

Dimitri Gielis

Oracle APEX Cheat Sheet



www.apexRnD.be



dgielis.blogspot.com



[@dgielis](https://twitter.com/dgielis)



dgielis@apexRnD.be

Dimitri Gielis

- ❖ Founder & CEO of APEX R&D
- ❖ 20+ years of Oracle Experience (OCP & APEX Certified)
- ❖ Oracle ACE Director
- ❖ “APEX Developer of the year 2009” by Oracle Magazine
- ❖ “Oracle Developer Choice award (ORDS)” in 2015
- ❖ Author Expert Oracle APEX
- ❖ Presenter at Conferences





www.apexRnD.be



www.apexofficeprint.com

http://dgielis.blogspot.com

@dgielis

Dimitri Gielis Blog (Oracle Application Express - APEX)
I created this Blog to share my knowledge especially in Oracle Application Express (APEX) and my feelings ...

FRIDAY, SEPTEMBER 26, 2014

APEX 4.2.6 patch available
Oracle released [APEX 4.2.6](#) yesterday, it's also available from My Oracle Support as a patch set 19595896.
Here's the table of bug fixes.
Table 6 Bugs Fixed in the Oracle Application Express 4.2.6.00.03 Patch Set

Bug Number	Description
18393429	Importing pre-4.2 application with data loading results in runtime error
18609856	APEX_WEB_SERVICE.CLOBBASE642BLOB converts incorrectly
18658967	Corrupted HTML mail messages are created from APEX_MAIL.SEND
18719750	ORA-01858 during application import
18733989	Errors picking up session with cookie when TOP CALL is not F, SHOW, or ACCEPT
19021052	Oracle Application Express 4.2.4 throwing error while generating check boxes
19313700	Check box fails with ORA-06502 after upgrade from release 4.0 to 4.2.4
19482775	ORA-06502 when importing application containing translations

Applying the patch took less than 7 minutes.

```
Session altered.  
...Validating Application Express  
...Database user "SYS", database schema "APEX_040200", user# "106" 20:48:40  
...Compiled 0 out of 3014 objects considered, 0 failed compilation 20:48:40  
...271 packages  
...263 package bodies  
...452 tables  
...11 functions  
...16 procedures  
...3 sequences  
...457 triggers  
...1320 indexes  
...211 views  
...0 libraries  
...6 types  
...0 type bodies  
...0 operators  
...0 index types
```

My next public appearances:

- [Oracle Open World: 27-SEP/2-OCT-2014](#)
- [ORCLAPEX-BE Meetup: 19-NOV-2014](#)
- [ODTUG KScope: 21/25-JUN-2015](#)

About Me

[Dimitri Gielis](#)

Passionate about Oracle and especially Oracle Application Express (APEX). Building applications with love @ APEX R&D.
[View my complete profile](#)

 **ORACLE**
ACE Director

Need APEX consulting, training, development, then visit:


Twitter
[Follow @dgielis](#)

A blue cartoon bird, resembling a Twitter bird, standing on a reflective surface. The bird is facing right, with its wings slightly spread and its beak open as if it's chirping. It has a yellow beak and yellow feet. The bird is standing on a light blue, reflective surface that shows a faint reflection of the bird below it.

Agenda

- ❖ Things I always have to look up in following domains:
 - ❖ Database
 - ❖ SQL
 - ❖ PL/SQL
 - ❖ APEX

Database



CREATE TABLE

```
1  create table employees (  
2      id                        number not null constraint employees_id_pk primary key,  
3      department_id            number  
4                                constraint employees_department_id_fk  
5                                references departments on delete cascade,  
6      name                     varchar2(50) not null,  
7      email                    varchar2(255),  
8      cost_center               number,  
9      date_hired                date,  
10     job                       varchar2(255),  
11     created                   date not null,  
12     created_by                varchar2(255) not null,  
13     updated                   date not null,  
14     updated_by                varchar2(255) not null  
15 );  
16  
17 create or replace trigger employees_biu  
18     before insert or update  
19     on employees  
20     for each row  
21 begin  
22     if :new.id is null then  
23         :new.id := to_number(sys_guid(), 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX');  
24     end if;  
25     if inserting then  
26         :new.created := sysdate;  
27         :new.created_by := user;  
28     end if;  
29     :new.updated := sysdate;  
30     :new.updated_by := user;  
31     :new.email := lower(:new.email);  
32 end employees_biu;  
33 /  
34
```



Quick SQL

Quick SQL

Saved Administration Help dimi ▼



AA2M.2

Save

Generate SQL

+ Worksheet



Clear

Load Samples

Syntax

Download

Settings

```
departments /insert 4
  name /nn
  location
  country
employees /insert 14
  name /nn vc50
  email /lower
  cost center num
  date hired
  job vc255

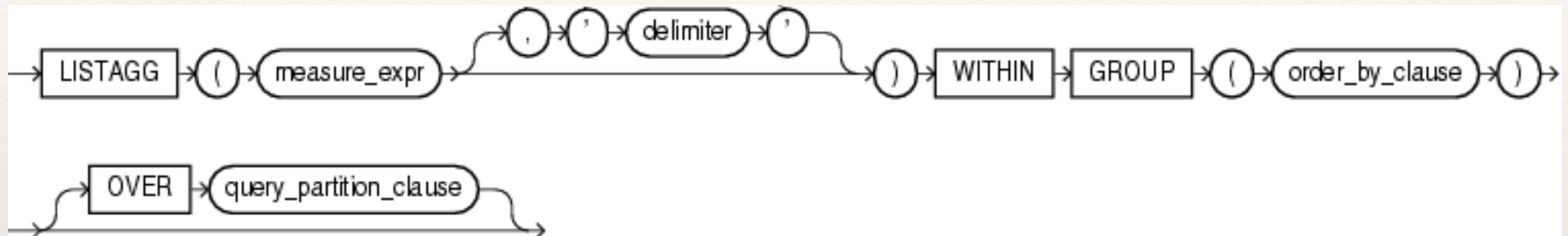
view emp_v departments employees
```

```
13
14 create table employees (
15     id                                     number not null constraint employee
16     department_id                       number
17                                         constraint employees_department_id_
18                                         references departments on delete ca
19     name                                varchar2(50) not null,
20     email                                varchar2(255),
21     cost_center                          number,
22     date_hired                           date,
23     job                                  varchar2(255),
24     created                              date not null,
25     created_by                           varchar2(255) not null,
26     updated                              date not null,
27     updated_by                           varchar2(255) not null
28 )
29 ;
30
31
32 -- triggers
33 create or replace trigger departments_biu
34     before insert or update
35     on departments
36     for each row
37 begin
38     if :new.id is null then
39         :new.id := to_number(sys_guid(), 'XXXXXXXXXXXXXXXXXXXXXXXXXXXXX
```

