

DOAG 2018 BIG DATA DAYS

20. und 21. September 2018 in Dresden

#BIGDATADAYS

Ausgewählte Oracle Datenbank Features im Data Warehouse Umfeld

Ulrike Schwinn (Ulrike.Schwinn@oracle.com, @uschwinn)

BU Core & Cloud Technologies
Oracle Deutschland B.V. & Co KG

Data Warehouse Features in der Oracle Datenbank

External Tables Extended Datentyp

VARCHAR2 JSON Database Filesystem

XMLTYPE XMLDB Sentiment Analysis

Approximate Funktionen Object Storage

Analytische Funktionen Temporäre Tabellen Partitionierung

Materialized Views Parallel Query Parallel DML, Komprimierung

Database Resource Manager Analytische Views

Pattern Matching Pivotierung Optimizer Plan Management

RDF Graph AUTONOMOUS

Approximate Funktionen in Oracle

- Herausforderungen bei Analysen von großen Datenmengen
 - das Sortieren/Gruppieren/Berechnen großer Datenmengen
 - Speicherplatz (Temp Tablespace)
 - zeitaufwendig und Ressource intensiv
- Seit Oracle Database 12.1: Approximate Funktionalität
 - Verwendung von Optimierungstechniken um (ungefähre) Ergebnisse mit hoher Genauigkeit zu berechnen
 - => deutlich reduzierter Ressource Verbrauch und um Faktoren schneller
- **Blogbeitrag:** <https://blogs.oracle.com/coretec/approximate-funktionen-in-der-oracle-datenbank>

12.1: COUNT (DISTINCT) vs. APPROX_COUNT_DISTINCT

```
SQL> select t1,  
count(distinct(t))  
from approx_t group by t1;
```

```
T COUNT(DISTINCT(T))
```

```
-----
```

a	1999999
b	2999999
x	999999

Elapsed: **00:00:07.93**

```
SQL> select t1,  
approx_count_distinct(t)  
from approx_t group by t1;
```

```
T APPROX_COUNT_DISTINCT(T)
```

```
-----
```

a	1939328
b	2944730
x	971092

Elapsed: **00:00:02.36**

Statt COUNT (DISTINCT) nun APPROX_COUNT_DISTINCT

The screenshot shows the Oracle SQL Developer interface. At the top, the status bar indicates a duration of 11,06000042 seconds and the user 'bumuc_122_sh'. The 'Query Builder' tab is active, displaying two SQL queries:

```
select t1, count(distinct(t)) from approx_t group by t1;  
select t1, approx count distinct(t) from approx_t group by t1;
```

Below the query editor, the 'Script Output' and 'Autotrace' tabs are visible. The 'Autotrace' tab shows a comparison of execution plans for the two queries. The left plan for the first query shows a 'HASH (GROUP BY APPROX)' step. The right plan for the second query shows a 'HASH (GROUP BY)' step, a 'VIEWVM_NWWW_1' step, and another 'HASH (GROUP BY)' step. Both plans include 'TABLE ACCESS (FULL) APPROX_T' and 'Other XML' steps.

At the bottom, the 'Autotrace' output is displayed in a table:

V\$STATNAME Name	V\$MYSTAT Value Autotrace 1	V\$MYSTAT Value Autotrace
consistent gets	10371	10371
consistent gets from cache	10371	10371
consistent gets pin	10371	10371
consistent gets pin (fastpath)	10371	10371
CPU used by this session	413	619
CPU used when call started	413	619
DB time	415	790

Einfach mit Initialisierungsparameter

```
SQL> alter session set APPROX_FOR_COUNT_DISTINCT =true;
```

```
Session altered.
```

```
SQL> select t1, count(distinct(t)) from approx_t group by t1;
```

```
T COUNT(DISTINCT(T))
```

```
- -----
```

```
a          1939328
```

```
b          2944730
```

```
x           971092
```

```
Elapsed: 00:00:02.29
```

Berechnung von Aggregationen

Ausgangsbeispiel

- Berechnung von Produktverkäufen pro Jahr, pro Monat oder pro Tag

```
SQL> SELECT t.calendar_year year,  
           t.calendar_month_number month,  
           t.day_number_in_month day, s.prod_id  
FROM sh.times t, sh.sales s  
WHERE t.time_id = s.time_id order by year, month, day;
```

YEAR	MONTH	DAY	PROD_ID
1998	1	1	37
1998	1	1	37
1998	1	1	37
1998	1	1	37
...			
1998	1	1	44
1998	1	1	44

12.2: Mit analytische Funktionen

- **APPROX_COUNT_DISTINCT_DETAIL**: eine Art Helper Funktion, die die zusammengefassten Inhalte (Synopsen) in einem BLOB ausgibt
 - Speicherung in Tabelle oder materialisierten View
- **APPROX_COUNT_DISTINCT_AGG**: aggregiert die vorberechneten Informationen
- **TO_APPROX_COUNT_DISTINCT**: liefert lesbare Ausgabe

