Cisco Unified Intelligence Center SQL Syntax

Cisco Unified Intelligence Center SQL Syntax

This topic provides the following information on using the SQL Syntax for Cisco Unified Intelligence Center:

- Guidelines
- Supported Data Types for Fields and Parameters
- Special Keywords for the SQL Parser (with Sample Queries)

Guidelines

- You cannot use comments in an SQL query.
- A database query must contain a select statement followed by one or more fields. For example: `SELECT [fields] FROM [tables] WHERE [...]`

  This sample query: `select CallTypeID, TimeZone from Call_Type_Interval where TimeZone = 240` creates fields CallTypeID and TimeZone.
- You should not use SELECT*, instead you must list all the fields you want to be returned in a SQL query
- An Anonymous Block must be a valid SQL statement that returns a result set. It may contain parameters named :[paramName], where a colon is always the first character of the parameter name and [paramName] is a remaining part of the parameter name.

  The parameter values entered by a user are substituted into the body of the anonymous block in place of the corresponding parameter names.
- Informix and SQL Server Stored Procedures are supported. Stored Procedures must return a result set. For Stored Procedures, parameters are used to pass the values when making a stored procedure call to the database to obtain the result set.
- Using the Datediff() function in a Where clause causes performance issues.
- There can be no unnamed fields in an SQL query. Each field needs an alias.
- Alias names must be unique.

- Informix stored procedures must contain a returning statement, and for each data type in the returning statement, there must be a corresponding alias specified with the letters AS.

  For example: RETURNING CHAR(32) AS returnID, CHAR(32) AS returnName, INTEGER AS returnRefreshrate, BOOLEAN as returnHistorical; And not: RETURNING CHAR(32, CHAR(32, INTEGER, BOOLEAN; If a user fails to provide an alias, the field name will just be fieldN, where N is the index of unnamed field, such as field1, field2, and so on.

- Informix stored procedure parameter names are prefixed with the 'at' character: *@param1*, *@param2 ...*

Supported Data Types for Fields and Parameters

- *BIGINT, DECIMAL, DOUBLE, FLOAT, INTEGER, NUMERIC, SMALLINT, REAL, TINYINT*
- *CHAR , LONGNVARCHAR, LONGVARCHAR, NCHAR, NVARCHAR, VARCHAR*
- *DATETIME*
- *BOOLEAN, BIT*

Special Keywords for the SQL Parser (with Sample Queries)

- **ALL** (SQL Server or Informix)—SELECT ALL CallTypeID from Call_Type_Interval
- **DISTINCT** (SQL Server or Informix)—SELECT DISTINCT CallTypeID from Call_Type_Interval
- **TOP** (SQL Server)—SELECT TOP 5 CallTypeID from Call_Type_Interval
- **FIRST** (Informix)—SELECT FIRST 5 ID FROM CUICDATASETINFO
- **UNIQUE** (Informix)—SELECT UNIQUE NAME FROM CUICGRID

Unified Intelligence Center supports these aggregate functions for both Informix and SQL Server: SUM, COUNT, MIN, MAX, and AVG.

In cases where a report definition field is an aggregate function (such as *sum(CallsHandled),* and that field is a key criteria field or an advanced filter, the supported syntax is:

```
SELECT (fields)FROM [tables]
WHERE [...]
GROUP BY [...]
HAVING [...] optional
ORDER BY [...] optional
```

Sample query:

```
select CallTypeID, TimeZone, sum(CallsHandled) as total, avg(CallsHandled) as average
from Call_Type_Interval
where TimeZone = 240
group by CallTypeID, TimeZone
having sum(CallsHandled) in(3, 5, 13) and avg(CallsHandled) > 0
order by CallTypeID
```