**UNIVERSIDAD TECNOLOGICA DE PANAMA**

**FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES**

**EVALUACION DE PROYECTO 2**

**Base de Datos 1**

|  |  |
| --- | --- |
| **Tema: Oracle, MySQL** | |
| **INTEGRANTES** | **NOTA FINAL** |
| **Cedeño Kevin** | **97** |
| **García Juan** | **97** |
| **Lo Yui** | **97** |
| **Zarate Fernando** | **97** |

|  |  |  |  |
| --- | --- | --- | --- |
| **TRABAJO ESCRITO 50%** | PONDERACIÓN | PUNTOS OBTENIDOS | Observaciones |
| * Índices (contenido, figuras, cuadros) | 2 | 2 |  |
| * Introducción | 3 | 3 |  |
| * Resumen Ejecutivo | 5 | 2 | El resumen parece una introducción. No contiene lo medular del trabajo desarrollado |
| * Cuerpo del trabajo | 30 | 30 |  |
| * Conclusiones | 5 | 5 |  |
| * Bibliografía | 5 | 5 |  |
| **TOTAL ACUMULADO** |  | 47 |  |

|  |  |  |  |
| --- | --- | --- | --- |
| **SUSTENTACIÓN 50%** | PONDERACIÓN | PUNTOS OBTENIDOS | Observaciones |
| * Contenido desarrollado | 35 | 35 |  |
| * Uso de recursos audiovisuales | 5 | 5 |  |
| * Tono de Voz | 5 | 5 |  |
| * Mantiene motivación del grupo | 2 | 2 |  |
| * Uso del tiempo | 3 | 3 |  |
| **TOTAL ACUMULADO** |  | 50 |  |

UNIVERSIDAD TECNOLOGICA DE PANAMA

FACULTAD DE INGENIERIA DE SISTEMAS COMPUTACIONALES

LICENCIATURA EN INGENIERIA DE SISTEMAS Y COMPUTACION

BASE DE DATOS I

MANEJADORES DE BASE DE DATOS

PRESENTADO A CONSIDERACIÓN DE:

ING. JEANNETTE JOHNSON DE HERRERA

INTEGRANTES

CEDEÑO, KEVIN

LO, YUI

GARCIA, JUAN

ZARATE FERNANDO

GRUPO: 1IL131

LUNES 22 DE JUNIO DEL 2020

# Índice de contenido

[Índice de contenido 2](#_Toc43705668)

[Índice de figuras 3](#_Toc43705669)

[Índice de cuadros 4](#_Toc43705670)

[Introducción 5](#_Toc43705671)

[Resumen ejecutivo 6](#_Toc43705672)

[Contenido del trabajo 7](#_Toc43705673)

[Oracle 7](#_Toc43705674)

[Sus características 7](#_Toc43705675)

[Ventajas y desventajas de su uso 8](#_Toc43705676)

[Ventajas 8](#_Toc43705677)

[Desventajas 8](#_Toc43705678)

[Diferencias con otros manejadores de su mismo tipo Relacional o NoSQL 9](#_Toc43705679)

[**MariaDB** 10](#_Toc43705680)

[**Cassandra** 11](#_Toc43705681)

[Técnicas empleadas 12](#_Toc43705682)

[**Creación de base de datos y tablas** 12](#_Toc43705683)

[Crear una Tabla 18](#_Toc43705684)

[MySQL 19](#_Toc43705685)

[Sus características 19](#_Toc43705686)

[Ventajas y desventajas de su uso 20](#_Toc43705687)

[Diferencias con otros manejadores de su mismo tipo Relacional o NoSQL 20](#_Toc43705688)

[Técnicas empleadas 21](#_Toc43705689)

[Conclusiones 28](#_Toc43705690)

[Bibliografía 29](#_Toc43705691)

# Índice de figuras

[Figura 1. MariaDB Ventajas 10](file:///C:\Users\Juan%20Camilo\Downloads\Proyecto%202-Oracle%20y%20MySQL-CedeñoKevin-LoYui-GarcíaJuan-ZarateFernando.docx#_Toc43550117)

[Figura 2. Cassandra 11](file:///C:\Users\Juan%20Camilo\Downloads\Proyecto%202-Oracle%20y%20MySQL-CedeñoKevin-LoYui-GarcíaJuan-ZarateFernando.docx#_Toc43550118)

[Figura 3. Funcionamiento Bases de Datos 18](#_Toc43550119)

[Figura 4. Creación de base de datos 22](#_Toc43550120)

[Figura 5. Mostrar base de datos 22](#_Toc43550121)

[Figura 6. Comando USE 23](#_Toc43550122)

[Figura 7. Comando CREATE TABLE 23](#_Toc43550123)

[Figura 8. Comando SHOW TABLE 23](#_Toc43550124)

[Figura 9. Sintáxis para definir columna 24](#_Toc43550125)

