



Dierk Lenz

SQL Macros und andere Performance- Verbesserungen in Oracle 21c

#ittage

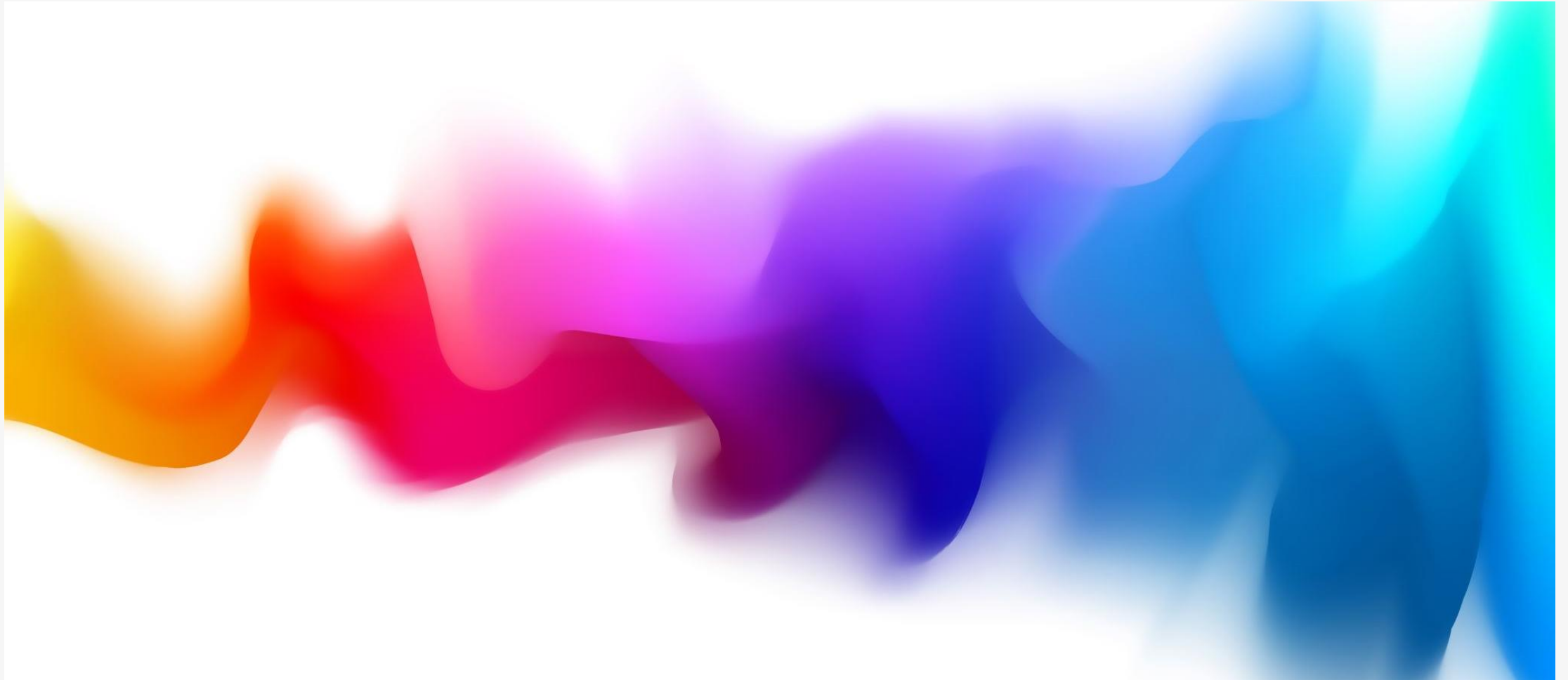
Herrmann & Lenz Services

Herrmann & Lenz Solutions



- Erfolgreich seit 1996 am Markt
- Firmensitz: Burscheid (bei Leverkusen)
- Beratung, Schulung und Betrieb/Fernwartung rund um das Thema Datenbanken
- Schwerpunktthemen: Hochverfügbarkeit, Tuning, Migrationen und Troubleshooting / Monitoring
- Herrmann & Lenz Solutions GmbH
 - Produkte: Monitoring Module, Taskzone, TimeTracking

Allgemeines zu Oracle 21c



Das Innovation Release Oracle 21c

- Folgt auf Oracle 19c
- Oracle 20c wurde pandemiebedingt gestrichen
- Innovation Release...
 - Ende Premier Support 30.6.2023
 - Kein Extended Support
- Zum Vergleich Oracle 19c (Long Term Release)
 - Ende Premier Support 30.4.2024
 - Ende Extended Support 30.4.2027
- Erstes Multitenant-only Release!

