



Universidad de Granada

[decsai.ugr.es](http://decsai.ugr.es)

# **Fundamentos de Bases de Datos**

Grado en Ingeniería Informática

**Seminario 3: SQL. Índices y optimización**



**DECSAI**

**Departamento de Ciencias de la  
Computación e Inteligencia Artificial**

- 1. Índices**
- 2. Vistas**
- 3. Clusters**



- 1. Índices**
2. Vistas
3. Clusters



### La sentencia CREATE INDEX

#### Sintaxis:

```
create [unique] index indice on { tabla ( columna[asc|desc],[  
columna[asc|desc]] ...) | cluster )  
[initrans n] [maxtrans n] [tablespace tablespace] [storage  
storage] [pctfree n] [nosort]
```

- **unique** significa que el valor de la clave verifica una condición de unicidad
- **nosort** significa que no hay que ordenar las filas cuando se crea el índice
- Se pueden crear varios índices por tabla
- Permiten mejorar las consultas cuando se accede a la tabla ordenada según el campo clave del índice y cuando consulta según dicho campo
- Permite crear índices compuestos de hasta 16 componentes
- Por defecto el orden es ascendente
- Los índices pueden ralentizar la actualización de las tablas

### La sentencia CREATE INDEX

#### Índices de clave invertida:

Invierten el orden de los bytes de la clave. Optimizan el rendimiento del acceso secuencial en configuraciones paralelas de Oracle.

#### Sintaxis:

```
create [unique] index indice on {tabla
(columna[asc|desc],[ columna[asc|desc]] ...) | cluster}
reverse
```

### La sentencia CREATE INDEX

#### Índices de mapa de bits (Bitmap).

Solo funcionan bien en atributos categóricos y son especialmente útiles cuando el dominio es pequeño

#### Sintaxis

```
create bitmap index indice on {tabla
(columna[asc|desc],[ columna[asc|desc]] ...) | cluster}
```

### La sentencia CREATE INDEX

#### Tablas organizadas por índices

Son tablas que están organizadas como arboles B de forma que las hojas de los arboles son la tuplas. Esta forma de la tabla se debe indicar como una clausula adicional en la sentencia CREATE TABLE

#### Sintaxis

```
CREATE TABLE [usuario.]nombre_tabla
    ({datos_columna | restricciones de tabla}
    [{datos_columna | restricciones de tabla}]...)
```

#### ORGANIZATION INDEX

La tabla debe tener especificada una llave primaria

***En el cuaderno de prácticas pueden encontrarse ejemplos de uso de todos estos tipo de índices***

1. Índices
2. **Vistas**
3. Clusters





### La sentencia CREATE VIEW

#### Sintaxis:

create view *vista* [(*alias* [ ,*alias*] ... )] as *consulta* [with check option [constraint *restriccion* ] ]

- Los **alias** nos permiten renombrar todas las columnas de la vista
- La consulta nos permite construir una visión de usuario tan compleja como queramos. Solo se impide la clausula “order by”
- “with check option” proporciona restricciones adicionales para la actualización mediante vistas

### La sentencia CREATE VIEW

- Una vista puede aparecer en cualquier sentencia “select”.
- Una vista puede ser objetivo en una sentencia de actualización; pero hay que tener en cuenta los problemas que la actualización mediante vistas de usuario puede generar.

***En el cuaderno de prácticas pueden encontrarse ejemplos de uso de todos estos tipo de índices***

1. Índices
2. Vistas
3. **Clusters**



### La sentencia CREATE CLUSTER

#### Concepto de Cluster:

- Un “cluster” es una forma de almacenamiento en la que se almacenan juntas la tuplas de distintas tablas que comparten uno o varios campos comunes y se consultan juntas.
- Los cluster se pueden indexar o crear mediante tablas hash

#### Ejemplo

- Si se van a consultar siempre conjuntamente (sacar listas de alumnos), la tabla asignaturas y la tabla matrícula se pueden almacenar juntas a través del campo código de asignatura.

Cada ocurrencia de asignatura se almacenaría conjuntamente con las ocurrencias de la tabla matrícula que le corresponden

### La sentencia CREATE CLUSTER

#### Sintaxis :

```
CREATE CLUSTER [usuario].cluster
(columna tipo_de_dato [, columna tipo_de_dato ]...)
[PCTFREE n], [PCTUSED n], [INITRANS n ],
[MAXTRAN n] [TABLESPACE nombre], [STORAGE
nombre],[SIZE n]
```

### La sentencia CREATE CLUSTER: Clusters indexados

**Ejemplo:** Creación del cluster e inserción de las tablas asigna y matricula

```
create cluster listas(asi# varchar(4))
create table asigna (asi# varchar(4) primary key ,
    ...)
    cluster listas(asi#)
```

```
create table matricula(asi# varchar(4) references
    asigna,
    ...)
    cluster listas(asi#)
```

**Creación del índice asociado:**  
 create index idx\_listas on cluster listas

### La sentencia CREATE CLUSTER: Clusters Hash

#### Sintaxis :

```
CREATE CLUSTER [usuario].cluster
(columna tipo_dato) [HASH is columna] SIZE <tamaño>
HASHKEY <cantidad_valores_distintos_de_la_clave>
```

La clausula **HASH** se usa cuando la clave de cluster es un valor entero uniformemente distribuido. En caso contrario ORACLE aplica su algoritmo de direccionamiento

**SIZE** mide el tamaño en bytes del espacio que van a ocupar las tuplas con del mismo valor de clave:

- Hay que tener en cuenta las tuplas de las dos tablas

- Hay que prever si van a haber colisiones

- Se debe estimar un 15% adicional.

**HASH KEY** estima cuantos valores distintos va a tomar la clave del cluster