Vorgehensweise

2. **APPROX_COUNT_DISTINCT_AGG** und **TO_APPROX_COUNT_DISTINCT**

```
SQL> SELECT year,  
          TO_APPROX_COUNT_DISTINCT(APPROX_COUNT_DISTINCT_AGG(daily_detail)) yearly  
FROM daily_prod_count GROUP BY year order by year;
```

YEAR	YEARLY
1998	60
1999	72
2000	72
2001	71

```
SQL> SELECT year, month,  
          TO_APPROX_COUNT_DISTINCT(APPROX_COUNT_DISTINCT_AGG(daily_detail)) monthly  
FROM daily_prod_count  
GROUP BY year, month order by year, month;
```

...

Top N Berechnungen

- Ab 18c neue Funktionen
 - APPROX_COUNT
 - APPROX_SUM
- Anwendung
 - Berechnet die (ungefähren) Top-n-Werte, die bei Verwendung mit der Funktion APPROX_RANK am häufigsten vorkommen
 - Einschränkung in der HAVING Klausel erforderlich

Ausgangssituation

```
SQL> select owner, object_type, count(*) zahl
      from test_objects
      group by owner, object_type
      order by 1,3 desc;
```

OWNER	OBJECT_TYPE	ZAHL
APEX_050100	INDEX	1337
APEX_050100	TRIGGER	501
APEX_050100	TABLE	442
APEX_050100	VIEW	283
APEX_050100	PACKAGE	279
APEX_050100	PACKAGE BODY	272
APEX_050100	TYPE	14
APEX_050100	PROCEDURE	13

...

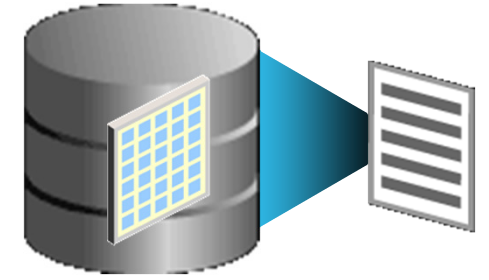
18c: 3 häufigsten Vorkommen von Objekttypen

APPROX_COUNT und APPROX_RANK

```
SQL> select owner, object_type, approx_count(*), approx_rank(partition by owner
order by approx_count(*) desc) rang
from test_objects
group by owner, object_type
having approx_rank(partition by owner order by approx_count(*) desc) <= 3
order by 1,3 desc;
```

OWNER	OBJECT_TYPE	APPROX_COUNT (*)	RANG
APEX_050100	INDEX	1337	1
APEX_050100	TRIGGER	501	2
APEX_050100	TABLE	442	3
APEX_LISTENER	SYNONYM	22	1
APPQOSSYS	TABLE	5	1
APPQOSSYS	SYNONYM	1	2
AUDSYS	INDEX PARTITION	24	1
AUDSYS	TABLE PARTITION	8	2

External Tables Funktionen



- Generell zum Laden und Entladen von Daten
- Datenaufbereitung mit Preprocessing möglich
- Zugriff auf heterogene Daten (z.B. XML, JSON, Daten in Hadoop)
- 12.2: Performanter Zugriff auf große Datenmengen
 - Über Partitionierung (Range, List, Composite Range, Composite List)
 - Über deklarative Constraints
- 12.2 Flexibilität durch Parameter Änderungen zur Laufzeit
- Blogbeiträge: https://blogs.oracle.com/coretec/?search_terms=external%20table

Beispiel: External Table mit List Partitionierung

```
create table ext_part
(last_name varchar2(25), gender varchar2(1), city varchar2(50),
 birth_year number)
ORGANIZATION EXTERNAL
( TYPE ORACLE_LOADER
  DEFAULT DIRECTORY home
  ACCESS PARAMETERS( RECORDS DELIMITED BY NEWLINE
                    FIELDS TERMINATED BY ';' optionally enclosed BY '"'
                    MISSING FIELD VALUES ARE NULL))
REJECT LIMIT UNLIMITED
PARTITION BY LIST (gender)
( PARTITION part_f VALUES ('F') LOCATION (home:'cust_f.csv'),
  PARTITION part_m VALUES ('M') LOCATION (home:'cust_m.csv'));
```