[Figura 10. Valores nulos 24](#_Toc43550126)

[Figura 11. Valores por defecto 24](#_Toc43550127)

[Figura 12.Claves primarias 25](#_Toc43550128)

[Figura 13. Columna autoincrementadas 25](#_Toc43550129)

[Figura 14. Comentario en MySQL 25](#_Toc43550130)

[Figura 15.Sintaxis de SELECT**Forma incondicional** 26](#_Toc43550131)

[Figura 16. Consulta incondicional 26](#_Toc43550132)

[Figura 17. Consulta específica 27](#_Toc43550133)

[Figura 18. Ejemplo SELECT 27](#_Toc43550134)

[Figura 19. Aplicar función sobre columnas 27](#_Toc43550135)

[Figura 20. Query a la base de datos MySQL 28](#_Toc43550136)

# Índice de cuadros

[Tabla 1. Parámetros de inicialización 14](#_Toc43550534)

# Introducción

En el entorno del mercado actual, la competitividad y la rapidez de maniobra de una empresa son imprescindibles para su éxito. Para conseguirlo existe cada vez una mayor demanda de datos y, por tanto, más necesidad de gestionarlos. Esta demanda siempre ha estado patente en empresas y sociedades, pero en estos años se ha disparado debido al acceso multitudinario a las redes integradas en Internet y a la aparición de los dispositivos móviles que también requieren esa información.

Es importante conocer los diferentes tipos de bases de datos, y las bases de datos que conforman estos tipos, por ello, en este trabajo se hará una introducción a dos bases de datos (MySQL y Oracle) presentando sus principales características, ventajas y desventajas, para que así los miembros del curso puedan diferenciar y generar una idea de cual base de datos usar dependiendo de la situación en la que uno se encuentre. Por último se mostrará como hacer distintos tipos de operaciones básicas en estas bases de datos, para familiarizar al lector con las diferentes sintáxis que existen en las bases de datos.

# Resumen ejecutivo

Hoy en día existen infinidades de bases de datos de diferentes tipos (Relacional, No relacionales, Orientadas a documentos, etc). En este proyecto se introducirá dos tipos de bases de datos tipo relacional (MySQL y Oracle).

Como primer punto de cada base de datos, se enumerará las principales características de la base de datos correspondiente, para después presentar sus ventajas y desventajas, después se hará una pequeña comparación con otra base de datos de tipo similar; para así reforzar la idea de cuales son sus puntos fuertes y débiles.

Por último se hará distintos tipos de querys básicos y creaciones de tablas de cada una, viendo las distintas técnicas utilizadas en cada una.

El resumen ejecutivo debe tener lo más importante de todo el trabajo. Esto parece más bien una introducción

# Contenido del trabajo

# Oracle

## Sus características

Oracle Database se caracteriza principalmente por ser un tipo sistema de gestión de base de datos de tipo objeto-relacional (ORDBMS, por sus iniciales en inglés de Object-Relacional Data Base Management System), desarrollado por Oracle corporation. Oracle Database es considerado como uno de los sistemas de bases de datos más completos sobresaliendo entre: Soporte de transacciones, estabilidad, escalabilidad, y soporte multiplataforma. (FORMATALENT, s.f.)

Características principales de Oracle

* Modelo relacional: los usuarios visualizan los datos en tablas con el formato filas/columnas.
* Herramienta de administración gráfica intuitiva y cómoda de utilizar.
* Control de acceso: tecnologías avanzadas para vigilar la entrada a los datos.
* Protección de datos: seguridad completa en el entorno de producción y de pruebas y gestión de copias de seguridad.
* Lenguaje de diseño de bases de datos muy completo (PL/SQL): permite implementar diseños "activos", que se pueden adaptar a las necesidades cambiantes de negocio.
* Alta disponibilidad: escalabilidad, protección y alto rendimiento para la actividad empresarial.
* Gestión de usuarios: agilidad en los trámites, reducción de costes y seguridad en el control de las personas que acceden a las aplicaciones y a los sistemas.

## Ventajas y desventajas de su uso

### Ventajas

Oracle es la primera empresa en desarrollar e implementar software empresarial 100% activado por Internet en toda su línea de productos: base de datos, aplicaciones comerciales y herramientas para el soporte de decisiones y el desarrollo de aplicaciones. Esta es una de sus numerosas ventajas, que se resumen en cinco puntos:

* Motor de base de datos objeto-relacional más usado a nivel mundial.
* Multiplataforma: puede ejecutarse desde un PC hasta una supercomputadora.
* Soporta todas las funciones que se esperan de un buen servidor.
* Software del servidor que puede ejecutarse en multitud de sistemas operativos: Linux, Mac, Windows, etc.
* Permite el uso de particiones para la mejora de la eficiencia, de replicación e incluso ciertas versiones admiten la administración de bases de datos distribuidas.
* El software del servidor puede ejecutarse en multitud de sistemas operativos.
* Existe incluso una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa.
* Oracle es la base de datos con más orientación hacía INTERNET.