Demo

SQL

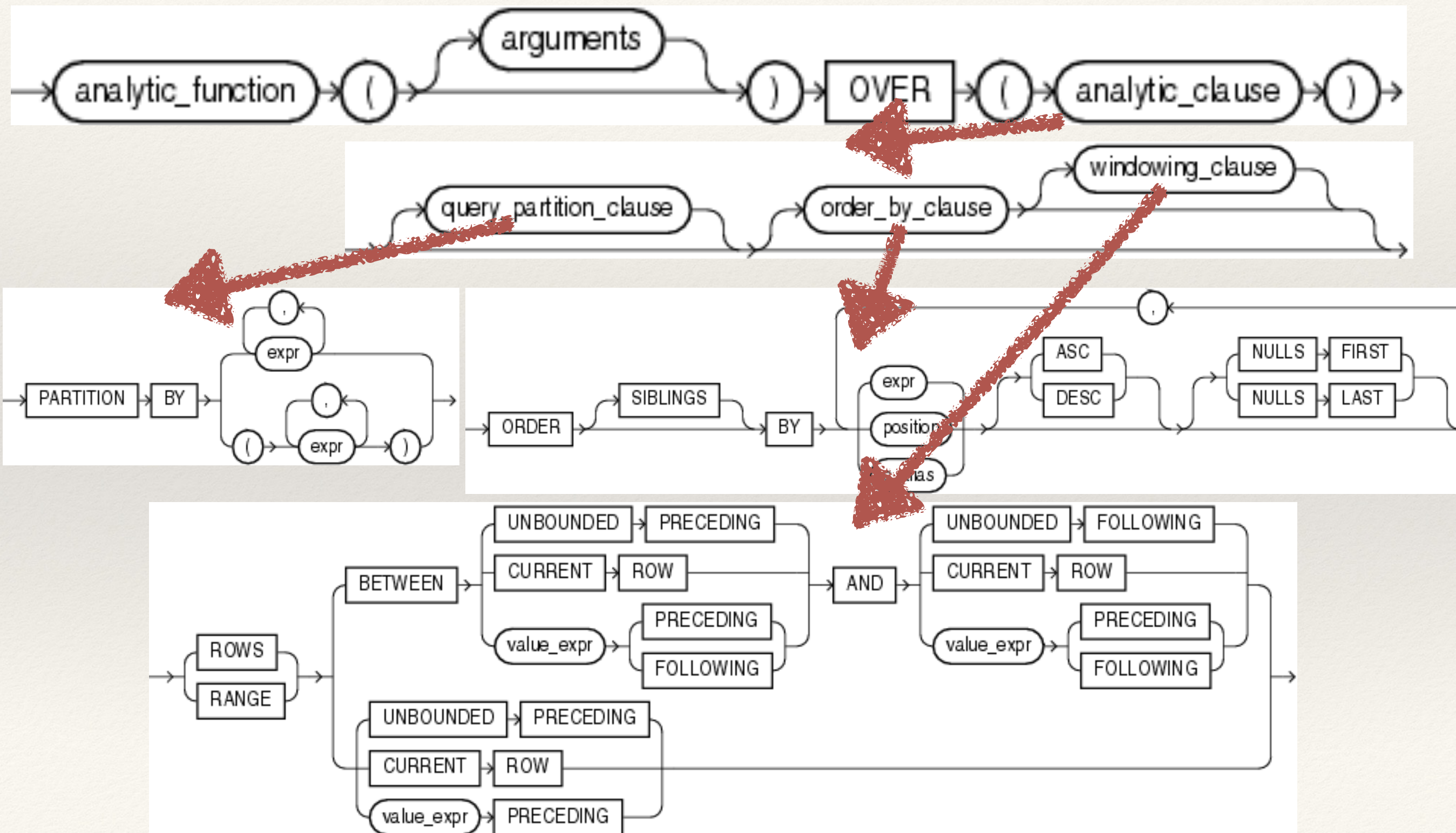
Get (comma) delimited string



```
SELECT LISTAGG(last_name, ';' )  
       WITHIN GROUP (ORDER BY hire_date, last_name) "Emp_list",  
       MIN(hire_date) "Earliest"  
FROM employees  
WHERE department_id = 30;
```

Emp_list	Earliest
Raphaely; Khoo; Tobias; Baida; Himuro; Colmenares	07-DEC-02

Analytical functions



LAG - Before

The following example provides, for each purchasing clerk in the `employees` table, the salary of the employee hired just before:

```
SELECT hire_date, last_name, salary,  
       LAG(salary, 1, 0 ) OVER (ORDER BY hire_date) AS prev_sal  
FROM employees  
WHERE job_id = 'PU_CLERK'  
ORDER BY hire_date;
```

HIRE_DATE	LAST_NAME	SALARY	PREV_SAL
18-MAY-03	Khoo	3100	0
24-JUL-05	Tobias	2800	3100
24-DEC-05	Baida	2900	2800
15-NOV-06	Himuro	2600	2900
10-AUG-07	Colmenares	2500	2600

LEAD - After

The following example provides, for each employee in Department 30 in the `employees` table, the hire date of the employee hired just after:

```
SELECT hire_date, last_name,  
       LEAD(hire_date, 1) OVER (ORDER BY hire_date) AS "Next Hired"  
FROM employees  
WHERE department_id = 30  
ORDER BY hire_date;
```

HIRE_DATE	LAST_NAME	Next Hired
07-DEC-02	Raphaely	18-MAY-03
18-MAY-03	Khoo	24-JUL-05
24-JUL-05	Tobias	24-DEC-05
24-DEC-05	Baida	15-NOV-06
15-NOV-06	Himuro	10-AUG-07
10-AUG-07	Colmenares	

ROW_NUMBER

The following example finds the three highest paid employees in each department in the `hr.employees` table. Fewer than three rows are returned for departments with fewer than three employees.

```
SELECT department_id, first_name, last_name, salary
FROM
(
  SELECT
    department_id, first_name, last_name, salary,
    ROW_NUMBER() OVER (PARTITION BY department_id ORDER BY salary desc) rn
  FROM employees
)
WHERE rn <= 3
ORDER BY department_id, salary DESC, last_name;
```



Oracle SQL Function Cheat Sheet

<http://www.databasestar.com/oracle-sql-functions>

Aggregate Functions

AVG ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
COUNT ([* | [DISTINCT | ALL] expression) [OVER (analytic_clause)]
MAX ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
MIN ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
SUM ([DISTINCT|ALL] expression) [OVER (analytic_clause)]

Conversion Functions

BIN_TO_NUM (expression_list)
CAST (expression AS type_name)
CAST (MULTISET (subquery) AS type_name)
COALESCE (expr1, expr2, [expr...])
CHARTOROWID (input_char)
FROM_TZ (timestamp_value, timezone_value)
HEXTORAW (charvalue)
NUMTODSINTERVAL (number, interval_unit)
NUMTOYMINTERVAL (number, interval_unit)
RAWTOHEX (charvalue)
RAWTONHEX (raw)
ROWIDTOCHAR (rowid)
ROWIDTONCHAR (rowid)
SCN_TO_TIMESTAMP (number)
TIMESTAMP_TO_SCN (timestamp)
TO_BINARY_DOUBLE (expression [, format [, nlsparam]])
TO_CHAR(input_value, [format_mask], [nls_parameter])
TO_CLOB (input_string)
TO_DATE(charvalue, [format_mask], [nls_date_language])
TO_DSINTERVAL (input_string [, nlsparam])
TO_LOB (long_value)
TO_MULTI_BYTE (string)
TO_NCHAR (input_string)
TO_NCHAR (input_datetime [, format [,nlsparam]])
TO_NCHAR (number [, format [, nlsparam]])
TO_NCLOB (lob_value)
TO_NUMBER (input_value, [format_mask], [nls_parameter])
TO_SINGLE_BYTE (input_string)
TO_TIMESTAMP (input_string, [format_mask], [nlsparam'])
TO_TIMESTAMP_TZ (input_string [, format_mask] [, nls_param])
TO_YMINTERVAL (input_string)
UNISTR (string)

Date and Time Functions

ADD_MONTHS (input_date, number_months)
CURRENT_DATE
CURRENT_TIMESTAMP ([precision])
DBTIMEZONE
LAST_DAY (input_date)
LOCALTIMESTAMP (timestamp_precision)
MONTHS_BETWEEN (date1, date2)
NEW_TIME (input_date, timezone1, timezone2)
NEXT_DAY (input_date, weekday)
SESSIONTIMEZONE
SYS_EXTRACT_UTC (datetime_with_timezone_value)
SYSDATE
SYSTIMESTAMP
TZ_OFFSET (timezone_name | time_value | SESSIONTIMEZONE | DBTIMEZONE)

Environment Functions

CON_DBID_TO_ID (container_dbid)
CON_GUID_TO_ID (container_guid)
CON_NAME_TO_ID (container_name)
CON_UID_TO_ID (container_uid)
ORA_INVOKING_USER
ORA_INVOKING_USERID ()
SYS_CONTEXT (namespace, parameter [, length])
SYS_GUID()
SYS_TYPEID (object_type_value)
UID
USER
USERENV (parameter)
SQLCODE
SQLERRM (error_number)

NLS Functions

NLS_CHARSET_DECL_LEN (byte_count, char_set_id)
NLS_CHARSET_ID (string_value)
NLS_CHARSET_NAME (number)

