SQL - Technical notes and limitations

⚠️ The information below is valid **only for the legacy PL/SQL and T-SQL analyzers embedded in CAST AIP**. See SQL - Available extensions for information about SQL related extensions.

ⓘ **Summary**: This section provides more detail about the support for SQL technologies and the way in which the they are supported by the **legacy PL/SQL and T-SQL analyzers embedded in CAST AIP**.

Microsoft SQL Server and Sybase ASE

Miscellaneous

- For duplicated objects (objects with the same name belonging to the same database but with different users), only one of the objects is managed.

- Support for Bracketed and Quoted Identifiers:
    - Names and symbols will be saved in the Analysis Service in their standard form, i.e.: the form in which they are displayed in the system catalog.
    - Certain character types are not supported when they are enclosed within a QUOTED IDENTIFIER: ' or \n (new line), \r (carriage return), \t (tabulation).
    - If the name of an object (or any other component with a qualified name) contains at least one non-supported character, the link to this object will not be saved in the Analysis Service.
    - If a database name contains non-supported characters, a warning will be emitted, but the synchronization process will continue (it will not be stopped). This warning can be interpreted by the potential risk of failure of query execution or object resolution. The presence of a quotation mark (") in a database name is NOT supported - if this were the case, the synchronization process would be stopped because of the failure of the save process.
- Most of the syntaxes related to administration commands (e.g. create database, create user, dbcc etc) are not supported.
- Grouped Stored Procedures are partially supported (only the first procedure of the group is taken into account).
- Cross server links are not created.

Inter-Application links

Please note that currently **inter-Application links** (i.e. links between source code in separate Applications) between the following combinations of technologies is **not supported**:

- Between Microsoft SQL Server T-SQL Analysis Units in different Applications
- Between Sybase ASE T-SQL Analysis Units in different Applications

If you do require link resolution, then all Analysis Units must belong to the **same Application**.

Dynamic queries

The Microsoft T-SQL Analyzer does not support the identification of links between objects involved in dynamic queries. In the following code, a table (ACCT) is being used in a dynamic query within the procedure (ACCOUNT_SEARCH), as such the analyzer will not identify a link between the table and the procedure.

```
CREATE
OR
REPLACE PROCEDURE ACCOUNT_SEARCH (p_inst_id }}{{IN CHAR, p_cust_id }}{{IN CHAR, {{ }}p_cust_nbr }}{{IN CHAR,
p_cursor }}{{OUT SYS_REFCURSOR) }}{{IS QRY VARCHAR2(1000) := }}{{'SELECT INST_ID, CUST_ID, CUST_NBR, CLOSE_DATE
FROM ACCT ';
```

Column Level Impact Analysis

The functionality known in previous releases of CAST AIP as the **Column Level Impact Analysis** (CLIA) for participating Microsoft SQL Server or Sybase ASE databases is no longer available (and has not been since CAST AIP 7.2.x). Since the embedded SQL analyzer generates objects for columns, you can use the following workaround to obtain CLIA-like results:

- For Column Level Impact Analysis when performing a column modification/deletion: you can put the selected column into a CAST Enlighten view and use the "show linked objects" function.
- For Column Level Impact Analysis when performing a column addition: you can put the table to which you want to add the column into a CAST Enlighten view and use the "show linked objects" function.

Unsupported Microsoft SQL-Server syntax for Microsoft SQL Server

The following syntax is **not supported**:

- ADD [COUNTER] SIGNATURE
- ALTER AUTHORIZATION
- ALTER DATABASE
- ALTER/DROP SCHEMA
- BACKUP CERTIFICATE
- BACKUP DATABASE/LOG: MIRROR TO < backup_device > [,...n] [...next-mirror] is not supported
- BACKUP/RESTORE SERVICE MASTER KEY
- BEGIN CONVERSATION TIMER
- BEGIN DIALOG [CONVERSATION]
- CREATE [PRIMARY] XML INDEX
- CREATE/ALTER/DROP APPLICATION ROLE
- CREATE/ALTER/DROP ASSEMBLY
- CREATE/ALTER/DROP ASYMMETRIC KEY
- CREATE/ALTER/DROP CERTIFICATE
- CREATE/ALTER/DROP CREDENTIAL
- CREATE/ALTER/DROP ENDPOINT
- CREATE/ALTER/DROP FULLTEXT CATALOG
- CREATE/ALTER/DROP FULLTEXT INDEX
- CREATE/ALTER/DROP MESSAGE TYPE
- CREATE/ALTER/DROP PARTITION FUNCTION
- CREATE/ALTER/DROP PARTITION SCHEME
- CREATE/ALTER/DROP QUEUE
- CREATE/ALTER/DROP REMOTE SERVICE BINDING
- CREATE/ALTER/DROP ROLE
- CREATE/ALTER/DROP ROUTE
- CREATE/ALTER/DROP SERVICE
- CREATE/ALTER/DROP USER
- CREATE/ALTER/DROP/OPEN/CLOSE SYMMETRIC KEY
- CREATE/ALTER/DROP/OPEN/CLOSE/BACKUP/RESTORE MASTER KEY
- CREATE/DROP CONTRACT
- CREATE/DROP EVENT NOTIFICATION
- DENY: Microsoft SQL-Server 2005 specific syntax is not supported (Microsoft SQL-Server 2000 syntax is fully supported)
- END CONVERSATION
- GET CONVERSATION GROUP
- GRANT: Microsoft SQL-Server 2005 specific syntax is not supported (Microsoft SQL-Server 2000 syntax is fully supported)
- MOVE CONVERSATION
- RECEIVE
- REVOKE: Microsoft SQL-Server 2005 specific syntax is not supported (Microsoft SQL-Server 2000 syntax is fully supported)
- SEND ON CONVERSATION
- SELECT: FOR <XML>
- WAITFOR ({ GET CONVERSATION GROUP ... | RECEIVE ... }) [, TIMEOUT timeout]
- WITH XMLNAMESPACES

Unsupported Sybase ASE 12.5.x syntax (see elements in bold)

- **MATCH FULL** criterion in CREATE/ALTER TABLE statements with a foreign key constraint:

- ALTER TABLE [database.[owner].]table_name ADD column_name datatype REFERENCES [[database.]owner.]table_name [(column_name)] [**MATCH FULL**]
- Following GRANT/REVOKE commands:
    - GRANT/REVOKE **DBCC dbcc_command [on {all | database }] to { user_list | role_list }**
    - GRANT/REVOKE **default**
    - GRANT SELECT ON **built-in**
    - GRANT **SET PROXY**
    - GRANT/REVOKE { **TRUNCATE TABLE** | { **UPDATE** | **DELETE** } **STATISTICS** }
- Following SET commands:
    - SET **BULK ARRAY**
    - SET **IDENTITY_INSERT**
- SQL-J procedure parameters initialized with default value:
    - CREATE PROC[EDURE] [owner.]procedure ([{[in | out | inout] @parameter_name datatype [= **default_value**]} ...
- Java data type style in SQL-J function parameters:CREATE FUNCTION querytext (xql **java.lang.String**, streamdoc **java.io.InputStream**) RETURNS **java.lang.String** ...
- **WITH STATUSONLY** clause in the KILL command:KILL spid_number **WITH STATUSONLY**
- **MOUNT**/**UNMOUNT DATABASE** commands.
- **FOR XML** clause in SELECT statements:
    - SELECT name, type FROM systypes WHERE name LIKE '%pattern%' **FOR XML**
- Square bracketed identifiers:
    - CREATE TABLE **[my table]**(**[c 1]** int, **[c 2]** char(10))
- Unicode prefix and related notations **u&**:
    - CREATE PROCEDURE my_procedure @v nchar = **u&'+000041'** AS ...
- Encryption features:
    - CREATE/ALTER/DROP **ENCRYPTION KEY**
    - GRANT/REVOKE **CREATE/DECRYPT ENCRYPTION**
    - SELECT .. INTO table_name (column_name **ENCRYPT [WITH [database][.][owner][.]key_name**, ...)
    - CREATE/ALTER TABLE ... **ENCRYPT [with [database][.][owner][.]keyname]**
- **XMLEXTRACT**, **XMLPARSE**, **XMLTEST** keywords.
- **DEFAULT** inside INSERT:
    - INSERT [INTO] <identifier> VALUES (..., **DEFAULT**, ...).
- Derived tables:
    - (SELECT ....) [AS alias [(colum)]] in FROM-clauses of SELECT, SELECT...INTO, CREATE VIEW, INSERT.

Oracle PL/SQL

Oracle 12c

RELIES_ON syntax

"RELIES_ON" in "RESULT_CACHE" clauses is deprecated since Oracle 12c. Therefore a syntax error will be seen in the analysis log when this syntax is encountered by the analyzer.