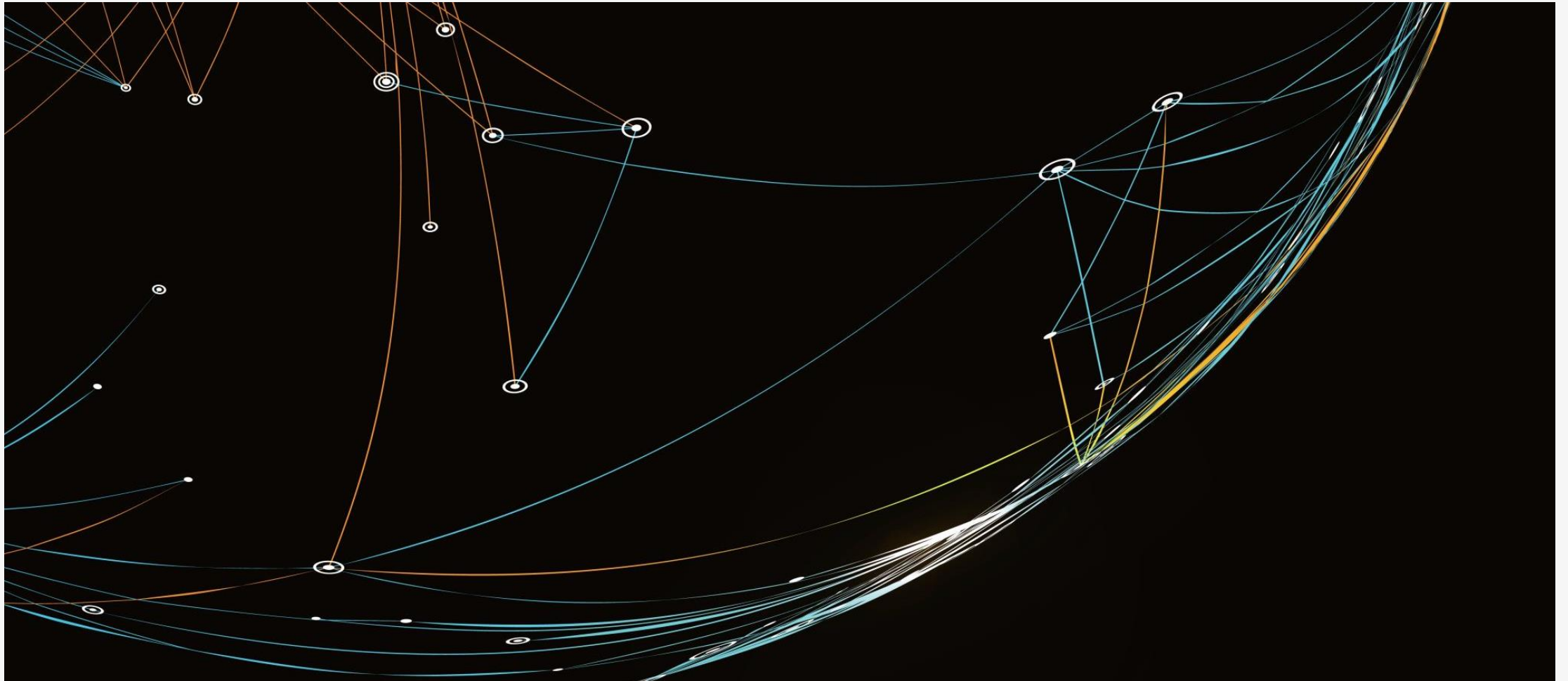
Warum Oracle21c einsetzen?

- Nicht, wenn Anwendungen mit Oracle19c laufen!
- Nur dann, wenn Eigenschaft oder Feature so wichtig, dass der zusätzliche Aufwand (erneuter Upgrade vor 30.6.2023) gerechtfertigt ist!

Oracle 21c testen?

- On-premises noch nicht verfügbar
- Ankündigung für 1. Halbjahr 2021
- Cloud! → Autonomous Database „Always Free“ Angebot

Performance Features in 21c



Allgemeines

- 21c offenbar kein Performance-Release
- Eher: Erweiterung durch sinnvolle Features
- Belastbare Daten erst mit Verfügbarkeit der On-premises-Version
- Auch Verbesserungen *behind the scenes* (z.B. Optimizer-Details) erst später erkennbar

SQL Quarantine

- Vorgabe von maximaler Laufzeit im Resource Manager
- Bei Überschreitung: Abbruch der Ausführung
- Neu mit SQL Quarantine:
 - Verwendeter Ausführungsplan wird in Quarantäne gesetzt
 - SQL kann erneut laufen, aber nicht mit dem Plan in Quarantäne
 - Vermeidung des wiederholten Ressourcenaufwands