Numeric and Maths Functions

ABS (number)
ACOS (number)
ASIN (number)
ATAN2 (number1 [/], number2)
BITAND (expr1, expr2)
CEIL (input_val)
CORR (expression1, expression2)
COS (number)
COSH (number)
COVAR_POP (expression1, expression2) [OVER (analytic_clause)]
COVAR_SAMP (expression1, expression2) [OVER (analytic_clause)]
CUME_DIST (expression1, ... expression_n) WITHIN GROUP (ORDER BY expression_order1, ... expression_order_n)
CUME_DIST() OVER ([query_partition_clause] ORDER BY order_clause)
DENSE_RANK (expr, [expr(n)]) WITHIN GROUP (ORDER BY (order_expr [ASC|DESC] [NULLS FIRST|LAST]))
DENSE_RANK() OVER ([query_partition_clause] order_by_clause)
EXP (number)
EXTRACT (date_component FROM expression)
FLOOR (input_number)
GREATEST (expr1, [expr_n])
LEAST (expr1, [expr_n])
LN (number)
LOG ([base,] expression)
MEDIAN (expr) [OVER (query_partition_clause)]
MOD (numerator, denominator)
ORA_HASH (expression [, max_bucket [, seed_value]])
PERCENT_RANK (expression) WITHIN GROUP (ORDER BY (expression_n [. DESC | ASC] [NULLS FIRST|LAST]))
PERCENT_RANK () OVER ([query_partition_clause] order_by_clause)
PERCENTILE_CONT (expression) WITHIN GROUP (ORDER BY expression [ASC | DESC] [OVER (query_partition_clause)
PERCENTILE_DISC (expression) WITHIN GROUP (ORDER BY expression [ASC | DESC] [OVER (query_partition_clause)
POWER (n2, n1)
RANK (expr) WITHIN GROUP (ORDER BY (order_expr [NULLS FIRST|LAST]))
RANK () OVER ([query_partition_clause] order_by_clause)
REMAINDER (n2, n1)
ROUND (input, roundto)
ROWNUM
ROW_NUMBER () OVER ([query_partition_clause] order_by_clause)
SIGN (number)
SIN (number)
SINH (number)
SQRT (number)
STANDARD_HASH (expression [, method])
STDDEV ([DISTINCT | ALL] expression) [OVER (analytic_clause)]
STDDEV_POP (expression) [OVER (analytic_clause)]
STDDEV_SAMP (expression) [OVER (analytic_clause)]
TAN (number)
TANH (number)
TRUNC (date, fmt)
TRUNC (number, decimals)
VAR_POP (expression) [OVER (analytic_clause)]
VAR_SAMP (expression) [OVER (analytic_clause)]
VARIANCE ([DISTINCT | ALL] expression) [OVER (analytic_clause)]
WIDTH_BUCKET (expression, min_value, max_value, num_buckets)

String and Character Functions

ASCII (charvalue)
ASCIISTR (charvalue)
CHR (number_code [USING NCHAR_CS])
COMPOSE (input_value)
CONCAT(string1, string2)
CONVERT (input_char, dest_char_set, [source_char_set])
DECODE (expression, search, result [, search, result]... [,default])
DECOMPOSE (input_string [CANONICAL|COMPATIBILITY])
DUMP (expression [, return_format] [, start_position] [, length])

INITCAP (input_string)
INSTR (string, substring, [start_position], [occurrence])
INSTR2 (string, substring, [start_position], [occurrence])
INSTR4 (string, substring, [start_position], [occurrence])
INSTRB (string, substring, [start_position], [occurrence])
INSTRC (string, substring, [start_position], [occurrence])
LISTAGG (measure_expr [, delimiter]) WITHIN GROUP (order_by_clause) [OVER query_partition_clause]
LENGTH (string_value)
LENGTH2 (string_value)
LENGTH4 (string_value)
LENGTHB (string_value)
LENGTHC (string_value)
LOWER (input_string)
LPAD(expr, length [, pad_expr])
LTRIM(input_string, [trim_string])
LNNVL (condition)
NCHR (number_code)
NLS_INITCAP (input_char [, nlsparam])
NLS_LOWER (input_char [, nlsparam])
NLS_UPPER (input_char [, nlsparam])
NLSSORT (input_char [, nlsparam])
NANVL (check_value, replace_value)
NVL (check_value, replace_value)
NVL2 (value_to_check, value_if_not_null, value_if_null)
NULLIF (expr1, expr2)
REGEXP_COUNT (source_char, pattern [, position [, match_pattern [, subexpression]]])
REGEXP_INSTR (source_char, pattern [, position [, occurrence [, return_option [, match_pattern [, subexpression]]]]])
REGEXP_REPLACE (source_char, pattern [, replace_string [, position [, occurrence [, match_parameter]]]])
REGEXP_SUBSTR (source_char, pattern [, position [, occurrence [, match_parameter]]]])
REPLACE (whole_string, string_to_replace, [replacement_string])
RPAD (expr, length [, pad_expr])
RTRIM (input_string, [trim_character])
SOUNDEX (string)
SUBSTR (string, start_position, [length])
TRANSLATE (source, from_string, to_string)
TRANSLATE (charvalue USING {CHAR_CS|NCHAR_CS})
TREAT (expression AS [REF] [schema.] type)
TRIM ([[LEADING | TRAILING | BOTH] trim_character FROM] trim_source)
UPPER (input_string)
VSIZE (expression)

Analytic Functions

FIRST_VALUE (expression [IGNORE NULLS]) OVER (analytic_clause)
LAST_VALUE (expression [IGNORE NULLS]) OVER (analytic_clause)
LAG (expression [, offset [, default]]) OVER ([query_partition_clause] order_by_clause)
LEAD (expression [, offset [, default]]) OVER ([query_partition_clause] order_by_clause)
NTILE (expression) OVER ([query_partition_clause] order_by_clause)
RATIO_TO_REPORT(expression) OVER ([query_partition_clause])

Other Functions

CASE [expression]
WHEN condition_1 THEN result_1
WHEN condition_n THEN result_n
ELSE result
END case_name

SYS_CONNECT_BY_PATH (column,
character_separator)

Grouping Functions

GROUP_ID ()
GROUPING (expression)
GROUPING_ID (expression1 [, expression_n])

Large Object Functions

BFILENAME (directory, filename)
EMPTY_BLOB ()
EMPTY_CLOB ()

Oracle SQL Function Cheat Sheet

<http://www.databasestar.com/oracle-sql-functions>

Aggregate Functions

AVG ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
 COUNT ([* | [DISTINCT | ALL] expression) [OVER (analytic_clause)]
 MAX ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
 MIN ([DISTINCT|ALL] expression) [OVER (analytic_clause)]
 SUM ([DISTINCT|ALL] expression) [OVER (analytic_clause)]

Conversion Functions

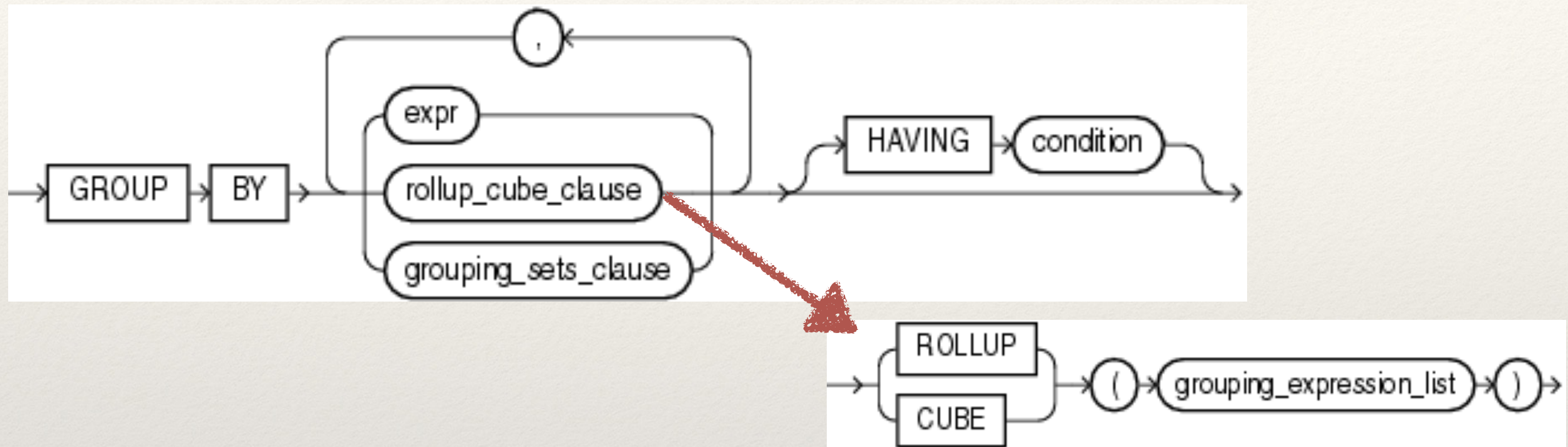
BIN_TO_NUM (expression_list)
 CAST (expression AS type_name)
 CAST (MULTISSET (subquery) AS type_name)
 COALESCE (expr1, expr2, [expr...])
 CHARTOROWID (input_char)
 FROM_TZ (timestamp_value, timezone_value)
 HEXTORAW (charvalue)
 NUMTODSINTERVAL (number, interval_unit)
 NUMTOYMINTERVAL (number, interval_unit)
 RAWTOHEX (charvalue)
 RAWTONHEX (raw)
 ROWIDTOCHAR (rowid)
 ROWIDTONCHAR (rowid)
 SCN_TO_TIMESTAMP (number)
 TIMESTAMP_TO_SCN (timestamp)
 TO_BINARY_DOUBLE (expression [, format [, nlsparam]])
 TO_CHAR (input_value, [format_mask], [nls_parameter])
 TO_CLOB (input_string)
 TO_DATE (charvalue, [format_mask], [nls_date_language])
 TO_DSINTERVAL (input_string [, nlsparam])
 TO_LOB (long_value)
 TO_MULTI_BYTE (string)
 TO_NCHAR (input_string)
 TO_NCHAR (input_datetime [, format [, nlsparam]])
 TO_NCHAR (number [, format [, nlsparam]])
 TO_NCLOB (lob_value)
 TO_NUMBER (input_value, [format_mask], [nls_parameter])
 TO_SINGLE_BYTE (input_string)

