Mind the Gap
Lücken und Muster erkennen mit Match_Recognize

18. November 2015

Robert Marz
Technical Architect

Robert Marz

Kunde

- Technical Architect mit datenbankzentrischem Weltbild

its-people

- Portfoliomanager Datenbanken
- Gesellschafter

DOAG

- Aktiv in der Datenbank Community themenverantwortlich Hochverfügbarkeit

@RobbieDatabee

www.its-people.de/blog

Robert.Marz @its-people.de

its-people auf einen Blick

Unternehmensphilosophie

- Zusammenschluss selbständiger IT-Experten unter einer gemeinsamen Marke
- Bündelung von umfassendem IT-Wissen
- Partnerschaft auf Augenhöhe
- Gemeinsam stark!

Unternehmensdaten

- Gründung: 2003
- Anzahl der Experten: 70

Leistungsangebot

DatenbankTechnologie
Plattform & Anwendungsentwicklung

Business Intelligence & DWH
Architektur, Design & Implementierung

Customer Data Integration
Beratung, Konzepte & Projekte

Best Practice ERP
Mediation, Integration, Migration

IT Management Services
Projekt-, Qualitäts-, Test-Mgmt

Beratung, Planung, Konzeption, Implementierung,
Support, Krisenbewältigung

Stärken

- Kompetenz, Exzellenz und Qualität aus praktischer Erfahrung
- Partnerschaft auf Gegenseitigkeit
- Kundenorientierung auf höchster Ebene
- Wertekodex als Geschäftsgrundlage
- Teams für komplexe Projektaufgaben
- Nähe zum Kunden
- Dienstleistungen von Menschen für Menschen

its-people
Ihr Partner für Projekte im Umfeld Datenbanken, Data Warehouse, Business Intelligence, Customer Data Integration und ERP

its-people Frankfurt
Lyoner Str. 44-48, 60528 Frankfurt
Tel.: +49 (69) 2475 210-0
E-Mail: frankfurt@its-people.de

its-people ERP Beratungsgesellschaft mbH
Lyoner Str. 44-48, 60528 Frankfurt
Tel.: +49 (69) 2475 19-80
E-Mail: erp@its-people.de

its-people Köln
Hohenzollernring 57, 50672 Köln
Tel.: +49 (221) 1602 5204
E-Mail: koeln@its-people.de

its-people München
Lichtenbergstr. 8, 85748 Garching
Tel.: +49 (89) 5484 2401
E-Mail: muenchen@its-people.de

its-people Hamburg
Am Strohhause 31, 20097 Hamburg
Tel.: +49 (40) 2360 8808
E-Mail: hamburg@its-people.de

Unsere Vorträge auf der DOAG 2015

Dienstag

Virtual Reality Analytics
12:00 - 12:45 Uhr Raum Oslo, Ebene 2
Jörg Osarek

SQL Developer – Lassen Sie sich anstecken!
12:00 - 12:45 Uhr Raum Sydney, Ebene 1
Sabine Heimsath

Kommandozeile 2.0 – SQLcl –
Ein Erfahrungsbericht
15.00 – 15.45 Uhr Raum Sydney, Ebene 1
Jens Behring

Divide et impera –
Session-Management im ETL
16.00 – 16.45 Uhr Raum Oslo, Ebene 2
Sven Bosinger

Mittwoch

Mind the Gap: Lücken und Muster erkennen mit Match-Recognize
12:00 - 12:45 Uhr Raum Helsinki, Ebene 2
Robert Marz

Donnerstag

Nächster Halt 12c: Migration bei der DB Netz
12:00 - 12:45 Uhr Raum Seoul, Ebene 3
Robert Marz

These: Werkverträge in der IT sind anachronistisch!
12:00 - 12:45 Uhr Raum Oslo, Ebene 2
Thomas Algermissen

Customer Analytics –
Erkenntnisse einer 20-jährigen Reise!
14.00- 14.45 Uhr Raum Oslo, Ebene 2
Frank Sommerer

2015 DOAG Konferenz + Ausstellung

Beispiel eins: Zusammenhängende Gruppen (1/3)