(Hernandez, 2017)

### Desventajas

Al igual que cualquier tipo de software, independiente a su uso puede llegar a presentar múltiples desventajas, como es el caso de Oracle:

* El mayor inconveniente de Oracle es quizás su precio. Incluso las licencias de Personal Oracle son excesivamente caras. Otro problema es la necesidad de ajustes. Un error frecuente consiste en pensar que basta instalar el Oracle en un servidor y enchufar directamente las aplicaciones clientes. Un Oracle mal configurado puede ser desesperantemente lento.
* También es elevado el coste de la información, y sólo últimamente han comenzado a aparecer buenos libros sobre asuntos técnicos distintos de la simple instalación y administración.
* Inhabilidad de implementar el procesamiento recursivo
* De acuerdo con “SQL para tontos” es un libro el nombre original es “SQL For Dummies”, una de las mayores desventajas de SQL es su incapacidad de ejecutar procesamientos recursivos. Podemos recordar que un procesamiento recursivo es un tipo de función de computadora (o programa) en el cual uno de los pasos o procedimientos vuelve a hacer correr el programa entero (o el procedimiento). SQL carece de construcciones de tipo lazo que son comunes en otros tipos de lenguajes de programación de alto nivel. No se pueden repetir acciones y no hay forma de definir construcciones repetitivas en SQL.
* incompatibilidad y complejidad
* Una de las mayores desventajas de Oracle SQL es la inconsistencia e incompatibilidad de datos en las áreas del tiempo y sintaxis de datos, concatenación de cadenas y sensibilidad de caracteres. El lenguaje es complejo, con un enfoque de palabras clave similar en estructura a COBOL (por las cifras en inglés de lenguaje común orientado a los negocios), con menos reglas de sintaxis y gramática.

## Diferencias con otros manejadores de su mismo tipo Relacional o NoSQL

Una de las mejores maneras de comparar Oracle con otros manejadores de su mismo tipo relacional es mediante el análisis de sus ventajas y desventajas que puede presentar cada uno. Nosotros analizaremos dos manejares que le hacen la competencia, estos son el MariaDB y Cassandra.

### **MariaDB**

A picture containing device, clock

Description automatically generatedEste SGBD es una derivación de MySQL que cuenta con la mayoría de las características de este e incluye varias extensiones. Nace a partir de la adquisición de MySQL por parte de Oracle para seguir la filosofía Open Source y tiene la ventaja de que es totalmente compatible con MySQL. (Marín, 2019)



Figura . MariaDB Ventajas

Ventajas

* Aumento de motores de almacenamiento
* Gran escalabilidad
* Seguridad y rapidez en transacciones
* Extensiones y nuevas características relacionadas con su aplicación para Bases de datos NoSQL.

Desventajas

* No tiene desventajas muy aparentes salvo algunas pequeñas incompatibilidades en la migración de MariaDB y MySQL o pequeños atrasos en la liberación de versiones estables

### **Cassandra**

Se caracteriza por utilizar almacenamiento clave-valor. Es un SGBD NoSQL distribuido y masivamente escalable. (Marín, 2019)

Figura . Cassandra

Dispone de un lenguaje propio para las consultas denominado CQL (Cassandra Query Languaje).

Ventajas

* Multiplataforma
* Propio lenguaje de consultas (CQL)
* Escalado lineal y horizontal
* Es un SGBD distribuido
* Utiliza una arquitectura peer-to-peer

Desventajas

* Al igual que MariaDB no tiene muchas desventajas. Es tan bueno que es utilizado por Facebook, Twitter, Instagram, Spotify o Netflix

## Técnicas empleadas

### **Creación de base de datos y tablas**

**Proceso para la creación de base de datos**

En Oracle existen varias maneras de crear una base de datos, por ejemplo:

* Base de datos con DBCA: este hace uso de la herramienta del asistente de configuración de base de datos de Oracle o por su definición original en inglés.

Oracle Database Configuration Assistant, esta es la manera más sencilla de crear la base de datos.

* Base de datos con la instrucción CREATE DATABASE: esta es la manera manual de crear una base de datos en Oracle, esta será la que explicaremos.

CREATE DATABASE

Esta es la instrucción SQL, que brinda un enfoque más manual para crear una base de datos que usar el Asistente de configuración de la base de datos Oracle (DBCA). Una ventaja de usar esta declaración sobre el uso de DBCA es que puede crear bases de datos desde scripts.

Acerca de la creación de una base de datos con la instrucción CREATE DATABASE

Cuando usa la CREATE DATABASE instrucción, debe completar acciones adicionales antes de tener una base de datos operativa. Estas acciones incluyen la creación de vistas en las tablas del diccionario de datos y la instalación de paquetes PL / SQL estándar. Realiza estas acciones ejecutando los scripts proporcionados. Los procesos serán para Windows.