- Mögliche Access Treiber: ORACLE_LOADER, ORACLE_HIVE und ORACLE_HDFS

Beispiel: External Table ohne LOCATION Parameter

```
create table ext_flex
( last_name varchar2(25), gender
  varchar2(1), city varchar2(50),
  birth_year number)
ORGANIZATION EXTERNAL
( type ORACLE_LOADER
  default directory home
  ACCESS PARAMETERS (
    RECORDS DELIMITED BY NEWLINE
    nologfile nobadfile
    FIELDS TERMINATED BY ';'
    optionally enclosed BY '"'
    MISSING FIELD VALUES ARE NULL ))
REJECT LIMIT UNLIMITED
```

```
SQL> SELECT * from ext_flex;
```

```
no rows selected
```

```
SQL> SELECT count(*) FROM ext_flex
      EXTERNAL MODIFY
      (LOCATION(dir_w:'cust_f.csv'));
```

```
      COUNT(*)
-----
          18325
```


18c: Inline External Tables

- Ohne vorheriges Anlegen von External Tables

```
SELECT * FROM EXTERNAL (  
    (first_name      varchar2(30),  
     last_name       varchar2(30),  
     hiredate        date,  
     department_name varchar2(30),  
     city            varchar2(30),  
     street_address  varchar2(30))  
    TYPE ORACLE_LOADER  
    DEFAULT DIRECTORY home_dir  
    ACCESS PARAMETERS (  
        RECORDS DELIMITED BY NEWLINE nobadfile nologfile  
        fields date_format date mask "dd.mm.yy")  
        LOCATION ('employees.csv') REJECT LIMIT UNLIMITED  
    ) employees_external  
where first_name='Nancy';
```

18c: In-Memory External Tables

- Voraussetzung^{**}: INMEMORY_SIZE aktivieren

```
SQL> alter system set QUERY_REWRITE_INTEGRITY=stale_tolerated;
SQL> CREATE TABLE sales_ext_inmem
      (last_name          varchar2(30),
       hiredate          date,
       department_name   varchar2(30))
      ORGANIZATION EXTERNAL
      (TYPE ORACLE_LOADER
       DEFAULT DIRECTORY home_dir
       ACCESS PARAMETERS (
        RECORDS DELIMITED BY NEWLINE nobadfile nologfile
        fields date_format date mask "dd.mm.yy")
       LOCATION ('employees.csv')) REJECT LIMIT UNLIMITED
      INMEMORY;
```

****** Lizenzierung beachten

18c: In-Memory External Tables

- POPULATE bzw. REPOPULATE erforderlich

```
SQL> EXEC DBMS_INMEMORY.POPULATE('SCOTT', 'SALES_EXT_INMEM');
```

```
SQL> SELECT * FROM dbms_xplan.display_cursor(format=>'BASIC');
```

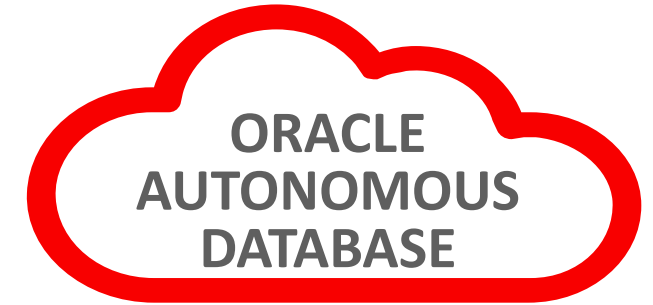
```
PLAN_TABLE_OUTPUT
```

```
-----  
EXPLAINED SQL STATEMENT:
```

```
-----  
select count(*) from sales_ext_inmem  
Plan hash value: 427401744
```

```
-----  
| Id  | Operation                                | Name          |  
-----  
|  0  | SELECT STATEMENT                          |               |  
|  1  |   SORT AGGREGATE                          |               |  
|  2  |    EXTERNAL TABLE ACCESS INMEMORY FULL | SALES_EXT_INMEM |  
-----
```

Object Storage und External Tables

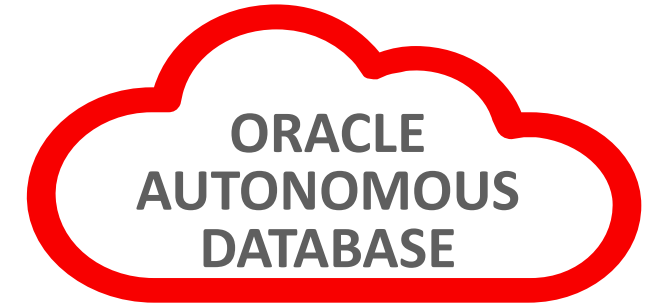


- Nur für Autonomous Data Warehouse Cloud (ADW) und Autonomous Transaction Processing (ATP)
- Vorgehen
 - Upload der Dateien in den Object Storage in einem dafür erzeugten Bucket

Objects

Upload Object	<input type="text" value="Search Objects by prefix"/>
CHANNELS_EXT.txt	...
gfk.dmp	...
sample_exdp_sh_sales.dmp	...
sample_sales.dat	...

