**EventDataHub**

**Descripción General**

1. **Propósito de la base de datos**

EventDataHub está diseñada para gestionar de manera eficiente los datos relacionados con eventos, ubicaciones, boletas, entradas, organizadores y participantes en conciertos y festivales. Su propósito principal es centralizar y organizar toda la información necesaria para la administración de eventos, facilita la planificación, coordinación y administración de eventos, optimiza la asignación de recursos y proporciona una interfaz para gestionar entradas y ubicar a los participantes.

1. **Usuarios finales**

Los usuarios finales incluyen organizadores de eventos y empresas de ventas de entradas. Los organizadores de eventos utilizan la base de datos para crear y gestionar eventos, asignan participantes (como artistas, presentadores, etc.), y controlan la disponibilidad y estado de las entradas. Además, pueden realizar un seguimiento del rendimiento de los eventos y obtener informes detallados sobre la asistencia y las ventas. Las empresas de ventas de entradas utilizan la base de datos para gestionar la venta de entradas, actualizar el estado de las entradas (por ejemplo, activas, utilizadas), y garantizar que la disponibilidad de las entradas esté siempre actualizada. También pueden generar informes de ventas y administrar devoluciones o cambios.

1. **Resumen del diseño**

La base de datos está estructurada en varias tablas interrelacionadas que representan entidades clave como eventos, ubicaciones, boletas, entradas, organizadores y participantes. La arquitectura de la base de datos incluye relaciones uno a muchos y muchos a muchos, con índices para optimizar consultas y procedimientos almacenados para automatizar tareas.

**Requisitos del Sistema**

1. **Hardware**

* **Requisitos Mínimos**:
  + Procesador: 2 GHz Dual Core
  + Memoria RAM: 4 GB
  + Espacio en Disco: 20 GB
  + Conexión de red básica
* **Requisitos Recomendados**:
  + Procesador: 3 GHz Octa Core
  + Memoria RAM: 16 GB
  + Espacio en Disco: 100 GB (con almacenamiento SSD para mejorar el rendimiento de las operaciones de E/S)
  + Conexión de red de alta velocidad y baja latencia

1. **Software**

* **Sistema Operativo**: Windows Server 2019 o superior
* **Gestor de Bases de Datos**: MySQL 8.0 o superior
* **Otros Software**: Herramientas de respaldo y recuperación (por ejemplo, MySQL Enterprise Backup), software de monitoreo de rendimiento (por ejemplo, Nagios, Zabbix), y herramientas de seguridad (por ejemplo, firewalls y sistemas de detección de intrusiones).

**Diseño Lógico de Datos**

1. **Diagramas ERD (Entity-Relationship Diagram):** Ver Archivo Adjunto, Diseño lógico EventDataHub

**Modelo de Datos**

1. **Diagramas ERD (Entity-Relationship Diagram):** Ver Archivo Adjunto, Modelo Entidad Relacion EventDataHub
2. **Descripción de las Tablas**

**Eventos**

Almacena información sobre los eventos organizados.

* **Nombre de la Tabla:** Eventos
* **Columnas:**
  + **EventoID:** INT, auto incrementable, PK.
  + **Nombre:** VARCHAR(25), not null, Nombre del evento.
  + **Descripción:** TEXT, Descripción del evento.
  + **Fecha:** DATE, not null, Fecha del evento.
  + **Hora:** TIME, not null, Hora del evento.
  + **UbicacionID:** INT, not null, Identificador de la ubicación, FK.
  + **Estado:** ENUM('Activo', 'Cancelado', 'SoldOut', 'Finalizado', 'Pospuesto'), not null, Estado del evento.
* **Llaves Primarias:** EventoID
* **Llaves Foráneas:** UbicacionID -> Ubicaciones(UbicacionID)
* **Índices:** INDEX(EventoID), INDEX(Estado), INDEX(Fecha)

**Boletas**

Almacena información sobre los diferentes tipos de boletas.

* **Nombre de la Tabla:** Boletas
* **Columnas:**
  + **BoletaID:** INT, auto incrementable, PK.
  + **Nombre:** VARCHAR(15), not null, UNIQUE, Nombre de la boleta.
* **Llaves Primarias:** BoletaID

**BoletasEvento**

Relaciona las boletas con los eventos y sus respectivos precios.