Paso 1: Se especifica un identificador de instancia (SID)

ORACLE\_SID la variable de entorno se utiliza para distinguir esta instancia de otras instancias de Oracle Database que puede crear más tarde y ejecutar simultáneamente en la misma computadora host.

set ORACLE\_SID=mynewdb

Paso 2: Asegurar de que las variables de entorno requeridas están establecidas

Dependiendo de su plataforma, antes de que pueda iniciar SQL \* Plus (como se requiere en un paso posterior), es posible que tenga que establecer variables de entorno, o al menos verificar que estén configuradas correctamente.

Por ejemplo, en la mayoría de las plataformas, se deben establecer ORACLE\_SID y ORACLE\_HOME, Además es aconsejable establecer la variable PATH para incluir el ORACLE\_HOME directorio /bin. En la plataforma Windows, OUI asigna automáticamente valores a ORACLE\_HOME y ORACLE\_SID en el registro de Windows. Si no creó una base de datos al instalarla, OUI no establece ORACLE\_SID en el registro y tendrá que establecer la variable de entorno ORACLE\_SID al crear la base de datos más adelante.

Paso 3: elija un método de autenticación del administrador de la base de datos

Debe estar autenticado y se le deben otorgar los privilegios del sistema adecuados para crear una base de datos.

Paso 4: Cree el archivo de parámetros de inicialización

Cuando se inicia una instancia de Oracle, lee un archivo de parámetros de inicialización. Este archivo puede ser un archivo de texto, que puede crearse y modificarse con un editor de texto, o un archivo binario, que es creado y modificado dinámicamente por la base de datos. El archivo binario, que se prefiere, se denomina archivo de parámetros del servidor. En este paso, creará un archivo de parámetros de inicialización de texto. En un paso posterior, creará un archivo de parámetros del servidor a partir del archivo de texto.

Tabla. Parámetros mínimos de inicialización recomendados

| **Nombre del parámetro** | **Obligatorio** | **Notas** |
| --- | --- | --- |
| DB\_NAME | Sí | Identificador de base de datos. Debe corresponder al valor utilizado en la instrucción CREATE DATABASE Máximo 8 caracteres. |
| CONTROL\_FILES | No | Muy recomendable. Si no se proporciona, la instancia de base de datos crea un archivo de control en la misma ubicación que el archivo de parámetros de inicialización. Proporcionar este parámetro le permite multiplexar archivos de control. Consulte "Creación de archivos de control inicial" para obtener más información. |
| MEMORY\_TARGET | No | Establece la cantidad total de memoria utilizada por la instancia y habilita la administración automática de memoria. Puede elegir otros parámetros de inicialización en lugar de este para un mayor control manual del uso de memoria. Consulte " Configuración manual de la memoria". |

Tabla 1. Parámetros de inicialización

Paso 5: (solo Windows) Cree una instancia

En la plataforma Windows, antes de poder conectarse a una instancia, debe crearla manualmente si aún no existe. El ORADIM comando crea una instancia de Oracle Database al crear un nuevo servicio de Windows.

oradim -NEW -SID *sid* -STARTMODE MANUAL -PFILE *file*

Paso 6: Conéctese a la instancia

Inicie SQL \* Plus y conéctese a su instancia de Oracle Database con el SYSDBA privilegio administrativo.

$ sqlplus /nolog

SQL> CONNECT SYS AS SYSDBA

Connected to an idle instance.

Connected to:

Oracle Database 19c Enterprise Edition Release 19.0.0.0.0 - Production Version 19.1.0.0.0

Paso 7: Crear un archivo de parámetros de servidor

El archivo de parámetros del servidor le permite cambiar los parámetros de inicialización con el comando ALTER SYSTEM y conservar los cambios en un cierre y arranque de la base de datos. El archivo de parámetros del servidor se crea a partir del archivo de inicialización de texto editado.

Paso 8: Iniciar la instancia

Inicie una instancia sin montar una base de datos.

CREATE SPFILE FROM PFILE;

Paso 9: Emita la instrucción CREATE DATABASE

Para crear la nueva base de datos, utilice la instrucción CREATE DATABASE

STARTUP NOMOUNT

Paso 10: Crear espacios de tablas adicionales

Para que la base de datos sea funcional, debe crear espacios de tablas adicionales para los datos de la aplicación.

