Oracle: Data Types

Character Data Types:

VARCHAR2(size [ BYTE | CHAR ]) Variable-length character string having maximum length size bytes or characters. Maximum size is 4000 bytes or characters, and minimum is 1 byte or 1 character. You must specify size for VARCHAR2.

BYTE indicates that the column will have byte length semantics; CHAR indicates that the column will have character semantics.

NVARCHAR2(size) Variable-length character string having maximum length size characters. Maximum size is determined by the national character set definition, with an upper limit of 4000 bytes. You must specify size for NVARCHAR2.

NCHAR(size) Fixed-length character data of length size characters. Maximum size is determined by the national character set definition, with an upper limit of 2000 bytes. Default and minimum size is 1 character.

CHAR ([size [BYTE | CHAR]]) Fixed-length character data of length size bytes. Maximum size is 2000 bytes or characters. Default and minimum size is 1 byte.

Number Data Types:

NUMBER [(precision [,scale])] Number having precision p and scale s. The precision p can range from 1 to 38. The scale s can range from -84 to 127.

BINARY_FLOAT 32-bit floating point number. This datatype requires 5 bytes, including the length byte.

BINARY_DOUBLE 64-bit floating point number. This datatype requires 9 bytes, including the length byte.

Long and Raw Data Types:

LONG Character data of variable length up to 2 gigabytes, or 231 -1 bytes.

LONG RAW Raw binary data of variable length up to 2 gigabytes.

RAW(size) Raw binary data of length size bytes. Maximum size is 2000 bytes. You must specify size for a RAW value.

Date/Time Data Types:

DATE Valid date range from January 1, 4712 BC to December 31, 9999 AD.

TIMESTAMP (fractional_seconds_precision) Year, month, and day values of date, as well as hour, minute, and second values of time, where fractional_seconds_precision is the number of digits in the fractional part of the SECOND datetime field. Accepted values of fractional_seconds_precision are 0 to 9. The default is 6.

TIMESTAMP (fractional_seconds_precision) WITH TIME ZONE All values of TIMESTAMP as well as time zone displacement value, where fractional_seconds_precision is the number of digits in the fractional part of the SECOND datetime field. Accepted values are 0 to 9. The default is 6.

TIMESTAMP (fractional_seconds_precision) WITH LOCAL TIME ZONE All values of TIMESTAMP WITH TIME ZONE, with the following exceptions:

- Data is normalized to the database time zone when it is stored in the database.
- When the data is retrieved, users see the data in the session time zone.

INTERVAL YEAR [(year_precision)] TO MONTH Stores a period of time in years and months, where year_precision is the number of digits in the YEAR datetime field. Accepted values are 0 to 9. The default is 2.

INTERVAL DAY [(day_precision)] TO SECOND [(fractional_seconds_precision)] Stores a period of time in days, hours, minutes, and seconds, where

- day_precision is the maximum number of digits in the DAY datetime field. Accepted values are 0 to 9. The default is 2.
- fractional_seconds_precision is the number of digits in the fractional part of the SECOND field. Accepted values are 0 to 9. The default is 6.

Row ID Data Types:

ROWID Base 64 string representing the unique address of a row in its table. This datatype is primarily for values returned by the ROWID pseudocolumn.

UROWID [(size)] Base 64 string representing the logical address of a row of an index-organized table. The optional size is the size of a column of type UROWID. The maximum size and default is 4000 bytes.
Large Object Data Types:

**CLOB** A character large object containing single-byte or multibyte characters. Both fixed-width and variable-width character sets are supported, both using the database character set. Maximum size is \((4 \text{ gigabytes} - 1) \times \text{ (database block size)}\).

**NCLOB** A character large object containing Unicode characters. Both fixed-width and variable-width character sets are supported, both using the database national character set. Maximum size is \((4 \text{ gigabytes} - 1) \times \text{ (database block size)}\). Stores national character set data.

