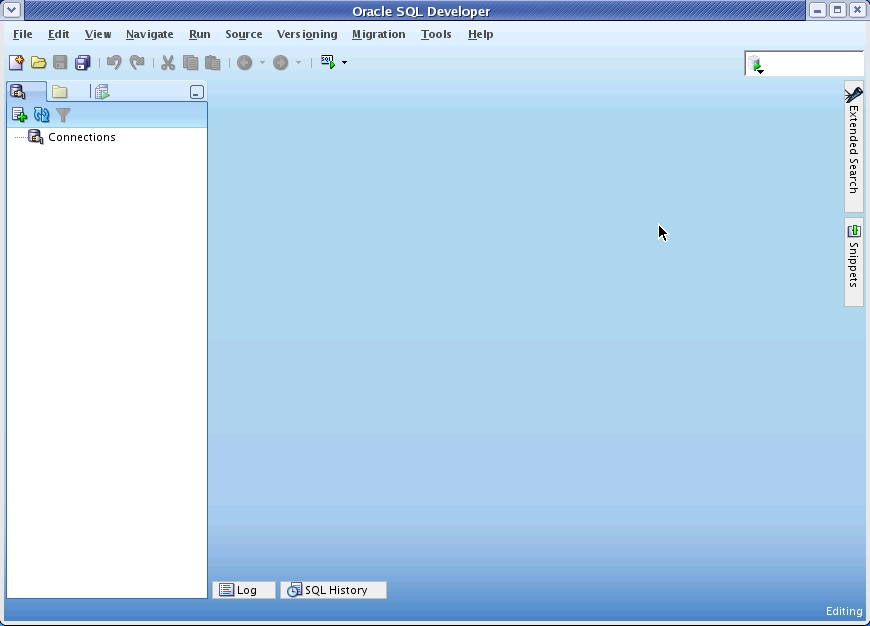
1. Oracle Graphical User Interface

* There are numerous graphical interfaces available from Oracle and other after-market sources
* At NAIT, students use Oracle’s SQL Developer (freely distributable)

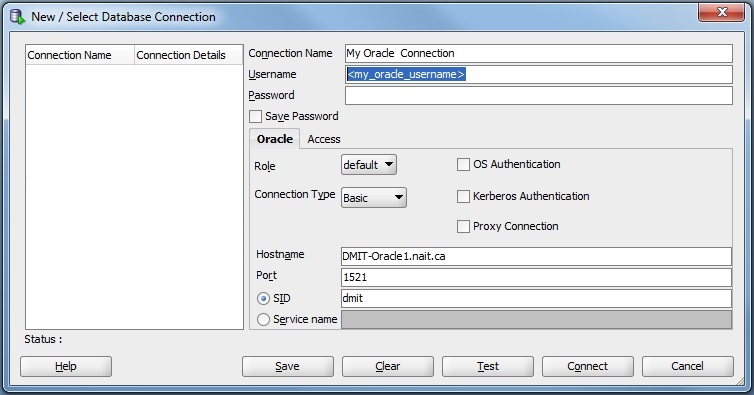
Windows: Start | All Programs | Oracle | SQL Developer | SQL Developer



* You now need to create a Connection to your database
* Right click on Connections, and choose “New Connection”

-- see below for sample

* Enter a Connection Name
* Enter your Oracle Username: (not to be confused with your NAIT username)
* Enter your Password (optional): Password
* Change the “Hostname” to: DMIT-Oracle1.nait.ca
* Leave the “Port” as 1521
* Change the “SID” to: dmit
* Click the “Test” button to check your connection: you should receive a message “Status: Success” if you connected successfully to your database
* Click the “Save” button to save your connection
* Click the “Cancel” button to close the connection creator



* To start an SQL session;
* Go to the menu and choose: Tools | SQL Worksheet, then choose your connection string, hit OK