* **Nombre de la Tabla:** BoletasEvento
* **Columnas:**
  + **BoletaEventoID:** INT, Auto Incrementable, PK,
  + **BoletaID:** INT, Identificador de la boleta, not null, FK.
  + **EventoID:** INT, Identificador del evento, not null, FK.
  + **Precio:** DECIMAL(10,2), not null, Precio de la boleta para el evento.
  + **CantidadDisponible:** INT, not null, Cantidad disponible de esta boleta.
* **Llaves Primarias:** BoletaEventoID
* **Llaves Foráneas:**
  + BoletaID -> Boletas(BoletaID)
  + EventoID -> Eventos(EventoID)
* **Índices:** INDEX(EventoID)

**Entradas**

Almacena las entradas compradas por los asistentes.

* **Nombre de la Tabla:** Entradas
* **Columnas:**
  + **EntradaID:** INT, Identificador único de la entrada, PK. auto incrementable
  + **BoletaEventoID:** INT, Identificador del evento, not null, FK.
  + **Estado:** ENUM('Activa', 'Utilizada'), not null, Estado de la entrada.
  + **Codigo:** BIGINT
* **Llaves Primarias:** EntradaID
* **Llaves Foráneas:**
  + EventoID -> Eventos(EventoID)
  + BoletaID -> Boletas(BoletaID)
* **Índices:** INDEX(Estado)

**Ubicaciones**

Almacena información sobre las ubicaciones donde se realizan los eventos.

* **Nombre de la Tabla:** Ubicaciones
* **Columnas:**
  + **UbicacionID:** INT, auto incrementable, Identificador único de la ubicación, PK.
  + **Nombre:** VARCHAR(30), Nombre de la ubicación.
  + **Direccion:** TEXT, not null, Dirección de la ubicación.
  + **CiudadID:** INT, identificador de la ciudad, not null, FK.
  + **Capacidad:** INT, Capacidad máxima de la ubicación.
  + **PrecioAlquiler:** DECIMAL(10,2), Precio de alquiler de la ubicación.
* **Llaves Primarias:** UbicacionID
* **Llaves Foráneas:**
  + CiudadID -> Ciudad(CiudadID)
* **Índices:** INDEX(Ciudad), INDEX(Capacidad), INDEX(PrecioAlquiler)

**Ciudades**

Almacena información sobre las ciudades.

* **Nombre de la Tabla:** Ciudades
* **Columnas:**
  + **CiudadID:** INT, Identificador único de la ciudad, auto incrementable, PK.
  + **Nombre:** VARCHAR(20), unique, not null, Nombre de la ciudad.
* **Llaves Primarias:** CiudadID

**Organizadores**

Almacena información sobre los organizadores de los eventos.

* **Nombre de la Tabla:** Organizadores
* **Columnas:**
  + **OrganizadorID:** INT, Identificador único del organizador, auto incrementable, PK.
  + **Nombre:** VARCHAR(25), not null, Nombre del organizador.
  + **Email:** VARCHAR(30), not null, Email del organizador.
  + **Teléfono:** VARCHAR(10), not null, Teléfono del organizador.
* **Llaves Primarias:** OrganizadorID

**EventoOrganizador**

Relaciona los eventos con sus organizadores y sus roles específicos.

* **Nombre de la Tabla:** EventoOrganizador
* **Columnas:**
  + **EventoID:** INT, Identificador del evento, not null, PK, FK.
  + **OrganizadorID:** INT, Identificador del organizador, not null, PK, FK.
  + **Inversion:** Decimal(10,2) Inversión del organizador en el evento.
* **Llaves Primarias:** EventoID, OrganizadorID
* **Llaves Foráneas:**
  + EventoID -> Eventos(EventoID)

**Contrataciones**

Almacena información sobre las contrataciones de participantes para los eventos.

* **Nombre de la Tabla:** Contrataciones
* **Columnas:**
  + **EventoID:** INT, Identificador del evento, not null,PK, FK.
  + **ParticipanteID:** INT, Identificador del participante, not null, PK, FK.
  + **Descripcion:** TEXT, not null, Descripción de la contratación.
  + **Precio:** DECIMAL (10,2), Precio acordado por la contratación.
* **Llaves Primarias:** EventoID, ParticipanteID
* **Llaves Foráneas:**
  + EventoID -> Eventos(EventoID)
  + ParticipanteID -> Participantes(ParticipanteID)
* **Índices:** INDEX (EventoID)

**Participantes**

Almacena información sobre los participantes en los eventos.