**BLOB** A binary large object. Maximum size is \((4 \text{ gigabytes} - 1) \times \text{ (database block size)}\).

**BFILE** Contains a locator to a large binary file stored outside the database. Enables byte stream I/O access to external LOBs residing on the database server. Maximum size is 4 gigabytes.
Oracle: ANSI Supported Data Types

CHARACTER [VARYING] (size)

{CHAR | NCHAR} VARYING (size)

VARCHAR (size)

NATIONAL {CHARACTER | CHAR} [VARYING] (size)

{NUMERIC | DECIMAL | DEC} [(precision [, scale])]

{INTEGER | INT | SMALLINT}

{FLOAT [(SIZE)] | DOUBLE PRECISION | REAL}
Oracle: Command Summary

1. SQL commands
   - DDL
     
     | ALTER CLUSTER | ALTER DATABASE | ALTER FUNCTION |
     | ALTER INDEX   | ALTER PACKAGE  | ALTER PROCEDURE |
     | ALTER PROFILE | ALTER RESOURCE COST | ALTER ROLE |
     | ALTER ROLLBACK SEGMENT | ALTER SEQUENCE | ALTER SNAPSHOT |
     | ALTER SNAPSHOT LOG | ALTER TABLE | ALTER TABLESPACE |
     | ALTER TRIGGER  | ALTER USER     | ALTER VIEW     |
     | ANALYZE       | AUDIT          | COMMENT        |
     | CREATE CLUSTER | CREATE CONTROLFILE | CREATE DATABASE |
     | CREATE DATABASE LINK | CREATE FUNCTION | CREATE INDEX |
     | CREATE PACKAGE | CREATE PACKAGE BODY | CREATE PROCEDURE |
     | CREATE PROFILE | CREATE ROLE     | CREATE ROLLBACK SEGMENT |
     | CREATE SCHEMA | CREATE SEQUENCE | CREATE SNAPSHOT |
     | CREATE SNAPSHOT LOG | CREATE SYNONYM | CREATE TABLE |
     | CREATE TABLESPACE | CREATE TRIGGER | CREATE USER |
     | CREATE VIEW   | DROP CLUSTER   | DROP DATABASE LINK |
     | DROP FUNCTION | DROP INDEX     | DROP PROCEDURE |
     | DROP PROFILE  | DROP ROLE      | DROP ROLLBACK SEGMENT |
     | DROP SEQUENCE | DROP SNAPSHOT  | DROP SNAPSHOT LOG |
     | DROP SYNONYM  | DROP TABLE     | DROP TABLESPACE |
     | DROP TRIGGER  | DROP USER      | DROP VIEW |
     | GRANT         | NOAUDIT        | RENAME |
     | REVOKE        | TRUNCATE       | UPDATE |

   - DML
     
     DELETE  EXPLAIN PLAN  INSERT  LOCK TABLE  SELECT

   - Transaction control
     
     COMMIT  ROLLBACK  SAVEPOINT  SET TRANSACTION

   - Session control
     
     ALTER SESSION  SET ROLE

   - System control
     
     ALTER SYSTEM

2. PL/SQL commands

   | CLOSE statement | EXIT statement | FETCH statement | GOTO statement |
   | IF statement    | LOOP statement | NULL statement  | OPEN statement |
   | RAISE statement | RETURN statement |
ALTER TABLE command

PURPOSE: To alter the definition of a table in one of these ways:

- to add a column
- to add an integrity constraint
- to redefine a column (datatype, size, default value)
- to modify storage characteristics or other parameters
- to enable, disable, or drop an integrity constraint or trigger
- to explicitly allocate an extent
- to allow or disallow writing to a table
- to modify the degree of parallelism for a table

SYNTAX:

```
ALTER TABLE [schema.]table
[ADD { {column datatype [DEFAULT expr] [column_constraint] ... |
| table_constraint}]
| ( {column datatype [DEFAULT expr] [column_constraint] ... |
| table_constraint} [, {column datatype [DEFAULT expr] [column_constraint] ... |
| table_constraint} ] ... ) } ]
[MODIFY {column [datatype] [DEFAULT expr] [column_constraint] ... |
| (column [datatype] [DEFAULT expr] [column_constraint] ... |
| [, column datatype [DEFAULT expr] [column_constraint] ... ] ... ) } ]
[DROP drop_clause] ...
```

ALTER VIEW command

PURPOSE: To recompile a view.