-- OR --

* Click on the Open SQL Worksheet icon start sql session (or type Ctrl-Shift-W) and choose which SQL connection you would like to use
* The left side of the interface will then show the available Connections while the right side will be split horizontally between the Command Window (upper section) and the Results Window (lower section)
* To enter and execute your code:
* Type your code in the Command Window (upper section)
* Press the Execute Statement (F9) icon execute statement to run a single command
* Press the Run Script (F5) icon run script to run all commands in the Command Window
* Press the Clear icon clear (in either window) to erase your code/results
* To modify the look and feel (i.e. to show line numbers, etc.), go to the menu item: Tools | Preferences
* You can resize the command/results window by dragging the separator up or down
* Please review the menu item Help for more in-depth information

DDL Constraints

* A limitation on the data done when the table is defined using DDL
* In this course, you are expected to name **ALL** constraints
* Constraints we will be using are divided into two groups: integrity (primary keys, foreign keys) and value (check, NULL or NOT NULL and default)
* There are two categories of constraints used in table creation
* Column constraints
* Defined at the same time as the column itself is defined
* Restricted to only one column
* Table constraints
* Defined independently of any column
* **May** apply to a single column or to multiple columns of the current table
* NULL/NOT NULL
* Can only be defined as a column constraint
* **NULL** is currently the default if a status is not specified, however in this course you are expected to specify NULL or NOT NULL for **ALL** columns
* Check constraint rules are case sensitive within single quotes
* The wildcards for the LIKE operator are the percent sign % (matches zero or multiple characters) and the underscore \_ (matches a single character), or you can use regular expressions [] (i.e. lists of characters or ranges of characters) using REGEXP\_LIKE
* DEFAULT
* Not actually a constraint, therefore there is **NO** constraint name
* Can only be defined on a column
* Must be defined before any constraints

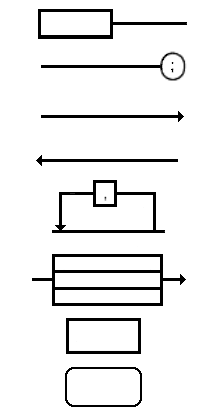
1. Data types (Partial List)

|  |  |  |
| --- | --- | --- |
| **TYPE** | **DESCRIPTION** | |
| NUMBER [(p [, s])] | Numeric data for fixed and floating-point numbers  Maximum guaranteed decimal precision is 38  Range is 1.0 x 10-130 to just less than 1.0 x 10126 | |
|  | Precision (p) | - total number of digits |
| Scale (s) | - number of digits to the right of the decimal point  - range of values: -84 to 127 |
|  | - both are optional but if scale is present precision must be present | |
| CHAR [(size)] | Fixed length character string of length size  Maximum size is 2000  Default and minimum size is 1 | |
| VARCHAR2 (size) | Variable length character string with a maximum length of size  Maximum is 4000  Minimum is 1  There is no default – you must always specify the size | |
| DATE | Data variable – valid range from January 1, 4712 BC to December 31, 9999 AD | |

|  |  |  |
| --- | --- | --- |
| LOB  (not used in this course) | Large Object  Up to 4 gigabytes | BLOB – Binary data  CLOB – Character data  BFILE – Binary file reference (file is outside the database)  NCLOB – Character data (supports 16 bit characters) |
| LONG | Character data of variable length up to 2 gigabytes (Provided for backward compatibility only – DO NOT USE – too many restrictions e.g. union, substring) | |

Syntax diagrams

* These syntax diagrams are drawn using a line-and arrow format to depict valid syntax
* In this depiction you follow a line to correct syntax for what you desire to do
* The following are the symbols used in the diagrams:



The beginning of the diagram

The end of the diagram

The diagram continues on the line below

The diagram is continued from the line above

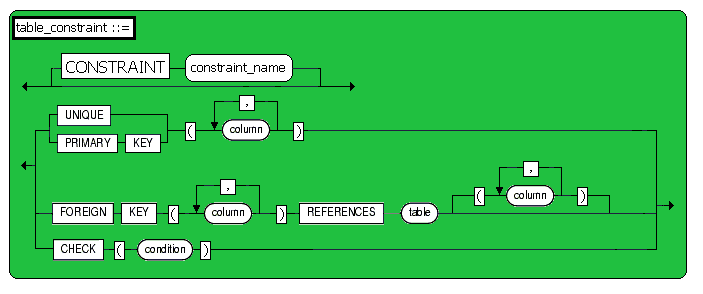
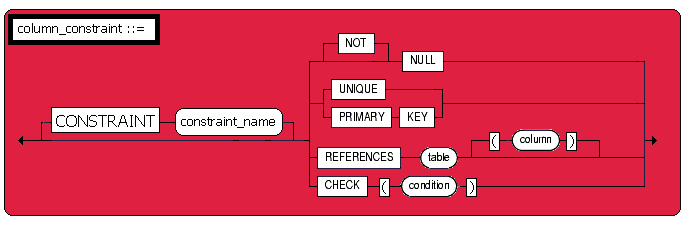
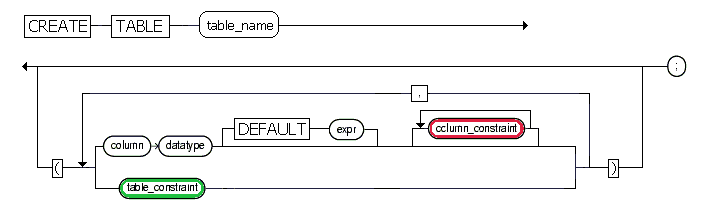
Loop Back

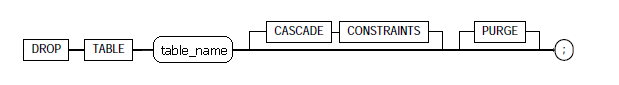
Choice between paths to follow

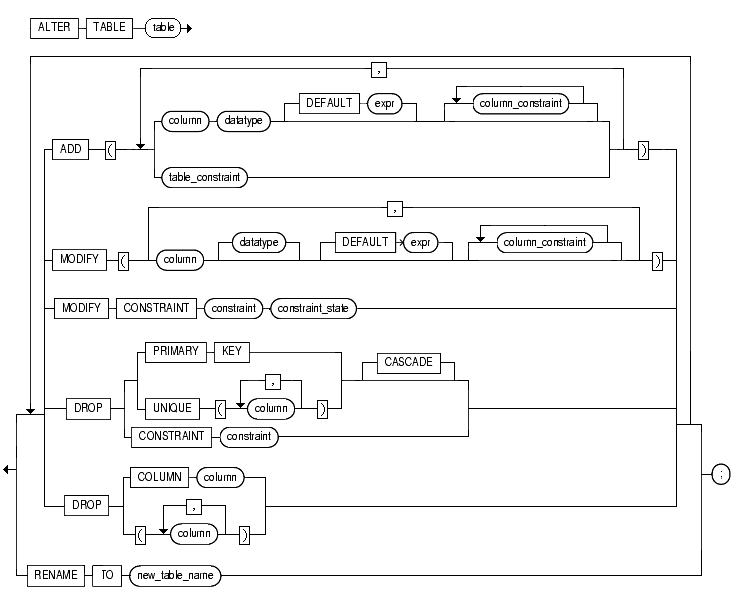
Key word or literal value

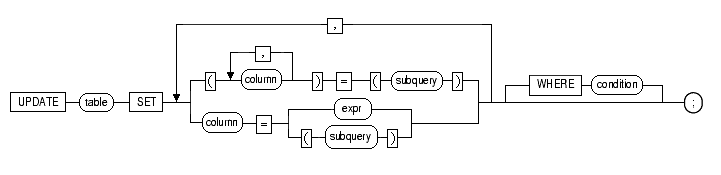
User input

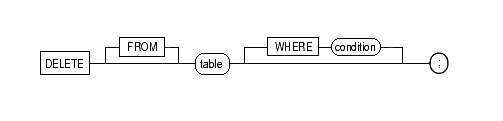
* If a diagram is too complex to be shown in one section (e.g. CREATE TABLE and SELECT), the corresponding sections will be labelled (e.g. table\_constraint ::= ) and enclosed in a rounded box
* If the diagram contains more than one possible path, you **must** select the **one** path that is appropriate to the action you desire (i.e. some paths do nothing, but they are still a path)

