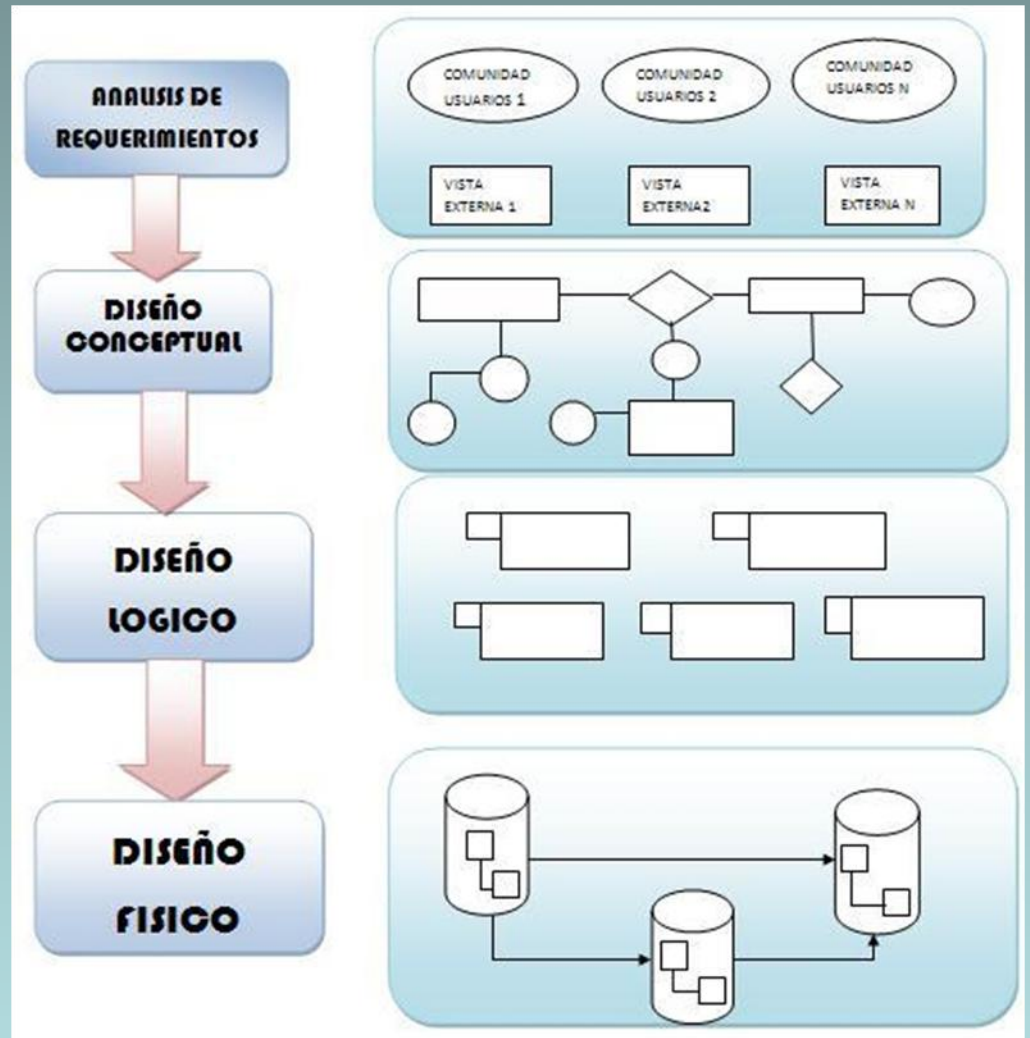


UT4- DISEÑO FÍSICO DE BASES DE DATOS.

LENGUAJE DDL

1-Introducción

- Nos encontramos en la última fase:



1.1-BD (Software)

- Es un programa residente en memoria, que se encarga de gestionar todo el tratamiento de entrada, salida, protección y elaboración de la información de interés del usuario

1.2 Tipos de BD

- Desde el punto de vista de la organización lógica:
 - a) Jerárquicas. (Progress)
 - b) Relacionales. (Oracle, Access, Mysql...)
- Desde el punto de vista de número de usuarios:
 - a) Monousuario (dBase, Access, Paradox...)
 - b) Multiusuario cliente/servidor (Oracle, Mysql...)
- Oracle es una base de datos relacional para entornos cliente/servidor.