Online Timezone File Upgrade

- Bisher bei Upgrades des Timezone Files: längere Downtime erforderlich, da Datenbank im Upgrade-Modus gestartet
- Nun Upgrade bei offener Datenbank möglich
- Initialisierungsparameter
`timezone_version_upgrade_online = [true | false]`
- Stolperfalle: abschließender Neustart der Instanz (zeitlich unabhängig) erforderlich

Defragmentierung von SecureFile LOBs

- Operation Shrink Space verfügbar für SecureFile LOBs
- Für LOBs explizit:

```
ALTER TABLE <tab> MODIFY LOB <lobc> SHRINK SPACE;
```

- Als Teil der CASCADE-Option:

```
ALTER TABLE <tab> SHRINK SPACE CASCADE;
```

SQL Macros

- Problem: PL/SQL-Funktionen in SQLs (SELECT-Liste, Filter)
 - Wechsel zwischen SQL- und PL/SQL-Engine
 - Extrem lange Laufzeiten, insbesondere bei großen Datenmengen
- Neu: SQL Macros!
 - Definition mit `PRAGMA SQL_MACRO (TYPE = { SCALAR | TABLE })`
- Idee: PL/SQL-Funktionen, die lediglich SQL-Funktionalität beinhalten, effizienter einbinden
- SQL Macros nicht als Performance-Feature präsentiert – können sich aber sehen lassen

Rückblick auf 12c: Pragma UDF

- Kaum bekannte Verbesserung, eingeführt in Oracle Database 12c: ein Pragma zur Kennzeichnung von *user-defined functions*

```
PRAGMA UDF;
```

- Hintergrund:
 - Falls PL/SQL-Funktion hauptsächlich in SQLs verwendet wird
 - Zur Laufzeit erforderliche Aufgaben werden vorgezogen zur Compile-Zeit
 - Möglicher Nachteil: ggfs. längere Laufzeit bei der Verwendung direkt aus PL/SQL

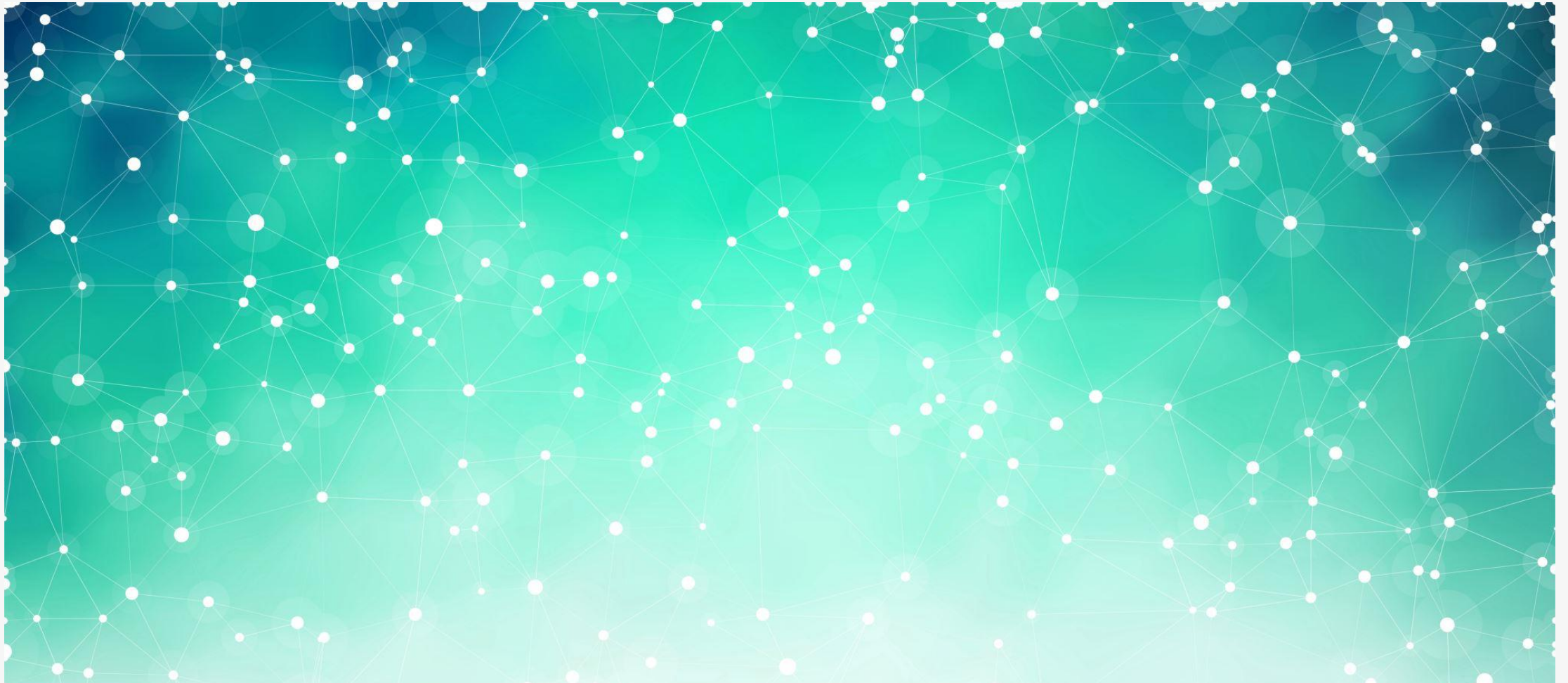
Was ist eine SQL Macro-Funktion?

