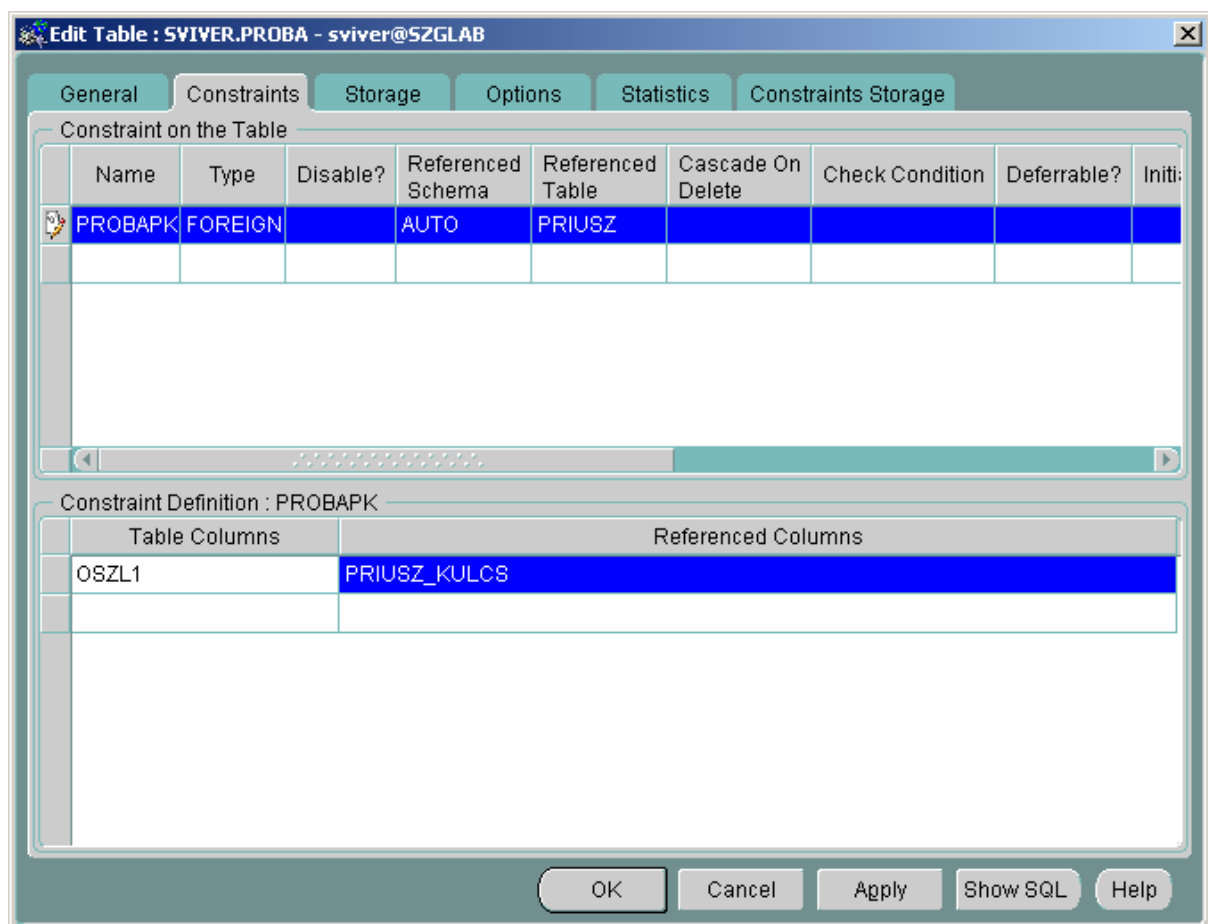## Database constraints in Oracle

Database constraints are constraints or simple rules regarding the database data aiming to maintain consistency on a database server level. Moreover, based on the constraints Oracle is able to optimize the queries more efficiently. Such constraint might be the syntax check of a record field (e.g. personal ID number contains 2 alphabetic and 6 numeric characters), but also the check of related records (in order to ensure that the data references are correct). Thus constraints are very useful, but can cause serious troubles during system modifications.

There might be several constraints that belong to a table. Basically there are two ways to define a constraint: either by entering the respective SQL command directly; or by using a graphical tool to create the command – thus eliminating the need to know the exact syntax. Upon creating a new or editing an existing table in Enterprise Manager constraints can be set on the Constraints tab.

As a matter of fact, if the NOT NULL keyword is part of the table definition – meaning that a field value cannot be NULL – can also be considered, as a constraint. However, this constraint is set during table field edition.

