Quick SQL Grammar

Apex Utilities Quick SQL Help Content

Datatypes

num, number	NUMBER
int, integer	INTEGER
d, date	DATE
ts, timestamp	TIMESTAMP
tstz, tswtz, timestamp with local time zone	TIMESTAMP WITH LOCAL TIMEZONE
char, vc, varchar, varchar2, string	VARCHAR2(4000)
vcNNN	VARCHAR2(NNN) NNN identifies a number between 1 and 32767.
vc(NNN)	VARCHAR2(NNN) NNN identifies a number between 1 and 32767.
vc32k	VARCHAR2(32767)
clob	CLOB
blob	BLOB
json	CLOB CHECK (<Column Name> IS JSON)
file	Adds a BLOB column and _FILENAME, _CHARSET, _MIMETYPE, _LASTUPD columns that enhance the ability for file download via a browser.

Table Directives

/api	Generate PL/SQL package API to query, insert, update, and delete data within a table.
/audit	Adds Oracle auditing, by default AUDIT ALL ON [TABLE NAME].
/auditcols, /audit cols, /audit columns	Automatically adds an UPDATED, UPDATED_BY, INSERTED, and INSERTED_BY columns and the trigger logic to set column values.
/colprefix	Prefix all columns of a given table with this value. Automatically adds an underscore if not provided.
/compress, /compressed	Table will be created compressed.
/insert NN	Generate **NN** SQL INSERT statement(s) with random data, for example: /INSERT 20. (Maximum = 1000)
/rest	Generate REST enablement of the table using Oracle REST Data Services (ORDS)
/select	Generate SQL SELECT statement after generating data for each table
/unique	Generate table level unique constraint

Column Directives

/idx, /index, /indexed	Will create a non unique index on the associated column.
/unique	Creates a unique constraint
/check	Creates a check constraint with comma or white space delimited values e.g. /check Yes, No

/constant	When generating data set this column to a constant value. For example /constant NYC.
/default	Adds default value if the column is null
/values	Comma separated list of values to use when generating data. For example /values 1, 2, 3, 4 or /values Yes, No.
/upper	Forces column values to upper case
/lower	Forces column values to lower case
/nn, /not null	Adds a not null constraint on the column
/between	Adds a between check constraint on the column, for example /between 1 and 100
/hidden, /invisible	Hidden columns are not displayed using select * from table.
/references, /reference, /fk	Foreign key references e.g. /references table_name. Note you can reference tables that are not part of your model.
/pk	Identifies column as the primary key of the table. It is recommended not manually specify primary keys and let this app create primary key columns automatically.
--, [comments]	Enclose comments using square brackets or using dash dash syntax

Views

Syntax:

```
view [view_name] [table name] [table name]...
```

Ensure the view name contains no spaces, ensure the table names contain no spaces. Delimit table names by a space or comma.
Example:

dept dname loc emp ename job view dept_emp emp dept

This syntax restricts views to conjunctive queries (i.e. containing equijoin predicates) only.

Settings

You can enter inline settings to explicitly set SQL syntax generation options. Alternatively, you can click **Settings** at the top of the right pane to declaratively set the generation options.

Entering settings directly into the Quick SQL Shorthand pane ensures the same SQL generation options are utilized even if you download the script and later paste it back. For example, enter the following to prefix all table names with TEST and generate for schema OBE:

```
# settings = { prefix: "test", schema: "OBE" }.
```

Alternatively, enter each setting on a separate line for the same result:

```
# prefix: "test"
# schema: "OBE"
```

Note: *The settings must start on a new line and begin with `# settings =` to enter multiple settings, or `#` to enter a single setting per line. All values are case insensitive. Brackets, spaces, and commas can be added for clarity but are ignored. To have all settings generated use `# verbose: true`.*

Setting	Description	Example	Default

APEX	This setting controls the syntax generated to support audit columns. Specifically if audit columns are enabled triggers are generated to maintain the user creating a row and the user last updating a row. When enabled the following function is used:
	`coalesce(sys_context('APEX$SESSION','APP_USER'),user)`
	When not enabled the following function is used:
	`user`
		# apex: true	false

API	Generate PL/SQL APIs on all tables for create, insert, update, delete and query.	# api: true	false
AUDITCOLS	Adds an additional created, created_by, updated and updated_by columns to every table created.	# auditcols: true	false
COMPRESS	When enabled creates all tables compressed. Default is not compressed.	# compress: true	false
CREATEDBYCOL	When Audit Columns are enabled the default column used to track the user who created a row is CREATED_BY. Use this setting to override default audit column name.	# createdByCol: "created_by_user"	created_by