Aufgabe:
Finde in einer sortierten Liste zusammenhängende Gruppen von Werten.
Bestimme Start, Ende und Anzahl der Elemente.
Zusammenhängend bedeutet: Abstand von 1

Beispiel eins: Zusammenhängende Gruppen (2/3)

Lösung mit Analytical Functions:

```
select min(wert) anfang
     , max(wert) ende
     , count(*)  anzahl
from (
select wert
     , row_number() over(order by wert) as zeilennr
     , wert - row_number() over(order by wert) as gruppe
  from b01
)
group by gruppe
order by anfang;
```

1	2	3	6	7	8	9	12	13	14	15

Zeilen Muster Definieren

Ein Muster (Schlüsselwort Pattern)

- Eine zusammenhängende Reihe von Zeilen
- Aufeinanderfolgende RegExp Ausdrücke:

```
PATTERN (A B*) -- „A“: 1 Zeile, „B*“ 0 oder mehr – so viele Zeilen wie möglich
```

Bedingungen definieren (Schlüsselwort Define)

```
DEFINE -- Für A nichts angegeben bedeutet TRUE
       B as wert = prev(wert)+1
```

Wieviel Zeilen sollen erzeugt werden?

```
ONE ROW PER MATCH | ALL ROWS PER MATCH
```

Ausgabespalten werden neu definiert und ersetzen die Eingabespalten

```
MEASURES a.wert anfang, last(wert) ende
         , count(*) anzahl
```

1	2	3	6	7	8	9	12	13	14	15

Anatomie von MATCH_RECOGNIZE

Eingangsdaten

Reihenfolge

Ausgabe: Spaltendefinition

Ausgabe: Wieviel Zeilen

Wie geht's weiter

Zeilenmuster

Bedingungen des Musters

```
select *
  from b01
 match_recognize (
  order by wert
  measures a.wert          anfang
         , last(wert)      ende
         , count(*)        anzahl
  one row per match
  after match skip past last row
  pattern (a b*)
  define b as wert = prev(wert)+1
);
```

Beispiel eins: Zusammenhängende Gruppen (3/3)

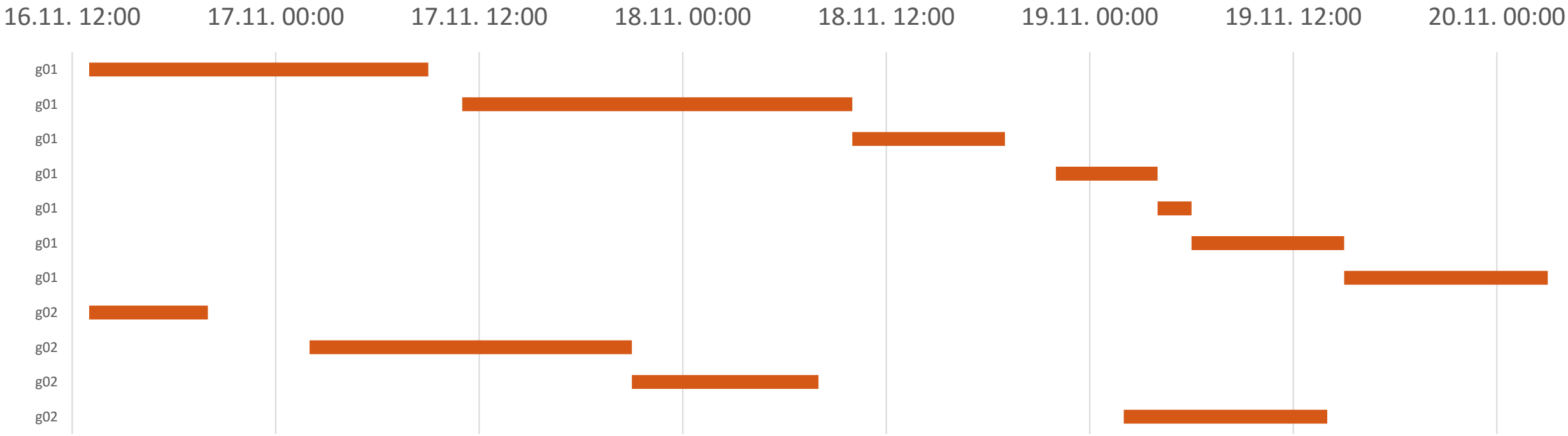
Lösung mit Match Recognize:

```
select *
  from b01
 match_recognize (
  order by wert
  measures a.wert          anfang
         , last(wert)      ende
         , count(*)        anzahl
  -- one row per match                    -- default
  -- after match skip past last row       -- default
  pattern (a b*)
  define b as wert = prev(wert)+1
);
```