CREATE DATABASE mynewdb

USER SYS IDENTIFIED BY sys\_password

USER SYSTEM IDENTIFIED BY system\_password

LOGFILE GROUP 1 ('/u01/logs/my/redo01a.log','/u02/logs/my/redo01b.log') SIZE 100M BLOCKSIZE 512,

GROUP 2 ('/u01/logs/my/redo02a.log','/u02/logs/my/redo02b.log') SIZE 100M BLOCKSIZE 512,

GROUP 3 ('/u01/logs/my/redo03a.log','/u02/logs/my/redo03b.log') SIZE 100M BLOCKSIZE 512

MAXLOGHISTORY 1

MAXLOGFILES 16

MAXLOGMEMBERS 3

MAXDATAFILES 1024

CHARACTER SET AL32UTF8

NATIONAL CHARACTER SET AL16UTF16

EXTENT MANAGEMENT LOCAL

DATAFILE '/u01/app/oracle/oradata/mynewdb/system01.dbf'

SIZE 700M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED

SYSAUX DATAFILE '/u01/app/oracle/oradata/mynewdb/sysaux01.dbf'

SIZE 550M REUSE AUTOEXTEND ON NEXT 10240K MAXSIZE UNLIMITED

DEFAULT TABLESPACE users

DATAFILE '/u01/app/oracle/oradata/mynewdb/users01.dbf'

SIZE 500M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED

DEFAULT TEMPORARY TABLESPACE tempts1

TEMPFILE '/u01/app/oracle/oradata/mynewdb/temp01.dbf'

SIZE 20M REUSE AUTOEXTEND ON NEXT 640K MAXSIZE UNLIMITED

UNDO TABLESPACE undotbs1

DATAFILE '/u01/app/oracle/oradata/mynewdb/undotbs01.dbf'

SIZE 200M REUSE AUTOEXTEND ON NEXT 5120K MAXSIZE UNLIMITED

USER\_DATA TABLESPACE usertbs

DATAFILE '/u01/app/oracle/oradata/mynewdb/usertbs01.dbf'

SIZE 200M REUSE AUTOEXTEND ON MAXSIZE UNLIMITED;

Paso 11: Ejecutar scripts para crear vistas de diccionario de datos

Ejecute los scripts necesarios para crear vistas de diccionario de datos, sinónimos y paquetes PL/SQL, y para admitir el correcto funcionamiento de SQL\*Plus.

@?/rdbms/admin/catalog.sql

@?/rdbms/admin/catproc.sql

$ORACLE\_HOME/perl/bin/perl $ORACLE\_HOME/rdbms/admin/catctl.pl -d

$ORACLE\_HOME/rdbms/admin -n number\_of\_processes -l output\_log\_directory catpcat.sql

@?/rdbms/admin/utlrp.sql

@?/sqlplus/admin/pupbld.sql

Paso 12: (Opcional) Ejecutar scripts para instalar opciones adicionales

Es posible que desee ejecutar otros scripts. Los scripts que ejecute están determinados por las características y opciones que elija utilizar o instalar.

Paso 13: Copia de seguridad de la base de datos

Realice una copia de seguridad completa de la base de datos para asegurarse de que tiene un conjunto completo de archivos desde los que recuperarse si se produce un error de medios.

Paso 14: (Opcional) Habilitar inicio automático de instancias

Es posible que desee configurar la instancia de Oracle para que se inicie automáticamente cuando se reinicie el equipo host. (Oracle, 2019)

ORADIM -EDIT -SID sid -STARTMODE AUTO -SRVCSTART SYSTEM [-SPFILE]

## Crear una Tabla

Para crear las tablas en Oracle se basan en los modelos entidad relación

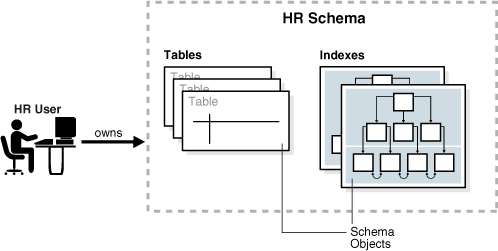


Figura 3. Funcionamiento Bases de Datos

La instancia para crear una tabla es: CREATE TABLE.

El siguiente ejemplo muestra la instrucción CREATE TABLE para la tabla de empleados en el esquema de muestra hr. La declaración especifica columnas como employee\_id, first\_name, etc., y especifica un tipo de datos como NUMBER o DATE para cada columna.

CREATE TABLE employees

(employee\_id NUMBER(6)

, first\_name VARCHAR2(20)

, last\_name VARCHAR2(25)

CONSTRAINT emp\_last\_name\_nn NOT NULL

, email VARCHAR2(25)

CONSTRAINT emp\_email\_nn NOT NULL

, phone\_number VARCHAR2(20)

, hire\_date DATE

CONSTRAINT emp\_hire\_date\_nn NOT NULL

, job\_id VARCHAR2(10)

CONSTRAINT emp\_job\_nn NOT NULL

, salary NUMBER(8,2)

, commission\_pct NUMBER(2,2)

, manager\_id NUMBER(6)

, department\_id NUMBER(4)

, CONSTRAINT emp\_salary\_min

CHECK (salary > 0)

, CONSTRAINT emp\_email\_uk

UNIQUE (email)

) ;

