Database objects

The following table describes the database objects displayed in SQL Query Tuner and contains information regarding each one, including object name, DBMS platform, and any notes pertaining to the specified object.

In SQL Query Tuner, database objects are stored in Data Source Explorer as subnodes of individual, pertinent databases.

Database Object	Notes
Check Constraints	A check constraint is a search condition applied to a table. When a check constraint is in place, Insert and Update statements issued against the table will only complete if the statements pass the constraint rules. Check constraints are used to enforce data integrity when it cannot be defined by key uniqueness or referential integrity restraints. A check condition is a logical expression that defines valid data values for a column.
Foreign Keys	A foreign key references a primary or unique key of a table (the same table the foreign key is defined on, or another table and is created as a result of an established relationship). Its purpose is to indicate that referential integrity is maintained according to the constraints. The number of columns in a foreign key must be equal to the number of columns in the corresponding primary or unique key. Additionally, the column definitions of the foreign key must have the same data types and lengths. Foreign key names are automatically assigned if one is not specified.
Groups	Groups are units that contain items. Typically, groups contain the result of a single business transaction where several items are involved. For example, a group is the set of articles bought by a customer during a visit to the supermarket.
Indexes	An index is an ordered set of pointers to rows in a base table. Each index is based on the values of data in one or more table columns. An index is an object that is separate from the data in the table. When an index is created, the database builds and maintains it automatically. Indexes are used to improve performance. In most cases, access to data is faster with an index. Although an index cannot be created for a view, an index created for the table on which a view is based can improve the performance of operations on that view. Indexes are also used to ensure uniqueness. A table with a unique index cannot have rows with identical keys.
Packages	A package is a procedural schema object classified as a PL/ SQL program unit that allows the access and manipulation of database information. A package is a group of related procedures and functions, together with the cursors and variables they use, stored together in the database for continued use as a unit. Similar to standalone procedures and functions, packaged procedures and functions can be called explicitly by applications or users. DB applications explicitly call packaged procedures as necessary with privileges granted, a user can explicitly execute any of the procedures contained in it. Packages provide a method of encapsulating related procedures, functions, and associated cursors and variables together as a unit in the database. For example, a single package might contain two statements that contain several procedures and functions used to process banking transactions. Packages allow the database administrator or application developer to organize similar routines as well as offering increased functionality and database performance. Packages provide advantages in the following areas: encapsulation of related procedures and variables, declaration of public and private procedures, variables, constraints and cursors, separation of the package specification and package body, and better performance. Encapsulation of procedural constructs in a package also makes privilege management easier. Granting the privilege to use a package makes all constructs of the package assessable to the grantee. The methods of package definition allow you to specify which variables, cursors, and procedures are: public, directly accessible to the users of a package, private, or hidden from the user of the package.
Primary Keys	A key is a set of columns used to identify or access a row or rows. The key is identified in the description of a table, index, or referential constraint. The same column can be part of more than one key. A unique key is a key that is constrained so that no two of its values are equal. The columns of a unique key cannot contain NULL values. The primary key is one of the unique keys defined on a table, but is selected to be the key of the first importance. There can only be one primary key on a table.

Procedures	A procedure is an application program that can be started through the SQL CALL statement. The procedure is specified by a procedure name, which may be followed by arguments enclosed within parenthesis.

The argument or arguments of a procedure are individual scalar values, which can be of different types and can have different meanings. The arguments can be used to pass values into the procedure, receive return values from the procedure, or both.

A procedure, also called a stored procedure, is a database object created via the CREATE PROCEDURE statement that can encapsulate logic and SQL statements. Procedures are used as subroutine extensions to applications, and other database objects that can contain logic.

When a procedure is invoked in SQL and logic within a procedure is executed on the server, data is only transferred between the client and the database server in the procedure call and in the procedure return. If you have a series of SQL statements to execute within a client application, and the application does not need to do any processing in between the statements, then this series of statements would benefit from being included in a procedure. |
| Tables | Tables are logical structures maintained by the database manager. Tables are composed of columns and rows. The rows are not necessarily ordered within a table.

A base table is used to hold persistent user data.

A result table is a set of rows that the database manager selects or generates from one or more base tables to satisfy a query.

A summary table is a table defined by a query that is also used to determine the data in the table. |
| Triggers | A trigger defines a set of actions that are performed when a specified SQL operation (such as delete, insert, or update) occurs on a specified table. When the specified SQL operation occurs, the trigger is activated and starts the defined actions.

Triggers can be used, along with referential constraints and check constraints, to enforce data integrity rules. Triggers can also be used to cause updates to other tables, automatically generate or transform values for inserted or updated rows, or invoke functions to perform tasks such as issuing alerts. |
| Unique Keys | A unique key is a key that is constrained so that no two of its values are equal. The columns of a unique key cannot contain null values. The constraint is enforced by the database manager during the execution of any operation that changes data values, such as INSERT or UPDATE. The mechanism used to enforce the constraint is called a unique index. Thus, every unique key is a key of a unique index. Such an index is said to have the UNIQUE attribute.

A primary key is a special case of a unique key. A table cannot have more than one primary key.

A foreign key is a key that is specified in the definition of a referential constraint.

A partitioning key is a key that is part of the definition of a table in a partitioned database. The partitioning key is used to determine the partition on which the row of data is stored. If a partitioning key is defined, unique keys and primary keys must include the same columns as the partitioning key, but can have additional columns. A table cannot have more than one partitioning key.

Clustered: A cluster composes of a group of tables that share the same data blocks, and are grouped together because they share common columns and are often used together.

Filegroup: Lets you select the filegroup within the database where the constraint is stored.

Fill Factor: Lets you specify a percentage of how large each constraint can become. |
| Views | A view provides an alternate way of looking at the data in one or more tables.

A view is a named specification of a result table and can be thought of as having columns and rows just like a base table. For retrieval purposes, all views can be used just like base tables.
You can use views to select certain elements of a table and can present an existing table in a customized table format without having to create a new table. |