timeseries ts, objectref o
        WHERE ts.tstype_id in (11,12,13,14,15,16,17,18,19,20)
              AND tst.tstype_id=ts.tstype_id
              AND ts.ts_id=o.ts_id)
  PIVOT (
    max(o.objref_id)
    for o.type in (
      'TrainMeteringPoint'      as "TRAINMETERINGPOINT",
      'MessageType'            as "MESSAGETYPE",
      'TractionUnit'           as "TRACTIONUNIT"
    )
  )
),
...
```

```

WITH tsn AS (
    SELECT *
    FROM (...)
    PIVOT (...)
),
SELECT energy_type, VALUE1, VALUE2, VALUE3, VALUE4
FROM (SELECT messagetype, tractionunit, trainmeteringpoint, energy_type,
           valuex, timeserie_id
      FROM tsn)
PIVOT (
    max(timeserie_id)
    for valuex in (
        'VALUE1' as "VALUE1",
        'VALUE2' as "VALUE2",
        'VALUE3' as "VALUE3",
        'VALUE4' as "VALUE4"
    )
)

```

| ENERGY_TYPE          | VALUE1 | VALUE2 | VALUE3 | VALUE4 |
|----------------------|--------|--------|--------|--------|
| -----                | -----  | -----  | -----  | -----  |
| METERED_TRAIN_ENERGY | 70310  | 70312  | 70311  | 70313  |
| METERED_TRAIN_ENERGY | 70318  | 70430  | 70319  | 70431  |
| METERED_TRAIN_ENERGY | 70314  | 70316  | 70315  | 70317  |
| ...                  |        |        |        |        |

| ENERGY_TYPE          | VALUE1 | VALUE2 | VALUE3 | VALUE4 |
|----------------------|--------|--------|--------|--------|
| -----                | -----  | -----  | -----  | -----  |
| METERED_TRAIN_ENERGY | 70310  | 70312  | 70311  | 70313  |
| METERED_TRAIN_ENERGY | 70318  | 70430  | 70319  | 70431  |
| METERED_TRAIN_ENERGY | 70314  | 70316  | 70315  | 70317  |
| ...                  |        |        |        |        |



**WITH**

**tsn AS (**

SELECT \*

FROM (...)

PIVOT (...)

**),**

**tsn2 AS (**

SELECT energy\_type, VALUE1, VALUE2, VALUE3, VALUE4

FROM (SELECT messagetype, tractionunit, trainmeteringpoint, energy\_type,  
valuex, timeserie\_id

FROM tsn)

PIVOT (

max(timeserie\_id)

for valuex in (

'VALUE1' as "VALUE1",

'VALUE2' as "VALUE2",

'VALUE3' as "VALUE3",

'VALUE4' as "VALUE4"

)

)

**)**

WITH

tsn AS (

SELECT \*

FROM (...)

PIVOT (...)

),

tsn2 AS (

SELECT energy\_type, VALUE1, VALUE2, VALUE3, VALUE4

FROM (SELECT ...

FROM tsn)

PIVOT ( ... )

)

SELECT energy\_type, value1, value2, value3, value4,

CASE WHEN (nvl2(VALUE1,1,0) + nvl2(VALUE2,1,0) +

nvl2(VALUE3,1,0) + nvl2(VALUE4,1,0)) in (2,4)

AND value1 is not null AND value2 is not null

THEN 1

ELSE 0

END is\_ok

FROM tsn2

| ENERGY_TYPE          | VALUE1 | VALUE2 | VALUE3 | VALUE4 | IS_OK |
|----------------------|--------|--------|--------|--------|-------|
| -----                | -----  | -----  | -----  | -----  | ----- |
| METERED_TRAIN_ENERGY | 70310  | 70312  | 70311  | 70313  | 1     |
| METERED_TRAIN_ENERGY | 70318  | 70430  | 70319  | 70431  | 1     |
| METERED_TRAIN_ENERGY | 70314  | 70316  | 70315  | 70317  | 1     |
| ...                  |        |        |        |        |       |

| ENERGY_TYPE          | VALUE1 | VALUE2 | VALUE3 | VALUE4 | IS_OK |
|----------------------|--------|--------|--------|--------|-------|
| -----                | -----  | -----  | -----  | -----  | ----- |
| METERED_TRAIN_ENERGY | 70310  | 70312  | 70311  | 70313  | 1     |
| METERED_TRAIN_ENERGY | 70318  | 70430  | 70319  | 70431  | 1     |
| METERED_TRAIN_ENERGY | 70314  | 70316  | 70315  | 70317  | 1     |
| ...                  |        |        |        |        |       |

| Id   | Operation                   | Name              | Rows  | Bytes | TempSpc | Cost (%CPU) | Time     |
|------|-----------------------------|-------------------|-------|-------|---------|-------------|----------|
| 0    | SELECT STATEMENT            |                   |       |       |         | 5783 (100)  |          |
| 1    | HASH GROUP BY PIVOT         |                   | 3640  | 5602K |         | 5783 (4)    | 00:01:10 |
| 2    | VIEW                        |                   | 3640  | 5602K |         | 5782 (4)    | 00:01:10 |
| 3    | HASH GROUP BY PIVOT         |                   | 3640  | 593K  | 656K    | 5782 (4)    | 00:01:10 |
| * 4  | HASH JOIN                   |                   | 3640  | 593K  |         | 5645 (4)    | 00:01:08 |
| 5    | INLIST ITERATOR             |                   |       |       |         |             |          |
| 6    | TABLE ACCESS BY INDEX ROWID | TIMESERIESTYPE    | 10    | 810   |         | 2 (0)       | 00:00:01 |
| * 7  | INDEX UNIQUE SCAN           | PK_TIMESERIESTYPE | 10    |       |         | 1 (0)       | 00:00:01 |
| * 8  | HASH JOIN                   |                   | 4004  | 336K  |         | 5643 (4)    | 00:01:08 |
| 9    | TABLE ACCESS FULL           | ENTITYREF         | 4004  | 297K  |         | 17 (0)      | 00:00:01 |
| * 10 | TABLE ACCESS FULL           | TIMESERIES        | 8703K | 83M   |         | 5598 (3)    | 00:01:08 |

Predicate Information (identified by operation id):