# MySQL

## Sus características

MySQL al ser basada en código abierto es fácilmente accesible y la inmensa mayoría de programadores que trabajan en desarrollo web han pasado usar MySQL en alguno de sus proyectos porque al estar ampliamente extendido cuenta además con una ingente comunidad que ofrece soporte a otros usuarios. (Ángel Robledano, 2019)

Características principales de MySQL:

* Trabaja con bases de datos relacionales: utiliza tablas múltiples que se interconectan entre sí para almacenar la información y organizarla correctamente.
* Funcionamiento en un modelo cliente y servidor: clientes y servidores se comunican entre sí de manera diferenciada para un mejor rendimiento.
* Compatibilidad con SQL: MySQL ofrece plena compatibilidad por lo que si has trabajado en otro motor de bases de datos no tendrás problemas en migrar a MySQL.
* Vistas: ofrece compatibilidad para poder configurar vistas personalizadas del mismo modo que podemos hacerlo en otras bases de datos SQL.
* Procedimientos almacenados: MySQL posee la característica de no procesar las tablas directamente, sino que a través de procedimientos almacenados es posible incrementar la eficacia de nuestra implementación.
* Desencadenantes: MySQL permite además poder automatizar ciertas tareas dentro de nuestra base de datos. En el momento que se produce un evento otro es lanzado para actualizar registros u optimizar su funcionalidad.
* Transacciones: una transacción representa la actuación de diversas operaciones en la base de datos como un dispositivo. El sistema de base de registros avala que todos los procedimientos se establezcan correctamente o ninguna de ellas.

## Ventajas y desventajas de su uso

Descritas las principales características de MySQL es fácil ver sus ventajas. MySQL es una opción razonable para ser usado en ámbito empresarial. Al estar basado en código abierto permite a pequeñas empresas y desarrolladores disponer de una solución fiable y estandarizada para sus aplicaciones. Por ejemplo, si se cuenta con un listado de clientes, una tienda online con un catálogo de productos o incluso una gran selección de contenidos multimedia disponible, MySQL ayuda a gestionarlo todo debida y ordenadamente. (Ángel Robledano, 2019)

Por otro lado, el de las desventajas, el principal problema al cual MySQL se enfrenta es a los altos volúmenes de datos, sobre todo si se realiza gran cantidad de escritura en sus tablas. En sistemas con mucho tráfico esto puede volverse un gran problema, razón por la cual se recomienda en esos casos el uso de sistemas como PostgreSQL, que están mejor adaptados para manejar grandes cantidades de datos.

Además de eso, otro gran problema de MySQL con el cual se topan a menudo los más novatos es que no es nada intuitivo, es decir, no es fácil darle uso si no sabemos lo que hacemos. Esto claro se puede remediar estudiando sobre este software, pero no deja de ser un aspecto bastante criticado de la herramienta.

## Diferencias con otros manejadores de su mismo tipo Relacional o NoSQL

Ventajas de PostgreSQL sobre MySQL:

PostgreSQL es una base de datos relacional de objetos, mientras que MySQL es una base de datos puramente relacional. Esto significa que PostgreSQL incluye características como la herencia de tablas y la sobrecarga de funciones, que pueden ser importantes para ciertas aplicaciones. PostgreSQL también se adhiere más a los estándares SQL. (Krasimir Hristozov, 2019)

Según Mihir Shah, una de la diferencia entre MongoDB y MySQL es que MongoDB tiene capacidad para manejar grandes datos no estructurados y es más rápido. Gran cantidades de personas están experimentando el rendimiento de MongoDB en el mundo real principalmente, porque permite a los usuarios consultar de una manera diferente que es más sensible a la carga de trabajo.

Los desarrolladores señalan que MySQL es bastante lento en comparación con MongoDB cuando se trata de lidiar con la gran base de datos. Por lo tanto, es una mejor opción para usuarios con un pequeño volumen de datos y está buscando una solución más general, ya que no puede hacer frente a grandes cantidades de datos no estructurados.

## Técnicas empleadas

**Creación de bases de datos y tablas**

**Crear una base de datos**

Para empezar, crearemos una base de datos para nosotros solos, y la llamaremos "prueba". Para crear una base de datos se usa una sentencia CREATE DATABASE:

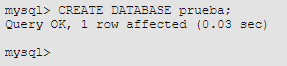


Figura 4. Creación de base de datos

Podemos averiguar cuántas bases de datos existen en nuestro sistema usando la sentencia SHOW DATABASES:

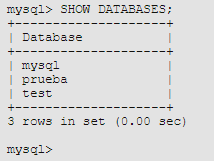


Figura 5. Mostrar base de datos

Para seleccionar una base de datos se usa el comando USE, que no es exactamente una sentencia SQL, sino más bien de una opción de MySQL:



Figura 6. Comando USE

**Crear una tabla**

En su forma más simple, la sentencia CREATE TABLE creará una tabla con las columnas que indiquemos. Crearemos, como ejemplo, una tabla que nos permitirá almacenar nombres de personas y sus fechas de nacimiento. Deberemos indicar el nombre de la tabla y los nombres y tipos de las columnas:

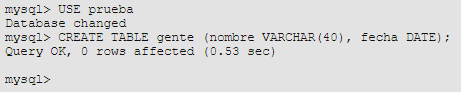


Figura 7. Comando CREATE TABLE

Hemos creado una tabla llamada "gente" con dos columnas: "nombre" que puede contener cadenas de hasta 40 caracteres y "fecha" de tipo fecha.