1	2	3	6	7	8	9	12	13	14	15

Beispiel zwei: Zeiträume Zusammenfassen (1/3)

Aufgabe:
Fasse angrenzende Zeiträume innerhalb einer Gruppe zu einem Satz zusammen.



Beispiel zwei: Zeiträume Zusammenfassen (2/3)

Lösung mit Analytical Functions:

```
with
  startzeitpunkte
  as ( select strt.gruppe
            , case
                when startzeit
                     = lag(endzeit)
                       over(partition by gruppe
                            order by startzeit
                           )
                then 0
              else 1
              end   gstart
            , strt.startzeit
            , strt.endzeit
         from b02 strt
     )
, gruppen
   as ( select stzp.gruppe
               , sum(stzp.gstart)
                  over( partition by gruppe
                        order by startzeit
                      ) gruppenid
               , stzp.gstart
               , stzp.startzeit
               , stzp.endzeit
           from startzeitpunkte stzp)
select gruppe
     , min(startzeit) startzeit
     , max(endzeit) endzeit
from gruppen
group by gruppe, gruppenid
order by gruppe, startzeit;
```

Beispiel zwei: Zeiträume Zusammenfassen (3/3)

Lösung mit Match Recognize:

In diesem Beispiel neu:

- Einsatz von PARTITION BY
- Zugriff auf eine Zeile außerhalb des Match mit NEXT() für das Measure „luecke“

```
select gruppe
     , startzeit
     , endzeit
     , luecke
  from b02
 match_recognize (
  PARTITION BY gruppe
  order by startzeit
  measures
      a.startzeit startzeit
    , endzeit endzeit
    , NEXT(startzeit) - endzeit luecke
  pattern(a b*)
  define b as startzeit = prev(endzeit)
);
```

Bonus Beispiel drei: Ask-Tom Special (1/3)

Aufgabe:
Ermittle aus Gruppen von Datensätzen jeweils die Differenz von Werten der neuesten beiden pro Gruppe.

TESTNR	WERT	ZEITPKT
1	5	12:42:00
1	7	12:54:39
1	2	14:48:30
2	16	13:19:57
2	20	15:01:09
3	-42	13:19:57

[Ask Tom Beispiel](https://asktom.oracle.com/pls/apex/f?p=100:11:::NO::P11_QUESTION_ID:9522851900346371735#followup-9522863800346005939
"SQL query that returns the difference between the latest")

Bonus Beispiel drei: Ask-Tom Special (2/3)

Lösung mit Analytic Functions

```
with ranked_data
  as ( select t.testnr
            , t.wert
            , t.zeitpkt
            , row_number() over (partition by t.testnr order by t.zeitpkt desc) as rk
            , lag(t.wert)  over (partition by t.testnr order by t.zeitpkt) as prev_wert
         from testlaeufe t
     )
select testnr
     , wert - prev_wert as delta
  from ranked_data
 where rk = 1
      and prev_wert is not null
;
```

Bonus Beispiel drei: Ask-Tom Special (3/3)

Lösung mit Match Recognize

```
select testnr
     , delta
  from testlaeufe t
  match_recognize(
    partition by testnr order by zeitpkt desc
    measures z1.wert - z2.wert delta
    pattern(^z1 z2)   -- Die erste und zweite Zeile
    define z1 as 1=1  -- Ein define-Ausdruck muss da sein
      -- , z2 as 1=1  -- Kann entfallen
    );
```

Patterns: Quantifiers & Co

Quantifier	Treffer
*	0 oder mehr
+	1 oder mehr
?	0 oder 1
{n,m}	Zwischen n & m

Quantifiers sind gierig (greedy):

```
PATTERN (X Y* Z)
```

Alle Zeilen, die für Y und Z Treffer ergeben, werden Y zugeschlagen

Der widerspenstige (reluctant) Quantifier „?“ als Suffix präferiert das folgende Pattern:

```
PATTERN (X Y*? Z)
```

Neue Optimizer Operations / Performance

Der Einsatz ist abhängig von dem verwendeten PATTERN.

