WELCOME

DBMS/Utility-Packages –

Explore the possibilities!

Axel vom Stein  **BSS Bohnenberg GmbH**

- Technical Project Manager, Head of IT (working for BSS since 2001)

-Oracle:

- since 2001
- started out as a developer (PL/SQL, C, C#)
- interest moved more towards DBA topics
- Oracle ACE

- Focus and soft spots:

- teaching trainees and young professionals
- HA - concepts
- migration projects, standardization
- everything you can automate
- spare time: diving, German Oracle User Group

Headquarters: Solingen (Germany)

Other Offices: Kattowice (Poland), Wels (Austria)

Founded: 1991

Employees: approx. 170

- Design and detailed planning of intralogistic systems and facilities
- Flowpicker® (patent for special high performance order picking)
- Warehouse Management Control System (WMCS)
- Material Flow Control (MFC)
- Automated highbays and mini-loads
- Automated Guided Vehicles (AGV's)
- Pick-by-light / pick-by-voice
- Forklift Control System (FCS)
- Visualization
- PLC – technology
- etc.

1.) Motivation

2.) Categories

3.) Summary

Discussion between (young) Developers (vastly simplified):


: PL/SQL– it‘s completly useless!


: Maybe but the DBA‘s need it for optimization!


: OK, to be more specific, it‘s useless for us!


: What in the world are the DBA‘s doing with it?


: Are you really interested?


: Yes/No, maybe it has some useful stuff for us …


: I have heard there are some predefined packages!


: OK, let‘s take a shot. Everything they can do we can do better …

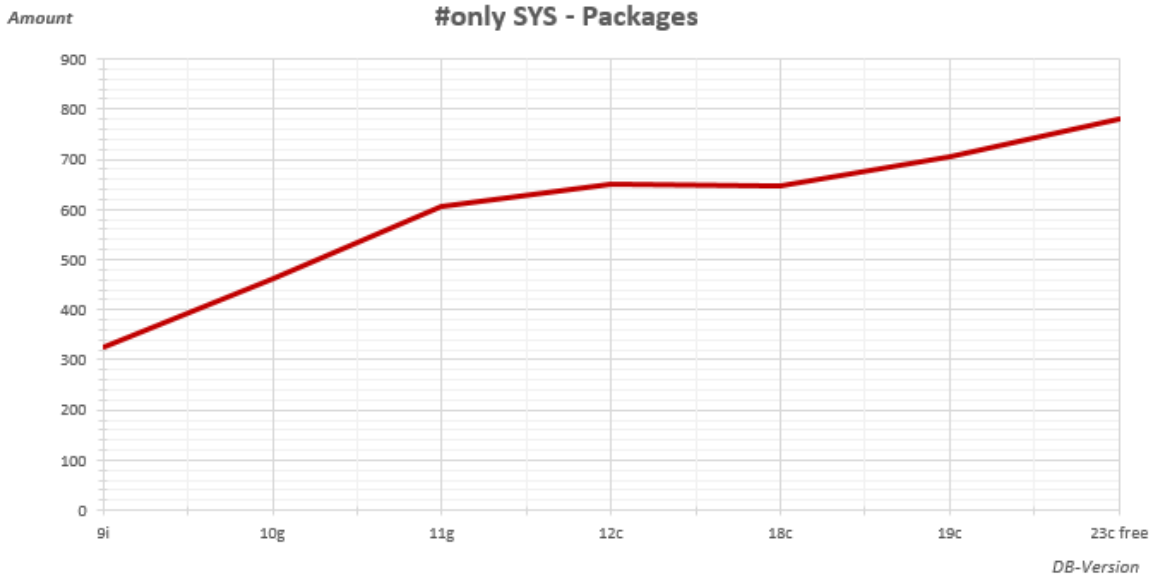
It seems there is something interesting for DBA‘s as well as for the Developers!

A presentation is a good start shedding light into the dark!

- **Many packages should be known by the DBA‘s to support Developers**
- **Some packages are interesting for automation tasks**
- **Several things are impossible without using PL/SQL (concerns DBA‘s and Developers)**
- **Often packages are named oddish and you can‘t guess what‘s inside**
- **You find routines inside packages which are also useful out of the scope**

Documentation (PL/SQL Packages and Types Reference) – Chapter 1 (depends on installation):

```
SELECT DISTINCT Owner, Object_Type, Object_Name FROM DBA_Objects
 WHERE Owner IN ('SYS', 'OUTLN', 'SYSTEM', 'CTXSYS', 'DBSNMP', 'LOGSTDBY_ADMINISTRATOR', 'ORDSYS', 'ORDPLUGINS',
                 'OEM_MONITOR', 'MDSYS', 'LBACSYS', 'DMSYS', 'WMSYS', 'OLAPDBA', 'OLAPSVR', 'OLAP_USER',
                 'OLAPSYS', 'EXFSYS', 'SYSMAN', 'MDDATA', 'SI_INFORMTN_SCHEMA', 'XDB', 'ODM')
AND Object_Type IN ('PACKAGE', 'TYPE')
   ORDER BY Owner, Object_Type, Object_Name
```



