### Using The Default Clause ====>
====================================

* If while inserting a record in the table ,we don’t provide any value for a column , then Oracle automatically inserts NULL value in it.

* In order, to change this behavior Oracle allows us to use DEFAULT clause to specify DEFAULT VALUE for a column while creating the table.

* This value can be literal value, an expression, or an SQL Function, such as SYSDATE .

* Oracle will insert it in the column when INSERT INTO statement does not provide a specific value.

# Syntax ==>
- CREATE TABLE table_name (
Column1 datatype (size) ,
Column2 datatype (size) ,
Column3 datatype (size) DEFAULT <value>);

## Renaming Constraints ===>
=============================

* We can use the RENAME CONSTRAINT command to change constraint name to a more descriptive name.

* To rename a constraint name we use the following command:
- ALTER TABLE Students RENAME CONSTRAINT ST_NM_NN TO STD_NAME_NN

## Disabling/Enabling Constraint ===>
======================================

* We can enable and disable constraints as necessary by using the ALTER TABLE command.

* By default, when a constraint is created, it is enabled, unless we explicitly disable it.

* We might want to disable constraints when updating massive volumes of data or inserting large amounts of data at once to decrease overall time for these operations.

* After the data manipulation is performed, we can re-enable the constraint.

* Syntax:
- ALTER TABLE <table_name> ENABLE/DISABLE Constraint <constraint_name>;

* Example: The following statement disables an existing primary key constraint named OD_ID_PK on the ORDERS table.
- ALTER TABLE Orders DISABLE CONSTRAINT OD_ID_PK;

##  Validate / NoValidate Clause ===>
=====================================

* By default ,when a constraint is applied on a table which already contains data , then Oracle verifies the existing data also before adding/enabling the constraint.

* Now if we want that Oracle should ignore existing data and apply constraint only on future data,then we can use a special clause called NOVALIDATE while enabling/adding the constraint.

* Overall , there are 4 combinations for this and they are:
- ENABLE VALIDATE
- ENABLE NOVALIDATE
- DISABLE VALIDATE
- DISABLE NOVALIDATE

* Syntax:
- ALTER TABLE <table_name> ENABLE/DISABLE VALIDATE/NOVALIDATE Constraint <constraint_name>;

* Example:
- The following statement enables an existing CHECK CONSTRAINT named VM_CT_CH on the VENDOR_MASTER table forcing it not to validate existing data.
- ALTER TABLE Vendor_Master ENABLE NOVALIDATE CONSTRAINT VM_CT_CH;

# Enable Validate: ==>
- This means that Oracle will validate the existing data and only if the data satisfies the condition of the constraint , the constraint will get enabled . This is also the default option for ENABLE clause.

# Enable NoValidate: ==>
- Forces Oracle not to validate the existing data but only apply the constraint on future data. However it only works with NOT NULL, CHECK and REFERENTIAL CONSTRAINT.

# Disable NoValidate: ==>
- This is the default option for DISABLE clause and by doing this we will disable the constraint and prohibit any kind of check on the incoming data.

# Disable Validate: ==>
- This is a special case in Oracle where the constraint is disabled but kept valid . In this situation no DML operation is allowed on the table because Oracle is not in a position to validate the data. Thus by doing this we make the table read only.

##  Simple Way Of Making Table Read Only ==>
============================================

* For any table we own, we can modify the data or alter the object.

* Oracle 11g introduced the ability to make a table read-only; this prevents us from performing any data manipulations or issuing any changes to the structure of the table.

* To make a table read-only or return it to write mode, we use the following syntax options.

- ALTER TABLE tablename READ ONLY;
- ALTER TABLE tablename READ WRITE;

## Obtaining Details About Constraints ===>

============================================

* Whenever we apply constraints on a table , then Oracle internally maintains it’s details in it’s DATA DICTIONARIES.

* For constraints , Oracle has 4 DATA DICTIONARIES:
- ALL_CONSTRAINTS
- ALL_CONS_COLUMNS
- USER_CONSTRAINTS
- USER_CONS_COLUMNS

* The first two contain details of all the constraints in the Oracle database , while the next two contain the details of the constraints of the current user only.

# USER_CONSTRAINTS: ==>

=======================

* It contains the following useful columns:

* CONSTRAINT_NAME: Stores the name of the constraint
* CONSTRAINT_TYPE: Stores a single character to indicate the type of the constraint . These characters are :
- P -> Primary Key
- U -> Unique
- C -> Check
- R -> Referential
- C -> Not Null

* SEARCH_CONDITION: Contains the condition of the constraint and only contains an entry for the Check Constraint.

* TABLE_NAME: Name of the table on which constraint is applied  DELETE_RULE: Contains the deletion rule for Referential Constraint.

# WAQ to display name and type of all the constraints applied on EMP table.

```
- select table_name, constraint_name
  2  from user_constraint
  3  where table_name = 'VENDOR_MASTER';
```

# USER_CONS_COLUMNS: ==>
========================

* It contains the following useful columns:
- CONSTRAINT_NAME: Stores the name of the constraint
- COLUMN_NAME: Stores a name of column on which the constraint has been applied
- POSITION: Contains the position of the column in the constraint
- TABLE_NAME: Name of the table on which constraint is applied

##  Removing Constraints ====>
==============================

* When a constraint is no longer needed, we can drop it with the ALTER TABLE command and the DROP clause.

* Syntax:
- ALTER TABLE <table_name> DROP CONSTRAINT <constraint_name>;

* Example:
* Dropping Check Constraint called EMP_SAL_CH applied on Emp table:
- ALTER TABLE Emp DROP CONSTRAINT EMP_SAL_CH;

* Dropping Foreign Constraint called EMP_DNO_FK applied on Emp table:
- ALTER TABLE Emp DROP CONSTRAINT EMP_DNO_FK;

* Dropping Primary Key Constraint called DEPT_DNO_PK applied on Dept table:
- ALTER TABLE Dept DROP CONSTRAINT DEPT_DNO_PK;

# Important Note

- We cannot drop a PRIMARY KEY or UNIQUE KEY constraint that is part of a REFERENTIAL INTEGRITY constraint without also dropping the FOREIGN KEY.

- To drop the referenced key and the foreign key together, we use the CASCADE clause., while dropping the constraint.

- If we omit CASCADE, then Oracle Database does not drop the PRIMARY KEY or UNIQUE constraint if any foreign key references it.