* **Nombre de la Tabla:** Participantes
* **Columnas:**
  + **ParticipanteID:** INT, Identificador único del participante, auto incrementable, PK.
  + **Nombre:** VARCHAR(25), not null, Nombre del participante.
  + **Email:** VARCHAR(30), not null, Email del participante.
  + **Teléfono:** VARCHAR(10), not null, Teléfono del participante.
  + **Tipo:** ENUM ('Artista', 'Conferencista', presentador, ‘Sonido’, ‘Bailarines’, ‘Seguridad’ ‘DJ’, ‘Productora’, 'Otro'), not null, Tipo de participante.
* **Llaves Primarias:** ParticipanteID
* **Indices:** INDEX(Tipo)

**Esquemas de Base de Datos**

1. **DDL (Data Definition Language):** Ver Archivo Adjunto, Script EventDataHub

**Procedimientos y Funciones Almacenadas**

* **TRIGGER Codigo:**

Este trigger se ejecuta antes de insertar una nueva entrada en la tabla Entradas. Su propósito es generar un código único para cada entrada. La lógica del trigger es la siguiente:

1. Se declaran tres variables locales: IdEvento, IdEntrada, y IdBoleta.
2. Se obtiene el EventoID correspondiente al BoletaEventoID de la nueva entrada.
3. Se determina el próximo ID de la entrada a ser insertada utilizando la tabla INFORMATION\_SCHEMA.
4. Se genera el código en el formato IdEvento + Fecha actual (YYYYMMDD) + IdEntrada + IdBoleta.
5. Se asigna este código a la nueva entrada.

* **TRIGGER ReduceCantidadDisponible:**

Este trigger se ejecuta después de insertar una nueva entrada en la tabla Entradas. Su propósito es reducir la cantidad disponible de boletas para un evento específico. La lógica del trigger es la siguiente:

Se actualiza la tabla BoletasEvento.

Se reduce en 1 la columna CantidadDisponible para el BoletaEventoID de la nueva entrada.

* **PROCEDURE AgruparEventosPorCiudades**

Agrupa eventos por ciudad. Dado el nombre de una ciudad, devuelve una lista de eventos que ocurren en esa ciudad, incluyendo el nombre del evento, la fecha, la hora y la ubicación. Es útil para obtener rápidamente una vista general de todos los eventos programados en una ciudad específica.

* **PROCEDURE ObtenerInfoDeEvento**

Obtiene todos los detalles de un evento específico. Dado el ID de un evento, devuelve el nombre del evento, descripción, fecha, hora, estado, nombre de la ubicación, dirección, y el total de boletas disponibles para el evento. Esto es útil para obtener una vista completa de la información y disponibilidad de un evento específico.

* **PROCEDURE EventosPorFecha**

Devuelve una lista de eventos que ocurren dentro de un rango de fechas especificado. Toma dos fechas como entrada: la fecha de inicio y la fecha de fin, y devuelve los eventos que ocurren entre esas fechas. Esto es útil para filtrar eventos por un periodo específico, como un mes o una semana.

* **PROCEDURE EntradasVendidas**

Devuelve el número total de entradas vendidas para un evento específico. Toma el ID de un evento como entrada y devuelve la cantidad de entradas vendidas para ese evento. Esto es útil para obtener un resumen rápido de las ventas de entradas para un evento específico.

**Consultas Comunes**

* **VIEW eventos del último mes**

Esta vista muestra los eventos que se han llevado a cabo en el último mes. Calcula el rango de fechas basado en el mes anterior al mes actual para determinar qué eventos están incluidos en este período. Permite a los usuarios revisar rápidamente los eventos recientes y es útil para informes de actividad o análisis de eventos pasados.

* **VIEW inversión total de los eventos**

Esta vista muestra la inversión total asociada a cada evento. Agrupa la inversión por nombre de evento, sumando todas las inversiones relacionadas con cada evento. Ideal para análisis financiero, esta vista permite ver cómo se distribuye la inversión total en eventos específicos y facilita la gestión presupuestaria.