DB – Version	# only SYS – Packages
9i	325
10g	461
11g	606
12c	649
18c	648
19c	704
23c free	781

Facts:

1.) PL/SQL is powerful and comprehensive

2.) There are whole presentations and/or training days for some of the packages

3.) I have been using many of the packages but not all of them

4.) If something is missing, there is a chance that it exists (or maybe not)

5.) Technical deep dive is impossible in 45-60 minutes

So:

1.) My personal choice due to my interest and know-how

2.) Short overview of possibilities with PL/SQL (hopefully not all known…)

3.) Many roads leed to Rome – here are some with PL/SQL

1.) Motivation

2.) Categories

- General
- Overview for each category
- Small examples

3.) Summary

- **Documentation:** Any package or type that is not described in the Oracle Database Documentation Library is not supported for direct customer use.

- **Round about 270 (19c) and 306 (23c) packages are documented, so more or less 40%**

I will show featues from undocumented packages (highlighted in red)

Partialy you need the EE (plus Option) → check the Database Licensing Guide

- **APEX:**
  - **If using APEX, it‘s worth to check these packages also for little helpers**
  - **These packages can always be found in the scheme of the resp. version e.g. `APEX_230100`**
  - **Synonyms on packages, e.g. `PUBLIC.APEX_STRING` on `APEX_230100.wwv_flow_string`**

Personal choice – some packages belong not only in one category:


- Network (UTL_TCP)


- Handling of files in the broadest sense (UTL_FILE)


- Backup, Recovery and Migration (DBMS_DATAPUMP)


- Tuning and Performance (DBMS_STATS)


- Communication between processes (DBMS_AQ)


- Management, Monitoring, Logging and Debugging (DBMS_SCHEDULER)


- Security and Patch (DBMS_FGA)


- Miscellaneous and little helpers (DBMS_DB_VERSION)

- **Network**

- **Handling of files in the broadest sense**

- **Backup, Recovery and Migration**

- **Tuning and Performance**

- **Communication between processes**

- **Management, Monitoring, Logging and Debugging**

- **Security and Patch**

- **Miscellaneous and little helpers**

`UTL_TCP`

- Socket-Client communication with other systems (TCP/IP interface from the DB)
- Create an own PL/SQL FTP-Client

`UTL_HTP`

- Communication with a webserver (build your own PL/SQL REST-Client)

`UTL_SMTP, APEX_MAIL (uses UTL_SMTP), UTL_ENCODE`

- Send mails
- Encoding mail body or images

`UTL_INADDR`

- Get IP/Hostnames

`DBMS_NETWORK_ACL_ADMIN, DBMS_NETWORK_ACL_UTILITY`

- Access priviliges on the network with PL/SQL

1.) Get the Host – IP of the DB – Server:

```
UTL_INADDR.Get_Host_Address
```

2.) Get the Client – IP (if DNS is set):

```
UTL_INADDR.Get_Host_Address('win11-client')
```

3.) Get the hostname of the DB – Servers (FQDN - if DNS is set):

```
UTL_INADDR.Get_Host_Name
```

4.) Get the hostname of a client (FQDN - if DNS is set):

```
UTL_INADDR.Get_Host_Name('172.16.17.11')
```

☹ It is not possible to get all IP-adresses of a DB – server …

5.) Access on IP‘s / ports more granular:

- **Without ACL‘s (Access Control List) no network communication (since 11g):**

```
ORA-24247: network access denied by access control list (ACL)
```

- **Views:**

```
                      11g: dba_network_acls, dba_network_acl_privileges
          Beginning with 12c: dba_host_aces,    dba_host_acls
```

```
DBMS_NETWORK_ACL_ADMIN.Append_Host_Ace(
    Host       => '192.168.1.1',
    Lower_Port => 5002,
    Upper_Port => 5002,
    Ace        => xs$Ace_Type ( Privilege_List => xs$Name_List('connect'), -- smtp, http, etc.
                                Principal_Name => ,COMMUNICATION',         -- ROLE, USER
                                Principal_Type => xs_Acl.PType_DB,
                                Start_Date     => SYSTIMESTAMP,
                                End_Date       => SYSTIMESTAMP + 1,
                                Granted        => TRUE) );
```