CREATED COL	When Audit Columns are enabled the default column used to track the user who created a row is CREATED. Use this setting to override default audit column name.	# createdCol: "created_date"	created
DATE	By default all DATE columns created using the Oracle DATE datatype. Use this setting to override this default. Valid values are: **date**, **timestamp**, **timestamp with time zone**,**TSWTZ**, **timestamp with local time zone**, **TSWLTZ**.	# date: "timestamp with local time zone"	date
DB	Specifies the database version the syntax should be compatible with. Valid values are: **11g**, **12c**, **19c**, **21c**	# db: "19c"	21c
DROP	Include SQL commands to drop each database object created.	# drop: true	false
LANGUAGE	Generate data used for insert statements using this language. The default is English. Supported languages include: **EN**, **DE**, **KO**, **JA**	# language: "EN"	EN
LONGVC	Allow longer identifiers to be used for database object names. Longer identifiers allow the maximum length a VARCHAR2 column datatype will be 32767 characters. When not set the maximum length of a VARCHAR2 column datatype will be 4000 characters.	# longVC: true	false
ONDELETE	This setting controls how foreign key ON DELETE settings. Valid values include: **cascade**, **restrict**, **set null**	# ondelete: "cascade"	cascade
OVERRIDESETTINGS	When enabled all application settings set via the user interface console are ignored and only settings set in the script will be used.	# overrideSettings: true	false
PK	Determines how the primary key will be set. Primary keys can be set using SYS_GUID, identity column or sequence. Valid values include: **guid**, **seq**, **identity**, **none**	# PK: "identity"	identity
PREFIX	Database object prefix. An underscore will be appended if not provided.	# prefix: "foo"	
PREFIXPKWITHTNAME	Prefix primary key database table columns with name of table. For example the primary key of the EMPLOYEE table would be EMPLOYEE_ID. Without setting the name of implicitly created primary key columns will be ID.	# prefixPKwithTname: true	false
GENPK	Automatically generate an ID primary key column for each table.	# genPK: false	true
RESETSETTINGS	Resets all application settings to default values. When included all application settings currently active for your session will be ignored.	# resetsettings	
ROWKEY	For each table created add a ROW_KEY column that generates an alphanumeric identifier. Values of the ROW_KEY column will be set by generated database table trigger logic.	# rowkey: true	false
TENANTID	For each table add a TENANT_ID column to support mutil-tenant applications. The value of this column is simply added to the table, maintaining this value will need to be provided by the developer.	# tenantID: true	false
ROWVERSION	For each table generated add a ROW_VERSION column that increments by 1 for each update. When enabled database table trigger logic will be generated to increment row versions on update.	# rowVersion: true	false
SCHEMA	Prefix object names with a schema name. The default is no schema prefix for object names.	# schema: "scott"	
SEMANTICS	You can choose no column semantics, or BYTE or CHAR semantics. **varchar2(4000)**, **varchar2(4000 byte)**, **varchar2(4000 char)**	# semantics: "char"	
UPDATED BYCOL	When enabling audit columns use this setting to override default audit column name.	# updatedByCol: "updated_by_user"	updated_by
UPDATED COL	When enabling audit columns use this setting to override default audit column name.	# updatedCol: "updated_dt"	updated
VERBOSE	Will show all settings, not just settings that are different from the default.	# verbose: true	false

Formal Grammar

```
qddl::= stmt+

stmt::= tree
      | view
      | '#' individual_setting
      |  'settings' '=' '{' individual_setting ( ',' individual_setting )* '}'

view::= 'view' view_name table_name+

view_name::= identifier
table_name::= identifier
column_name::= identifier

tree::= node+
tableNode::= indentation tableName tableDirective*
columnNode::= indentation columnName columnDirective* datatype*

indentation::= INDENT | DEDENT | SAMELEVEL

tableDirective::= '/'
       ('api'
      |'audit'|'auditcols'|'audit cols'|'audit columns'
      |'colprefix'
      |'compress'|'compressed'
      |'insert' integer
      |'rest'
      |'select'
      |'unique' )

columnDirective::= '/'
      ('idx'|'index'|'indexed'
      |'unique'
      |'check'
      |'constant'
      |'default'
      |'values'
      |'upper'
      |'lower'
      |'nn'|'not null'
      |'between'
      |'hidden'|'invisible'
      |'references'|'reference'
      |'fk'|'pk' )

datatype::=
       'num'|'number'
       |'int'|'integer'
       |'d'|'date'
       |'ts'|'timestamp'
       |'tstz'|'tswtz'|'timestamp' 'with' 'local' 'time' 'zonechar'
       |'vc'|'varchar'|'varchar2'|'string'
       |'vc' integer | 'vc' '(' integer ')'
       | 'vc32k'
       | 'clob'|'blob'|'jsonfile'

individual_setting::=
      ( 'apex'|'api'|'audit'
      |'cols'|'compress'|'createdbycol'|'createdcol'
      |'date'|'db'|'drop'
      |'language'|'longvc'
      |'ondelete'|'overridesettings'
      |'pk'|'prefix'|'prefixpkwithtname'
      |'genpk'
      |'resetsettings'|'rowkey'
      |'tenantid'|'rowversion'
      |'schema'|'semantics'
      |'updatedbycol'|'updatedcolverbose' ) ':' (string_literal| 'true' | 'false')
```

Railroad Syntax Diagram

diagram.xhtml