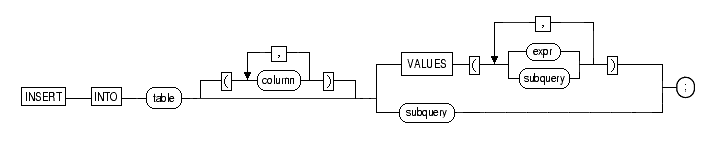


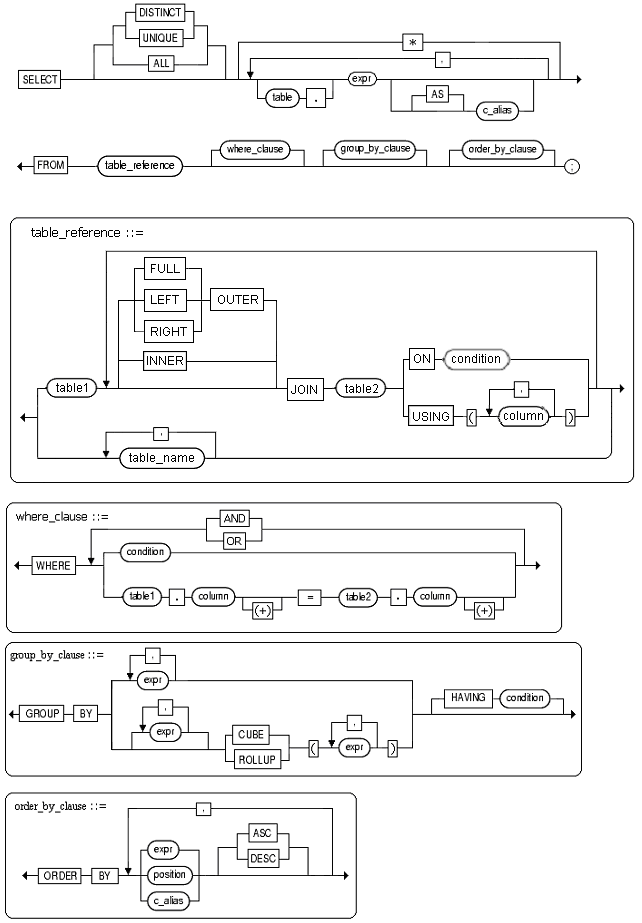












1. DDL

* Cascade deletion
* Oracle can maintain referential integrity by automatically removing dependent children if you remove the parent (use only on the **explicit** advice of a **Senior DBA**)
* "ON DELETE CASCADE" is coded in the child table immediately following the foreign key constraint
* For example in the emp table the first table constraint line would become

CONSTRAINT FK\_Emp\_DeptNo\_Dept\_DeptNo FOREIGN KEY (DeptNo)

REFERENCES Dept (DeptNo) ON DELETE CASCADE,

* Alter table command can
* Add a column
* Modify a column (datatype, size, default value)
* If decreasing a column size the column must be empty
* Add an integrity constraint
* Drop an integrity constraint
* Drop a column
* Rename the table

1. DML

* Select statement
* Must have a from clause - Can use the pseudo table “dual” (a single row “table” owned by the SYSTEM user and available to all users)

SELECT sysdate FROM dual;