- Network


- Handling of files in the broadest sense


- Backup, Recovery and Migration


- Tuning and Performance


- Communication between processes


- Management, Monitoring, Logging and Debugging


- Security and Patch


- Miscellaneous and little helpers

`UTL_FILE, DBMS_BACKUP_RESTORE`

- read/write files local

`DBMS_FILE_TRANSFER`

- read/write files remote, convert endianness

`UTL_COMPRESS, APEX_ZIP (uses UTL_COMPRESS)`

- zip/unzip files

`DBMS_LOB, DBMS_JSON, APEX_JSON`

- processing of LOB‘s and JSON

`DBMS_UTILITY, APEX_STRING`

- string operations

DBMS_BACKUP_RESTORE

not documented, but very powerful

- round about 260 routines, many overloaded (mostly for Backup & Restore)
- PL/SQL interface for RMAN
- Oracle internal use e.g. `$ORACLE_HOME/cfgtoollogs/dbca/orabase/rmanRestoreDatafiles.sql`
- usable also in (no)mount status
- code is inside the Oracle binaries
- stunning routines (even without relation to RMAN) …
- Delete files (oops …):

```
EXECUTE DBMS_BACKUP_RESTORE.DeleteFile('$ORACLE_BASE/diag/rdbms/<sid>/<sid>/trace/alert_ORCL.log');
```

Read folder recursively without any Oracle directory (access to it is necessary) :

```
DECLARE
  sDummy VARCHAR2(512); sPath VARCHAR2(512);
  TYPE   File_tab IS TABLE OF VARCHAR2(4000) INDEX BY BINARY_INTEGER;
  tFile  File_tab;

BEGIN
  sPath := '/home/user/list_the_files';

  DBMS_BACKUP_RESTORE.SearchFiles(
    NS      => sDummy,
    Pattern => sPath);

  SELECT t.Fname_Krbmsft BULK COLLECT INTO tFile FROM SYS.x$krbmsft t;

  FOR i IN tFile.First .. tFile.Last LOOP DBMS_OUTPUT.Put_Line(tFile(i)); END LOOP;
END;
/

OUTPUT:
/home/user/list_the_files/level1/file1.nix
/home/user/list_the_files/level2/file2.nix
```


- Network


- Handling of files in the broadest sense


- Backup, Recovery and Migration


- Tuning and Performance


- Communication between processes


- Management, Monitoring, Logging and Debugging


- Security and Patch


- Miscellaneous and little helpers

DBMS_BACKUP_RESTORE, DBMS_REPAIR, DBMS_ROWID

- find corrupt blocks of tables, indices and repair them
- information about ROWID‘s

DBMS_LOGMNR, DBMS_LOGMNR_D

- interface to Logminer (SQL on online and archivelog)

DBMS_DATAPUMP, DBMS_METADATA

- Migrations, Import/Export of tables, users, etc.
- PL/SQL API for impdp, expdp, perfect to automate tasks in combination with DBMS_SCHEDULER
- Possibilites to write messages in the logfile, monitor the process of Datapump, etc.

DBMS_TTS, DBMS_TDB

- Needed for transportable Tablespace or rather transportable DB (RMAN CONVERT cross plattform)

Remarkable:

150.1 DBMS_REPAIR Overview

The DBMS_REPAIR package is intended for use by database administrators only. It is not intended for use by application developers.

1.) Copy archivelog (e.g. from ASM to /home/oracle) with stamp in the control file:

```
Preparation: build string – e.g. LOG_ARCHIVE_FORMAT = <ORACE_SID>_%d_%T_%r_%S

ORCL_b9256de2_0002_1014301666_0000001477
```

NAME

DBID convert from Integer to Hex

THREAD# 4-digits

RESETLOGS_ID

SEQUENCE# 10-digits

source: v$database

source: v$archived_log

```
DECLARE
  nOutStamp     NUMBER;
  nOutRecId     NUMBER;
  sOutFullName  VARCHAR2(512);
BEGIN

DBMS_BACKUP_RESTORE.CopyArchivedLog
  (
   Arch_RecId => 10943,
   Arch_Stamp => 1023973230,
   FName      => '/home/oracle/ORCL_b9256de2_0002_1014301666_0000001477.alg',
   Full_Name  => sOutFullName,
   RecId      => nOutRecId,
   Stamp      => nOutStamp
  );

DBMS_OUTPUT.Put_Line('OutRecId:' || nOutRecId);
END;
/
```

→ Delete archivelog or set status (D,A,X) with stamp in CF is similar

2.) Set or delete RMAN configuration:

```
DECLARE
RECNO NUMBER;
BEGIN
  RECNO := DBMS_BACKUP_RESTORE.SETCONFIG('CONTROLFILE AUTOBACKUP','ON');
END;
/

EXECUTE DBMS_BACKUP_RESTORE.RESETCONFIG;
```

3.) Find block or object number for a ROWID:

```
DECLARE
  Object  NUMBER;
  Block   NUMBER;
  Row_ID  ROWID;
BEGIN
  SELECT ROWID INTO Row_ID FROM Schema.Table WHERE PK-Columne = 1;
  Object := DBMS_ROWID.ROWID_Object(Row_ID);
  Block  := DBMS_ROWID.ROWID_Block_Number(Row_ID);
  DBMS_OUTPUT.Put_Line('object number: ' || Object || ' block number:' || block);
END;
/
```