- MATCH RECOGNIZE(BUFFER)
- MATCH RECOGNIZE(SORT)
- MATCH RECOGNIZE(SORT DETERMINISTIC FINITE AUTO)
  - Kommt zum Einsatz, wenn kein Zugriff auf vorherige Zeilen nötig ist (Kein Backtracking)

Keith.Laker@oracle.com:

Match_recognize ist 1.5 – 1.9 mal schneller als der Einsatz von Window-Functions und skalliert hervoragend.

(Quelle: https://blogs.oracle.com/datawarehousing/entry/sessionization_analysis_with_12c_sql)

Beispiel vier: Behälterproblem - Feste Größe (1/3)

Ab demnächst im ist-people BLOG
http://www.its-people.de/blog

Beispiel fünf: Behälterproblem - Feste Anzahl (1/3)

Ab demnächst im ist-people BLOG
http://www.its-people.de/blog

Beispiel sechs: Vorsicht bei wiederholenden Bedingungen

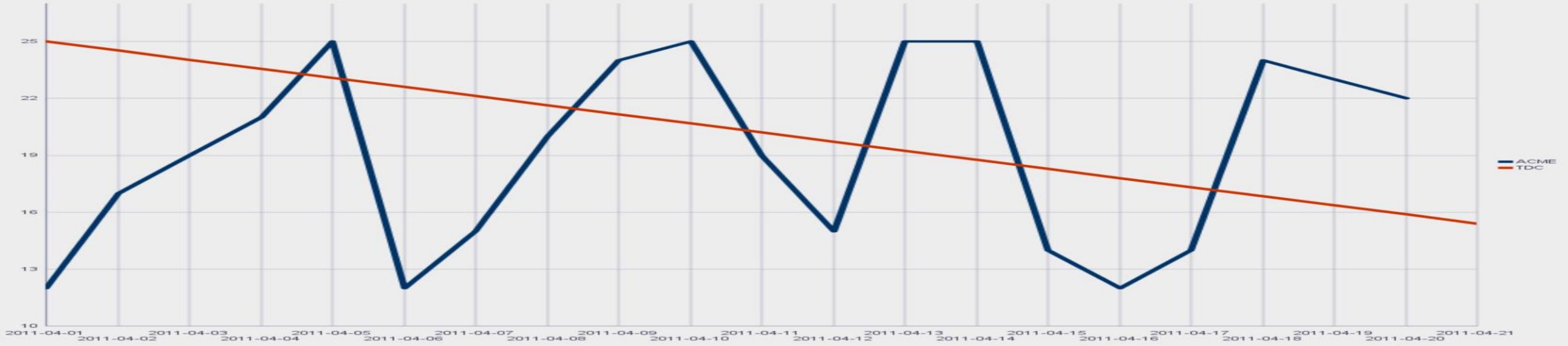
Regular Expression Maschinen testen ALLE möglichen Kombination – das beeinflusst die Laufzeit

```
select 'match'
  from ( select level n
           from dual
         connect by level <= 100
       )
  match_recognize
  ( pattern(a b* c)
    define b as n > prev(n)
         , c as n = 0
  )
;

-- 0,007 Sekunden Laufzeit
```

```
select 'match'
  from ( select level n
           from dual
         connect by level <= 100
       )
  match_recognize
  ( pattern(a b* b* b* c)
    define b as n > prev(n)
         , c as n = 0
  )
;

-- 3,131 Sekunden Laufzeit
```

Beispiel sieben: Stockticker (1/3)

Aufgabe:
Finde das „W“: SQL for Pattern Matching "Example 20-4 Pattern Match for a W-Shape“
https://docs.oracle.com/database/121/DWHSG/pattern.htm#DWHSG8970



Beispiel sieben: Stockticker (2/3)

