



This work is licensed under a [Creative Commons Attribution-NonCommercial-ShareAlike 2.5 License](http://creativecommons.org/licenses/by-nc-sa/2.5/).

## Oracle PL/SQL Cheatsheet

Symbols		Control Flow	
; Semicolon.	Statement terminator	IF..THEN..ELSE..ENDIF; As usual.	
% Percent sign	Attribute indicator (cursor attributes %ISOPEN and indirect declaration attributes like %ROWTYPE). Also use multibyte wildcard symbol, as in SQL	LOOP .. IF (condition) THEN EXIT END IF; .. END LOOP;	Equivalent to if (condition) then
_ Single underscore	Single-byte wildcard symbol, as in SQL	WHILE cond LOOP..END LOOP;	while () {};
: Colon	Host variable indicator, such as :block in Oracle Forms	FOR var IN n..m LOOP .. END LOOP;	for thing in range(n,m) {}
** Double asterisk	Exponentiation operator	EXECUTE function_name;	Function call FUNCTION name (parameter type,..) ..body.. END
< > and !=	Not equals"		Opens cursor, loops across until %NOTFOUND.
Double vertical bar	Concatenation operator	Cursor for.	FOR variables IN cursor LOOP..E LOOP;
<< and >>	Label delimiters		
: =	Assignment operator		
=>	Association operator for positional notation		
--	Double dash: single-line comment indicator	Implicit cursor named by developer.	Think of it as a select statement then name.
/* and */	Beginning and ending multiline comment block delimiters	Implicit cursor is called SQL	IF SQL%NOTFOUND THEN ..
<b>Data Types.</b>			
Database types	Definition	Declaring an explicit cursor.	DECLARE CURSOR employee_csr SELECT empid, salary FROM emplo BEGIN .. OPEN employee_cursor LOOP FETCH employee_cursor INTO my_ my_salary; EXIT WHEN employee_csr%NOTFO ..do stuff.. ENDLOOP;
NUMBER	Used to store any number		Obtains next record from cursor. Ca into individual variables (as above) RECORD.
CHAR(N), VARCHAR2(N)	Used for storing text	Executing a cursor	TYPE t_emp IS RECORD (T_Salary t_empid number); my_emprec t_emp; CURSOR employee_csr IS SELECT empid, salary FROM employee;
DATE	Oracle system date		OPEN employee_csr; LOOP
LONG	Stores large blocks of text		FETCH employee_csr INTO my_emp EXIT WHEN employee_csr%NOTFO IF my_emprec.t_empid ..
LONG RAW	Stores large blocks of binary data		Declaring parameters to be used at time. DECLARE .. CURSOR employee_csr (low_end VARCHAR2 high_end VARCHAR2) IS SELECT empid, salary FROM emplo
ROWID	Smaller binary data store		
MLSLABEL	Used for row identifier		
	Security label		
Non database types.	DEC, DECIMAL, REAL, DOUBLE- PRECISION, INTEGER, INT, SMALLIN NATURAL, POSITIVE, NUMERIC, BIN/ INTEGER, CHARACTER, VARCHAR, BOOLEAN, TABLE, RECORD	FETCH	
<b>PLSQL Module types</b>			
Procedure	A non-formal function that can accept parameters via value or reference. Similar to a function.	Declaring an explicit cursor using a record.	
Function	A classical function that returns one value. Usually contains declaration, execution and exception sections.	Executing explicit cursor using record.	
Package	A library, consisting of a specification function/prototype signatures, and a body with actual code. eg		
Trigger	Code attached to a table that fires on certain conditions.		
<b>Module Sections or Blocks</b>			
Variable Declaration	DECLARE employee-id employee.empid%TYPE,	Cursor Parameters.	

	CONSTANT number := 3.14, ratio RE BEGIN..		WHERE substr(lastname,1,1) BETW UPPER(low_end) AND UPPER(high_
Executable Section	.. BEGIN select * into my_employee where employee.emid = 42; END; ..	INVALID_CURSOR	Occurs when you attempt to c cursor that has not been open
Exception Handler.	END; EXCEPTIONS .. END;	CURSOR_ALREADY_OPEN	Occurs when you attempt to c cursor the second time
	<b>Package Syntax</b>	DUP_VAL_ON_INDEX	Unique or primary key constr violation
Specification	PACKAGE package_name IS [ declarations of variables and types [ specifications of cursors ] [ specifications of modules ] END [ package_name ]; PACKAGE BODY package_name IS [ declarations of variables and types [ specification and SELECT statement cursors ]	TOO_MANY_ROWS	More than one row was opbtai a single row subquery, or ano context when Oracle was expe one row.
Body	[ specification and body of modules ] [ BEGIN executable statements ] [ EXCEPTION exception handlers ] END [ package_name ];	ZERO_DIVIDE	An attempt to divide by zero.
		ROWTYPE_MISMATCH	An attempt to FETCH a cursor incompatible variable type.
		INVALID_NUMBER	An char type was referenced a number.
		OTHERS	Special catchall exception.
			<b>Pragmas</b>
		EXCEPTION_INIT	Tells the compiler to associate a particular error number with ar identifier you have declared as exception in your program.
		RESTRICT_REFERENCES	Tells the compiler the purity lev (freedom from side effects) of a packaged program.
General SQL*Plus script	.sql	SERIALLY_REUSABLE	Tells the PL/SQL runtime engin package-level data should not p between references to that dat Chapter 25, Tuning PL/SQL Applications for more informati
Testing script	.tst		
Stored procedure	.sp		
Stored function	.sf		
Stored package body	spb		
Stored package specification	.sps		

### Filename Extensions

### Implicit cursor attributes.

%NOTFOUND	True if fetch did not return row.
%ROWCOUNT	Number of rows processed by this cursor
%FOUND	Opposite of %NOTFOUND
%ISOPEN	If currently open for processing then true.

### Transaction processing

Same Options as SQL	COMMIT, ROLLBACK, SAVEPOINT
Transaction begins at execution of first change of data.	Rollbacks go to last COMMIT or SAVE_POINT
DBMS_TRANSACTION	A package with functions for transaction control.

### Exception Handling

Predefined	Relates to an oracle error. No need to invoke. Just catch. EXCEPTION WHEN NO_DATA_FOUN THEN DBMS_OUTPUT.PUT_LINE('No data found');
User defined.	Need to be declared, tested and handled in their respective blocks. DECLARE My_salary_null EXCEPTION; .. EBGIN.. IF my_emp_record.salary IS NULL THEN RAISE my_salary_null; END IF; EXCEPTION.. WHEN my_salary_null THEN DBMS_OUTPUT.PUT_LINE('Salary column was null for employee'); END

Pragmas. Associate a predefined error with a  
exception handler. eg to have  
my\_salary\_null catch Oracle error -1400  
DECLARE  
PRAGMA EXCEPTION INIT(my\_salary\_null,  
-1400);