- Network


- Handling of files in the broadest sense


- Backup, Recovery and Migration


- Tuning and Performance


- Communication between processes


- Management, Monitoring, Logging and Debugging


- Security and Patch


- Miscellaneous and little helpers

DBMS_ADVISOR

- general advisor routines, not outsourced in specific other advisor packages
- Management of advisor tasks (view dba_advisor_definitions)
- many routines are deprecated, outsourced in other advisor packages:

```
DBMS_ADDM             - Automatic Database Diagnostic Monitor
DBMS_SQLPA            - SQL Performance Analyzer
DBMS_SQLDIAG          - SQL Repair Advisor
DBMS_SQLTUNE          - SQL Tuning Advisor
DBMS_AUTO_SQLTUNE     - SQL Tuning Advisor with Autotasks
DBMS_COMPRESSION      - Compression Advisor tables, Blob's
DBMS_SPACE            - Segment growth and storage requirement
DBMS_INMEMORY_ADVISE – Can Workload benefit from In-Memory feature (23c)
```

DBMS_PARALLEL_EXECUTE

- execute incremental updates on tables parallel (also in SE)

`DBMS_STATS`

- Round about 160 routines for management of optimizer statistics

`DBMS_XPLAN, DBMS_OUTLN, DBMS_SPM`

- Show explain plan, manipulate with …
- Outlines:
  - exists minimum since version 8i (also possible in SE)
  - deprecated (since 11g)
  - desupported in a future release → use `DBMS_SPM`
- SQL Plan Management:
  - since 11g, up to 12.2 only in EE
  - since 18c also in SE2 with some limitations (see licence guide)
  - Migration from `OUTLN` → `SPM` with `DBMS_SPM.MIGRATE_STORED_OUTLINE`

`DBMS_TRACE`

- PL/SQL - trace with order of calling, definition sub program and trace level

`DBMS_PROFILER (up to 11g), DBMS_HPROF`

- time measurement of each PL/SQL row, sub program
- Hierachical Profiler:
  - beginning with 18c tables in user session
  - SQL and PL/SQL separate time measurement

`DBMS_RESOURCE_MANAGER, DBMS_RESOURCE_MANAGER_PRIVS, DBMS_SESSION`

- Resource Manager (see link in notes)

`DBMS_CONNECTION_POOL, DBMS_RESULT_CACHE`

- Administration of Connection – Pools, SQL and PL/SQL Function Result – Cache


- Network


- Handling of files in the broadest sense


- Backup, Recovery and Migration


- Tuning and Performance


- Communication between processes


- Management, Monitoring, Logging and Debugging


- Security and Patch

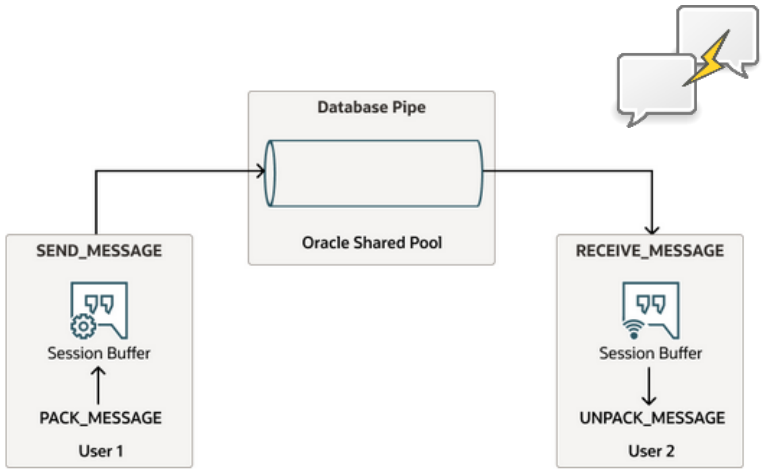

- Miscellaneous and little helpers

`DBMS_AQ, DBMS_AQADM, DBMS_AQELM`

- Advanced Queueing

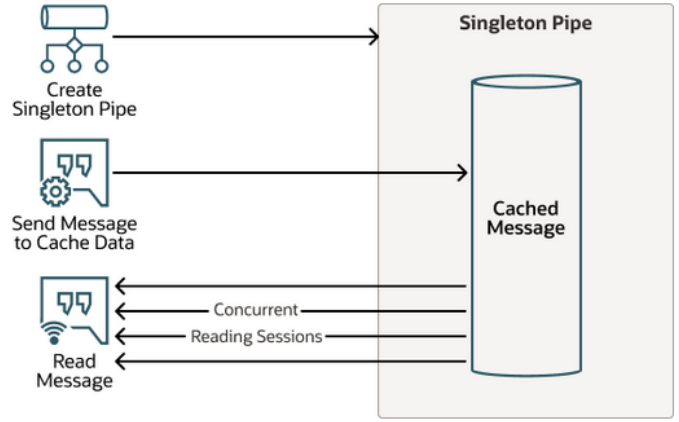
`DBMS_PIPE, DBMS_OUTPUT`

- logging or monitoring without writing to files or DB
- only in RAM (Instance Crash, DB Shutdown)
- if pipe is full (timeout or wait, must be solved programatically)
- enhancements in 23c (singleton pipes)