* String matching **IS** case sensitive
* Headings with spaces must be enclosed in double quotes (e.g. "valid column alias")
* Insert statement
* Sysdate, null and sequences can appear in a VALUES clause, default can NOT
* In this course, you are required to use a column list on all insert statements
* Delete statement
* Delete records
* Can delimit records to be deleted by specifying a where clause
* Update statement
* Update records
* Can delimit records to be updated by specifying a where clause

1. Functions and operators

**Mathematical Operators**

|  |  |
| --- | --- |
| Purpose | Operator |
| Addition | + |
| Subtraction | - |
| Multiplication | \* |
| Division | / |

**Comparison Operators**

|  |  |
| --- | --- |
| Purpose | Operator |
| Equality | = |
| Inequality | !=, ^=, <> |
| Greater than | > |
| Less than | < |
| Greater than or equal to | >= |
| Less than or equal to | <= |
| List operator | [NOT] IN (value1, value2…) |
| Between two values | [NOT] BETWEEN x AND y |
| Compares a value to every value in a following list or subquery – must be preceded by one of the following:  =, !=, ^=, <>, >, <, >=, <= | ANY, SOME, ALL |
| Null | IS [NOT] NULL |
| Pattern Matching | LIKE  (Wildcards % >= 0 characters;  \_ = 1 character) |
| Regular Expression | REGEXP\_LIKE (string, ‘[A-Z]’) |

**Logical Operators**

|  |  |
| --- | --- |
| Purpose | Operator |
| Negation | NOT |
| True if both conditions are true | AND |
| True if either condition is true | OR |

**Number Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Absolute value | ABS (m) |
| Rounding | ROUND (m [, n]) |
| Truncation | TRUNC (m [, n]) |
|  | m – original number  n – number of decimal places to round or truncate to |

**Character/String Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Concatenation | str1 || str2 or  CONCAT (str1, str2) |
| Convert first letter to upper case, balance of string to lower case | INITCAP (str) |
| Convert all letters to lower case | LOWER (str) |
| Convert all letters to upper case | UPPER (str) |
| Remove characters (chars) from left side of a string (chars defaults to a space) | LTRIM (str [, chars]) |
| Remove characters (chars) from right side of a string (chars defaults to a space) | RTRIM (str [, chars]) |
| Left pads a string to n characters long using chars (chars defaults to a space) | LPAD (str, n [, chars]) |
| Right pads a string to n characters long using chars (chars defaults to a space) | RPAD (str, n [, chars]) |
| Returns portion of a string starting at position m for a length on n characters - if n is not specified, returns the balance of the string (Oracle starts counting strings at one, not zero) | SUBSTR (str, m [, n]) |
| Location of a string (str2) within another string (str1) starting at position m for n occurrences - m and n default to 1 (Oracle starts counting strings at one, not zero) | INSTR (str1, str2 [, m [, n]]) |
| Length of a string | LENGTH (str) |

**Date Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Add a number of days to a date | Date + number |
| Add a number of months to a date, number may be negative | ADD\_MONTHS (date, number) |
| Last day of the month | Last\_day (date) |
| Number of months, as a decimal, between two dates | MONTHS\_BETWEEN (date1, date2) |
| Changes date/time to equivalent in another time zone (e.g. MST, PST) | NEW\_TIME (date, zone1, zone2) |
| Date of the next day of the week specified following the specified date | NEXT\_DAY (date, day of week) |
| Rounds date to the format specified | ROUND (date [, format]) |
| Current date/time | sysdate |
| Truncates date to format specified | TRUNC (date [, format]) |
| Date Formats (see the Help files for more formats) | dd – day  mm – month (2 digits)  mon – month (first 3 letters)  month – month (complete word)  yyyy – year  mi – minutes  ss – seconds |