- PL/SQL-Funktion, die als SQL_MACRO ausgewiesen ist
 - Typ SCALAR:
 - Nutzung in SELECT-Liste oder WHERE-Bedingung
 - RETURN-Typ VARCHAR2, CHAR oder CLOB
 - Typ TABLE:
 - Nutzung in FROM-Klausel
- Rückgabewert ist ein String, der als SQL-Ausdruck (SCALAR) oder SQL-Befehl (TABLE) ausgewertet wird
- SQL Macros vom Typ TABLE entspricht einer parametrisierten View!
- Praktisch (in SQL und PL/SQL): Der Quote-Operator q – Vermeidung von doppelten Anführungszeichen in Strings
Statt 'TO_NUMBER(''7'')' q '{TO_NUMBER('7')}'
(gibt es auch schon länger...)

SQL Macro Einschränkungen

- SQL Macros innerhalb von Views werden mit Owners Rights ausgeführt
- AUTHID kann nicht angegeben werden; Ausführung grundsätzlich mit Invokers Rights
- Beim Aufruf aus PL/SQL wird der definierte String zurückgegeben (nicht mit SQL interpretiert!)

Demo



Beispiel 1

```
CREATE OR REPLACE FUNCTION m1 (vin pls_integer)
RETURN VARCHAR2 SQL_MACRO (SCALAR)
AS
BEGIN
    RETURN
    q' {
        TO_CHAR(vin + 1)
    }';
END;
/
```

```
SELECT /*+ no_result_cache */ count(m1(gencol))
FROM (
    SELECT LEVEL gencol
    FROM dual
    CONNECT BY LEVEL <= 1000000
)
;
```

Beispiel 2

```
CREATE OR REPLACE FUNCTION get(nrows IN pls_integer)
RETURN VARCHAR2 SQL_MACRO(TABLE)
AS
BEGIN
    RETURN q' {
        SELECT LEVEL gencol, 'Blabla' as txt
        FROM dual
        CONNECT BY LEVEL <= get.nrows
    }';
END;
/

SELECT * FROM get(105);
```

Vergleich PL/SQL Pur, Pragma UDF, SQL Macro (wie Beispiel 1)

	Test 1	Test 2	Test 3	Mittel
Pur	1,417	1,542	1,495	1,485
Pragma UDF	0,576	0,526	0,536	0,546
SQL Macro	0,373	0,380	0,370	0,374

Achtung: Das ist ein Ad-Hoc-Test; kein Benchmark!!!

Fazit SQL Macros

- Aufgeräumter Code
- Parametrisierte Views
- Beschleunigung von PL/SQL-in-SQL-Code möglich
- UDF-Pragma für frühere Versionen und komplexeren Code
- Noch eine gute Nachricht für Oracle 19c
 - SQL Macros vom Typ TABLE verfügbar ab 19c (RU 19.7)
 - TYPE-Klausel weglassen

Das wichtigste Feature



Neue Deutsche Sortierreihenfolgen

- XGERMAN_S und XGERMAN_DIN_S
- Berücksichtigen das große „ß“!

(Nein, das ist kein Scherz.)

- OK, nicht ganz so wichtig für die Performance...

Fazit

- Interessante neue Features in Oracle 21c
- Eher Produktpflege als große Neuigkeiten
- Aktuelle Empfehlung: Oracle 19c rules!

Vielen Dank für Ihre Aufmerksamkeit!

Fragen & Kontakt

- Mail: dierk.lenz@hl-services.de
- Web: www.hl-services.de
- Blog: blog.hl-services.de
- Twitter: @ora1578