Oracle 11G R1

The following syntax introduced in Oracle 11g R1 is supported (other new syntax is not supported):

- New dataypes:
    - BINARY_DOUBLE
    - BINARY_FLOAT
    - BINARY XML
    - SIMPLE_INTEGER
- PL/SQL CONTINUE statement
- Compound trigger type
- REGEXP Built-ins
    - REGEXP_LIKE
    - REGEXP_REPLACE
    - REGEXP_SUBSTR
    - REGEXP_INSTR
    - REGEXP_COUNT
- NON ANSI JOIN relations between tables
- PIVOT/UNPIVOT
- LISTAGG

Calling external procedures implemented in a C library from a PL/SQL function

Calling external procedures implemented in a C library from a PL/SQL function is not supported and will cause syntax errors. For details refer: https://docs.oracle.com/cd/B14117_01/appdev.101/b10795/adfns_ex.htm#1006511.

OPEN FOR statement

The OPEN FOR statement for CURSORS (available since version 10G) is not supported.

Conditional compilation directives

Conditional compilation directives (i.e. directives starting with $$ or $, such as $$PLSQL_UNIT, $if, $else, $end etc.) are ignored during an analysis and therefore syntax errors do occur, and the object (function, procedure, trigger,..) is not saved in the CAST Analysis Service.

Links between objects belonging to different schemas not in the same Oracle instance

"Public DB links" are extracted and saved in the Analysis Service, but remote object resolution is not performed by a standard analysis. For instance, a PL/SQL procedure using a remote table through a DB Link on a remote Oracle instance, will have a missing link to this table. A custom workaround is possible that will create this missing link in some situations. Please contact CAST Support for more information.

Superfluous Parentheses

> ⓘ   Note that this limitation does not apply to CAST AIP 8.3.5 where superfluous parentheses are now supported.

A syntax error is reported by the analyzer when analyzing syntactically correct code containing superfluous parentheses. A superfluous parenthesis is a parenthesis that can be omitted without changing the result of the query. The following cases have been identified causing the issue:

Case 1: SELECT statement between parentheses followed by another SELECT statement between parentheses:

```
...
... AS    (
             SELECT ...
          )
          ( <==== Causing syntax error
             SELECT ...
          )
```

Case 2: FROM clause containing an ORDER BY clause or a WITH clause followed by parentheses:

```
... FROM
        (
           (
             ...
           ) ORDER BY
        ) <==== Causing syntax error
```

or:

```
... FROM
        (
           (
             ...
             WITH ...
           )
        ) <==== Causing syntax error
```

Wrapped PL/SQL source code

Wrapping is the Oracle process of hiding PL/SQL source code. The CAST Database Extractor (either embedded in the CAST Delivery Manager Tool or sta ndalone) cannot access the wrapped item's source code, and instead generates a generic body for these objects. These objects can be analyzed by the CAST PL/SQL Analyzer, but with limitations:

- All "fan-out" Quality Rules/metrics (for example "Avoid Artifacts with High Fan-Out - 7778") will not be taken into account
- All Quality Rules/metric based on properties (for example "Number of Code Lines - 10151") will not be taken into account
- For wrapped package/type headers and bodies, the sub-object declaration is not available, therefore the analyzer cannot create these objects
- As wrapped objects have no fan-out, they will have no direct or indirect link to End Points and Data Functions. Therefore, you may consider them as Transaction End Points

Inter-Application links

Please note that currently **inter-Application links** (i.e. links between source code in separate Applications) between the following combinations of technologies is **not supported**:

- Between Oracle Server PL/SQL Analysis Units in different Applications
- Between Oracle Forms Analysis Units and Oracle Server PL/SQL Analysis Units in different Applications

If you do require link resolution, then all Analysis Units must belong to the **same Application**.

Invalid Oracle Server objects

The analyzer does not analyze objects that are marked as invalid in the Oracle Server system catalogue. Invalid objects are still visible in CAST Enlighten (with the property "Object is invalid") but no links will be created from or to these objects. An entry in the analysis log will be created as follows:

```
Object <object name> is found as invalid - Please check your code and eventually recompile it.
```

Oracle objects containing the word LOG as an alias

When the word "LOG" is used as an alias, the SQL analyzer will raise a syntax error during the attempted analysis of the object. The object that contains the alias will not be analyzed once the syntax error has been raised. To workaround this issue, edit the object and replace the alias "LOG" with another alias name.

Fine Grain Impact Analysis

The functionality known in previous releases of CAST AIP as the **Fine Grain Impact Analysis** (FGIA) for participating Oracle schemas is no longer available (and has not been since CAST AIP 7.2.x). There is no workaround for this feature.

Line of Code (LOC) Count

Object type declarations (for example stored procedures declared in a package header) are not included in the Line Of Code (LOC) count when using the SQL Analyzer.