Numeric and Maths Functions

ABS (number)
 ACOS (number)
 ASIN (number)
 ATAN2 (number1 [/,] number2)
 BITAND (expr1, expr2)
 CEIL (input_val)
 CORR (expression1, expression2)
 COS (number)
 COSH (number)
 COVAR_POP (expression1, expression2) [OVER (analytic_clause)]
 COVAR_SAMP (expression1, expression2) [OVER (analytic_clause)]
 CUME_DIST (expression1, ... expression_n) WITHIN GROUP (ORDER BY expression_order1, ... expression_order_n)
 CUME_DIST() OVER ([query_partition_clause] ORDER BY order_clause)
 DENSE_RANK (expr, [expr(n)]) WITHIN GROUP (ORDER BY (order_expr [ASC|DESC] [NULLS FIRST|LAST]))
 DENSE_RANK() OVER ([query_partition_clause] order_by_clause)
 EXP (number)
 EXTRACT (date_component FROM expression)
 FLOOR (input_number)
 GREATEST (expr1, [expr_n])
 LEAST (expr1, [expr_n])
 LN (number)
 LOG ([base,] expression)
 MEDIAN (expr) [OVER (query_partition_clause)]
 MOD (numerator, denominator)
 ORA_HASH (expression [, max_bucket [, seed_value]])
 PERCENT_RANK (expression) WITHIN GROUP (ORDER BY (expression_n [. DESC | ASC] [NULLS FIRST|LAST]))
 PERCENT_RANK () OVER ([query_partition_clause] order_by_clause)
 PERCENTILE_CONT (expression) WITHIN GROUP (ORDER BY expression [ASC | DESC] [OVER (query_partition_clause)])
 PERCENTILE_DISC (expression) WITHIN GROUP (

INITCAP (input_string)
 INSTR (string, substring, [start_position], [occurrence])
 INSTR2 (string, substring, [start_position], [occurrence])
 INSTR4 (string, substring, [start_position], [occurrence])
 INSTRB (string, substring, [start_position], [occurrence])
 INSTRC (string, substring, [start_position], [occurrence])
 LISTAGG (measure_expr [, delimiter]) WITHIN GROUP (order_by_clause) [OVER query_partition_clause]
 LENGTH (string_value)
 LENGTH2 (string_value)
 LENGTH4 (string_value)
 LENGTHB (string_value)
 LENGTHC (string_value)
 LOWER (input_string)
 LPAD (expr, length [, pad_expr])
 LTRIM (input_string, [trim_string])
 LNNVL (condition)
 NCHR (number_code)
 NLS_INITCAP (input_char [, nlsparam])
 NLS_LOWER (input_char [, nlsparam])
 NLS_UPPER (input_char [, nlsparam])
 NLSSORT (input_char [, nlsparam])
 NANVL (check_value, replace_value)
 NVL (check_value, replace_value)
 NVL2 (value_to_check, value_if_not_null, value_if_null)
 NULLIF (expr1, expr2)
 REGEXP_COUNT (source_char, pattern [, position [, match_pattern [, subexpression]]])
 REGEXP_INSTR (source_char, pattern [, position [, occurrence [, return_option [, match_pattern [, subexpression]]]]])
 REGEXP_REPLACE (source_char, pattern [, replace_string [, position [, occurrence [, match_parameter]]]])
 REGEXP_SUBSTR (source_char, pattern [, position [, occurrence [, match_parameter]]])
 REPLACE (whole_string, string_to_replace, [replacement_string])
 RPAD (input_string, length [, pad_string])

Grouping set: ROLLUP



ROLLUP

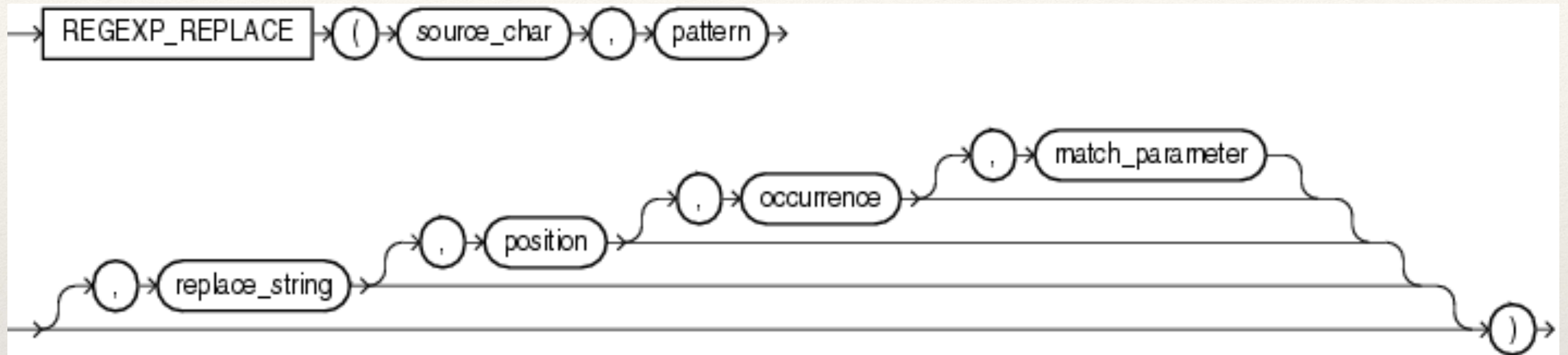
ROLLUP

In addition to the regular aggregation results we expect from the `GROUP BY` clause, the `ROLLUP` extension produces group subtotals from right to left and a grand total. If "n" is the number of columns listed in the `ROLLUP`, there will be n+1 levels of subtotals.

```
SELECT fact_1_id,  
       fact_2_id,  
       SUM(sales_value) AS sales_value  
FROM   dimension_tab  
GROUP BY ROLLUP (fact_1_id, fact_2_id)  
ORDER BY fact_1_id, fact_2_id;
```

FACT_1_ID	FACT_2_ID	SALES_VALUE
1	1	4363.55
1	2	4794.76
1	3	4718.25
1	4	5387.45
1	5	5027.34
1		24291.35
2	1	5652.84
2	2	4583.02
2	3	5555.77
2	4	5936.67
2	5	4508.74
2		26237.04
		50528.39

Regular Expressions



Get number out of string - replace non numeric by nothing

```
select regexp_replace(column_name, '[^[:digit:]]', '')  
from mytable;
```