Object Storage und External Tables



- Anlegen der Credentials und dann External Table über DBMS_CLOUD

```
BEGIN dbms_cloud.create_external_table (  
  table_name => 'COSTS_EXT' ,  
  credential_name => 'OBJ_STORE_CRED' ,  
  file_uri_list => 'https://swiftobjectstorage.de-region1.oraclecloud.com/v1/tenant_name/bucket_name/costs.dat' ,  
  format => json_object('ignoremissingcolumns' value 'true' ,  
                        'dateformat' value 'YYYY-MM-DD' ,  
                        'blankasnull' value 'true') ,  
  column_list => 'PROD_ID NUMBER ,  
                 TIME_ID DATE ,  
                 PROMO_ID NUMBER ,  
                 CHANNEL_ID NUMBER ,  
                 UNIT_COST NUMBER(10,2) ,  
                 UNIT_PRICE NUMBER(10,2) ' ) ;  
  
END ;  
/
```

Oracle Database 18c

- **Das erste jährlich** erscheinende Release der Oracle Datenbank
 - Neue Nummerierung – früher Oracle Database 12c Release 2 (12.2.0.2)
- Cloud-First Release auf Oracle Cloud und Engineered Systems
- Verfügbarkeit On-Premise: Linux, Solaris und Microsoft Windows
 - In Kürze auch Oracle Database 18c XE
- "Release Schedule of Current Database Releases" MOS Doc 742060.1
- Ziele in 18c: inkrementelle Änderungen und Fokus auf Stabilität

Ausgewählte Neuigkeiten in 18c

- Online Operationen
 - Online Merge von Partitionen und Subpartitionen
 - Online Modify: Änderung der Partition Strategie
- In-Memory Column Store
 - In-Memory External Tables (**)
 - Automatic In-Memory
- Hints
 - OPTIMIZER_IGNORE_HINTS
 - OPTIMIZER_IGNORE_PARALLEL_HINTS

** Lizenzierung beachten

Ausgewählte Neuigkeiten in 18c

- Index Zugriffe
 - MemOptimized Rowstore für schnelle Hash Index Zugriffe (**)
- Development
 - Private temporäre Tabellen
 - Transaktions- und Session-spezifische Variante
 - JSON: SQL Erweiterungen
 - Neue Funktionen wie TREAT, JSON_EQUAL, JSOXML, XMLTOJSON ...
 - Polymorphe Table Funktion
 - Implementierung über das Package Framework DBMS_TF
 - SQL*Plus Erweiterungen
 - SQL_ID, Anpassung an Bildschirmgröße

Und nicht vergessen in 12.2 ...

- Optimizer Statistics Advisor
- Index Usage Statistics
- Partitionierung (Read Only Partitionen, Mehr Online, Partitions-Wartungsoperationen mit Datenfilterung)
- Analytische Views
- Realtime Materialisierte Views
- Index mit High Compression
- In-Memory Column Store (Active Data Guard, Join Groups für Spalten, Expression Statistics Store, In-Memory JSON)
- JSON (Data Guide), SQL Funktionen mit Fehlerkontrolle, Oracle Text (Sentiment Analyse)

Wo ausprobieren?

LiveSQL.oracle.com



- <http://livesql.oracle.com>
- SQL Scratch Pad in der Oracle Cloud
- Browser und OTN Account nötig
- Kostenlos
- “Save und Share” von SQL Beispielen
- Interessante 18c Tutorials

Weitere Informationen

- Deutschsprachige Blogs
 - <https://blogs.oracle.com/coretec>
 - 18c Themenkategorie mit aktuell über 20 Beiträgen unter <https://blogs.oracle.com/coretec/datenbank-18c>
 - <http://blogs.oracle.com/cloudtec-de/>
- Oracle Technology Monthly (News und Thema des Monats)
 - <https://apex.oracle.com/pls/apex/germancommunities/events/event/2981/index.html>
 - https://www.youtube.com/playlist?list=PLHA_TOeNI7PYtnDhAy186sI-C3LjrzX
- Handbücher:
 - <https://docs.oracle.com/en/database/oracle/oracle-database/18/index.html>
- Kontakt: Ulrike.Schwinn@oracle.com @uschwinn

Integrated Cloud

Applications & Platform Services

ORACLE®