LO3 Bases de Datos

Índice

[**P4**](#_x61pb88edj2h) **2**

[**M4**](#_afn2i4h533tn) **7**

[**D2**](#_mh6rxonsnd70) **10**

[**Bibliografía**](#_70lv20v2ts7e) **11**

## 

## 

## P4

**Testear el sistema de bases de datos con respecto a las cuentas de**

**usuario y los requisitos del sistema operativo.**

**1. Resuelva y explique las diez consultas presentadas en el P2 apartado 5 (y cinco consultas más, elegidas entre el conjunto de consultas resueltas en los distintos apartados anteriores) con una Tabla de valor introducido, valor esperado, y valor real devuelto; suponiendo que tiene privilegios de Administrador.**

| **Consulta introducida** | **Valor esperado** | **Valor devuelto** |
| --- | --- | --- |
| Listado de propietarios con su nombre, apellidos y el número total de viviendas que poseen:  SELECT p.nombre, p.ape1, p.ape2, COUNT(v.vivienda\_id) AS total\_viviendas  FROM propietario p  LEFT JOIN vivienda v ON p.propietario\_id = v.id\_propietario  GROUP BY p.propietario\_id  ORDER BY total\_viviendas DESC; | Una tabla que contenga el nombre nombre y apellidos de cada propietario junto al número de viviendas que tiene cada vivienda |  |
| Total de alquileres realizados por cada agencia inmobiliaria:  SELECT agencia.agencia\_id, COUNT(alquiler.alquiler\_id) AS total\_alquileres  FROM agencia  JOIN vivienda ON agencia.agencia\_id = vivienda.id\_agencia  JOIN alquiler ON vivienda.vivienda\_id = alquiler.id\_vivienda  GROUP BY agencia.agencia\_id; | Una tabla que indique el id de cada agencia y cuantos alquileres gestiona. |  |
| Informe sobre el promedio de importe mensual de los alquileres de viviendas en una determinada población:  SELECT AVG(a.importmes) as promedio\_alquiler  FROM alquiler a  left join vivienda v on a.alquiler\_id = a.id\_vivienda  WHERE cp LIKE '28%'; | Una tabla que calcule la media de importes mensuales de los alquileres dentro de la población de Madrid |  |
| Listado de viviendas con un determinado número de habitaciones, ordenadas por metros cuadrados:  SELECT v.vivienda\_id, v.id\_agencia, v.numhab, v.MC as viviendas\_ordenadas  FROM vivienda v  WHERE numhab = 2  ORDER BY v.MC; | Tabla llamada viviendas\_ordenadas que muestre la clave primaria y la foránea de agencia de las viviendas con dos habitaciones ordenadas por los metros cuadrados de manera ascendente. |  |
| Informe sobre el promedio de edad de los inquilinos en una determinada población:  SELECT AVG(DATEDIFF(CURDATE(), inquilino.fenac)) AS promedio\_edad  FROM inquilino  WHERE cp LIKE '28%'; | Tabla que muestre la media de edad de los inquilinos. No tenemos un dato que sea la edad, pero se pueden sacar los años con la diferencia del año de nacimiento y la fecha actual.La poblacion escogida es madrid |  |
| ¿Cuántos Propietarios tengo en la empresa? | Tabla que muestre el número de propietarios que hay registrados |  |
| Queremos un listado de los inquilinos, incluyendo su nombre, sus  apellidos y su teléfono de contacto (si lo tuviera), ordenado por  apellido. |  |  |
| Listado de alquileres que no sean renovaciones:  select a.alquiler\_id, a.renovacion, a.fecha\_inicio from alquiler a where renovacion is null; | Un listado o tabla en el que aparezca el id del alquiler junto a la renovación y fecha de inicio de cada uno |  |
| Informe sobre los alquileres, ordenados por su fecha de inicio:  select a.alquiler\_id, a.fecha\_inicio from alquiler a order by a.fecha\_inicio ASC; | Listado de los alquileres ordenados por su fecha de inicio donde aparezca la fecha de inicio y el id de cada alquiler ordenado de manera ascendente |  |
| Listado de agencias con todos sus datos ordenados según cuántas viviendas gestionan de manera descendente  SELECT agencia.\*, COUNT(\*) AS num\_alquileres  FROM agencia  JOIN vivienda ON agencia.agencia\_id = vivienda.id\_agencia  GROUP BY agencia.agencia\_id  ORDER BY num\_alquileres DESC; |  |  |

**2. Describa problemas con los perfiles propuestos en el apartado M2.2.**

Existen algunos problemas con los perfiles propuestos en la M2:

-Inquilino: El perfil de inquilino puede llegar a tener problemas de seguridad.Aunque se restrinjan tablas y algunos campos, puede tener información confidencial acerca de los contratos de alquiler y detalles de la propiedad. Además, no se especifica la forma de verificación de usuario para garantizar que solo accedan a su información.