Anchors		Sample Patterns	
^	Start of line +	([A-Za-z0-9-]+)	Letters, numbers and hyphens
\A	Start of string +	(\d{1,2}\V\d{1,2}\V\d{4})	Date (e.g. 21/3/2006)
\$	End of line +	([^\s]+(?:\.(jpg gif png))\.\2)	jpg, gif or png image
\Z	End of string +	(^[1-9]{1}\$ ^[1-4]{1}[0-9]{1}\$ ^50\$)	Any number from 1 to 50 inclusive
\b	Word boundary +	(#?([A-Fa-f0-9]){3}([A-Fa-f0-9]){3})?)	Valid hexadecimal colour code
\B	Not word boundary +	((?=\d)(?=[a-z])(?=[A-Z]).{8,15})	8 to 15 character string with at least one upper case letter, one lower case letter, and one digit (useful for passwords).
\<	Start of word	(\w+@[a-zA-Z_]+?\.[a-zA-Z]{2,6})	Email addresses
\>	End of word	(\<\/?[^\>]+\>)	HTML Tags
Character Classes		Note These patterns are intended for reference purposes and have not been extensively tested. Please use with caution and test thoroughly before use.	
\c	Control character		
\s	White space		
\S	Not white space		
\d	Digit		
\D	Not digit	Quantifiers *0 or more + *?0 or more, ungreedy + +1 or more + +?1 or more, ungreedy + ?0 or 1 + ??0 or 1, ungreedy + {3}Exactly 3 + {3,}3 or more + {3,5}3, 4 or 5 + {3,5}?3, 4 or 5, ungreedy +	
\w	Word		
\W	Not word		
\xhh	Hexadecimal character hh		
\Oxxx	Octal character xxx		
POSIX Character Classes		Special Characters \Escape Character + \nNew line + \rCarriage return + \tTab + \vVertical tab + \fForm feed + \aAlarm [\b]Backspace \eEscape \N{name}Named Character	
[:upper:]	Upper case letters		
[:lower:]	Lower case letters		
[:alpha:]	All letters		
[:alnum:]	Digits and letters		
[:digit:]	Digits	Ranges .Any character except new line (\n) + (a b)a or b + (...)Group + (?:...)Passive Group + [abc]Range (a or b or c) + [^abc]Not a or b or c + [a-q]Letter between a and q + [A-Q]Upper case letter + between A and Q + [0-7]Digit between 0 and 7 + \n nth group/subpattern +	
[:xdigit:]	Hexadecimal digits		
[:punct:]	Punctuation		
[:blank:]	Space and tab		
[:space:]	Blank characters		
[:cntrl:]	Control characters	Note Ranges are inclusive.	
[:graph:]	Printed characters		
[:print:]	Printed characters and spaces		
[:word:]	Digits, letters and underscore		
Assertions		Pattern Modifiers gGlobal match iCase-insensitive mMultiple lines sTreat string as single line xAllow comments and white space in pattern eEvaluate replacement UUngreedy pattern	
?=	Lookahead assertion +		
?!	Negative lookahead +		
?<=	Lookbehind assertion +		
?!= or ?<!	Negative lookbehind +		
?>	Once-only Subexpression	Metacharacters (must be escaped) ^[]. \${}* ()\+) ? <>	
?()	Condition [if then]		
?()	Condition [if then else]		
?#	Comment		
Note Items marked + should work in most regular expression implementations.		Available free from AddedBytes.com	

Want to learn more about Regular Expressions?

☆ **Regular Expressions: Say What?** 

Alex Nuijten

 Wednesday, 25th Apr, 15:45 (45 min)

 Düsseldorf 1 ★ 8

 *SQL & PL/SQL*

Demo

PL/SQL

Documentation

```
...
/**
 * Returns true/false if APEX developer is enable
 * Supports both APEX 4 and 5 formats
 *
 * @issue #12 Initial creation
 * @issue 23 Some major update
 * @issue 46
 *
 * @example
 * select *
 * into l_temp
 * from dual;
 *
 * @param p_app_id APEX application ID
 * @param p_page_id APEX page ID
 * @param p_session_id
 *
 * @author Martin Giffy D''Souza
 * @created 29-Dec-2015
 * @return true/false
 */
function is_developer
    return boolean
as
begin
...

```

Documentation generation

- ❖ Use snippets
- ❖ Use extension e.g. PL/SQL in Visual Studio Code

Error Package

Error Handling	
Default Error Display Location	Inline with Field and in Notification 
Error Handling Function	<code>aop_error3_pkg.apex_error_handling</code> 

Create reusable components !!

Demo

APEX

APEX Dictionary

❖ SELECT * FROM APEX_DICTIONARY

ORACLE

Application Express

App Builder

SQL Workshop

Team Development

Packaged Apps

Search Application

?

DIMI

Workspace Utilities \ Application Express Views

Report View

Tree View

Q

Go

Actions

Reset

View	Comment	Parent View
APEX_APPLICATIONS	Applications defined in the current workspace or database user.	APEX_WORKSPACES
APEX_APPLICATION_ALL_AUTH	All authorization schemes for all components by Application	APEX_APPLICATIONS
APEX_APPLICATION_AUTH	Identifies the available Authentication Schemes defined for an Application	APEX_APPLICATIONS
APEX_APPLICATION_AUTHORIZATION	Identifies Authorization Schemes which can be applied at the application, page or component level	APEX_APPLICATIONS
APEX_APPLICATION_BC_ENTRIES	Identifies Breadcrumb Entries which map to a Page and identify a pages parent	APEX_APPLICATIONS
APEX_APPLICATION_BREADCRUMBS	Identifies the definition of a collection of Breadcrumb Entries which are used to identify a page Hierarchy	APEX_APPLICATIONS
APEX_APPLICATION_BUILD_OPTIONS	Identifies Build Options available to an application	APEX_APPLICATIONS
APEX_APPLICATION_CACHING	Applications defined in the current workspace or database user.	APEX_APPLICATIONS
APEX_APPLICATION_COMPUTATIONS	Identifies Application Computations which can run for every page or on login	APEX_APPLICATIONS
APEX_APPLICATION_GROUPS	Application Groups defined per workspace. Applications can be associated with an application group.	APEX_APPLICATIONS
APEX_APPLICATION_ITEMS	Identifies Application Items used to maintain session state that are not associated with a page	APEX_APPLICATIONS
APEX_APPLICATION_LISTS	Identifies a named collection of Application List Entries which can be included on any page using a region of type List. Display attributes are controlled using a List Template.	APEX_APPLICATIONS
APEX_APPLICATION_LIST_ENTRIES	Identifies the List Entries which define a List. List Entries can be hierarchical or flat.	APEX_APPLICATION_LISTS
APEX_APPLICATION_LOCKED_PAGES	Locked pages of an application	APEX_APPLICATIONS

About

Application Express Views are data dictionary views that expose the metadata for applications. In Application Express Views, you cannot only see the view, but you can also see the data they contain.

Workspace Utilities

App Builder Defaults

Manage News

Workspace Themes

Application Groups

Application Express Views

Cross Application Reports

Export ...

Substitution Strings

- ❖ Bind variable: :ITEM
- ❖ Substitution String: &ITEM.
- ❖ PL/SQL: v('ITEM')
- ❖ Template: #ITEM#



Substitution Strings

- ❖ **HTML escapes reserved HTML characters** APEX_ESCAPE.HTML
 - ❖ &ITEM!HTML.
 - ❖ #REPORT_COLUMN!HTML#
- ❖ **ATTR escapes reserved characters in a HTML attribute**
 - ❖ #REPORT_COLUMN!ATTR# APEX_ESCAPE.HTML_ATTRIBUTE
- ❖ **removes HTML tags from the output and escapes reserved HTML characters**
 - ❖ &ITEM!STRIPHTML. www_flow_utilities.striphtml
 - ❖ #REPORT_COLUMN!STRIPHTML#