```

4 - access("TST"."TIMESERIETYPE_ID"="TS"."TIMESERIETYPE_ID")
7 - access(("TST"."TIMESERIETYPE_ID"=40 OR "TST"."TIMESERIETYPE_ID"=41 OR
          "TST"."TIMESERIETYPE_ID"=42 OR "TST"."TIMESERIETYPE_ID"=43 OR "TST"."TIMESERIETYPE_ID"=44 OR
          "TST"."TIMESERIETYPE_ID"=45 OR "TST"."TIMESERIETYPE_ID"=46 OR "TST"."TIMESERIETYPE_ID"=47 OR
          "TST"."TIMESERIETYPE_ID"=48 OR "TST"."TIMESERIETYPE_ID"=49))
8 - access("ER"."TIMESERIE_ID"="TS"."TIMESERIE_ID")
10 - filter(("TS"."TIMESERIETYPE_ID"=40 OR "TS"."TIMESERIETYPE_ID"=41 OR "TS"."TIMESERIETYPE_ID"=42
          OR "TS"."TIMESERIETYPE_ID"=43 OR "TS"."TIMESERIETYPE_ID"=44 OR "TS"."TIMESERIETYPE_ID"=45 OR
          "TS"."TIMESERIETYPE_ID"=46 OR "TS"."TIMESERIETYPE_ID"=47 OR "TS"."TIMESERIETYPE_ID"=48 OR
          "TS"."TIMESERIETYPE_ID"=49))

```

```
mig_run.open_fixed_energy_cursor(l_cur);
```

```
LOOP
```

```
  FETCH l_cur INTO l_energy_type, l_value1, l_value2, l_value3, l_value4, l_is_ok;
```

```
  EXIT WHEN l_cur%NOTFOUND;
```

```
  IF NOT mig_run.is_processed(l_value1, l_value2, l_value3, l_value4) THEN
```

```
    IF l_is_ok = 1 THEN
```

```
      mig_run.insert_energy_type(l_energy_type, l_value1, l_value2, l_value3, l_value4, ...);
```

```
      l_loop_status:= 'SUCCESS';
```

```
    ELSE
```

```
      mig_run.insert_error_series(l_value1, l_value2, l_value3, l_value4,...);
```

```
      l_loop_status:= 'ERROR';
```

```
    END IF;
```

```
    mig_util.log(l_runnr, l_log_type, 'SUB', l_loop_status, l_energy_type, l_start_dt,  
                l_value1, l_value2, l_value3, l_value4, l_rowcount, l_seconds_ins);
```