In the Enterprise Manager the Constraints tab is practically divided into two parts. The upper part (Constraint on the Table) is used to define the constraints, while the lower part is used to define the mapping between the actual constraint and the table columns (Constraint Definition). The contents of the lower part are always related to one entry in the upper part, thus upon entering data in the lower part the user must pay attention to select the appropriate constraint in the upper part.

The table in the upper part of the Constraints tab contains the following columns:

Name – Here the user can assign a name to the constraint. If the name is not defined, the system automatically generates a more or less random string. The name must not contain accented letters or spaces. Even though defining a constraint name is not mandatory, it is highly recommended for the following reason. In case an operation is not executable because it is constraint-violating the system refers to the constraint using the constraint name. With a descriptive name the problem is apparent, without making the user to spend valuable time locating it.

Type – The type of the constraint which can have the following values in Oracle: UNIQUE, PRIMARY, FOREIGN, CHECK. The constraint type basically defines the further columns to be filled out. For more details, please see below.

Disable – It is a Yes / No flag. When checked, the constraint is temporarily disabled – the system acts as if the constraint was not there, at all. It is recommended to temporarily disable a constraint upon importing a massive amount of not constraint-violating data thus making data processing faster.

Referenced Schema – It is used with FOREIGN type constraints. See below.

Referenced Table – It is used with FOREIGN type constraints. See below.

Cascade On Delete – It is used with FOREIGN type constraints. See below.

Check Condition – It is used with CHECK type constraints. See below.

Deferrable – A Yes / No flag. When checked, the user (or the connecting application) can delay the constraint, which means that upon constraint violation the system does not give an error message right after the command violating the constraint but only at the end of the transaction, at COMMIT. The main advantage of using this flag is that the programmer has to perform error check only at only one occasion at the end of the transaction – and not after every single operation. This flag provide only an option though, which has to be activated by the developer at the beginning of the transaction.

Initially Deferred – A Yes / No flag which only has an effect if the Deferrable flag was set to yes. In case both the Deferrable and the Initially Deferred flags are set to true the system automatically postpones constraint checking without the user specifically asking for this behavior.

No validate – A Yes / No flag. When set to true the system is only checking the new and the modified records; when set to false the system guarantees the constraint compliance of the already existing records, as well.

RELY (Do Not Enforce) – A Yes / No flag. Setting this flag results in the system creating a "belief" or "RELY" constraint, meaning that the system does not check if the existing records (and the new and the modified) comply with the constraint or not but relies solely on the user to comply with the constraint. In these cases constraints are only used for query optimization hints, however processing speed increases since data checks are eliminated.

CHECK constraint

In case of a CHECK constraint the field Check Condition must be filled out. Herein a logical expression can be defined - similar to a WHERE clause in a query. When inserting new or modifying an existing record the system checks the value of the logical expression and in case it is false, the execution might be denied. Unfortunately sub-queries - more complicated queries, referencing other tables - cannot be embedded in the logical expression. Therefore, the logical expression can only refer to the fields that belong to the constraint's table.

UNIQUE constraint

A UNIQUE constraint can be used to define a key to the table. The creation of a UNIQUE constraint results in the followings: firstly, the combination defined by the key fields values must be unique in the table; secondly, based on the key fields an index is automatically generated to increase query speed. The field(s) belonging to the key must be enumerated in the Table Columns column of the table in the lower part.

PRIMARY constraint

A PRIMARY constraint marks a primary key. It is very similar to the UNIQUE constraint mentioned above, but involves more restrictions. On the first hand, while a table might have several UNIQUE constraints only one PRIMARY constraint is allowed. On the other hand, while a UNIQUE constraint allows NULL as a key field value a PRIMARY constraint ensures that primary key fields are always assigned a value other than NULL. Finally, a primary key can definitely identify a single table record, thus it is referenced by foreign keys from other tables. The process of creating a PRIMARY constraint is similar to that of a UNIQUE constraint automatically generating an index in the background.

FOREIGN constraint

The FOREIGN constraint is the most interesting and complicated constraint in the Oracle repertoire. The FOREIGN constraint marks a foreign key. A foreign key basically means that certain fields in the given table are referring to a record in another table based on certain fields of that table. The referenced table is usually called the parent, while the referencing table is called the child table. The constraint that belongs to a foreign key is defined in the child table.
Upon foreign key definition the Referenced Schema and Referenced Table columns are used to store which table records of which schema will be referenced from the current table - that is the parent table is defined here.
Then the table in the lower part of the screen is used to list the mapping(s) between the referencing and the referenced field(s). The referencing field of the child table goes into the Table Columns; the referenced field of the parent table goes into the Referenced Columns.
In case the Cascade On Delete parameter of a constraint is set a deletion chain is created; meaning that when a record is deleted in the parent table the record(s) referencing this record in the child table are automatically deleted.