Substitution Strings

❖ **JS escapes reserved characters in a JavaScript**

❖ **&ITEM!JS.**

`APEX_ESCAPE.JS_LITERAL`

❖ **#REPORT_COLUMN!JS#**

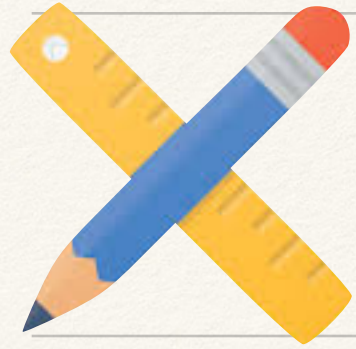
Built-in Substitution Strings

- ❖ APEX\$ROW_NUM
- ❖ APEX\$ROW_SELECTOR
- ❖ APEX\$ROW_STATUS
- ❖ APP_ID
- ❖ APP_ALIAS
- ❖ APP_AJAX_X01, ... APP_AJAX_X10
- ❖ APP_DATE_TIME_FORMAT
- ❖ APP_IMAGES
- ❖ APP-NLS_DATE_FORMAT
- ❖ APP-NLS_TIMESTAMP_FORMAT
- ❖ APP-NLS_TIMESTAMP_TZ_FORMAT
- ❖ APP_PAGE_ALIAS
- ❖ APP_PAGE_ID
- ❖ APP_REQUEST_DATA_HASH
- ❖ APP_SESSION
- ❖ APP_SESSION_VISIBLE
- ❖ APP_TITLE
- ❖ APP_UNIQUE_PAGE_ID
- ❖ APP_USER
- ❖ AUTHENTICATED_URL_PREFIX
- ❖ BROWSER_LANGUAGE
- ❖ CURRENT_PARENT_TAB_TEXT
- ❖ DEBUG
- ❖ HOME_LINK
- ❖ IMAGE_PREFIX
- ❖ JET_BASE_DIRECTORY
- ❖ JET_CSS_DIRECTORY
- ❖ JET_JS_DIRECTORY
- ❖ LOGIN_URL
- ❖ LOGOUT_URL
- ❖ PRINTER_FRIENDLY
- ❖ PROXY_SERVER
- ❖ PUBLIC_URL_PREFIX
- ❖ REQUEST
- ❖ Using REQUEST
- ❖ SCHEMA OWNER
- ❖ SQLERRM
- ❖ SYSDATE_YYYYMMDD
- ❖ THEME_DB_IMAGES
- ❖ THEME_IMAGES
- ❖ WORKSPACE_IMAGES
- ❖ WORKSPACE_ID

Environment variables

```
owa_util.print_cgi_env
```

```
owa_util.get_cgi_env('VARIABLE_NAME')
```



APEX URL

f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly

APEX URL

f?p=**APP**:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly

- Application Id: 1
- Application Id: &APP_ID.
- Application Alias: MY_APP
- Application Alias: &APP_ALIAS.

The application alias "PTRACK" can not be converted to a unique application ID.

- Application in a specific Workspace: &c=WORKSPACE

APEX URL

f?p=App:**PAGE**:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly

- Page Id: 1
- Page Id: &APP_PAGE_ID.
- Unique Page View Id: &APP_UNIQUE_PAGE_ID.
(use in request or debug - you can not use it to navigate as it's unique for the view)
- Page Alias: MY_PAGE
- Page Alias: &APP_PAGE_ALIAS.

APEX URL

f?p=App:Page:**SESSION**:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly

- Session Id: &SESSION.
- Session Id: &APP_SESSION_ID.
- Session 0: 0
- Session 0: &APP_SESSION_VISIBLE.

APEX URL

f?p=App:Page:Session:**REQUEST**:Debug:ClearCache:itemNames:itemValues:PrinterFriendly

- Application Button
- Request: APPLICATION_PROCESS=XYZ
- Interactive Report: IR[static id]_SAVED_REPORT
- Clone an APEX session: APEX_CLONE_SESSION

APEX URL

f?p=App:Page:Session:Request:**DEBUG**:ClearCache:itemNames:itemValues:PrinterFriendly

- YES
- NO
- Leveln (n between 1 and 9)
- &DEBUG.

APEX URL

f?p=App:Page:Session:Request:Debug:**CLEARCACHE**:itemNames:itemValues:PrinterFriendly

- Page Number(s): comma separated
- Collection Names
- RP: Reset Region Pagination
- APP: All Pages, All Application Items, Remove User Sort
- SESSION: All used apps in that session

APEX URL

f?p=App:Page:Session:Request:Debug:Clearcache:**ITEMNAMES:ITEMVALUES**:PrinterFriendly

- Items comma separated
- Values comma separated; not allowed to have “:”
when using comma use backslash /12,34/
- IR columns: IREQ_CUST_FIRST_NAME

APEX URL

f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:**PrinterFriendly**

- YES

- NO

APEX URL

f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly
&p_lang=en

- Choose language

APEX URL

f?p=App:Page:Session:Request:Debug:ClearCache:itemNames:itemValues:PrinterFriendly
&p_trace=YES

- Enable tracing: generates a trace file
- YES
- NO

Generate an APEX URL

❖ apex_util.prepare_url(
 'f?p=&APP_ID.:HOME:&APP_SESSION.::::P1_X:' || :P1_X
);

Generate an APEX URL

❖ apex_page.get_url(
 p_page => 1,
 p_items => 'P1_X,P1_Y',
 p_values => 'somevalue,othervalue'
);

```
FUNCTION GET_URL (  
    p_application      IN VARCHAR2 DEFAULT NULL,  
    p_page             IN VARCHAR2 DEFAULT NULL,  
    p_session          IN NUMBER     DEFAULT APEX.G_INSTANCE,  
    p_request          IN VARCHAR2 DEFAULT NULL,  
    p_debug            IN VARCHAR2 DEFAULT NULL,  
    p_clear_cache      IN VARCHAR2 DEFAULT NULL,  
    p_items            IN VARCHAR2 DEFAULT NULL,  
    p_values           IN VARCHAR2 DEFAULT NULL,  
    p_printer_friendly IN VARCHAR2 DEFAULT NULL,  
    p_trace            IN VARCHAR2 DEFAULT NULL )  
RETURN VARCHAR2;
```

PL/SQL APIs for URLs

- ❖ APEX_UTIL.URL_ENCODE
- ❖ APEX_UTIL.HOST_URL
- ❖ APEX_UTIL.REDIRECT_URL
- ❖ APEX_MAIL.GET_INSTANCE_URL
- ❖ APEX_MAIL.GET_IMAGES_URL
- ❖ API to generate URLs for use in APEX:
https://livesql.oracle.com/apex/livesql/file/content_CPSSPPZRWWADCX4KAMR6ZZB62.html

JavaScript and URLs

- ❖ `apex.navigation.dialog`
- ❖ `apex.navigation.openInNewWindow`
- ❖ `apex.navigation.popup`
- ❖ `apex.navigation.redirect`

- ❖ show inline dialog -> `openModal('dialog_auto')`

JavaScript and URLs

❖ apex.server.url ({
 p_request: "DELETE",
 x01: "test",
 pageItems: "#P1_DEPTNO,#P1_EMPNO" });

❖ apex.server.process ("MY_PROCESS", {
 x01: "test",
 pageItems: "#P1_DEPTNO,#P1_EMPNO"
}, {
 success: function(pData)
 // do something here
 }
});

=> APEX_APPLICATION.G_X01

Referencing Custom Items

- ❖ If you need to create your own items, you can access them after a page is submitted by referencing **APEX_APPLICATION.G_FXX** (01-50)

```
FOR i IN 1..3 LOOP
APEX_ITEM.TEXT(P_IDX      => 1,
  p_value      =>'array element '||i ,
  p_size       =>32,
  p_maxlength  =>32);
END LOOP;
```

```
FOR i IN 1..APEX_APPLICATION.G_F01.COUNT LOOP
  http.p('element '||I||' has a value of '||APEX_APPLICATION.G_F01(i));
END LOOP;
```

APEX_MAIL

```
DECLARE
    l_id NUMBER;
BEGIN
    l_id := APEX_MAIL.SEND(
        p_to          => 'fred@flintstone.com',
        p_from        => 'barney@rubble.com',
        p_subj        => 'APEX_MAIL with attachment',
        p_body        => 'Please review the attachment.',
        p_body_html   => '<b>Please</b> review the attachment');
    FOR c1 IN (SELECT filename, blob_content, mime_type
                FROM APEX_APPLICATION_FILES
                WHERE ID IN (123,456)) LOOP

        APEX_MAIL.ADD_ATTACHMENT(
            p_mail_id    => l_id,
            p_attachment => c1.blob_content,
            p_filename   => c1.filename,
            p_mime_type  => c1.mime_type);
    END LOOP;
    COMMIT;
END;
/
```

Download document

```
sys.http.flush;  
sys.http.init;  
owa_util.mime_header('application/octet-stream', false);  
sys.http.p('Content-length: ' || dbms_lob.getlength(l_blob));  
sys.http.p('Content-Disposition:attachment;filename="output.pdf"');  
owa_util.http_header_close;  
wpg_docload.download_file(l_blob);  
apex_application.stop_apex_engine;
```

APEX Builder Built-in Help

Universal Theme Cheat Sheet



Oracle Application Express 5

Universal Theme

Universal Theme is a responsive, versatile, and customizable user interface for your Application Express apps. It is designed uniquely for Application Express 5 to make it easy for developers to build beautiful, modern applications, at any scale, for any purpose, that work on any device.