-Agencia Inmobiliario:Este perfil también tiene problemas de seguridad ya que permite a la agencia acceder a datos de propietarios y alquileres que ellos no gestionan ya que no se especifica a qué registros de la tabla puede acceder y a cuáles no.

-DBA:El DBA tiene acceso a todos los datos, lo que lo convierte en el perfil más poderoso de la BBDD.Es importante que el DBA tenga fuertes barreras de seguridad antes de iniciar sesion como este, de lo contrario, si el DBa es malintencionado, puede causar grandes daños en la integridad y seguridad de los datos. Por lo que solo podrían ser personas de confianza y con una formación adecuada de la BBDD.

**3. Especifique características mínimas del Sistema informático, para dar cabida a la aplicación que se está generando para este supuesto, tanto a la parte Backend que se está acometiendo, como para la parte Frontend que irá asociada.**

Para la parte Backend, las características mínimas del sistema informático requeridas para dar cabida a la aplicación son las siguientes:

-Servidor web: debe contar con un servidor web compatible con el lenguaje de programación utilizado para el desarrollo de la aplicación. En este caso, si se está utilizando PHP, se puede utilizar Apache o Nginx.

-Base de datos: es necesario contar con un sistema de gestión de bases de datos compatible con el lenguaje de programación utilizado para el desarrollo de la aplicación. En este caso, si se está utilizando PHP, se puede utilizar MySQL o PostgreSQL.

-Sistema operativo: el servidor debe contar con un sistema operativo compatible con el servidor web y la base de datos utilizados. En general, se puede utilizar cualquier sistema operativo compatible con el lenguaje de programación utilizado.

-Recursos de hardware: el servidor debe contar con suficientes recursos de hardware, como memoria RAM y espacio de almacenamiento, para manejar la carga de trabajo esperada. Es importante considerar la escalabilidad del sistema en caso de que la carga de trabajo aumente en el futuro.

Para la parte Frontend, las características mínimas del sistema informático requeridas son las siguientes:

-Navegador web: el sistema debe ser compatible con los navegadores web modernos y populares, como Chrome, Firefox, Safari y Edge.

-Sistema operativo: el sistema debe ser compatible con los sistemas operativos comunes, como Windows, MacOS y Linux.

-Recursos de hardware: el sistema debe contar con suficientes recursos de hardware, como memoria RAM y procesador, para ejecutar la aplicación de manera fluida.

## 

## M4

**Asesorar sobre la eficacia del testeo, incluyendo una explicación sobre la manera que se lleva a cabo, de testear la base de datos.**

**Explicación de la etapa de pruebas de forma general en proyectos de desarrollo de aplicaciones.**

La etapa de pruebas se divide en diferentes fases y se llevan a cabo diferentes tipos de pruebas para asegurarse de que el software está listo para ser entregado. Algunas de las fases y tipos de pruebas que se pueden realizar son:

1. Pruebas unitarias: se realizan en el nivel de código para comprobar que cada unidad o módulo funciona correctamente y cumple con los requisitos establecidos.
2. Pruebas de integración: se realizan para comprobar que las diferentes partes del software funcionan correctamente en conjunto.
3. Pruebas de sistema: se realizan para comprobar que el software funciona correctamente en el entorno en el que será utilizado.
4. Pruebas de aceptación: se realizan para comprobar que el software cumple con los requisitos y expectativas del usuario final.
5. Pruebas de carga: se realizan para comprobar la capacidad de la aplicación para manejar grandes volúmenes de datos o usuarios simultáneos.
6. Pruebas de seguridad: se realizan para comprobar la seguridad del software y detectar posibles vulnerabilidades.