`DBMS_ALERT`

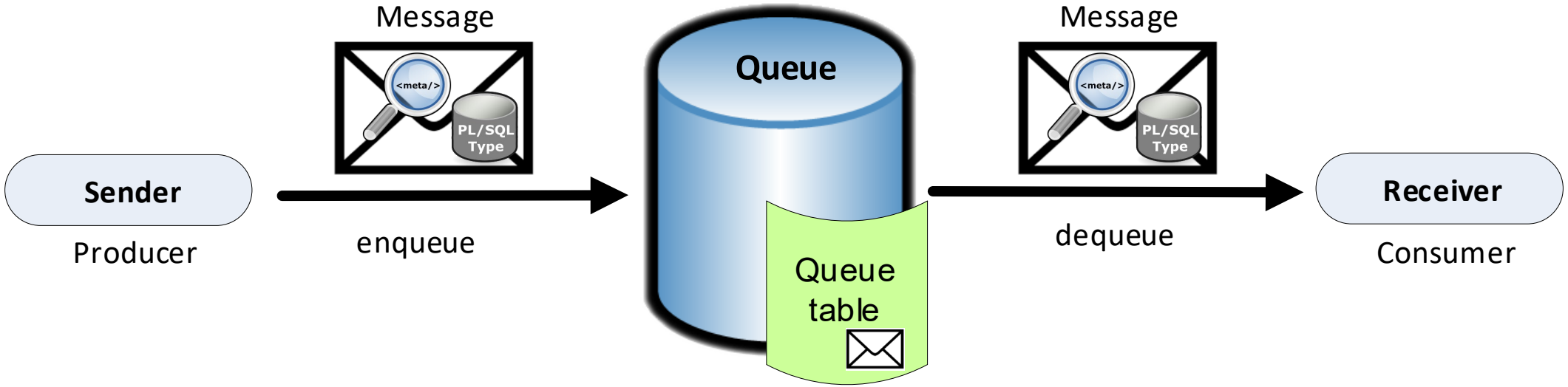
- asynchron messages with alerts (e.g. in combination with triggers)
- if alert is not catched fast enough, the next alert will be entered



`DBMS_CHANGE_NOTIFICATION`

- event based handling (also available with C# applications)

- **P2P (Point to Point) or Single Consumer Queue with one Queue:**



- **more complex: More than one Queue and/or Multi Consumer Queue**

- **Needed rights:**
  - **execute on `DBMS_AQADM` (create, drop, manage own queue)**
  - **excecute on `DBMS_AQ` (enqueue, dequeue own queues)**

Six steps to build a queue, enqueue and dequeue a message:

1.)
```
CREATE TYPE DOAG.Message_typ AS OBJECT
      (
       Subject  VARCHAR2(30),
       Text     VARCHAR2(80)
      );
```

2.)
```
DBMS_AQADM.CREATE_QUEUE_TABLE
        (
         Queue_Table        => 'DOAG.MESSAGE_QTABLE',
         Queue_Payload_Type => 'DOAG.MESSAGE_TYP'
        );
```

Select on DBA_QUEUE_TABLES:

OWNER	DOAG
QUEUE_TABLE	MESSAGE_QTABLE
OBJECT_TYPE	DOAG.MESSAGE_TYP
SORT_ORDER	ENQUEUE_TIME
RECIPIENTS	SINGLE
MESSAGE_GROUPING	NONE

3.)
```
DBMS_AQADM.CREATE_QUEUE
      (
       Queue_Name  => 'MESSAGE_QUEUE',
       Queue_Table => 'DOAG.MESSAGE_QTABLE'
      );
```

4.)
```
DBMS_AQADM.START_QUEUE
      (
       Queue_Name => 'DOAG.MESSAGE_QUEUE'
      );
```

Select on DBA_QUEUES:

OWNER	DOAG
NAME	MESSAGE_QUEUE
QUEUE_TABLE	MESSAGE_QTABLE
QUEUE_TYPE	NORMAL_QUEUE
MAX_RETRIES	5
RETRY_DELAY	0
ENQUEUE_ENABLED	YES
DEQUEUE_ENABLED	YES
RETENTION	0

5.)
```
DECLARE
   Enqueue_Options     SYS.DBMS_AQ.Enqueue_Options_t;
   Message_Properties  SYS.DBMS_AQ.Message_Properties_t;
   Message_Handle      RAW(16);
   Message             DOAG.Message_typ;

BEGIN

   Message := Message_typ ( 'First try','Easy or not?' );
   DBMS_AQ.ENQUEUE
      (
       Queue_Name          => 'DOAG.MESSAGE_QUEUE',
       Enqueue_Options     => Enqueue_Options,
       Message_properties  => Message_Properties,
       Payload             => Message,
       Msgid               => Message_Handle  -- out parameter
      );

   COMMIT;
END;
/
```

6.)
```
DECLARE
   Dequeue_Options      SYS.DBMS_AQ.Dequeue_Options_t;
   Message_Properties   SYS.DBMS_AQ.Message_Properties_t;
   Message_Handle       RAW(16);
   Message              DOAG.Message_typ;

BEGIN

   DBMS_AQ.DEQUEUE
      (
       Queue_Name          => 'DOAG.MESSAGE_QUEUE',
       Dequeue_Options     => Dequeue_Options,
       Message_Properties  => Message_Properties,
       Payload             => Message,
       Msgid               => Message_Handle
      );

   DBMS_OUTPUT.PUT_LINE ('Message → ' || Message.Subject || ': ' || Message.Text );
COMMIT;
END;
/
```

Message → First try: Easy or not?


