

## HOJA 3 DE EJERCICIOS.

### Ejercicio 3.1

Para una aplicación se han previsto 200 gigas de almacenamiento en disco. Justo después de la carga inicial se ocuparán 100 y, después, cada mes se espera un crecimiento de 10 gigas. A final de año se realizarán operaciones con la base de datos que darán como resultado el vaciado de una gran parte de la información almacenada. Un mes antes de poner en marcha la aplicación el distribuidor de los discos sugiere que, probablemente, solo sea capaz de suministrar 150 gigas al poner en marcha la aplicación, y el resto tres meses después. Dado que la previsión de crecimiento está dentro de esos márgenes opina que no hay motivo de preocupación. ¿Es correcto este planteamiento desde el punto de vista de administración? Suponemos que no hay un sistema RAID o un sistema automático de reconfiguración. Como cambiaría esto el enfoque?

### Ejercicio 3.2

En una aplicación que lleva funcionando un año los usuarios se quejan de que “de un tiempo a esta parte la aplicación va cada vez más lenta”. Los administradores responden que no hay razón para que ocurra eso ya que el nivel de transacciones se ha mantenido constante durante el último año, y si antes no iba lenta ahora no tiene por que ir. ¿Tiene fundamento esta respuesta?

### Ejercicio 3.3

Una base de datos modela una relación 1:N entre dos entidades A (1) y B (N) en una base de datos relacional mediante dos tablas. Con objeto de mejorar el rendimiento hay dudas si desnormalizar (toda la información relativa a A y B estará en una sola tabla) o crear una estructura cluster (no un índice cluster). Sabemos que la volatilidad de B es muy alta, frente a la de A que es baja. ¿Cuál sería la decisión en función de la volatilidad de B? Dar una razón al menos.

### Ejercicio 3.4

Dado el siguiente esquema de base de datos

```
CREATE TABLE PROPIETARIO (  
    P# Integer NOT NULL,  
    Dni Char(8) NOT NULL,  
    Nombre Char(15) NOT NULL,  
    Apellidos Char(20) NOT NULL,  
    PRIMARY KEY P#);  
  
CREATE TABLE CASAS (  
    P# Integer NOT NULL,  
    Calle Char(8) NOT NULL,  
    Numero Char(15) NOT NULL,  
    Piso Char(20) NOT NULL,  
    Localidad Char(30) NOT NULL,  
    Ciudad Char(25) NOT NULL,  
    ValorCatastral Integer,  
    PRIMARY KEY (Calle, Numero, Piso, Localidad, Ciudad),  
    FOREIGN KEY (P#) REFERENCES PROPIETARIO (P#));
```

Se pide:

- Indicar como quedaría el esquema físico si aplicáramos una técnica de *clustering* (agrupamiento) a estas dos tablas. ¿En qué casos sería aconsejable utilizar esta técnica?
- Con objeto de mejorar el rendimiento en un sistema de información, se plantea realizar una desnormalización en el esquema de la base de datos que forma parte de dicho sistema. ¿Significa esto modificar el diseño físico? ¿Por qué?
- Suponiendo que actualmente la BD ocupa 10000 páginas de 2048 bytes, ¿podemos asegurar que al utilizar páginas de 1024 bytes, la base de datos pasará a ocupar 20000 páginas?
- Si la consulta más frecuente que se realizase sobre estas dos tablas fuese:

```
SELECT Dni, apellidos, nombre, calle, numero, piso, localidad
FROM Propietarios, Casas
WHERE Propietarios.P# = Casas.P# And
      Ciudad = 'Madrid'
ORDER BY Localidad, Calle
```

¿Recomendaría algún índice para acelerar esta consulta? ¿Cuál? y ¿por qué?

### Ejercicio 3.5

Para una BD del Ayuntamiento de Alpepete se han definido las siguientes tablas:

```
CREATE TABLE PROPIETARIO (
  P#          Integer NOT NULL,
  Dni         Varchar (8) NOT NULL,
  Nombre      Varchar (16) NOT NULL,
  Apellidos   Varchar (22) NOT NULL,
  PRIMARY KEY P#);

CREATE TABLE CASAS (
  P#          Integer NOT NULL,
  Calle       Varchar (30) NOT NULL,
  Numero      Integer NOT NULL,
  ValorCatastral Integer,
  PRIMARY KEY (Calle, Numero));
```

Se pide:

- La tabla propietario tiene actualmente 200.000 filas, suponiendo que la cabecera y pie de página juntos ocupan 24 bytes, y que la cabecera de registro ocupa 50 bytes, determinar el número de páginas necesarias para almacenar la tabla, para un tamaño de página de 1 Kbyte. El tipo de datos Integer utiliza 4 bytes.
- ¿Cuántas páginas de fichero índice primario serán necesarias, para un índice con 20 bytes para el rango de claves y 10 para el direccionamiento de páginas, manteniéndose el mismo tamaño cabecera y pie de página que para los datos y con 20 bytes de cabecera de registro índice?
- ¿Cuántas páginas de fichero índice secundario serán necesarias, para un índice con 10 bytes para la clave y 10 para la dirección, manteniéndose el mismo tamaño de cabecera y pie de página que para los datos y con 30 bytes de cabecera de registro índice?
- ¿Qué índices recomendarías? y ¿por qué?