CREATING ORACLE TABLES

To create a table in ORACLE, you use the CREATE table statement. The general form of a CREATE TABLE statement can be studied using the following examples:

```
CREATE TABLE vijay.item
  (Item_id NUMBER CONSTRAINT pk_item PRIMARY KEY,
   Item_name VARCHAR2(10) CONSTRAINT nn_itemname NOT NULL
   CONSTRAINT upper_itemname
   CHECK (item_name = UPPER(item_name)),
   Ccenter_id VARCHAR2(10) CONSTRAINT fk_ccenter
   REFERENCES scott.costcenter(ccenter_id),
   Entry_date DATE DEFAULT SYSDATE,
   Item_cost NUMBER(10,2) CONSTRAINT ck_itemprice
   CHECK (item_price < 1500))
TABLESPACE inventory
STORAGE (INITIAL 6144
         NEXT 6144
         MINEXTENTS 1
         MAXEXTENTS 5
         PCTINCREASE 5);
```

In this example we have chosen to name the schema in which the table is created as in vijay.item. If this is omitted the table will be created in the user's own schema.

<table>
<thead>
<tr>
<th>CONSTRAINT</th>
<th>identifies the integrity constraint by the name constraint. Oracle uses this name to store the definition in the data dictionary. The data dictionary table is USER_CONSTRAINTS. There are two types of integrity constraints: column constraint and table constraint. Table constraint can impose rules in any column(s) of the table being created. A column constraint define a rule for the current column being defined.</th>
</tr>
</thead>
<tbody>
<tr>
<td>UNIQUE</td>
<td>designates a column or combination of columns as a unique key. You cannot define UNIQUE constraints on index-organized tables.</td>
</tr>
<tr>
<td>PRIMARY KEY</td>
<td>defines a column or combination of columns as the table's primary key.</td>
</tr>
<tr>
<td>FOREIGN KEY</td>
<td>defines a column or combination of columns as the foreign key in a referential integrity constraint.</td>
</tr>
<tr>
<td>REFERENCES</td>
<td>identifies the foreign key column that references the primary key of some other tables. This establishes the relationship between two tables.</td>
</tr>
</tbody>
</table>
**ON DELETE CASCADE** indicates that on deleting a column in the referenced table (the other tables) corresponding rows in the table being created will be deleted.

**NULL** allows null values for a column.

**NOT NULL** specifies that a column cannot contain null values.

**CHECK** stipulates a condition that each row in the table must satisfy.

**USING INDEX** specifies parameters for the index Oracle uses to enable a UNIQUE or PRIMARY KEY constraint. Use this clause only when enabling UNIQUE and PRIMARY KEY constraints.

**AS subquery** inserts the rows returned by the subquery into the table upon its creation.

You will generally create the reference tables (parent tables) first before creating the tables that references the parent table.

A **tablespace** is an allocation of space in the database that can contain any of the following segments: data segments, index segments, rollback segments, temporary segments. All databases have at least one tablespace, SYSTEM, which Oracle creates automatically when you create the database. When you create a tablespace, it is initially a read-write tablespace. After creating the tablespace, you can subsequently use the ALTER TABLESPACE command to take it offline or online, add datafiles to it, or make it a read-only tablespace.

The storage clause specifies the storage characteristics of a table. This clause is also used in creating other oracle schema objects such as clusters, indexes, rollback segments and tablespaces.

Some of the keywords and parameters in specifying a storage clause are explained below:

<table>
<thead>
<tr>
<th><strong>INITIAL</strong></th>
<th>specifies the size in bytes of the object's first extent. Oracle allocates space for this extent when you create the schema object. You can use K or M to specify this size in kilobytes or megabytes. The default value is the size of 5 data blocks. The minimum value is the size of 2 data blocks. The maximum value depends on your operating system. Oracle rounds values up to the next multiple of the data block size for values less than 5 data blocks, and rounds up to the next multiple of 5 data blocks for values greater than 5 data blocks.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NEXT</strong></td>
<td>specifies the size in bytes of the next extent to be allocated to the object. You can use K or M to specify the size in kilobytes or megabytes. The default value is the size of 5 data blocks. The minimum value is the size of 1 data block. The maximum value depends on your operating system. Oracle rounds values up to the next multiple of the data block size for values less than 5 data blocks, and rounds up to the next multiple of 5 data blocks for values greater than 5 data blocks.</td>
</tr>
</tbody>
</table>

- 2 -
multiple of the data block size for values less than 5 data blocks. For values greater than 5 data blocks, Oracle rounds up to a value that minimizes fragmentation.

**PCTINCREASE** specifies the percent by which the third and subsequent extents grow over the preceding extent. The default value is 50, meaning that each subsequent extent is 50% larger than the preceding extent. The minimum value is 0, meaning all extents after the first are the same size. The maximum value depends on your operating system.

You cannot specify PCTINCREASE for rollback segments. Rollback segments always have a PCTINCREASE value of 0.

Oracle rounds the calculated size of each new extent up to the next multiple of the data block size.

MINEXTENTS specifies the total number of extents to allocate when the object is created. This parameter enables you to allocate a large amount of space when you create an object, even if the space available is not contiguous. The default and minimum value is 1, meaning that Oracle only allocates the initial extent, except for rollback segments for which the default and minimum value is 2. The maximum value depends on your operating system.

If the MINEXTENTS value is greater than 1, then Oracle calculates the size of subsequent extents based on the values of the INITIAL, NEXT, and PCTINCREASE parameters.

MAXEXTENTS specifies the total number of extents, including the first, that Oracle can allocate for the object. The minimum value is 1 (except for rollback segments, which always have a minimum value of 2). The default and maximum values depend on your data block size.