ROLE OF CONSTRAINTS

Creating table from an existing Table

In addition to creating table as above, we can also create a new table from the existing table also .We apply the AS sub-query clause to both create the table and insert rows returned from the subquery. For example:

```
SQL> Create table emp1
     AS
     Select empno, ename, hiradate, sal from EMP where comm IS NULL;
```

This statement will create a new table emp1 which contains the rows returned by another table emp which satisfies the condition comm IS NULL.

Follow the following guidelines while creating a table from another table:

- If column specifications are given, the number of columns must equal the number of columns in the subquery select statement.
- If no column specifications are given, the columns of the new table are the same as the column names in the subquery.
- The select statement in the subquery will insert the records into the new table created.

Role of Constraints to achieve data integrity

To understand the role of constraints in database let us consider a case, when we appear for some interview, there are certain constraints for our qualification. Person who satisfies the given qualification constraints are eligible for interview, so these restrictions are called constraints which are to be enforced to select the correct candidate. Such limitations have to be enforced on the data to achieve the integrity (correctness).

The data, which does not, satisfies the conditions will not be stored, this ensures that the data stored in the database is valid and has gone through the integrity rules which prohibit the wrong data to be entered into the database.

Oracle allows us to apply constraints on single column or more than one column through the SQL syntax that will check data for integrity. While creating a table, constraints can be placed on the values that will be entered into the column(s). SQL will reject any value that violates the criteria that were defined.

Categories of Constraints

There are two categories of constraints, these are:

- Column constraints
- Table constraints

Column Constraints

When we define constraints along the definition of the column while creating or altering the table structure, they are called as column constraints. Column constraints are applied only to the individual columns. These constraints are applied to the current column. A column level constraint cannot be applied if the data constraint spans across multiple columns in a table.

For example: Empno of EMP table is a primary key, which is column level constraint.

Table Constraints

Table constraints are applied to more than one column of a table. These constraints are applied after defining all the table columns when creating or altering the structure of the table. A table level constraint can be applied if the data constraint spans across multiple columns in a table.

For example: in case of bank database the account number field of account holder table is not sufficient to uniquely identify the entities because two person can have joint account and hence repeating the value of account number once for each account holder,so in this case we have to apply primary key constraint on combination of account number field and name field to achieve the uniqueness. Now, in this case the primary key

constraint is applied on more than one column of the table, so this is the table level constraint.

Constraints can be added to the table at the time of creating a table with the create table command or later on with the help of ALTER TABLE command. All the details of constraints are stored in data dictionary. Each constraint is assigned a name by itself. Constraints are stored in the system tables by the Oracle engine .We can also give more user friendly names to the constraints so that they can be easily referenced, otherwise the name is automatically generated of the form SYS_Cn where n is unique number.