SYNTAX:

```
ALTER VIEW [schema.]view
COMPILE
```

CREATE INDEX command

PURPOSE: To create an index on one or more columns of a table or a cluster. An index is a database object that contains an entry for each value that appears in the indexed column(s) of the table or cluster and provides direct, fast access to rows.

SYNTAX:

```
CREATE INDEX [schema.]index
ON [schema.]table (column [ASC|DESC][, column [ASC|DESC]] ...)
```
CREATE SEQUENCE command

PURPOSE: To create a sequence. A sequence is a database object from which multiple users may generate unique integers. You can use sequences to automatically generate primary key values.

SYNTAX:

CREATE SEQUENCE [schema.]sequence
  [INCREMENT BY integer]
  [START WITH integer]
  [MAXVALUE integer | NOMAXVALUE]
  [MINVALUE integer | NOMINVALUE]
  [CYCLE | NOCYCLE]
  [ORDER | NOORDER]

CREATE TABLE command

PURPOSE: To create a table, the basic structure to hold user data, specifying this information:

- column definitions
- integrity constraints
- the table’s tablespace
- storage characteristics
- an optional cluster
- data from an arbitrary query

SYNTAX:

CREATE TABLE [schema.]table
  ( {column datatype [DEFAULT expr] [column_constraint] ...}
   | table_constraint}
  [, {column datatype [DEFAULT expr] [column_constraint] ...}
   | table_constraint} ]...)
  [AS subquery]

Constraints

1. Column constraint
   
   {[NOT] NULL
    | {UNIQUE | PRIMARY KEY}
    | REFERENCES [schema.]table [(column)]
    | ON DELETE {CASCADE | SET NULL}
    | CHECK (condition) }

2. Table constraint
   
   { {UNIQUE | PRIMARY KEY} (column [, column] ...) 
    | FOREIGN KEY (column [, column] ...) 
    REFERENCES [schema.]table [(column [, column] ...)]
    | ON DELETE {CASCADE | SET NULL}
    | CHECK (condition) }
CREATE TRIGGER command
PURPOSE: To define a stored procedure associated with a table.

SYNTAX:

CREATE [OR REPLACE] TRIGGER [user.]trigger
{BEFORE | AFTER}
{DELETE | INSERT | UPDATE [OF column [, column]...}
ON [user.]table
[REFERENCING {OLD [AS] old | NEW [AS] new}]
[FOR EACH ROW]
[WHEN (condition)]
block

CREATE VIEW command
PURPOSE: To define a view, a logical table based on one or more tables or views.

SYNTAX:

CREATE [OR REPLACE] [FORCE | NOFORCE] VIEW [schema.]view
[(alias [,alias]...)]
AS subquery
[WITH CHECK OPTION [CONSTRAINT constraint]]

DELETE command
PURPOSE: To remove rows from a table or from a view’s base table.

SYNTAX:

DELETE [FROM] [schema.]{table | view} [alias]
[WHERE condition]

DROP clause
PURPOSE: To remove an integrity constraint from the database.

SYNTAX:

DROP {PRIMARY KEY
    | UNIQUE (column [, column] ...)
    | CONSTRAINT constraint }
[CASCADE]

DROP INDEX command
PURPOSE: To remove an index from the database.

SYNTAX:

DROP INDEX [schema.]index
**DROP TABLE** command

PURPOSE: To remove a table and all its data from the database.