**Investigación y desarrollo sobre las distintas pruebas que se pueden realizar a las BBDD antes de ponerlas en producción.**

Antes de poner una base de datos en producción es importante realizar una serie de pruebas para garantizar su correcto funcionamiento y evitar posibles problemas en el futuro. A continuación se describen algunas de las pruebas más comunes que se realizan:

1. Pruebas de integridad: se realizan para comprobar que los datos ingresados en la base de datos son coherentes y cumplen con las restricciones definidas en las tablas y relaciones. Estas pruebas pueden incluir verificación de claves primarias y foráneas, comprobación de valores nulos o duplicados, entre otros.
2. Pruebas de rendimiento: se realizan para medir el tiempo de respuesta de la base de datos ante diferentes escenarios de uso, como la realización de consultas complejas o la actualización masiva de datos. Estas pruebas permiten identificar cuellos de botella y optimizar el rendimiento de la base de datos.
3. Pruebas de seguridad: se realizan para comprobar que la base de datos está protegida contra posibles ataques y vulnerabilidades. Estas pruebas pueden incluir comprobaciones de acceso no autorizado, encriptación de datos, permisos de usuarios y auditorías de registros.
4. Pruebas de recuperación ante fallos: se realizan para comprobar que la base de datos es capaz de recuperarse ante posibles fallos o interrupciones del sistema. Estas pruebas pueden incluir simulaciones de fallos de hardware, pérdida de conexión o fallos en la alimentación eléctrica.
5. Pruebas de migración: se realizan cuando se requiere migrar la base de datos a otro entorno o plataforma. Estas pruebas permiten comprobar que los datos y la estructura de la base de datos se han migrado correctamente y que el sistema sigue funcionando de manera adecuada.

**Proposiciones sobre pruebas a realizar en el diseño propuesto en el apartado M2.**

La proposición más básica para probar el diseño propuesto es hacer multitud de consultas a las bases de datos antes de lanzarla al público. Con cada una de estas queris esperamos un valor o un resultado, que sería el correcto, si este valor esperado es exactamente el mismo que recibimos significa que la base de datos cumple con su principal objetivo: Satisfacer la demanda de información con los datos introducidos en la tabla.

De lo contrario, si el valor esperado difiere del valor recibido puede significar dos cosas:

-Que la queri está mal formulada, por lo que habría que replantearse de nuevo hasta dar con la correcta para que la prueba en el diseño sea efectiva.

-Si estamos seguros que la queri está bien formulada y no recibimos el valor

esperado, debemos replantearnos el diseño de la base de datos hasta encontrar el fallo (quizá con otras consultas lograríamos encontrar el fallo) hasta encontrarlo y

solucionarlo.

La forma más efectiva de encontrar el fallo es hacer consultas que involucren varias tablas con el comando JOIN en sql, ya que respecto al diseño son los fallos más comunes.

## 

## 

## 

## 

## D2

**Evaluar la eficacia de la solución de bases de datos implementada con**

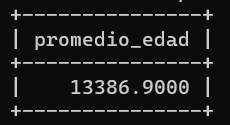
**respecto a los requerimientos de usuario y a los requisitos del sistema y**

**sugerir mejoras.**

**Identifique algunos problemas entre la solución propuesta en el apartado M2 y los datos obtenidos en el apartado P4. Describa, justifique y proponga una solución a cada uno de ellos.**

El único error encontrado es esta consulta:

-Informe sobre el promedio de edad de los inquilinos en una determinada población. El error consiste en que proporciona un número que no se adecua con el esperado. Se esperaría una media de edad (un número entero que referencie a los años)como por ejemplo, “23 ” y la consulta nos devuelve lo siguiente:



Y nos devuelve lo que parece ser un número decimal que aparentemente no tiene nada que ver con una media de edad de personas.

Lo que propongo para solucionar este error es hacer un nuevo campo que guarde la diferencia entre la fecha de nacimiento de los inquilinos y la fecha actual y que se vaya actualizando diariamente. Aparte, en el momento de hacer la consulta se cogen solo los años al hacer la diferencia entre estos dos datos y hacer la media con los números enteros resultantes, de forma que la respuesta a la consulta sería un número entero que representase los años.

## Bibliografía

<https://docs.microsoft.com/es-es/sql/sql-server/install/hardware-and-software-requirements-for-installing-sql-server?view=sql-server-ver15>

<https://dev.mysql.com/doc/mysql-installation-excerpt/8.0/en/system-requirements.html>

<https://www.occamagenciadigital.com/blog/conoce-las-fases-de-un-proyecto-de-desarrollo-de-software>

<https://www.atlassian.com/es/continuous-delivery/software-testing/types-of-software-testing>

<https://www.ibm.com/docs/es/was-zos/8.5.5?topic=servers-testing-production-phases>

<https://learn.microsoft.com/es-es/ef/core/testing/testing-with-the-database>