```
    commit;
```

```
  END IF;
```

```
END LOOP;
```

```
CLOSE l_cur;
```

```

PROCEDURE insert_energy_type(energy_type_i varchar2, v1_i number, v2_i number,
                             v3_i number, v4_i number, rowcount_io IN OUT number) AS
BEGIN
    INSERT /*+ APPEND */ INTO timeseriedata_new(...)
    SELECT ... FROM timeseriedata
    WHERE ts_id = v1_i;

    rowcount_io := sql%rowcount;

    INSERT INTO ts_mig_log
    VALUES (energy_type_i, v1_i, v2_i, v3_i, v4_i, sysdate);
END;

```

But now we only insert VALUE1, and need to go back and do an UPDATE later.

```

WITH
  v1_data AS (
    SELECT tsd_id, ..., value value1, ts_id
    FROM (SELECT t1.*,
               row_number() over (partition by fromtime order by reffromy desc) rn
            FROM timeseriedata t1 WHERE ts_id=v1_i)
    WHERE rn=1),
  v2_data AS (
    SELECT *
    FROM (SELECT value value2, fromtime,
               row_number() over (partition by fromtime order by reffromy desc) rn
            FROM timeseriedata WHERE ts_id=v2_i)
    WHERE rn=1),
  v3_data AS (
    SELECT *
    FROM (SELECT value value3, fromtime,
               row_number() over (partition by fromtime order by reffromy desc) rn
            FROM timeseriedata WHERE ts_id=v3_i)
    WHERE rn=1),
  v4_data AS (
    SELECT *
    FROM (SELECT value value4, fromtime,
               row_number() over (partition by fromtime order by reffromy desc) rn
            FROM timeseriedata WHERE ts_id=v4_i)
    WHERE rn=1)

```



```

WITH
    v1_data AS (
        SELECT tsd_id, ..., value value1, ts_id
        FROM (SELECT t1.*, row_number() over (partition by fromtime order by reffromy desc) rn
              FROM timeseriedata t1 WHERE ts_id=v1_i)
        WHERE rn=1),
    v2_data AS (
        SELECT *
        FROM (SELECT value value2, fromtime, row_number() over (...) rn
              FROM timeseriedata WHERE ts_id=v2_i)
        WHERE rn=1),
    v3_data AS (
        SELECT *
        FROM (SELECT value value3, fromtime, row_number() over (...) rn
              FROM timeseriedata WHERE ts_id=v3_i)
        WHERE rn=1),
    v4_data AS (
        SELECT *
        FROM (SELECT value value4, fromtime, row_number() over (...) rn
              FROM timeseriedata WHERE ts_id=v4_i)
        WHERE rn=1)
SELECT v1.tsd_id, v1.fromtime, ... v1.reffromy, v1.value1, v2.value2, v3.value3, v4.value4
FROM v1_data v1, v2_data v2, v3_data v3, v4_data v4
WHERE v1.fromtime=v2.fromtime(+) AND v2.fromtime=v3.fromtime(+)
      AND v3.fromtime=v4.fromtime(+);

```

|                       |  |  |  | Id         | Operation                   | Name           | Rows | Bytes |
|-----------------------|--|--|--|------------|-----------------------------|----------------|------|-------|
| SUBSTR(SQL_TEXT,1,20) |  |  |  | EXECUTIONS | BUFFER_GETS                 | ROWS_PROCESSED |      |       |
| WITH /* DUMMY*/       |  |  |  | 1          | 440659                      | 205700         | 87   | 71688 |
|                       |  |  |  | 2          | VIEW                        |                | 87   | 3393  |
| * 3                   |  |  |  |            | WINDOW SORT PUSHED RANK     |                | 87   | 2349  |
| 4                     |  |  |  |            | TABLE ACCESS BY INDEX ROWID | TIMESERIEDATA  | 87   | 2349  |
| * 5                   |  |  |  |            | INDEX RANGE SCAN            | IDX_TSD_BULK   | 87   |       |
| * 6                   |  |  |  |            | HASH JOIN RIGHT OUTER       |                | 87   | 68295 |
| * 7                   |  |  |  |            | VIEW                        |                | 87   | 3393  |
| * 8                   |  |  |  |            | WINDOW SORT PUSHED RANK     |                | 87   | 2349  |
| 9                     |  |  |  |            | TABLE ACCESS BY INDEX ROWID | TIMESERIEDATA  | 87   | 2349  |
| * 10                  |  |  |  |            | INDEX RANGE SCAN            | IDX_TSD_BULK   | 87   |       |
| * 11                  |  |  |  |            | HASH JOIN RIGHT OUTER       |                | 87   | 64902 |
| * 12                  |  |  |  |            | VIEW                        |                | 87   | 3393  |
| * 13                  |  |  |  |            | WINDOW SORT PUSHED RANK     |                | 87   | 2349  |
| 14                    |  |  |  |            | TABLE ACCESS BY INDEX ROWID | TIMESERIEDATA  | 87   | 2349  |
| * 15                  |  |  |  |            | INDEX RANGE SCAN            | IDX_TSD_BULK   | 87   |       |
| * 16                  |  |  |  |            | VIEW                        |                | 87   | 61509 |
| * 17                  |  |  |  |            | WINDOW SORT PUSHED RANK     |                | 87   | 8787  |
| 18                    |  |  |  |            | TABLE ACCESS BY INDEX ROWID | TIMESERIEDATA  | 87   | 8787  |
| * 19                  |  |  |  |            | INDEX RANGE SCAN            | IDX_TSD_BULK   | 87   |       |

IDX\_TSD\_BULK: TIMESERIEDATA(ts\_id, fromtime, totime, reffromy)

