Oracle PL/SQL Cheatsheet

Symbols

; Semicolon.	Statement terminator
% Percent sign	Attribute indicator (cursor attributes %ISOPEN and indirect declaration attributes like %ROWTYPE). Also use multibyte wildcard symbol, as in SQL
_ Single underscore	Single-byte wildcard symbol, as in SQL
: Colon	Host variable indicator, such as :bloc in Oracle Forms
** Double asterisk	Exponentiation operator
< > and !=	Not equals"
|| Double vertical bar	Concatenation operator
<< and >>	Label delimiters
:=	Assignment operator
=>	Association operator for positional no
--	Double dash: single-line comment indicator
/* and */	Beginning and ending multiline comm block delimiters

Data Types.

Database types	Definition
NUMBER	Used to store any number
CHAR(N), VARCHAR2(N)	Used for storing text
	Oracle system date
DATE	Stores large blocks of text
LONG	Stores large blocks of binary data
LONG RAW	Smaller binary data store
ROWID	Uesd for row identifier
MLSLABEL	Security label
Non database types.	DEC, DECIMAL, REAL, DOUBLE-PRECISION, INTEGER, INT, SMALLIN' NATURAL, POSITIVE, NUMERIC, BINA INTEGER, CHARACTER, VARCHAR, BOOLEAN, TABLE, RECORD

PLSQL Module types

Procedure	A non-formal function that can accep paremeters via value or reference. Si in form to a function.
Function	A classical function that returns one Usually contains declaration, executic and exception sections.
Package	A library, consisting of a specification function/prototype signatures, and a with actual code. eg
Trigger	Code attached to a table that fires on certian conditions.

Module Sections or Blocks

Variable Declaration	DECLARE employee-id employee.empid%TYPE,

Control Flow

IF..THEN..ELSE..ENDIF;	As usual.
LOOP .. IF (condition) THEN EXIT END IF; .. END LOOP:	Equivalent to if (conition) then t
WHILE cond LOOP..END LOOP;	while () {};
FOR var IN n..m LOOP .. END LOOP;	for thing in range(n,m) {}
EXECUTE function_name;	Function call FUNCTION name (parameter type,..) ..body.. END
Cursor for.	Opens cursor, loops across until %NOTFOUND. FOR variables IN cursor LOOP..E LOOP;

Explicit Cursor Handling

Implict cursor named by developer.	Think of it as a select statement th name.
Implict cusror is called SQL	IF SQL%NOTFOUND THEN ..
Declaring an explicit cursor.	DECLARE CURSOR employee_crsr SELECT empid, salary FROM emplo BEGIN ..
Executing a cursor	OPEN employee_cursor LOOP FETCH employee_cursor INTO my_ my_salary; EXIT WHEN employee_crsr%NOTF(..do stuff.. ENDLOOP;
FETCH	Obtains next record from cursor.Ca into individual variables (as above) RECORD.
Declaring an explicit cursor using a record.	TYPE t_emp IS RECORD (T_Salary t_empid number); my_emprec t_emp; CURSOR employee_crsr IS SELECT empid, salary FROM employee;
Executing explicit cursror using record.	OPEN employee_cursr; LOOP FETCH emloyee_crsr INTO my_emp EXIT WHEN employee_crsr%NOTF(IF my_emprec.t_empid ..
Cursor Parameters.	Declaring parameters to be used at time. DECLARE .. CURSOR employee_crsr(low_end VARCHAR2 high_end VARCHAR2) IS SELECT empid, salary FROM emplo

	CONSTANT number : = 3.14, ratio RE BEGIN..
Executable Section	.. BEGIN select * into my_employee where employee.emid = 42; END; ..
Exception Handler.	END; EXCEPTIONS .. END;

Package Syntax

Specification	PACKAGE package_name IS [declarations of variables and types [specifications of cursors] [specifications of modules] END [package_name];
Body	PACKAGE BODY package_name IS [declarations of variables and types [specification and SELECT statement cursors] [specification and body of modules] [BEGIN executable statements] [EXCEPTION exception handlers] END [package_name];

Filename Extensions

General SQL*Plus script	.sql
Testing script	.tst
Stored procedure	.sp
Stored function	.sf
Stored package body	spb
Stored package specification	.sps

Implict 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

WHERE substr(lastname,1,1) BETW
UPPER(low_end) AND UPPER(high_

Common exceptions

INVALID_CURSOR	Occurs when you attempt to c cursor that has not been open
CURSOR_ALREADY_OPEN	Occurs when you attempt to o cursor the second time
DUP_VAL_ON_INDEX	Unique or primary key constra violation
TOO_MANY_ROWS	More than one row was opbtai a single row subquery, or ano context when Oracle was expe one row.
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.

Pragmas

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 le (freedom from side effects) of a packaged program.
SERIALLY_REUSABLE	Tells the PL/SQL runtime engin package-level data should not [between references to that data Chapter 25, Tuning PL/SQL Applications for more informati

| 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); |