Podemos consultar cuántas tablas y qué nombres tienen en una base de datos, usando la sentencia SHOW TABLES:

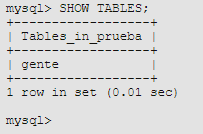


Figura 8. Comando SHOW TABLE

Pero tenemos muchas más opciones a la hora de definir columnas. Además del tipo y el nombre, podemos definir valores por defecto, permitir o no que contengan valores nulos, crear una clave primaria, indexar...

La sintaxis para definir columnas es:



Figura 9. Sintáxis para definir columna

Veamos cada una de las opciones por separado.

Valores nulos

La opción por defecto es que se permitan valores nulos, NULL, y para que no se permitan, se usa NOT NULL. Por ejemplo:



Figura 10. Valores nulos

Valores por defecto

Para cada columna también se puede definir, opcionalmente, un valor por defecto. Por ejemplo, si queremos que el valor por defecto para poblacion sea 5000, podemos crear la tabla como:



Figura 11. Valores por defecto

Claves primarias

También se puede definir una clave primaria sobre una columna, usando la palabra clave KEY o PRIMARY KEY.

Sólo puede existir una clave primaria en cada tabla, y la columna sobre la que se define una clave primaria no puede tener valores NULL. Si esto no se especifica de forma explícita, MySQL lo hará de forma automática.

Por ejemplo, si queremos crear un índice en la columna nombre de la tabla de ciudades, crearemos la tabla así:



Figura 12.Claves primarias

Columnas autoincrementadas

En MySQL tenemos la posibilidad de crear una columna autoincrementada, aunque esta columna sólo puede ser de tipo entero.

Si al insertar una fila se omite el valor de la columna autoinrementada o si se inserta un valor nulo para esa columna, su valor se calcula automáticamente, tomando el valor más alto de esa columna y sumándole una unidad. Esto permite crear, de una forma sencilla, una columna con un valor único para cada fila de la tabla.

Generalmente, estas columnas se usan como claves primarias 'artificiales'. MySQL está optimizado para usar valores enteros como claves primarias, de modo que la combinación de clave primaria, que sea entera y autoincrementada es ideal para usarla como clave primaria artificial:

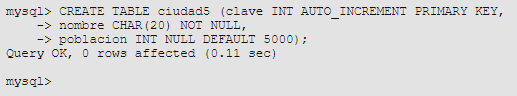


Figura 13. Columna autoincrementadas

Comentarios

Adicionalmente, al crear la tabla, podemos añadir un comentario a cada columna. Este comentario sirve como información adicional sobre alguna característica especial de la columna, y entra en el apartado de documentación de la base de datos:

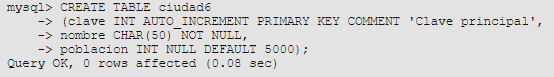


Figura 14. Comentario en MySQL

**Consultas**

**Selección de datos**

La sintaxis de SELECT es compleja, pero en este trabajo no explicaremos todas sus opciones. Una forma más general consiste en la siguiente sintaxis:

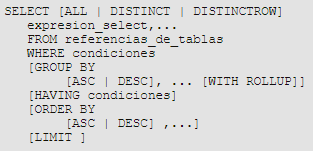


Figura 15.Sintaxis de SELECT**Forma incondicional**

La forma más sencilla es la que hemos usado hasta ahora, consiste en pedir todas las columnas y no especificar condiciones.

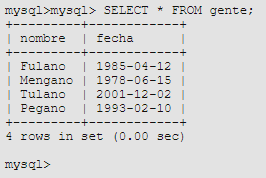


Figura 16. Consulta incondicional

**Limitar las columnas: proyección**

Mediante la sentencia SELECT es posible hacer una proyección de una tabla, seleccionando las columas de las que queremos obtener datos. En la sintaxis que hemos mostrado, la selección de columnas corresponde con la parte "expresion\_select". En el ejemplo anterior hemos usado \*, que quiere decir que se muestran todas las columnas.