* **VIEW eventos activos**

Esta vista muestra todos los eventos que están actualmente activos. Filtra los eventos basándose en el estado Activo. Permite a los usuarios y administradores ver qué eventos están en curso o planificados para su ejecución en el presente. Es útil para la programación y la gestión operativa de eventos.

* **SELECT Organizadores**

Esta consulta muestra todos los detalles de los organizadores registrados en la base de datos. Permite revisar la información de contacto y detalles de los organizadores, útil para la coordinación y comunicación con los organizadores de eventos.

* **SELECT Participantes**

Esta consulta muestra todos los detalles de los participantes en los eventos, incluidos sus datos de contacto y tipo. Permite obtener información sobre los participantes involucrados en los eventos, facilitando la gestión y coordinación de los roles asignados en cada evento.

* **SELECT eventos y sus ubicaciones**

Esta consulta combina información sobre eventos y sus ubicaciones para proporcionar una visión completa de los eventos y los lugares donde se llevan a cabo. Útil para obtener un panorama general de los eventos en relación con sus ubicaciones, lo que facilita la planificación y la logística de los eventos.

**Administración y Mantenimiento**

1. **Copia de seguridad y recuperación**

EventDataHub opta por realizar copias de seguridad regulares para asegurar la integridad de los datos en caso de pérdida o corrupción. Como expertos en el área, se recomienda realizar una copia de seguridad completa al menos cada 3 días para garantizar que todos los datos recientes estén protegidos. Estas copias pueden hacerse a partir de las 12 a.m. para minimizar el impacto en el rendimiento durante el horario laboral.

Hace uso de ‘Data Export’ en MySQL para realizar las copias de seguridad e Import ‘Data Import’ para restaurarla.

Ver **Referencias** para el tutorial.

1. **Monitoreo del rendimiento**

Para el monitoreo de rendimiento **EventDataHub** recomienda hacer las siguientes consultas al menos una vez por semana para identificar y solucionar posibles problemas de rendimiento y asegurarse de que el sistema funcione de manera óptima.

* **Ver las conexiones activas**

**SHOW STATUS LIKE 'Threads\_connected'**: Esta consulta muestra el número de conexiones activas en la base de datos en el momento de la consulta. Threads\_connected es una variable que indica cuántos hilos (threads) están actualmente conectados al servidor de base de datos.

Permite monitorizar cuántas conexiones están activas en la base de datos. Es útil para identificar si estás cerca del límite máximo de conexiones permitidas, lo que podría indicar una sobrecarga o problemas de rendimiento. Si el número de conexiones está muy alto, podría ser necesario optimizar la base de datos o revisar las aplicaciones que están conectándose.

* **Lista de procesos**

**SHOW PROCESSLIST**: Esta consulta muestra una lista de todos los procesos actuales que están siendo ejecutados en el servidor de base de datos. Incluye información sobre el ID del proceso, el usuario que lo está ejecutando, el estado, y el tiempo que lleva ejecutándose.

Permite revisar qué consultas o procesos están en ejecución, cuáles están en espera o bloqueadas, y qué recursos están consumiendo. Es útil para identificar procesos largos o problemáticos que podrían estar afectando el rendimiento de la base de datos.

1. **Mantenimiento Rutinario**

* **Optimizar tablas**

Realiza una optimización de la tabla especificada. Reorganiza los datos y los índices de la tabla, y puede reducir el espacio de almacenamiento y mejorar el rendimiento de las consultas.

* **Actualizar estadísticas de una tabla**

Actualiza las estadísticas de una tabla, que son utilizadas por el optimizador de consultas para elegir los mejores planes de ejecución. Las estadísticas incluyen información sobre la distribución de valores en las columnas y los índices de la tabla.

* **Verificar integridad de los datos**

Verifica la integridad de una tabla específica, comprobando la estructura de la tabla y los índices para asegurarse de que no haya errores o corrupción.

**Seguridad**

1. **Control de acceso**

En **EventDataHub**, la gestión del acceso a la base de datos se realiza a través de roles, que son conjuntos de permisos agrupados que se pueden asignar a usuarios. Esta metodología facilita la administración de permisos y asegura que cada usuario tenga acceso solo a las funcionalidades necesarias para su rol específico. Además, se implementan varias medidas de seguridad para garantizar la protección y la integridad de los datos, así como para controlar el acceso de los usuarios. Estas medidas incluyen la expiración periódica de contraseñas y restricciones de conexión según el rol del usuario.