- Network


- Handling of files in the broadest sense


- Backup, Recovery and Migration


- Tuning and Performance


- Communication between processes


- Management, Monitoring, Logging and Debugging


- Security and Patch


- Miscellaneous and little helpers

`DBMS_SCHEDULER, DBMS_CREDENTIAL, DBMS_JOBS`

- very powerful, useful for automation (complex time based plans possible)
- with use of `DBMS_CREDENTIAL` deposit access data (external Jobs, Filewatcher, etc.)
- `DBMS_JOBS` deprecated since 12.2

beginning with 19c `DBMS_JOB.SUBMIT` creates a schedule with name: `DBMS_JOB$_<Nummer>`

- Blog Mike Dietrich (`DBMS_JOB` – Behaviour Change in Oracle 19c …)

`DBMS_REDEFINITION`

- Reorganization fo tables

`DBMS_MONITOR, DBMS_SERVER_ALERT`

- DB can send alerts to other 100 metrics
- good combination with monitoring solutions like NAGIOS, ZABBIX

DBMS_APPLICATION_INFO

- Support for performance analysis, debugging und logging
- Set self-defined values (Procedures `Set_Module, Set_Client_Info, Set_Action`)
- `Client_Info, Module_Name, Action_Name` are immediately available after calling
- Columns in view `v$session`

  → possibility to locate endless loop (use a loop counter in `Set_Action`)
- Values can be read with PL/SQL (Procedures `Read_Client_Info, Read_Module`)

  → Protocol in case of errors

Long running operations:

```
DECLARE
  rIndex BINARY_INTEGER; slno BINARY_INTEGER;
BEGIN
rIndex := DBMS_APPLICATION_INFO.Set_Session_Longops_nohint; -- new record

  FOR i IN 1..10 LOOP
    DBMS_APPLICATION_INFO.Set_Session_Longops
      (
       rindex      => rIndex,      -- reference to record
       slno        => slno,        -- internal use
       op_name     => 'Example',   -- name
       target      => 20,          -- e.g. table id
       context     => i+10,        -- free for use
       sofar       => i,           -- loop counter
       totalwork   => 20,          -- loop max
       target_desc => 'Loop',
       units       => 'Units'
      );
    DBMS_SESSION.Sleep(5);
  END LOOP;

END;
/
```

```
SELECT t.Opname, t.message, t.sofar, t.totalwork
  FROM v$session_longops t
 WHERE opname = 'Example'
```

OPNAME	MESSAGE	SOFAR	TOTALWORK
Example	Example: Loop 20: 3 out of 20 Units done	3	20

 - Network

 - Handling of files in the broadest sense

 - Backup, Recovery and Migration

 - Tuning and Performance

 - Communication between processes

 - Management, Monitoring, Logging and Debugging

 - Security and Patch

 - Miscellaneous and little helpers

`DBMS_FGA`

- Policies for Fine Grained Auditing

`DBMS_ASSERT`

- Validate strings (impeding SQL – Injection)

`DBMS_CREDENTIAL, DBMS_CRYPTO, UTL_MATCH`

- Encryption for passwords and so on

`DBMS_REGISTRY, DBMS_REGISTRY_SYS, DBMS_REGISTRY_SIMPLE`

- Procedures for Upgrade, validation Catalog, etc.

`DBMS_QOPATCH`

- Information to Patch (history, current)

DBMS_PRIVILEGE_CAPTURE

- Documenation <18c: Database Vault Administrators Guide, >= 19c Security Guide
- Starting with 12.2 in EE without Database Vault (Database Licensing Guide), even in XE >= 18c
- Basic concept:
  1.) create capture
  2.) activate capture
  3.) collect used privileges of a foreign session
  4.) deactivate capture
  5.) generate report and check views (~15) e.g. DBA_PRIV_CAPTURES
  6.) remove capture
- Create_Capture with types (G_DATABASE, G_ROLE, G_CONTEXT)
- G_CONTEXT with conditions, e.g. SYS_CONTEXT('USERENV', 'SESSION_USER')='TEST'

1.) Restrict access to database links:

```
SELECT * FROM TRUSTED_SERVERS;
```

	TRUST	NAME
1	Trusted	All

```
EXECUTE DBMS_DISTRIBUTED_TRUST_ADMIN.DENY_ALL;

SELECT * FROM TRUSTED_SERVERS;
```

	TRUST	NAME
1	Untrusted	All

But: connect from PDB to another PDB in the same CDB is possible

2.) Validate scheme name (analog for object names):

```
SELECT DBMS_ASSERT.Schema_Name('ANTON') FROM dual;

 ORA-44001: invalid schema
 ORA-06512: at "SYS.DBMS_ASSERT", line 355
```