1.3 Conocimientos necesarios

- Para un programador de bases de datos el conocimiento mínimo debe ser de:
 - Conocimiento básico de las estructuras internas de Oracle.
 - Lenguaje SQL
 - Utilidades básicas: (Toad, Export, Import...)
 - PL/SQL
 - Tareas simples de administración

1.4 Componentes de las bases de datos ORACLE

- Motor: el programa ejecutable que debe estar en memoria para manejar la base de datos. (Levantado o Bajado)
- Servicio de red: Es un programa que se encarga de establecer las conexiones
- Listener (escuchador): Es un programa residente en memoria que se encarga de recibir las llamadas que llegan a la base de datos desde la red, y de pasárselas a esta
- Configuración. Tsnames.ORA

2- SQL

- El lenguaje de consulta estructurado o SQL (por sus siglas en inglés *structured query language*) es un lenguaje declarativo (decimos lo que debe hacer pero no como) de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en estas.

2.1 TIPOS

- DDL: Lenguaje Definición de Datos
- DML: Lenguaje de Manipulación de Datos.
Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo efectuar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre ella.
- DCL: Lenguaje de Control de Datos

2.2 Elementos Básicos en ORACLE

- Palabras reservadas
- Objetos de la base de datos
- Tipos de datos
- Literales
- Comentarios

2.2.1 Objetos de la BD

- Podemos distinguir dos opciones:
 - a) Objetos que no pertenecen a un esquema
 - b) Objetos que pertenecen a un esquema
 - Las palabras esquema (schema) y usuario (user) son equivalente en Oracle.
 - Cada esquema en Oracle es propiedad de un único usuario, con el mismo nombre que el esquema.
 - Ej: Podríamos encontrar esquemas (o usuarios) para contabilidad, recursos humanos... Cada uno de estos esquemas contendrá los objetos de su incumbencia (tablas, procedimientos...)

Esquema/Usuario

- Normalmente se usa la denominación esquema cuando contiene tablas y procedimientos comunes para muchos usuarios, y la palabra usuario cuando los objetos del esquema son más privados. Siguiendo el ejemplo anterior, “contabilidad” sería un esquema, y cada uno de los programadores que trabajen sobre las tablas de contabilidad sería un usuario (con su propio esquema).
- Inicialmente, los usuarios/esquemas únicamente pueden acceder a sus propios objetos, salvo que otros usuarios (o un usuario con privilegios del sistema) le otorguen permisos para poder acceder a sus objetos

2.2.1.1 Objetos que no pertenecen a un esquema

- Contextos
- Directorios
- Ficheros de parámetros
- Perfiles
- Roles
- Segmentos de Rollback
- Tablespaces y datafiles
- Usuarios/Esquemas

2.2.1.2 Objetos que pertenecen a un esquema

- Clusters
- Constraints
- Database links
- Triggers
- Dimensiones
- Tablas organizadas por índice
- Índices
- Objetos Java
- Vistas materializadas
- Objetos
- Tablas de Objetos
- Vistas de objetos
- Operadores
- Paquetes PL/SQL
- Secuencias
- Procedimientos y funciones almacenadas PL/SQL
- Sinónimos
- Tablas
- Vistas

2.2.2 Tipo de Datos

- En la implementación SQL de Oracle se pueden usar muchos tipos de datos, a la hora de definir objetos de la base de datos

2.2.2.1 Valor NULL

- Un NULL es un valor no disponible, no asignado, desconocido o no aplicable.
- Un NULL no tiene tipo de datos asociado, puesto que no es nada, pero cualquier columna de un tipo de datos dado puede contener un NULL, es decir, puede contener la ausencia de valor de ese tipo de datos.
- En relación con el tipo de datos de la columna, cualquier columna puede contener NULL. Las restricciones NOT NULL, y PRIMARY KEY imponen a la columna sobre las que se aplican la obligatoriedad de no admitir NULL como posible valor.
- Por defecto toda columna o variable es NULL.