* **Roles**

**Administrador:**

* + - Crea un rol con permisos completos en la base de datos **EventDataHub**. Los administradores pueden realizar cualquier operación, como crear, modificar o eliminar tablas y datos. Además, tienen la capacidad de otorgar estos permisos a otros roles o usuarios.
    - **Permisos:** ALL PRIVILEGES.
    - Los administradores configuran y gestionan la seguridad general del sistema, asegurando que los permisos se otorguen de manera correcta y eficiente.

**Desarrollador:**

* + - Establece un rol con permisos amplios que permiten al desarrollador realizar una variedad de operaciones, incluyendo la creación y modificación de estructuras de la base de datos y la manipulación de datos.
    - **Permisos:** SELECT, INSERT, UPDATE, DELETE, CREATE, DROP, INDEX, ALTER en la base de datos
    - Los Ddesarrolladores trabajan en la implementación y mantenimiento de la base de datos, utilizando permisos avanzados para realizar sus tareas.

**Analista:**

* + - Define un rol con acceso de solo lectura a todas las bases de datos en el sistema. Ideal para usuarios que necesitan consultar datos sin hacer modificaciones.
    - **Permisos:** SELECT en todas las bases de datos.
    - Los analistas consultan los datos necesarios para sus análisis sin modificar la base de datos.

**Junior:**

* + - Asigna un rol con permisos para leer, insertar, actualizar y eliminar datos en todas las bases de datos. Está diseñado para usuarios con necesidades de acceso más amplias que el rol de analista, pero sin los permisos completos del desarrollador.
    - **Permisos:** SELECT, INSERT, UPDATE, DELETE en todas las bases de datos.
    - Los juniors realizan tareas que requieren permisos más amplios, como la manipulación de datos, pero no necesitan permisos administrativos.

**Pasante:**

* + - Define un rol con permisos limitados a solo lectura para ciertas tablas específicas en la base de datos. Es adecuado para usuarios que requieren acceso a datos sin la posibilidad de modificar o eliminar información.
    - **Permisos:** SELECT en las tablas Eventos, Entradas y Ubicaciones.
    - Los pasantes obtienen acceso restringido a datos específicos para aprendizaje o tareas limitadas.
* **Eliminación de Role**

**EventDataHub** Permite eliminar un rol específico cuando ya no es necesario. Por ejemplo, el rol de pasante se elimina con el comando DROP ROLE 'pasante', liberando los recursos y permisos asociados a ese rol.

* **Expiración de Contraseñas**

Para mejorar la seguridad, se establece que las contraseñas de los usuarios expiren cada 90 días. Esto obliga a los usuarios a actualizar sus contraseñas regularmente, reduciendo el riesgo de acceso no autorizado debido a contraseñas comprometidas o antiguas.

* **Conexiones Restringidas por Rol**

Los administradores pueden conectarse a la base de datos desde cualquier host. Esto les permite gestionar y administrar la base de datos de manera más flexible, incluso desde ubicaciones remotas.

Los demás roles están restringidos a conectarse solo a través del localhost. Esta medida asegura que solo se pueda acceder a la base de datos desde el servidor local, aumentando la seguridad al limitar el acceso externo.

1. **Auditoría:**

La auditoría en **EventDataHub** se centra en registrar y revisar las actividades que se realizan en la base de datos. Esto incluye el seguimiento de todas las consultas SQL ejecutadas y la identificación de consultas que pueden afectar negativamente al rendimiento del sistema. Estos procedimientos permiten detectar y resolver problemas, mejorar la seguridad y garantizar el cumplimiento de las políticas de la base de datos.

