

Índice

Normas del examen	2
Ejercicio 1 : <i>Scripts</i> de inicio y parada de Oracle (1 punto)	2
Ejercicio 2 : Arrancar automáticamente Oracle cuando se inicie el servidor (2 puntos)	2
Ejercicio 3 : Crea usuarios de base de datos (2 puntos)	2
Ejercicio 4 : Almacena información periódicamente en la base de datos (4 puntos)	3
Ejercicio 5 : Envía un correo periódicamente (2 punto)	4
Instrucciones de entrega	4

Normas del examen

Es difícil evaluar el manejo de *scripts* sin realizar un examen en el ordenador, pero también es difícil condensar un examen en solo dos horas. Por eso, el examen se plantea como una práctica, que el profesor corregirá en clase.

El desarrollo de este ejercicio será como el de otras prácticas. La única diferencia está en que la nota de esta práctica se tendrá en cuenta en el apartado *exámenes* en vez de en el apartado *prácticas* al calcular la calificación del trimestre.

Ejercicio 1 : *Scripts* de inicio y parada de Oracle (1 punto)

Crea dos *scripts* para iniciar y parar **Oracle** en `/home/alumno/scripts/oraclestart.sh` y `/home/alumno/scripts/oraclestop.sh`

Ejercicio 2 : Arrancar automáticamente Oracle cuando se inicie el servidor (2 puntos)

- **Oracle** debe levantarse cuando la máquina se inicie, y apagarse cuando la máquina se cierre.
- Oracle se iniciará solo si se indica en el fichero `/etc/oratab`
- En el fichero `/home/alumno/logs/oracle.log` se dejará una traza de cuando se arrancó y se paró la máquina, y si fue necesario arrancar o parar *Oracle*. Por ejemplo:

5

```
2017-02-10-12:40:00 - Máquina arrancada
2017-02-10-12:40:01 - Oracle arrancando porque /etc/oratab indica Y
2017-02-10-12:40:20 - Oracle arrancado
2017-02-10-12:41:00 - Máquina parando
2017-02-10-12:41:01 - Oracle parando
2017-02-10-12:41:20 - Oracle parado
```

Listado 1: Ejemplo de `/home/alumno/logs/oracle.log` cuando **Oracle** se arranca

5

```
2017-02-10-12:40:00 - Máquina arrancada
2017-02-10-12:40:01 - Oracle no se arranca porque /etc/oratab indica N
2017-02-10-12:41:00 - Máquina parando
2017-02-10-12:41:01 - Oracle parando
2017-02-10-12:41:10 - Oracle parado
```

Listado 2: Ejemplo de `/home/alumno/logs/oracle.log` cuando **Oracle** no se arranca

Ejercicio 3 : Crea usuarios de base de datos (2 puntos)

Crea un script de nombre `/home/alumno/scripts/nuevo-usuario-oracle.sh` que cree un nuevo usuario de oracle. Si se invoca sin parámetros, o con más de dos, mostrará el texto de ayuda del listado 3

```
Crea un usuario nuevo de oracle, con permisos connect y resource.
Si el usuario ya existe, lo desbloquea y le cambia la contraseña.
```

```
Uso: nuevo-usuario-oracle.sh <usuario> <contraseña>
```

Listado 3: Ayuda del script nuevo-usuario-oracle.sh

Ejercicio 4 : Almacena información periódicamente en la base de datos (4 puntos)

Programa un *script* para que cada minuto almacene en la tabla DF la información del comando `df -k`. Esta tabla (listado 4) tendrá como columnas:

- hora: Hora de lanzamiento del comando
- sistema: Nombre del tipo de sistema de ficheros
- tamaño: Tamaño en KB del sistema de ficheros
- usado: Tamaño usado, en KB
- montado: Punto de montaje

```
5 create table DF(
    hora varchar(40),
    sistema varchar(40),
    tamaño varchar(40),
    usado varchar(40),
    montado varchar(40)
);
```

Listado 4: Creación de la tabla DF

```
5 Filesystem      1K-blocks      Used Available Use% Mounted on
udev              4002180          0   4002180   0% /dev
tmpfs              804488       19756    784732   3% /run
/dev/sda1        237874840 183034916 42733532 82% /
10 tmpfs           4022440     437328   3585112 11% /dev/shm
tmpfs              5120           4        5116   1% /run/lock
tmpfs           4022440          0   4022440   0% /sys/fs/cgroup
/dev/sdb5        689521880 595546232 58926896 91% /home/windows
cgmfs              100           0         100   0% /run/cgmanager/fs
10 tmpfs           804488         88    804400   1% /run/user/1000
```

Listado 5: Ejemplo de salida del comando `df -k`

Pistas para realizar el *script*:

- Los *heredocs* pueden contener variables: <http://superuser.com/questions/456615/how-to-pass-variables-to-a-heredoc-in-bash>
- Cortar columnas con `awk`: <https://www.cyberciti.biz/tips/processing-the-delimited-files-using-cut-and-awk.html>

- Leer líneas una por una y meterlas en una variable: <http://stackoverflow.com/questions/10929453/read-a-file-line-by-line-assigning-the-value-to-a-variable>
- Quitar la primera línea de la salida de `df -k` con el comando `tail`: <https://linuxconfig.org/commands-on-how-to-delete-a-first-line-from-a-text-file-using-bash-shell>
- El *script* debería seguir los siguientes pasos:
 1. Quitar la primera línea de la salida de `df -k` con `tail -n +2`
 2. Leer cada línea con `while` y `read`
 3. Sacar los campos de cada línea con `awk`
 4. Ejecutar una sentencia SQL con los datos extraídos

Ejercicio 5 : Envía un correo periódicamente (2 punto)

- Programa un *script* para que cada minuto envíe un correo con la información promedio del comando `df -k`. Puedes usar como base para la consulta el listado 6.
- El correo se enviará a `alvarogonzalez.profesor@gmail.com`
- Con copia a `alvaro@alvarogonzalez.no-ip.biz`
- Indica tu nombre en el asunto del correo

```
5 select
    sistema, avg(tamano), avg(usado), montado
  from
    DF
  group by
    sistema, montado;
```

Listado 6: Consulta tipo para extraer información promedio

Instrucciones de entrega

- El ejercicio se realizará y entregará de manera individual.
- El profesor comprobará el funcionamiento del sistema, hasta el día 14 de Febrero.
- No importa que el correo llegue a `alvarogonzalez.profesor@gmail.com`, lo que importa es que se envíe (el profesor mirará los ficheros de traza para comprobarlo)