**Conversion Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Date to varchar2 | TO\_CHAR (date [, format])  e.g. to\_char(‘01/02/03’, ‘yy/mm/dd’) |
| Number to varchar2 | TO\_CHAR (number [, format])  e.g. to\_char(123456, ‘99999’) |
| Varchar2 to date | TO\_DATE (str [, format])  e.g. to\_date(‘2035-15-11’, ‘yyyy-dd-mm’) |
| Varchar2 to number | TO\_NUMBER (str [, format])  e.g. to\_number(‘$12,345.67’, ‘$99,999.99’) |

**Other Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Returns a new value based on a source value compared to one or more comparison values (similar to the CASE operator in Transact-SQL) | DECODE (source,  compare1, output1  [, compare2, output2 [,…]  [, default]) |
| The greatest value in a list | GREATEST (expr1, expr2,…) |
| The least value in a list | LEAST (expr1, expr2,…) |
| Returns the second value (expr2) if the first value (expr1) is null, otherwise returns the first value (expr1) | NVL (expr1, expr2) |
| Returns an integer that uniquely identifies the current user | UID |
| Returns the current Oracle user’s login | USER |

**Aggregate Functions**

|  |  |
| --- | --- |
| Purpose | Function |
| Average | AVG ([DISTINCT|ALL] number) |
| Number of rows in the query (note: will not count rows whose values are NULL) | COUNT (\*| [DISTINCT|ALL] expr) |
| Maximum value | MAX ([DISTINCT|ALL] number) |
| Minimum value | MIN ([DISTINCT|ALL] number) |
| Sum | SUM ([DISTINCT|ALL] number) |

1. Sequences

* In SQL Server you can use identity to create sequential numbers for a primary key; in Oracle you use sequences, but not just for a primary key
* A sequence is simply a database object from which multiple users may generate unique integers
* Typically used to automatically generate primary key values

CREATE SEQUENCE Sequence\_Name

INCREMENT BY x

START WITH y

NOCACHE;

Note: x and y are integers that can be positive or negative

Example:

CREATE SEQUENCE ORDIDINCREMENT BY 1START WITH 622

NOCACHE;

* To access use the pseudo-columns CURRVAL and NEXTVAL

Example:

SELECT ordid.nextval FROM dual; Output: 622

SELECT ordid.currval FROM dual; Output: 622

SELECT ordid.nextval FROM dual; Output: 623

SELECT ordid.nextval FROM dual; Output: 624

SELECT ordid.currval FROM dual; Output: 624

* Pseudo-columns can also be used in other SQL statements (e.g. INSERT)

Example:

INSERT INTO ORDER

(ord\_id, order\_date, comm\_plan, cust\_id, ship\_date, total)

VALUES

(ordid.nextval, sysdate, 'A', 108, NULL, 100.00);

* Dropping a sequence:

DROP SEQUENCE Sequence\_Name;

1. Transactions

* Allow a series of SQL statement to succeed or fail as a logical unit
* In Oracle a transaction begins with

1. The first SQL statement issued after the previous transaction finishes, or
2. The first SQL statement after connecting to the database

* You do NOT explicitly state the beginning of a transaction but you **must** specify the end
* COMMIT [WORK]

1. Makes all changes made done during the transaction permanent
2. Allows other sessions to see the changes made during the transaction
3. Releases all locks acquired during the transaction

* ROLLBACK [WORK]

1. Cancels changes made during the transaction
2. Releases all locks acquired during the transaction

* Disconnecting from the database
* If a session disconnects from the database without ending the transaction, the transaction can either be committed or rolled back
* Command-line mode SQL\*Plus **normally** issues a commit, SQL\*Plus Worksheet **normally** issues a rollback or a commit depending on the version, and SQL Developer **normally** issues a rollback
* If you do not manually commit or rollback, you are not **guaranteed** of a commit or a rollback

1. Data dictionary

* All data about the database is stored in the data dictionary
* There are tables that contain specific information: User\_Tables, User\_Constraints, User\_Source, User\_Triggers;
* Describe these tables, then select the appropriate columns

Example:

DESC User\_Tables;

SELECT Table\_Name

FROM User\_Tables;