2.2.2.2 Tipos en Oracle

Tipos de datos Oracle				
Alfanuméricos	Numéricos	Fecha	Binarios	Otros
CHAR VARCHAR2 VARCHAR NCHAR NVARCHAR2 LONG	NUMBER FLOAT	DATE	RAW LONG RAW BLOB CLOB NLOB BFILE	ROWID

2.2.2.2.1 Los más usados

- Alfanuméricos:
 - **VARCHAR2(n)**. Cuando se declara un dato de este tipo, éste puede contener caracteres (codificados según el juego de caracteres de la sesión) hasta una longitud total de n posiciones. La longitud máxima para las columnas de este tipo de datos es de 4000 posiciones. Es obligatorio especificar la longitud de una columna varchar2
- Numérico:
 - **NUMBER(p, s)**.
 - Donde p indica el número total de dígitos (hasta 38),
 - s indica la escala, o número de dígitos a la derecha del punto decimal (hasta 127).
 - Si al almacenar un valor en una columna de tipo number se excede la precisión (número de dígitos), Oracle da un error, si se excede la escala (más decimales de los que soporta) Oracle redondea el valor a la capacidad de la columna.

- Fecha:
 - **Date:** Una columna de tipo date almacena información sobre fecha y hora. Para cada dato de tipo date, Oracle almacena información sobre siglo, año, mes, día, hora, minuto y segundo.
 - Se puede especificar un valor de fecha mediante un literal (que Oracle intentará convertir) o mediante la función TO_DATE.
 - '02/01/1975'
 - TO_DATE('20/04/2003 09:20','DD/MM/YYYY HH:MI')

- **Lob:** Los tipos de datos LOB (Large OBject), se usan para almacenar datos de gran tamaño, como ficheros (BFILE), texto (CLOB), vídeo o imágenes (BLOB).
- **ROWID:** Toda fila de toda tabla tiene una dirección o identificación única en Oracle, conocida como ROWID

2.2.3 Literales

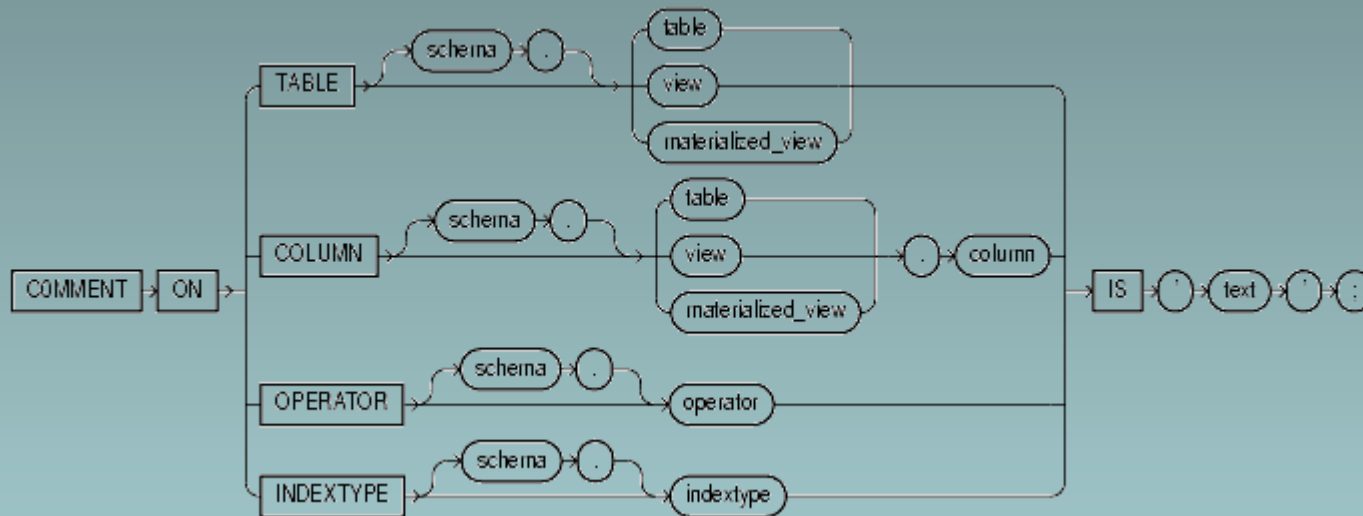
- Los literales de texto en SQL de Oracle están compuestos por una cadena de caracteres entrecomillados
- 'Hola', '1900'
- Si hay que poner comilla simple se pone otra delante:
- 'L''apòstrof'

2.2.4 Comentarios

- `--comentario de una línea.`
- `/* comentario de
varias líneas*/`

Lo que vaya dentro de comentario no se tiene en cuenta para nada. Suelen ser cosas aclarativas de lo que estás haciendo.