* **Activación del Registro General:**
* Permite registrar todas las consultas que se ejecutan en la base de datos, incluyendo las consultas SELECT, INSERT, UPDATE y DELETE. Esto es útil para una revisión completa de las operaciones realizadas en la base de datos.
* Activa el registro general, que guarda un log de todas las operaciones realizadas, permitiendo la revisión exhaustiva de cada consulta.
* Las actividades registradas incluyen todas las consultas y comandos ejecutados en la base de datos. Este log es fundamental para una auditoría detallada y para entender el uso general de la base de datos.
* **Activación del Registro de Consultas Lentas:**
  + Registra consultas que tardan más de un tiempo determinado en ejecutarse. Esto ayuda a identificar y optimizar consultas que pueden estar afectando el rendimiento de la base de datos.
  + Habilita el registro de consultas lentas, facilitando la identificación de consultas que requieren optimización para mejorar el rendimiento general de la base de datos.
  + Solo se registran las consultas que exceden un umbral de tiempo especificado. Esto ayuda a enfocar el análisis en las consultas que tienen un impacto significativo en el rendimiento.
* **Visualización de Archivos de Registro:**
  + **SHOW PROCESSLIST**: Muestra una lista de los procesos actuales en la base de datos. Permite a los administradores ver las consultas en ejecución, así como su estado y otros detalles importantes como el tiempo que llevan ejecutándose. Esto es útil para identificar consultas que pueden estar bloqueando otros procesos o afectando el rendimiento general.
  + **SHOW GLOBAL STATUS**: Proporciona una visión general del estado actual de la base de datos, incluyendo métricas y estadísticas globales sobre su rendimiento y actividad. Ofrece información valiosa sobre la carga del sistema, el uso de recursos y otras estadísticas clave que pueden ayudar a evaluar la salud y eficiencia de la base de datos.

**Anexos**

1. **• Glosario de términos:**

* **PK:** Primary Key
* **FK:** Foreign Key

1. **• Referencias:**

**Documentacion MySQL:** [MySQL :: MySQL Documentation](https://dev.mysql.com/doc/)

**Proceso para backup:** [backup\_mysql.pdf (wordpress.com)](https://elclickizquierdo.wordpress.com/wp-content/uploads/2014/12/backup_mysql.pdf)

**EventoParticipante:** La tabla EventoParticipante almacena la relación entre los eventos y los participantes, incluyendo el precio que un participante cobra dependiendo del lugar y la cantidad de gente.

**EventoParticipante:** La columna Descripcion en EventoParticipante detalla los acuerdos específicos de las contrataciones, como "20 minutos en el escenario", "25 guardias", "5 cámaras, un dron", "Invitado especial de Brugal".

**BoletasEvento:** La tabla BoletasEvento se usa para crear la relación entre el tipo de boleta (General, Special Guest, VIP, o cualquier otro nombre que se desee crear) y el evento, así también poner el precio de la boleta y la cantidad inicial de boletas disponibles.

**Cantidad Disponible:** La columna CantidadDisponible en BoletasEvento se va reduciendo cada vez que se compra una entrada (es decir, cada vez que se crea una entrada de ese evento y tipo de boleta). Esto se realiza a través del trigger ReducirCantidadDisponible.

**Código en Entradas:** En la tabla Entradas, la columna Codigo se genera concatenando los valores de EntradaID, EventoID, UsuarioID, FechaCompra y BoletaID. Por ejemplo:

* Id del evento: 1
* Fecha de compra de la entrada: 2024-07-30
* Id de la entrada: 23
* Id de la boleta: 2

El código resultante sería: 120240730232

1. **• Historial de cambios:**

**24-07-2024:** Creación de la versión 1 de la documentación.

**26-07-2024:** Se cambió el nombre de Localidad a Ubicación. Se agregó la columna CantidadDisponible en Boletas.

**30-07-2024:** Se creó BoletasEvento y se movieron las columnas de Precio y EventoID en Boletas a BoletasEvento. Se eliminó la tabla Usuarios. Se eliminaron EventoID, UsuarioID y FechaCompra en Entradas. Se creó la tabla Ciudades y se hizo foreign key desde Ubicaciones. Se eliminó Asientos en Ubicaciones.

**31-07-2024:** Se sustituyó la columna Rol por Inversión en EventoOrganizador.

**02-08-2024:** Se eliminó la columna Descripción en Boletas y se movió la columna CantidadDisponible a BoletasEvento. Se crearon las consultas comunes.

**03-08-2024:** Se añadieron detalles de hardware y sistema, roles y usuarios, registros, auditoría y administración.

**04-08-2024:** Se añadieron descripciones de los triggers Codigo y ReducirCantidadDisponible, así como demás vistas, consultas y procedimientos almacenados.

**05-08-2024:** Se añadieron detalles generales de la base de datos, anexos y documentos adjuntos.