3.) How wide strings differ:

- e.g. for password – complexity, string match

```
distance = smallest amount of insert, delete, update to transfer string 1 → string 2

SELECT UTL_MATCH.Edit_Distance('TEST', 'STTE') FROM DUAL; -- returns 4
```

- next to Levenshtein also Jaro-Winkler Distance

4.) Patch information with SQL:

- e.g. for automatic deployment, select version, patch, etc.

```
SELECT xmltransform(DBMS_QOPATCH.Is_Patch_Installed('33806152'), DBMS_QOPATCH.Get_OPatch_xslt) FROM DUAL;

 Patch Information:
          33806152:   applied on 2022-04-20T23:08:50+02:00
```

 - Network

 - Handling of files in the broadest sense

 - Backup, Recovery and Migration

 - Tuning and Performance

 - Communication between processes

 - Management, Monitoring, Logging and Debugging

 - Security and Patch

 - Miscellaneous and little helpers

1.) Dissolve a view:

```
output CLOB;
BEGIN
  DBMS_UTILITY.Expand_SQL_Text(input_sql_text  => 'SELECT * FROM v$option',
                               output_sql_text => output);
  DBMS_OUTPUT.Put_Line(output);
 END;
/
```

2.) Beginning with 18c – no GRANT ON DBMS_LOCK necessery for using SLEEP:

```
DBMS_SESSION.Sleep;
```

3.) Get Serial# with PL/SQL:

```
DBMS_DEBUG_JDWP.Current_Session_Serial; -- (Java Debug Wire Protocol)
```

4.) Insert some text into alert.log or trace (1-trace, 2-alert.log, 3-both):

```
DBMS_SYSTEM.ksdwrt(2, 'My Message');
```

5.) Activate trace in foreign session:

```
EXEC DBMS_SYSTEM.Set_Sql_Trace_In_Session(SID, SERIAL#, TRUE);
```

6.) Measure the CPU time (e.g. in own PL/SQL code):

```
EXEC DBMS_UTILITY.GET_CPU_TIME
```

7.) String operations :

```
DECLARE
  sListe VARCHAR2(50); iAmount BINARY_INTEGER; tListe DBMS_UTILITY.lname_array;
BEGIN
  sListe := 'A,B,C,D';
  DBMS_UTILITY.Comma_to_Table (
     list   => sListe,
     tablen => iAmount,
     tab    => tListe);
  FOR i IN 1 .. iAmount LOOP DBMS_OUTPUT.Put_Line(i || ' : ' || tListe(i)); END LOOP;
END;
/
```

Output:

```
1 : A
2 : B
3 : C
4 : D
```

8.) Filter domain from URL:

```
SELECT APEX_STRING_UTIL.get_domain('https://docs.oracle.com/en/database/index.html') FROM DUAL;
```

Output: `docs.oracle.com`

9.) Display filesize human readable:

```
SELECT APEX_STRING_UTIL.to_display_filesize(1024*1024) FROM DUAL;
```

Output: `1M`

10.) Show initials of a name:

```
BEGIN
  sys.dbms_output.put_line(APEX_STRING.get_initials(p_str => 'Axel vom Stein', p_cnt => 3));
END;
/
```

Output: `AVS`

11.) Show time since a given date:

```
SELECT APEX_UTIL.get_since(sysdate-1) FROM DUAL;
```

Output: `24 hours ago`

12.) Dissolve tnsnames.ora string:

```
SELECT DBMS_TNS.Resolve_Tnsname('MyDatabase') FROM DUAL;
```

13.) Dissolve database links:

```
SELECT DB_LINK, DBMS_TNS.Resolve_Tnsname(Host) FROM DBA_DB_LINKS;
```

14.) Display external tables or directories:

```
DECLARE external BOOLEAN;
  BEGIN
    external := DBMS_TDB.CHECK_EXTERNAL;
  END;
/

Output:
The following external tables exist in the database: SYS.OPATCH_XML_INV
The following directories exist in the database:     SYS.XMLDIR,
                                                     SYS.XSDDIR,
                                                     SYS.ORA_DBMS_FCP_LOGDIR, …
```

15.) Format errror messages in different languages:

```
DECLARE
   s varchar2(200); i pls_integer;
BEGIN
   i:= UTL_LMS.get_message(16839, 'rdbms', 'ora', 'dutch', s);
   dbms_output.put_line('Message in dutch is: '||s);
END;
/

Output:
Message in dutch is: Pluggable database moet zich in de aangekoppelde modus bevinden.
```

16.) Insert some text into alert.log or trace:

```
DBMS_USERDIAG.TRACE('BSS: Into Alert.log', 1);

Output:
tail -3f alert_FREE.log
2023-12-03T10:24:57.650579+00:00
FREEPDB1(3):BSS: Into Alert.log
```