Statement aus der Doku: 16 Sekunden

```
SELECT *
FROM Ticker MATCH_RECOGNIZE (
  PARTITION BY symbol
  ORDER BY tstamp
  MEASURES
    MATCH_NUMBER() AS match_num,
    CLASSIFIER()  AS  var_match,
    STRT.tstamp AS start_tstamp,
    FINAL LAST(UP.tstamp) AS end_tstamp
  ALL ROWS PER MATCH
  AFTER MATCH SKIP TO LAST UP
  PATTERN (STRT DOWN+ UP+ DOWN+ UP+)
  DEFINE
    DOWN AS DOWN.price < PREV(DOWN.price)
    , UP AS UP.price > PREV(UP.price)

  ) MR
ORDER BY MR.symbol, MR.match_num, MR.tstamp;
```

Beispiel sieben: Stockticker (3/3)

STRT-Bedingung definiert: 0,016 Sekunden

```
SELECT *
FROM Ticker MATCH_RECOGNIZE (
  PARTITION BY symbol
  ORDER BY tstamp
  MEASURES
    MATCH_NUMBER() AS match_num,
    CLASSIFIER()  AS  var_match,
    STRT.tstamp AS start_tstamp,
    FINAL LAST(UP.tstamp) AS end_tstamp
  ALL ROWS PER MATCH
  AFTER MATCH SKIP TO LAST UP
  PATTERN (STRT DOWN+ UP+ DOWN+ UP+)
  DEFINE
    DOWN AS DOWN.price < PREV(DOWN.price)
    , UP AS UP.price > PREV(UP.price)
    , STRT as price >=  nvl(prev(price),0)
  ) MR
ORDER BY MR.symbol, MR.match_num, MR.tstamp;
```

Tipps & Tricks (1/2)

Nachdenken

- Bedingungen und Pattern so genau wie möglich formulieren

Testen

- Normale Werte
- Grenzwerte
- Sonderfälle – Datenmengen ohne Treffer

Debuggen

- Measures Funktionen
  - classifier()
  - match_number()
- All rows per match with unmatched rows

Tipps & Tricks (2/2)

MEASURES

- Spalten MÜSSEN Aliase haben
- DISTINCT & LISTAGG nicht zulässig

Achtung

- “?” Kennzeichnet JDBC Bind Variablen
- “?” ist ein Pattern Quantifier
- Kein Workaround bisher