SYNTAX:

```
DROP TABLE [schema.]table
[CASCADE CONSTRAINTS]
```

**DROP VIEW** command

PURPOSE: To remove a view from the database.

SYNTAX:

```
DROP VIEW [schema.]view
```

**GRANT** command

PURPOSE: To control access to database objects.

SYNTAX:

```
GRANT object_privilege [, object_privilege] ... 
[(column [, column] ...)]
ON [user.]object
TO {user | role | PUBLIC} [, {user | role | PUBLIC}] ...}
[WITH GRANT OPTION]
```

Privileges include SELECT, DROP, ALTER, DELETE, INSERT, ...

**INSERT** command

PURPOSE: To add rows to a table or to a view's base table.

SYNTAX:

```
INSERT INTO [schema.]{table | view | subquery }
[ (column [, column] ...)]
{VALUES (expr [, expr] ...) | subquery}
```
**SELECT** command

PURPOSE: To retrieve data from one or more tables, views, or snapshots.

SYNTAX:

```
SELECT [DISTINCT | ALL] { *
    | { [schema.]table | view }.*
    | expr } [ [AS] c_alias ]
[, { [schema.]table | view }.*
    | expr } [ [AS] c_alias ] ] ...
FROM [schema.]table | view | subquery }
[, [schema.]... ] ...
[WHERE condition ]
[ GROUP BY expr [, expr] ... [HAVING condition] ]
[(UNION | UNION ALL | INTERSECT | MINUS) SELECT command ]
[ORDER BY {expr|position} [ASC | DESC]
[, {expr|position} [ASC | DESC]] ...]
[FOR UPDATE [OF [schema.]table | view].]column
[, [[schema.]table | view].]column ] ... [NOWAIT] ]
```

**UPDATE** command

PURPOSE: To change existing values in a table or in a view’s base table.

SYNTAX:

```
UPDATE [schema.]table | view | subquery } [alias]
SET { (column [, column] ...) = (subquery)
  | column = { expr | (subquery) } }
[, { (column [, column] ...) = (subquery)
  | column = { expr | (subquery) } } ] ...
[WHERE condition]
```
Oracle: Views

• View is a virtual table
  Derived from others tables (either base or virtual)

• 3 types of views, determined by how they are created:
  1. Row subset
     Consist of a subset of rows from a single table
  2. Column subset
     Consist of a subset of columns from a single table
  3. Join
     Consist of a subset of rows and columns from the result of joining 2 or more tables together

• When view is created, subquery is stored in the catalog
  – The defining subquery not executed until the view is accessed by a query
  – The SELECT of the query is ”merged” with the SELECT of the view, creating a composite query that is executed
  – Views are always up to date as a result

• View updates
  – Views can be updated like base tables
  – But not all views are updatable
  – General problems:
    * Problems caused when view does not contain primary key
      1. Insertion anomalies
      2. Deletion anomalies
      3. Update anomalies
    * Invisible results
    * Columns consisting of derived attributes cannot be updated
      (this includes columns representing aggregate function results, grouping, etc.)
    * Ambiguous updates in join views
1. The dictionary
   - Oracle’s dictionary contains information about *everything* stored in the DB
   - Descriptions of these objects accessed via
     - `DICTIONARY(Table_Name, Comments)` view
     - Synonym is DICT
   - Descriptions about the columns of the dictionary objects accessed via
     - `DICT_COLUMNS(Table_Name, Column_Name, Comments)` view

2. The catalogs
   - There are 3 catalogs available:
     (a) `USER_CATALOG(Table_Name, Table_Type)`
        - Displays info about the tables, views, synonyms, sequences owned by a user
        - Synonym is CAT
     (b) `ALL_CATALOG(Table_Name, Table_Type)`
        - Displays info re *selectable* objects a user has access to
        - Includes a column listing the owner
     (c) `DBA_CATALOG(Table_Name, Table_Type)`
        - Displays info re all objects in the DB
        - Columns as per ALL_CATALOG