23c Documentation:

```
DBMS_USERDIAG.TRACE(
message IN VARCHAR2,
alert IN BOOLEAN DEFAULT FALSE);
```

23c Free:

```
DBMS_USERDIAG.TRACE
(message IN VARCHAR2,
alert IN BINARY_INTEGER DEFAULT 0);
```

1.) Motivation

2.) Categories

3.) Summary

Summary: My overview of interesting Packages

DBMS_FGA
*DBMS_RLS
DBMS_PRIVILEGE_CAPTURE
DBMS_ASSERT
DBMS_CREDENTIAL
*DBMS_REDACT
DBMS_DISTRIBUTED_TRUST_ADMIN
DBMS_CRYPTO
UTL_MATCH
DBMS_REGISTRY
DBMS_REGISTRY_SYS
DBMS_REGISTRY_SIMPLE
DBMS_QOPATCH
*DBMS_SQL_FIREWALL

DBMS_BACKUP_RESTORE
DBMS_REPAIR
DBMS_LOGMNR
DBMS_LOGMNR_D
DBMS_ROWID
DBMS_DATAPUMP
DBMS_METADATA
DBMS_TTS
DBMS_TDB
*DBMS_DG
*DBMS_FLASHBACK
*DBMS_DBCOMP
*DBMS_CLOUD
*DBMS_DICTIONARY_CHECK

UTL_TCP
DBMS_NETWORK_ACL_ADMIN
DBMS_NETWORK_ACL_UTILITY
UTL_INADDR
UTL_TCP
UTL_SMTP
APEX_MAIL
UTL_ENCODE
DBMS_CLOUD

DBMS_XPLAN
DBMS_SESSION
DBMS_SYSTEM
DBMS_STATS
DBMS_APPLICATION_INFO
DBMS_TRACE
DBMS_PROFILER
DBMS_HPROF
DBMS_RESOURCE_MANAGER
DBMS_RESOURCE_MANAGER_PRIVS
DBMS_CONNECTION_POOL
DBMS_RESULT_CACHE
DBMS_ADVISOR
DBMS_INMEMORY_ADVISE
DBMS_ADDM
DBMS_SQLPA
DBMS_SQLDIAG
DBMS_SQLTUNE
DBMS_AUTO_SQLTUNE
DBMS_COMPRESSION
DBMS_OUTLN (dep.)
DBMS_SPM
APEX_UTIL
*DBMS_SEARCH

DBMS_UTILITY
DBMS_USERDIAG
DBMS_DEBUG (dep.)
DBMS_DEBUG_JDWP
*DBMS_DB_VERSION
*DBMS_PREPROCESSOR
*DBMS_RANDOM
*DBMS_DDL
*DIUTIL
UTL_LMS
*DBMS_STAT_FUNCS
*DBMS_DST
*DBMS_SQL

DBMS_CHANGE_NOTIFICATION
DBMS_AQ
DBMS_AQADM
DBMS_AQELM
DBMS_PIPE
DBMS_OUTPUT
UTL_COMPRESS
APEX_ZIP
DBMS_ALERT

UTL_FILE
DBMS_BACKUP_RESTORE
DBMS_FILE_TRANSFER
UTL_COMPRESS
APEX_ZIP
DBMS_LOB
DBMS_JSON
*DBMS_JSON_SCHEMA
APEX_JSON
APEX_STRING

DBMS_SCHEDULER
DBMS_JOB
DBMS_CREDENTIAL
DBMS_REDEFINITION
DBMS_SESSION
DBMS_APPLICATION_INFO
DBMS_SERVER_ALERT
DBMS_SYSTEM
*DBMS_WARNINGS
*DBMS_PLSQL_CODE_COVERAGE
*UTL_CALL_STACK
*DBMS_ERRLOG
DBMS_MONITOR
DBMS_USERDIAG

Summary: Conclusion

- There are many more packages to explore (today I only showed a short overview)
- PL/SQL is important for both roles (DBA and Developers)
- features and possibilities can partly not be used to the full potential without PL/SQL
- Many roads leed to Rome, some with PL/SQL ☺
- to be continued ???

Summary: Links and Notes

- **PL/SQL Packages and Types Reference (19c):**

  https://docs.oracle.com/en/database/oracle/oracle-database/19/arpls/index.html

- **Oracle Application Express API Reference (23.1):**

  https://docs.oracle.com/en/database/oracle/application-express/23.1/aeapi/index.html

- **Presentation for AQ, Resource – Manager:**

  https://docplayer.org/115043345-Ein-fast-vergessenes-feature-oracle-advanced-qeueing.html

  https://docplayer.org/143303464-Resource-manager-der-beginn-einer-wunderbaren-freundschaft.html

- **Book:**

  Paulo Portugal – Advanced Oracle DBMS Packages (2012) (sadly only up to 11g)

axel.vomstein@bss.gmbh