```

INSERT /*+ APPEND */
  INTO timeseriedata_new(TSD_ID,...,VALUE1, VALUE2, VALUE3, VALUE3)
WITH
  v1_data AS (
    SELECT tsd_id, ..., value value1,ts_id
    FROM (SELECT t1.*, row_number() over (partition by fromtime order by reffromy desc) rn
          FROM timeseriedata t1 WHERE ts_id=v1_i)
    WHERE rn=1),
  v2_data AS (
    SELECT *
    FROM (SELECT value value2, fromtime, row_number() over (...) rn
          FROM timeseriedata WHERE ts_id=v2_i)
    WHERE rn=1),
  ...
  v4_data AS (
    SELECT *
    FROM (SELECT value value4, fromtime, row_number() over (...) rn
          FROM timeseriedata WHERE ts_id=v4_i)
    WHERE rn=1)
SELECT v1.tsd_id,v1.fromtime,... v1.reffromy, v1.value1, v2.value2, v3.value3, v4.value4
FROM v1_data v1, v2_data v2, v3_data v3, v4_data v4
WHERE v1.fromtime=v2.fromtime(+) AND v2.fromtime=v3.fromtime(+)
AND v3.fromtime=v4.fromtime(+);

```

```

PROCEDURE insert_energy_type(energy_type_i varchar2, v1_i number, v2_i number,
                               v3_i number, v4_i number, rowcount_io IN OUT number) AS
BEGIN
    INSERT /*+ APPEND */ INTO timeseriedata_new(...)
    WITH
        v1_data AS (...;

    rowcount_io := sql%rowcount;

    INSERT INTO ts_mig_log
        VALUES (energy_type_i, v1_i, v2_i, v3_i, v4_i, sysdate);
END;

```

Now the TIMESERIEDATA is inserted with both VALUE1, VALUE2, VALUE3 and VALUE4!

```

mig_run.open_fixed_energy_cursor(l_cur);

LOOP
  FETCH l_cur INTO l_energy_type, l_value1, l_value2, l_value3, l_value4, l_is_ok;
  EXIT WHEN l_cur%NOTFOUND;

  IF NOT mig_run.is_processed(l_value1, l_value2, l_value3, l_value4) THEN
    IF l_is_ok = 1 THEN
      mig_run.insert_energy_type(l_energy_type, l_value1, l_value2, l_value3, l_value4, ...);
      l_loop_status:= 'SUCCESS';
    ELSE
      mig_run.insert_error_series(l_value1, l_value2, l_value3, l_value4,...);
      l_loop_status:= 'ERROR';
    END IF;

    mig_util.log(l_runnr, l_log_type, 'SUB', l_loop_status, l_energy_type, l_start_dt,
                l_value1, l_value2, l_value3, l_value4, l_rowcount, l_seconds_ins);

    commit;
  END IF;
END LOOP;
CLOSE l_cur;

```

Alternative 2 – would be something like:

```
INSERT /*+ APPEND */
  INTO timeseriedata_new(tsd_id, fromtime, totime, reffromy, reftoy,...
                        value1, value2, value3, value4, ts_id)
  SELECT tsd_id, fromtime, totime, reffromy, reftoy, value1, null, null, null, ts_id
  FROM (SELECT tsd.tsd_id, tsd.fromtime, tsd.totime, tsd.reffromy, tsd.reftoy,
              tsd.value1, null, null, null, tsd.ts_id,
              row_number() over (partition by tsd.ts_id, tsd.fromtime
                                order by tsd.reffromy desc) rn
        FROM timeseriedata tsd, timeseries ts
        WHERE tsd.ts_id = ts.ts_id
              AND ts.tstype_id in (40,41,42,43))
 WHERE rn=1;
```

Alternative 2 – would be something like:

```
UPDATE timeseriedata_new n SET
value2=(SELECT value
        FROM (SELECT value, row_number() over (partition by fromtime
                                                order by reffromy desc) rn
              FROM timeseriedata o
              WHERE o.timeserie_id = (SELECT value2 FROM timeserie_et_mig t
                                      WHERE t.value1 = n.timeserie_id)
              AND o.fromtime = n.fromtime)
        WHERE rn=1),
value3=(SELECT value
        FROM (SELECT value, row_number() over (partition by fromtime
                                                order by reffromy desc) rn
              FROM timeseriedata o
              WHERE o.timeserie_id = (SELECT value3 FROM timeserie_et_mig t
                                      WHERE t.value1 = n.timeserie_id)
              AND o.fromtime = n.fromtime)
        WHERE rn=1),
value4=(SELECT value
        FROM (SELECT value, row_number() over (partition by fromtime
                                                order by reffromy desc) rn
              FROM timeseriedata o
              WHERE o.timeserie_id = (SELECT value4 FROM timeserie_et_mig t
                                      WHERE t.value1 = n.timeserie_id)
              AND o.fromtime = n.fromtime)
        WHERE rn=1);
```

**Table: TRAINRUNS**

| TypeName  | TSTYPE_ID | ENAME   | Into   | New Energy Type |
|-----------|-----------|---------|--------|-----------------|
| Energy 11 | 21        | act gen | value1 | Energy New 5    |
| Energy 12 | 21        | act con | value2 |                 |
| Energy 13 | 22        |         | value1 | Energy New 6    |
| Energy 14 | 23        |         | value1 | Energy New 7    |

Very few rows in TIMESERIEDATA for each TS\_ID

Using a FOR loop for each TS\_ID would be very chatty



```

WITH
  et_names AS (
    SELECT ename, guid FROM energytype GROUP BY ename, guid
  ),
  tr_ids AS (
    SELECT  value1, value2, (nvl2(VALUE1,1,0) + nvl2(VALUE2,1,0)) checksum
    FROM (
      SELECT (debitaccount || operatingday || pricecategory ||
              tmstractionunitset || tmstrain) id, ts_id, et.ename
      FROM trainrun tr, et_names et
      WHERE tr.energytype=et.guid
            AND tr.ts_id IN (
              SELECT ts_id FROM timeseries ts WHERE ts.tstype_id IN (21))
            AND energytype IN (SELECT guid FROM et_names)
      )
    PIVOT (
      max(timeserie_id)
      for (ename) in (
        'act con' as "VALUE1",
        'act gen' as "VALUE2"
      )
    )
  )

```

```

WITH
  et_names AS (
    SELECT ename, guid FROM energytype GROUP BY ename, guid
  ),
  tr_ids AS (
    SELECT value1, value2, (nvl2(VALUE1,1,0) + nvl2(VALUE2,1,0)) checksum
    ...
  )
v1_data as (
  SELECT tsd_id,..., value value1, id_value2, checksum
  FROM (SELECT t1.*, tr_ids.value2 id_value2, tr_ids.checksum,
    row_number() over (partition by ts_id, fromtime
      order by reffromy desc) rn
    FROM timeseriedata t1, tr_ids
    WHERE t1.ts_id= tr_ids.value1
      AND 1_today_epoch between t1.reffromy and t1.reftoy)
  WHERE rn=1),

```