3. Various catalog view details
   NOTE: The majority have ALL and DBA versions comparable to USER/ALL/DBA_CATALOG
   (a) `USER_OBJECTS (OBJ)`
      - Displays info re *all* objects (e.g., packages, procedures, indexes, ...)
      - Columns:
        - `Object_Name`
        - `Object_Id`
        - `Object_Type`
        - `Created`
        - `Last_DDL_Time`
        - `Timestamp`
        - `Status`
   (b) `USER_TABLES (TABS)`
      - Displays detailed info re objects
      - Columns:
        - `Ident`
        - `Space-related`
        - `Stats-related`
        - `Other`
        - `Table_Name`
        - `Tablespace_Name`
        - `Num_Rows`
        - `Degree`
        - `Cluster_Name`
        - `Blocks`
        - `Instances`
        - `Pct_Free`
        - `Empty_Blocks`
        - `Cache`
        - `Pct_Used`
        - `Avg_Space`
        - `Table_Lock`
        - `Ini_Trans`
        - `Chain_Cut`
        - `Max_Trans`
        - `Avg_Row_Len`
        - `Initial_Extent`
        - `Next_Extent`
        - `Min_Extents`
        - `Max_Extents`
   (c) `USER_UPDATABLE_COLUMNS`
   (d) `USER_VIEWS`
      - Displays detailed info re objects
      - Columns:
        - `View_Name`
        - `Text_Length`
        - `Text`
(e) USER_CONSTRAINTS

- Displays info re all objects (e.g., packages, procedures, indexes, ...)
- Columns:

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Owner</td>
</tr>
<tr>
<td>Constraint_Name</td>
</tr>
<tr>
<td>Constraint_Type</td>
</tr>
<tr>
<td>C: CHECK</td>
</tr>
<tr>
<td>P: PRIMARY KEY</td>
</tr>
<tr>
<td>R: FOREIGN KEY</td>
</tr>
<tr>
<td>U: UNIQUE</td>
</tr>
<tr>
<td>V: WITH CHECK OPTION</td>
</tr>
<tr>
<td>Table_Name</td>
</tr>
<tr>
<td>Search_Condition</td>
</tr>
<tr>
<td>Search condition (with CHECK)</td>
</tr>
<tr>
<td>R_Owner</td>
</tr>
<tr>
<td>Owner of table ref’d by foreign key</td>
</tr>
<tr>
<td>R_Constraint_Name</td>
</tr>
<tr>
<td>Name of constraint ref’d by foreign key</td>
</tr>
<tr>
<td>Delete_Rule</td>
</tr>
<tr>
<td>Action taken when primary key deleted</td>
</tr>
<tr>
<td>Status</td>
</tr>
<tr>
<td>Status of constraint (ENABLED, DISABLED)</td>
</tr>
</tbody>
</table>

4. Additional catalog views:

USER_CONS_COLUMNS
USER_TAB_COMMENTS
USER_COL_COMMENTS
USER_INDEXES (IND)
USER_IND_COLUMNS
USER_CLUSTERS (CLU)
USER_CLU_COLUMNS
USER_DB_LINKS
USER_SNAPSHOTS
USER_SNAPSHOT_LOGS
USER_TRIGGERS
USER_SOURCE
USER_ERRORS
USER_OBJECT_SIZE
USER_ARGUMENTS
USER_TABLESPACES
USER_TS_QUOTAS
USER_SEGMENTS
USER_EXTENTS
USER_FREE_SPACE
USER_USERS
USER_RESOURCE_LIMITS
USER_TAB_PRIVS
USER_COL_PRIVS
USER_SYS_PRIVS
USER_ROLE_PRIVS
USER_AUDIT_OBJECT
USER_AUDIT_SESSION
USER_AUDIT_STATEMENT
USER_AUDIT_TRAIL
USER_OBJ_AUDIT_OPTS
USER_DEF_AUDIT_OPTS