Getting Started



Design



Components



Migration Guide



Icons



Reference

<https://apex.oracle.com/ut>

APEX Builder Keyboard Cheat Sheet

Actions

Function	Mac	PC
Save	Ctrl+Option+S	Ctrl+Alt+S
Save and Run Page	Ctrl+Option+R	Ctrl+Alt+R
Undo	Ctrl+Z	Ctrl+Z
Redo	Ctrl+Y	Ctrl+Y

Region Actions

Display From Here	Ctrl+Option+D	Ctrl+Alt+D
Display From Page	Ctrl+Option+T	Ctrl+Alt+T
Restore/Expand	Option+F11	Alt+F11
Toggle Empty Region Positions	Ctrl+Option+E	Ctrl+Alt+E

Miscellaneous

Help	Option+F1	Alt+F1
Messages	Ctrl+F1	Ctrl+F1
Page Search	Ctrl+Option+F	Ctrl+Alt+F
Keyboard Shortcuts	Option+Shift+F	Alt+Shift+F1

Regions

Function	Mac	PC
Rendering	Option+1	Alt+1
Dynamic Actions	Option+2	Alt+2
Processing	Option+3	Alt+3
Page Shared Components	Option+4	Alt+4
Grid Layout	Option+5	Alt+5
Property Editor	Option+6	Alt+6
Gallery Regions	Option+7	Alt+7
Gallery Items	Option+8	Alt+8
Gallery Buttons	Option+9	Alt+9



APEX Plug-in Cheat Sheet

Apex Plugins Cheatsheet

apex_plugin: functions

Returns the name attribute which has to be used for a HTML input element if you want that the value of the element is stored in session state when the page is submitted. If you have a HTML input element which returns multiple values (eg. select list with multiple="multiple") you have to set p_is_multi_value. Note: This function has to be called before you write something to the HTTP buffer with HTP.P(RN)

```
function get_input_name_for_page_item (
  p_is_multi_value in boolean )
  return varchar2;
```

apex_plugin: plugin

```
type t_plugin is record (
  name      varchar2(45),
  file_prefix varchar2(4000),
  attribute_01 varchar2(32767), ... );
```

apex_plugin: item

```
type t_page_item is record (
  id          number,
  name        varchar2(255),
  label       varchar2(4000),
  plain_label varchar2(4000),
  format_mask varchar2(255),
  is_required boolean,
  lov_definition varchar2(4000),
  lov_display_extra boolean,
  lov_display_null boolean,
  lov_null_text varchar2(255),
  lov_null_value varchar2(255),
  lov_cascade_parent_items varchar2(255),
  ajax_items_to_submit varchar2(255),
  ajax_optimize_refresh boolean,
  element_width number,
  element_max_length number,
  element_height number,
  element_attributes varchar2(2000),
  element_option_attributes varchar2(4000),
  escape_output boolean,
  attribute_01 varchar2(32767), ... );
```

```
type t_page_item_render_result is record (
  is_navigable boolean default false,
  navigable_dom_id varchar2(255) );
```

```
type t_page_item_validation_result is record (
  message      varchar2(32767),
  display_location varchar2(40), /* if not set the app default is used */
  page_item_name varchar2(255) ); /* if not set the validated page item
                                   name is used */
```

Render function

```
function <name of function> (
  p_item      in apex_plugin.t_page_item,
  p_plugin    in apex_plugin.t_plugin,
  p_value     in varchar2,
  p_is_readonly in boolean,
  p_is_printer_friendly in boolean )
  return apex_plugin.t_page_item_render_result;
```

Returns the AJAX identifier which has to be used for the on-demand call of a plug-in. Note: if the plug-in doesn't have an AJAX callback configured, null will be returned!

```
function get_ajax_identifier return varchar2;
```

apex_plugin: DA

```
type t_dynamic_action is record (
  id          number,
  action      varchar2(50),
  attribute_01 varchar2(32767), ... );
```

```
type t_dynamic_action_render_result is record (
  javascript_function varchar2(32767),
  ajax_identifier     varchar2(255),
  attribute_01        varchar2(32767), ... );
```

Render function

```
function <name of function> (
  p_dynamic_action in apex_plugin.t_dynamic_action,
  p_plugin         in apex_plugin.t_plugin )
  return apex_plugin.t_dynamic_action_render_result;
```

apex_plugin: region

```
type t_region is record (
  id          number,
  static_id   varchar2(255),
  name        varchar2(255),
  type        varchar2(255),
  source      varchar2(32767),
  error_message varchar2(32767),
  attribute_01 varchar2(32767), ... );
```

Render function

```
function <name of function> (
  p_region      in apex_plugin.t_region,
  p_plugin      in apex_plugin.t_plugin,
  p_is_printer_friendly in boolean )
  return apex_plugin.t_region_render_result;
```

apex_plugin: process

```
type t_process is record (
  id          number,
  name        varchar2(255),
  success_message varchar2(32767),
  attribute_01 varchar2(32767), ... );
```

```
type t_process_exec_result is record (
  success_message varchar2(32767) );
```

Execute function

```
function <name of function> (
  p_process in apex_plugin.t_process,
  p_plugin  in apex_plugin.t_plugin )
  return apex_plugin.t_process_exec_result;
```

apex_plugin_util

```
function escape (
  p_value in varchar2,
  p_escape in boolean ) return varchar2;
```

Note: if both values are null, TRUE will be returned as well.

```
function is_equal (
  p_value1 in varchar2,
  p_value2 in varchar2 ) return boolean;
```

Returns the position in the list where p_value is stored. If it's not found null will be returned.

```
function get_position_in_list (
  p_list in wwv_flow_global.vc_arr2,
  p_value in varchar2 ) return number;
```

Outputs the value in an escaped form. It's also taking care of chunking big strings into smaller outputs.

```
procedure print_escaped_value (
  p_value in varchar2 );
```

Outputs a hidden field to store the page item value if the page item is rendered readonly but not printer friendly

```
procedure print_hidden_if_readonly (
  p_item_name in varchar2,
  p_value     in varchar2,
  p_is_readonly in boolean,
  p_is_printer_friendly in boolean,
  p_id_postfix in varchar2 default null );
```

Outputs a hidden protected field to store the page item value.

```
procedure print_hidden_protected (
  p_item_name in varchar2,
  p_value     in varchar2,
  p_id_postfix in varchar2 default null );
```

Outputs a SPAN tag for a display only field.

```
procedure print_display_only (
  p_item_name in varchar2,
  p_display_value in varchar2,
  p_show_line_breaks in boolean,
  p_escape     in boolean,
  p_attributes in varchar2,
  p_id_postfix in varchar2 default '_DISPLAY' );
```

Outputs an OPTION tag. The values are always escaped.

```
procedure print_option (
  p_display_value in varchar2,
  p_return_value in varchar2,
  p_is_selected in boolean,
  p_attributes in varchar2,
  p_escape     in boolean default true );
```

Returns a Javascript array with all the page items which are really on the current and on page 0. p_bind_variable_list are the bind variables returned by wwv_flow_utilities.get_binds

```
function get_dependent_fields (
  p_bind_variable_list in wwv_flow_global.vc_arr2 )
  return varchar2;
```

Outputs a JSON response based on the result of a two column LOV in the format:

```
[{"d":"display","r":"return"}, {"d":"...", "r":"..."}]
```

p_component_name is the name of the page item or report column, ... which is used in case an error has to be displayed.

Note: The HTTP header is initialized with mime type "application/json"

```
procedure print_lov_as_json (
  p_sql_statement in varchar2,
  p_component_name in varchar2,
  p_escape        in boolean,
  p_support_legacy_lov in boolean default false /*internal use only! */);
```

* For a complete reference check the apex_plugin and apex_plugin_util packages

apex_plugin_util

Executes the query and returns the values for each column.