```

WITH
  et_names AS (
    SELECT ename, guid FROM energytype GROUP BY ename, guid
  ),
  tr_ids AS (
    SELECT value1, value2, (nvl2(VALUE1,1,0) + nvl2(VALUE2,1,0)) checksum
    ...
  )
v1_data as (
  SELECT tsd_id,..., value value1, id_value2, checksum
  ...
),
v2_data as (
  SELECT ts_id,..., value1, value2, null value3, null value4, id_value2, checksum
  FROM (SELECT /*+ USE_NL(t2 v1) INDEX(t2 idx_tsd_bulk */ v1.*, t2.value value2,
    row_number() over (partition by t2.ts_id, t2.fromtime
    order by t2.reffromy desc) rn
    FROM v1_data v1, timeseriedata t2
  WHERE t2.ts_id= v1.id_value2
    AND t2.fromtime = v1.fromtime
    AND 1_today_epoch between t2.reffromy and t2.reftoy)
  WHERE rn=1)

```

```

WITH
    et_names AS (
        SELECT ename, guid FROM energytype GROUP BY ename, guid
    ),
    tr_ids AS (
        SELECT value1, value2, (nvl2(VALUE1,1,0) + nvl2(VALUE2,1,0)) checksum
        ...
    )
v1_data as (
    SELECT tsd_id,..., value value1, id_value2, checksum
    ...
),
v2_data as (
    SELECT ts_id,..., value1, value2, null value3, null value4, id_value2, checksum
    ...
)
SELECT v2.*, row_number() over (partition by v2.timeserie_id order by fromtime ) ins_log
FROM v2_data v2;

```

```

INSERT /*+ APPEND */
  ALL
  WHEN (checksum=2) THEN
    INTO timeseriedata_new(tsd_id,...,value1,value2,value3,value4)
      VALUES (tsd_id,...,value1,value2,value3,value4)
  WHEN (ins_log=1) THEN
    INTO ts_mig_log(energy_type,value1,value2,value3,value4,completed)
      VALUES('Energy New 5',ts_id, ID_VALUE2, null, null, sysdate)
  WHEN (checksum<>2) THEN
    INTO timeseriedata_err(tsd_id,...,value1,value2,value3,value4,energy_type)
      VALUES (tsd_id,...,value1,value2,value3,value4, 'Energy New 5')
WITH
  et_names    AS (...),
  tr_ids      AS (...),
  v1_data     AS (...),
  v2_data     AS (...)
SELECT v2.*, row_number() over (partition by v2.timeserie_id order by fromtime ) ins_log
FROM v2_data v2;

```

## **Task: Dropping TIMESERIES**

From TRAINRUNS, FIXED and TIMESERIES table  
Which has been move into VALUE2, VALUE3 and VALUE4

```

-----
-- Drop old records from TRAINRUNS (TRR)
-----

DECLARE
    CURSOR c_ts IS
        SELECT * FROM TRAINRUNS
        WHERE ts_id IN (
            SELECT value
            FROM (SELECT value2, value3, value4
                  FROM ts_mig_log
                  WHERE value2 IS NOT NULL)
            UNPIVOT EXCLUDE NULLS (
                VALUE FOR COL IN (value2, value3, value4))
        );

    TYPE ts_aat IS TABLE OF c_ts%ROWTYPE;
    l_ts      ts_aat;
BEGIN
    ...
END;

```

```

-----
-- Drop old records from TRAINRUNS (TRR)
-----

DECLARE
    ...
BEGIN
    OPEN c_ts;
    LOOP
        FETCH c_ts BULK COLLECT INTO l_ts LIMIT l_bulk_limit;
        EXIT WHEN l_ts.count = 0;

        FORALL i IN 1..l_ts.COUNT
            INSERT /*+ APPEND_VALUES */ INTO TRAINRUNS_BACKUP
                VALUES(l_ts(i).ts_id, ...);
        FORALL i IN 1..l_ts.COUNT
            DELETE FROM flexible WHERE ts_id=l_ts(i).ts_id;

        COMMIT;
    END LOOP;
END;

```



## The only things left are ...

*Copying data for tstype\_id in (22,23), but this is straight forward:*

```
INSERT /*+ APPEND */ INTO timeseriedata_new  
SELECT ... FROM timeseriedata  
WHERE tstype_id IN (22,23);
```

*switching new and old table, but this is also pretty straight forward 😊*



[lasse.jenssen@rightconsulting.no](mailto:lasse.jenssen@rightconsulting.no)

Twitter: @lasjen

<http://www.rightconsulting.no>