2.2.4.1 Comentar Tablas



--comentario sobre tabla

COMMENT ON TABLE DAM1.DEPA_DEPARTAMENTOS IS

'Tipos de departamentos que pueden existir en los centros de los almacenes.'

--comentario sobre columna

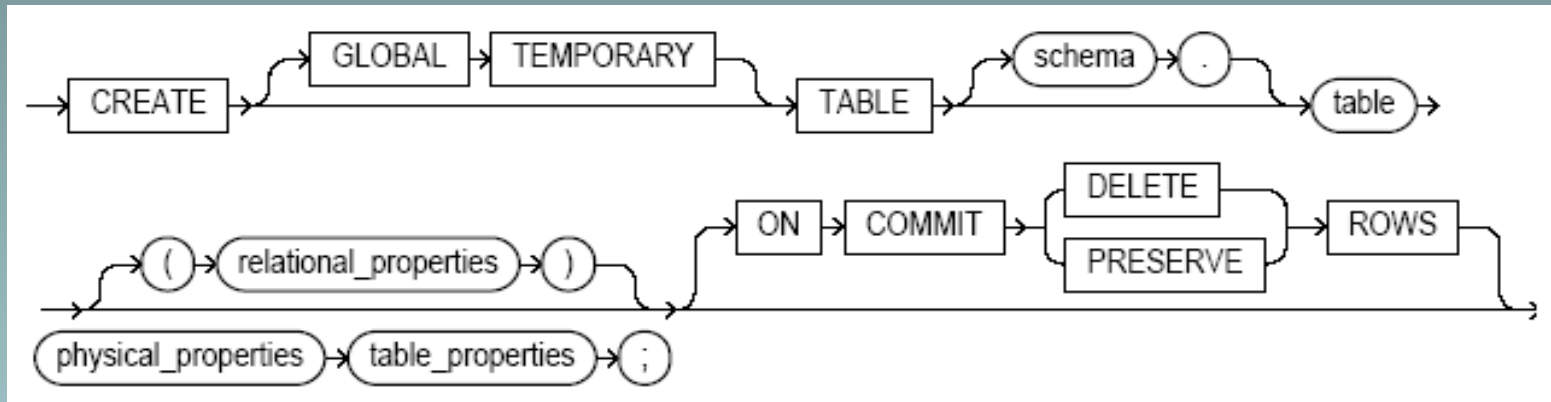
COMMENT ON COLUMN DAM1.DEPA_DEPARTAMENTOS.COD_DEPA IS

'Clave primaria alfanumérica.'

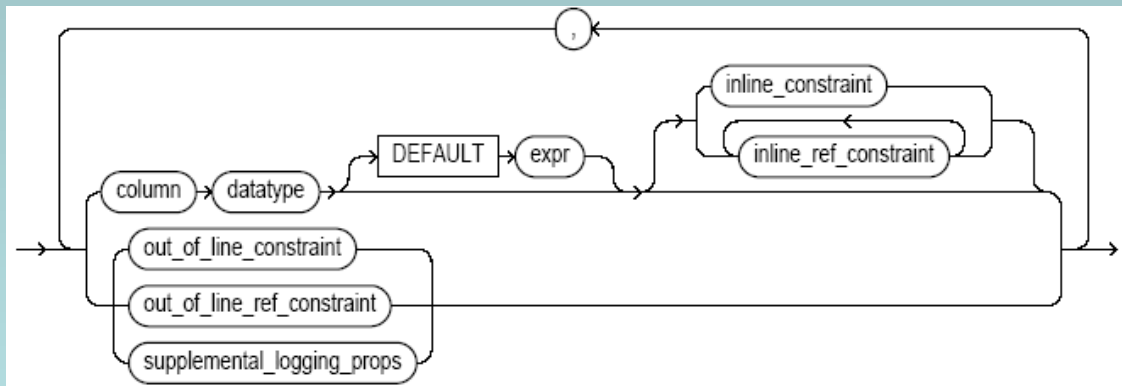
3 - DDL

- Son sentencias que se usan para la definición de objetos de la base de datos. Entre estas sentencias están las de creación (create), modificación (alter, replace) y borrado (drop) de objetos de la base de datos y las usadas para proporcionar integridad y consistencia a la base de datos (constraints y checks).
- Todas estas instrucciones actualizan y mantienen lo que se denomina el diccionario de datos (tablas que contienen información sobre los objetos de la base de datos).