Note: All column values are returned as a string

p_sql_statement: SQL statement which should be executed.

p_min_columns and *p_max_columns*: Used to check if the SQL statement complies to the number of required columns.

p_component_name: The name of the page item or report column which is used in case an error has to be displayed.

p_search_type: Use one of the *c_search_** constants.

p_search_column_no: Column which should be used to restrict the SQL statement. Has to be in range of *p_min_column* and *p_max_column*.

p_search_string: Value which should be used to restrict the query.

p_first_row: Skips all rows before the specified row. Can be used for pagination.

p_max_rows: Limits the number of returned rows.

```
function get_data (
    p_sql_statement    in varchar2,
    p_min_columns      in number,
    p_max_columns      in number,
    p_component_name    in varchar2,
    p_search_type       in varchar2 default null,
    p_search_column_no  in varchar2 default 2,
    p_search_string     in varchar2 default null,
    p_first_row         in number  default null,
    p_max_rows          in number  default null,
    p_support_legacy_lov in boolean  default false /* internal use only! */)
return t_column_value_list;
```

Advanced version of *get_data*, returns the column values with original data types.

p_data_type_list: If provided checks that the data type for each column matches to the specified data type in the array. Use the constants *c_data_type_** for available data types.

```
function get_data2 (
    p_sql_statement    in varchar2,
    p_min_columns      in number,
    p_max_columns      in number,
    p_data_type_list    in wwv_flow_global.vc_arr2 default c_empty_data_type_list,
    p_component_name    in varchar2,
    p_search_type       in varchar2 default null,
    p_search_column_no  in varchar2 default 2,
    p_search_string     in varchar2 default null,
    p_first_row         in number  default null,
    p_max_rows          in number  default null)
return t_column_value_list2;
```

Gets the display value for the value specified in *p_search_string*.

p_component_name: The name of the page item or report column, ... which is used in case an error has to be displayed.

p_display_column_no: Column which should be returned from the SQL statement.

Has to be in range of *p_min_column* and *p_max_column*.

p_search_column_no: Column which should be used to restrict the SQL statement.

Has to be in range of *p_min_column* and *p_max_column*.

p_search_string: Value which should be looked up.

p_display_extra: If set to TRUE and a value isn't found, the search value will be added to the result instead.

Note: In the case multiple rows are returned by the query, just the first one is returned.

```
function get_display_data (
    p_sql_statement    in varchar2,
    p_min_columns      in number,
    p_max_columns      in number,
    p_component_name    in varchar2,
    p_display_column_no in binary_integer default 1,
    p_search_column_no  in binary_integer default 2,
    p_search_string     in varchar2,
    p_display_extra      in boolean  default true,
    p_support_legacy_lov in boolean  default false /* for internal use only! */)
return varchar2;
```

Gets the display values for the values specified in *p_search_value_list*.

```
function get_display_data (
    p_sql_statement    in varchar2,
    p_min_columns      in number,
    p_max_columns      in number,
    p_component_name    in varchar2,
    p_display_column_no in binary_integer default 1,
    p_search_column_no  in binary_integer default 2,
    p_search_value_list in wwv_flow_global.vc_arr2,
    p_display_extra      in boolean  default true,
    p_support_legacy_lov in boolean  default false /* for internal use only! */)
return wwv_flow_global.vc_arr2;
```

Returns a jQuery selector based on a comma delimited string of page item names.

```
function page_item_names_to_jquery (
    p_page_item_names in varchar2 )
return varchar2;
```

Returns a cleaned up version of a comma delimited string of page item names

where all spaces are removed, colons are replaced with a comma and the page item names are converted to uppercase. This is important for JavaScript code, because page item names are case sensitive in the browser.

```
function cleanup_page_item_names (
    p_page_item_names in varchar2 )
return varchar2;
```

Writes the data of the page item meta data into the HTTP buffer.

```
procedure debug_page_item (
    p_plugin  in wwv_flow_plugin.t_plugin,
    p_page_item in wwv_flow_plugin.t_page_item );
procedure debug_page_item (
    p_plugin      in wwv_flow_plugin.t_plugin,
    p_page_item   in wwv_flow_plugin.t_page_item,
    p_value        in varchar2,
    p_is_readonly  in boolean,
    p_is_printer_friendly in boolean );
```

Writes the data of the region meta data into the HTTP buffer.

```
procedure debug_region (
    p_plugin in wwv_flow_plugin.t_plugin,
    p_region in wwv_flow_plugin.t_region );
procedure debug_region (
    p_plugin      in wwv_flow_plugin.t_plugin,
    p_region      in wwv_flow_plugin.t_region,
    p_is_printer_friendly in boolean );
```

Writes the data of the dynamic action meta data into the HTTP buffer.

```
procedure debug_dynamic_action (
    p_plugin      in wwv_flow_plugin.t_plugin,
    p_dynamic_action in wwv_flow_plugin.t_dynamic_action );
```

Writes the data of the process meta data into the HTTP buffer.

```
procedure debug_process (
    p_plugin in wwv_flow_plugin.t_plugin,
    p_process in wwv_flow_plugin.t_process );
```

apex_javascript

Adds the script tag to load a javascript library.

p_name: has to be specified without .js

p_directory: has to have a trailing slash

p_version: version identifier which should be added to the library name

p_skip_extension: if true the extension .js is NOT added

```
procedure add_library(
    p_name      in varchar2,
    p_directory  in varchar2 default wwv_flow.g_image_prefix||'javascript/',
    p_version    in varchar2 default c_apex_version,
    p_skip_extension in boolean  default false );
```

Adds a javascript code snippets to the HTML output which is executed at the onload event.

If an entry with the same key exists it will be ignored.

If *p_key* is null the snippet will always be added.

p_code: javascript code snippet. eg: \$s('P1_TEST', 'abc');

p_key: name of the key. eg: APEX_WIDGET

```
procedure add_onload_code(
    p_code in varchar2,
    p_key  in varchar2 default null);
```

Adds a code snippet which is included inline into the HTML output

eg. You can use this procedure to add new functions or global variable declarations.

Note: if you want to execute code, you should use *add_onload_code*

If an entry with the same key exists it will be ignored.

If *p_key* is null the snippet will always be added.

p_code: javascript code snippet. eg: \$s('P1_TEST', 123);

p_key: identifier for the code snippet. If specified and a code snippet with the same name has already been added the new code snippet will be ignored.

```
procedure add_inline_code(
    p_code in varchar2,
    p_key  in varchar2 default null);
```

Escapes a text so that it can be used in JavaScript without problems.

Replaces \ with \\, / with \/, " with \u0022, ' with \u0027, tab with \t, chr(10) with \n.

```
function escape (
    p_text in varchar2 ) return varchar2;
```

apex_css

Adds the style tag to load a css library.

p_name: has to be specified without .js

p_directory: has to have a trailing slash

p_version: version identifier which should be added to the library name

p_skip_extension: if true the extension .css is NOT added

```
procedure add_file (
    p_name      in varchar2,
    p_directory  in varchar2 default wwv_flow.g_image_prefix||'css/',
    p_version    in varchar2 default c_apex_version,
    p_skip_extension in boolean  default false );
```

Adds a CSS style snippet which is included inline into the HTML output

eg. You can use this procedure to add new css style declarations.

If an entry with the same key exists it will be ignored.

If *p_key* is null the snippet will always be added.

p_style: CSS style snippet. eg: #test { color:#fff }

p_key: identifier for the style snippet. If specified and a style snippet with the same name has already been added the new style snippet will be ignored.

```
procedure add (
    p_css in varchar2,
    p_key in varchar2 default null );
```

Demo

Recap

- ❖ Things I always have to look up in following domains:
 - ❖ Database
 - ❖ SQL
 - ❖ PL/SQL
 - ❖ APEX

Q&A

APEX RD
research & development



www.apexRnD.be



dgielis.blogspot.com



@dgielis



dgielis@apexRnD.be

Consulting, Development, Training

- ❖ Looking for consulting, training and development in Oracle Application Express (APEX)?
- ❖ Contact : www.apexRnD.be
- ❖ Mail : info@apexRnD.be