Pero podemos usar una lista de columnas, y de ese modo sólo se mostrarán esas columnas:

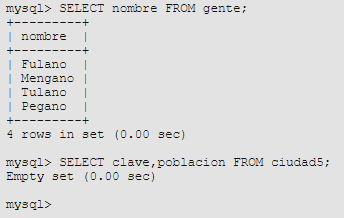


Figura 17. Consulta específica

Las “expresiones\_select” no se limitan a nombres de columnas de tablas, pueden ser otras expresiones, incluso aunque no correspondan a ninguna tabla:

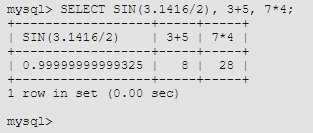


Figura 18. Ejemplo SELECT

Figura. SELECT con operaciones

También podemos aplicar funciones sobre columnas de tablas, y usar esas columnas en expresiones para generar nuevas columnas:

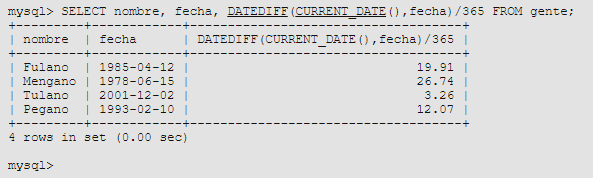


Figura 19. Aplicar función sobre columnas

**Alias**

Aprovechemos la ocasión para mencionar que también es posible asignar un alias a cualquiera de las expresiones select. Esto se puede hacer usando la palabra AS, aunque esta palabra es opcional:

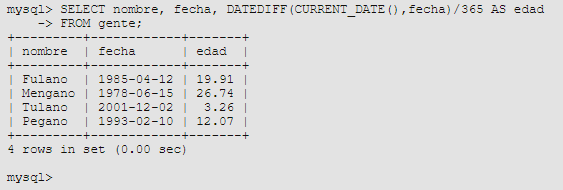


Figura 20. Query a la base de datos MySQL

Figura. Renombrar columna

# Conclusiones

Como hemos podido ver, MySQL es un sistema de gestión de bases de datos relacionales, es más usado del mundo de hecho, y se caracteriza por ser gratuito, multiplataforma, es compatible con varias interfaces gráficas, tiene soporte para casi una docena de motores de almacenamiento e incluye funciones de respaldo, revisión de tablas, reparación, optimización y replicación.

# Bibliografía

* FORMATALENT. (.). Formación en Oracle: Historia y Características., de FormaTalent Business School Sitio web: https://formatalent.com/formacion-en-oracle-historia-y-caracteristicas/
* Madayeli Perez Hernandez. (2017). ventajas y desventajas de MySQL, oracle, visual foxpro y access. 15 noviembre, 2017, de mape309site Sitio web: https://mape309site.wordpress.com/2017/11/15/ventajas-y-desventajas-de-mysql-oracle-visual-foxpro-y-access/
* “5 grandes ventajas de Oracle**”** [Marcial Varela;](https://www.cursosfemxa.es/autor/449-marcialvarela)Martes, 22 Marzo 2016 10:13 https://www.cursosfemxa.es/blog/5-grandes-ventajas-de-oracle
* Rafael Marín. (2019). Los gestores de bases de datos más usados en la actualidad. 16/04/2019, de Revista Digital Sitio web: https://revistadigital.inesem.es/informatica-y-tics/los-gestores-de-bases-de-datos-mas-usados/
* Ángel Robledano. (2019). Qué es MySQL: Características y ventajas. 18 de junio de 2020, de openwebinars Sitio web: <https://openwebinars.net/blog/que-es-mysql/#:~:text=Caracter%C3%ADsticas%20de%20MySQL&text=La%20m%C3%A1s%20evidente%20es%20que,la%20informaci%C3%B3n%20y%20organizarla%20correctamente.&text=En%20bases%20de%20datos%20de,Procedimientos%20almacenados>.
* Anónimo. (2019). MySQL. junio 19, 2020, de TecnoMagazine Sitio web: <https://tecnomagazine.net/2019/11/12/mysql/>
* Hristozov, K.. (2019). MySQL vs PostgreSQL -- Choose the Right Database for Your Project. junio 19, 2020, de okta Sitio web: https://developer.okta.com/blog/2019/07/19/mysql-vs-postgres#:~:text=Postgres%20is%20an%20object%2Drelational,more%20closely%20to%20SQL%20standards.
* Shah, M. (2019). MongoDB vs MySQL: A Comparative Study on Databases. junio 19, 2020, de SIMFORM Sitio web: <https://www.simform.com/mongodb-vs-mysql-databases/#:~:text=MongoDB%20vs%20MySQL%3A%20Performance%20%26%20Speed,is%20more%20sensitive%20to%20workload>.
* Anónimo. (s.f.). Curso de SQL. junio 19, 2020, de AulaClic Sitio web: <https://www.aulaclic.es/sql/>
* Oracle. (2019). Oracle DataBase Documentation. 2020, de Docs Oracle Sitio web: https://docs.oracle.com/en/database/oracle/oracle-database/index.html