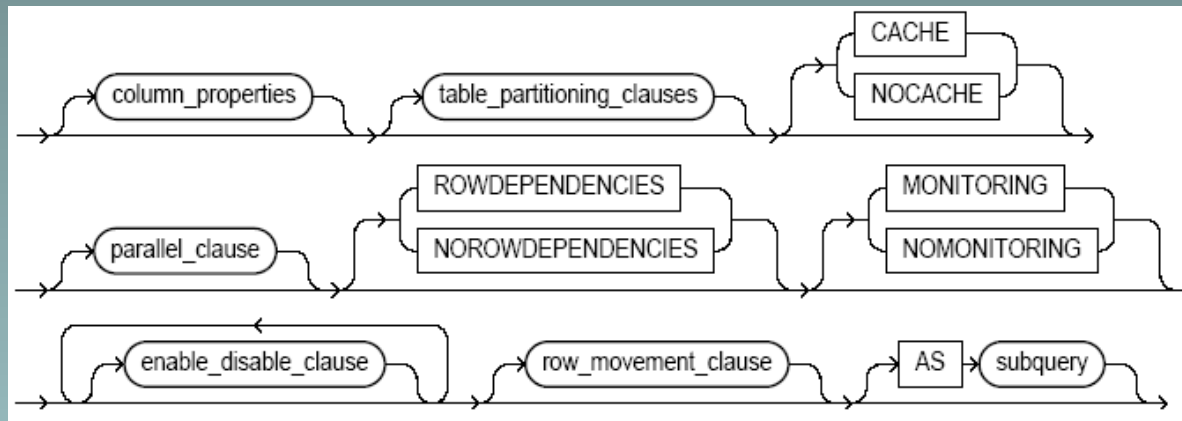
3.1 Create Table



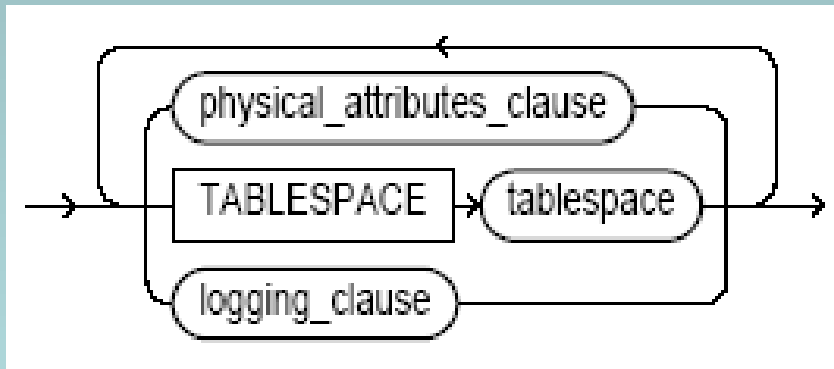
Relational_properties:



- Tabla properties



- *Physical_properties*



```

CREATE TABLE nombre_tabla(
    COLUMNA TIPO
        [NOT NULL]
        [UNIQUE]
        [PRIMARY KEY]
        [DEFAULT valor]
        [REFERENCES Nomtabla][columna[,columna]][ON DELETE
            CASCADE]]
        [CHECK condicion],
    COLUMNA TIPO [NOT NULL],
    . . .
{CONSTRAINT nombre_clave_primaria PRIMARY KEY (columnas_clave)}
{CONSTRAINT nombre_clave_foránea
    FOREIGN KEY(columnas_clave) REFERENCES tabla_detalle(
        columnas_clave )
    {ON DELETE CASCADE} }
)
{TABLESPACE tablespace_de_creación}
{STORAGE( INITIAL XX{K|M} NEXT XX{K|M} )}

```

```
CREATE TABLE FACTURA(  
    REFERENCIA VARCHAR2(10) NOT NULL,  
    DESCRIPCION VARCHAR2(50),  
    C_PAIS NUMBER(3),  
    C_CLIENTE NUMBER(5),  
    IMPORTE NUMBER(12),  
    CONSTRAINT PK_FACTURA PRIMARY KEY( REFERENCIA ),  
    CONSTRAINT FK_CLIENTE FOREIGN KEY(C_PAIS,C_CLIENTE) REFERENCES  
    CLIENTE( C_PAIS, C_CLIENTE)  
    ON DELETE CASCADE  
)  
TABLESPACE tab_facturas  
STORAGE( INITIAL 1M NEXT 500K );
```

ALTER TABLE

- Sirve para modificar cualquier objeto existente en la base de datos.
- Los objeto que más se modifican son las tablas, los índices y los objetos PL/SQL.