Leseliste

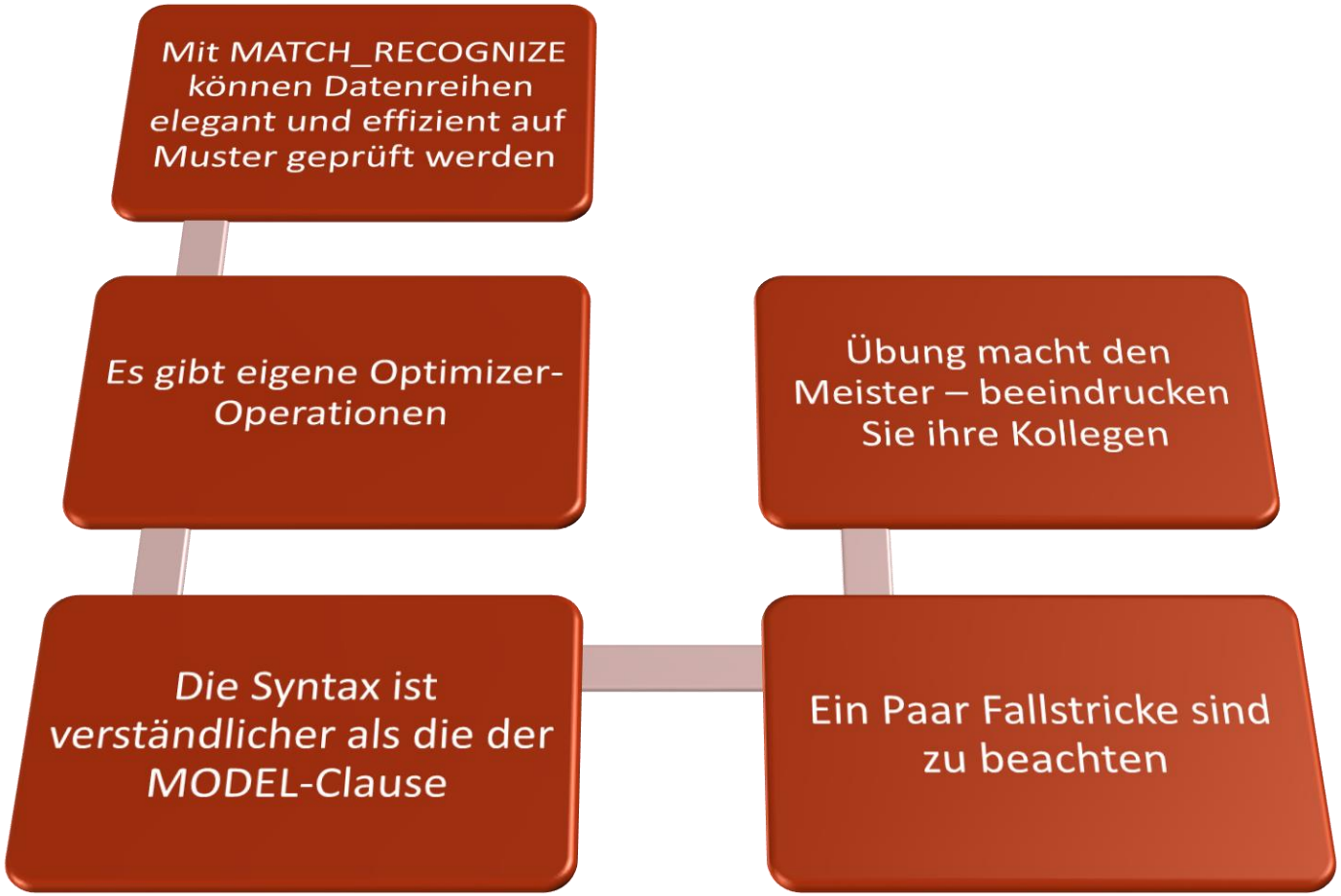
- Stew Ashton – „An Oracle Programmer“
  @StewAshton
  https://stewashton.wordpress.com/category/match_recognize/
  Die einige Beispiele in diesem Vortrag basieren auf seiner Vorarbeit.
  Die Verwendung wurde freundlicherweise von ihm genehmigt.

- Database Data Warehousing Guide - Kapitel 20: SQL for Pattern Matching
  https://docs.oracle.com/database/121/DWHSG/pattern.htm

- The Data Warehouse Insider - MATCH_RECOGNIZE and the Optimizer
  By Klaker-Oracle
  https://blogs.oracle.com/datawarehousing/entry/match_recognize_and_the_optimizer

- The Data Warehouse Insider - Sessionization analysis with 12c SQL pattern matching is super fast
  By Klaker-Oracle
  https://blogs.oracle.com/datawarehousing/entry/sessionization_analysis_with_12c_sql

Fazit


Mit MATCH_RECOGNIZE können Datenreihen elegant und effizient auf Muster geprüft werden
Es gibt eigene Optimizer-Operationen
Übung macht den Meister – beeindrucken Sie ihre Kollegen
Die Syntax ist verständlicher als die der MODEL-Clause
Ein Paar Fallstricke sind zu beachten

Wann	Von Donnerstag, 21. Januar 2016 12:00 Uhr bis Freitag, 22. Januar, 12:00 Uhr	
Wo	GHOTEL hotel & living Würzburg	
Themen	RAC & Dataguard in der Praxis	Vorträge RAC Attack Workshop (in deutsch) Dataguard Workshop Networking
URL	http://www.doag.org/termine/termine.php?tid=507125	

we make the difference
www.its-people.de

Herzlichen Dank für Ihre Aufmerksamkeit !

Ihre Fragen ?

its-people GmbH
Frankfurt Tel. 069 2475 2100
Hamburg Tel. 040 2360 8808
Köln Tel. 0221 1602 5204
München Tel. 089 5484 2401

its-people ERP Beratungsgesellschaft mbH
Frankfurt Tel. 069 2475 1980

www.its-people.de **info@its-people.de**