ALTER. Sintaxis

- ALTER TABLE nombretabla
 - [ADD (columna[,columna]...)
 - [MODIFY (columna[,columna]...)
 - [DROP COLUMN (columna[, columna]...)
 - [ADD CONSTRAINT restriccion]
 - [ADD CONSTRAINT restriccion];

ALTER. Ejemplos

```
--añade dos columnas a una tabla
ALTER TABLE KTMP_VALORES
ADD (ANYO_CONTABLE_VALO number(4),
     REF_CONTABLE_VALO  number(20))
```

```
--modifica la definición de columnas de una tabla
ALTER TABLE reov_registro_op_valor
MODIFY (fecha_operacion_reov NOT NULL,
        id_fiin number(12))
```

```
--añade una restricción a una tabla
ALTER TABLE ktmp_temporal
ADD CONSTRAINT FK_KTMP_EPER_EPER FOREIGN KEY(id_eper)
REFERENCES EPER_ELEMENTOS_PERSONA(id_eper)
```

ALTER TABLE. Restricciones

- La integridad referencial es una herramienta imprescindible de las bases de datos relacionales. Pero provoca varios problemas. Por ejemplo, si borramos un registro en la tabla principal que está relacionado con uno o varios de la secundaria ocurrirá un error, ya que de permitírse nos borrar el registro ocurrirá fallo de integridad (habrá claves secundarios refiriéndose a una clave principal que ya no existe).
- Por ello Oracle nos ofrece dos soluciones a añadir tras la cláusula REFERENCES:
 - ❑ **ON DELETE SET NULL.** Coloca nulos todas las claves secundarias relacionadas con la borrada.
 - ❑ **ON DELETE CASCADE.** Borra todos los registros cuya clave secundaria es igual que la clave del registro borrado.
- Existe también la opción de ON UPDATE CASCADE o ON UPDATE SET NULL, pero ORACLE no las implementa.

DROP

- Sirve para ELIMINAR objetos de la base de datos.
- Los objetos más comúnmente eliminados son índices y tablas temporales.

DROP. Ejemplos

--elimina un índice

DROP INDEX IX_VALO_REF_EXTERNA

--elimina una tabla

DROP TABLE ktmp_temporal

--elimina un usuario o esquema

DROP USER ktmp_temporal

Junto con DROP se puede especificar la cláusula **CASCADE CONSTRAINTS**, lo haría que eliminara todos los objetos que hicieran referencia al objeto eliminado, convirtiendo esta instrucción en algo muy peligroso, que sólo debe ejecutar el DBA, salvo en tablas e índices temporales, que podrían ser eliminados por usuarios.

TRUNCATE

Esta instrucción es un híbrido entre instrucciones DDL y DML. Sólo se ejecuta sobre tablas y no modifica la definición de las mismas en cuanto a estructura, pero elimina todos los datos de la tabla sobre la que se ejecuta. TRUNCATE emite automáticamente una instrucción COMMIT, y libera el espacio ocupado por la tabla en el sistema de ficheros reduciéndolo al inicial.

```
TRUNCATE TABLE TMP_FRACCIONES
```

SECUENCIAS

- Una secuencia (sequence) es un objeto de base de datos que genera números secuenciales.

SECUENCIAS. Sintaxis

- Sintaxis:
 - CREATE SEQUENCE nombre_secuencia
 - {START WITH entero}
 - {INCREMENT BY entero}
 - {MAXVALUE entero | NOMAXVALUE}
 - {MINVALUE entero | NOMINVALUE }
 - {CYCLE | NOCYCLE};

CREATE INDEX

- CREATE {UNIQUE} INDEX
nombre_índice
 - ON tabla(columnas_indexadas)
 - {TABLESPACE tab_indices}
 - {STORAGE(INITIAL XX{K|M} NEXT XX{K|M}
))}