AEROTECK- ECKERS

INDICE

INTRODUCCION

Este es un proyecto consiste en el diseño e implementación de una base de datos para la gestión de vuelos de una aerolínea ficticia llamada AEROTECK.

El sistema permite registrar pasajeros, vuelos, aviones, reservas, tripulación y su asignación en los distintos viajes. Su objetivo es optimizar la administración de operaciones, seguimiento de personal y disponibilidad de vuelos, asegurando eficiencia y confiabilidad en su información.

SITUACION PROBLEMÁTICA

Las aerolíneas requieren el uso de un sistema centralizado que permita manejar de forma segura y eficiente toda la información relacionada con vuelos, reservas y personal.

Sin una base de datos estructurada, se generan problemas como doble asignación de asientos, errores en asignación de tripulación, falta de trazabilidad en pasajeros y vuelos, dificultad para generar reportes y estadísticas, etc.

La implementación de una base de datos relacional bien diseñada busca dar solución a estos problemas, asegurando integridad de datos, eficiencia en las operaciones y capacidad de análisis.

OBJETIVOS DEL PROYECTO

- Diseñar e implementar una base de datos que centralice la información de la aerolínea.
- Facilitar el control y la gestión de reservas, pasajeros, vuelos y tripulación.
- Evitar errores comunes como la doble asignación de asientos o tripulantes.
- Permitir el acceso rápido a la información para generar reportes y estadísticas.
- Mejorar la eficiencia operativa de la aerolínea mediante una estructura de datos bien definida.

MODELO DE NEGOCIO

Aeroteck es una aerolínea que opera en vuelos nacionales e internacionales. Cuenta con una flota de aviones de distintos modelos y capacidades, y con un equipo de tripulantes compuesto por pilotos, azafatas y copilotos.

Con la base de datos se busca centralizar y ordenar toda la información de la empresa, para que esta funcione de manera eficiente y coordinada, reduciendo errores y optimizando los recursos.

Se podrá utilizar en distintas áreas de la empresa, como en las siguientes:

- Recursos humanos: Gestión en la asignación de tripulantes por vuelos, asegurando la disponibilidad de cada uno, evitando solapamientos o ausencias.
- Operaciones: programación y control de vuelos, con su hora de salida y llegada, y la selección de aviones para cada viaje.
- Atención al cliente: Se podrá administrar las reservas, controlar los asientos ocupados y disponibles, y el registro de los pasajeros.

Descripción de tablas

Pasajeros:

Campos	Tipo de dato	Clave	Restricciones
Id_pasajero	INT	PRIMARY KEY	AUTO_INCREMENT, NOT NULL
Nombre	VARCHAR(50)		NOT NULL
Apellido	VARCHAR(50)		NOT NULL
Dni	INT	UNIQUE	NOT NULL
Email	VARCHAR(50)	UNIQUE	Puede ser NULL
Nacionalidad	VARCHAR(30)		DEFAULT 'Argentina'

Aviones:

Campos	Tipo de dato	Clave	Restricciones
Id_avion	INT	PRIMARY KEY	AUTO_INCREMENT, NOT NULL
Modelo	VARCHAR(30)		NOT NULL
Capacidad	INT		NOT NULL
Ano_fabricacion	INT		NOT NULL
Matricula	VARCHAR(20)	UNIQUE	NOT NULL

Roles_tripulación:

Campos	Tipo de dato	Clave	Restricciones
Id_rol	INT	PRIMARY KEY	AUTO_INCREMENT, NOT NULL
Nombre_rol	VARCHAR(30)	UNIQUE	NOT NULL

Vuelos:

Campos	Tipo de dato	Clave	Restricciones
Id_vuelo	INT	PRIMARY KEY	AUTO_INCREMENT, NOT NULL
Origen	VARCHAR(30)		NOT NULL
Destino	VARCHAR(30)		NOT NULL
Fecha_salida	DATE		NOT NULL
Hora_salida	TIME		NOT NULL
Id_avion	INT	FOREIGN KEY	NOT NULL, REFERENCE aviones(id_avion)
fecha_llegada	DATE		
hora_llegada	TIME		
Estado	ENUM(('PROGRAMADO', 'EN VUELO', 'CANCELADO', 'COMPLETADO')		DEFAULT ('PROGRAMADO')

Reservas:

Campos	Tipo de dato	Clave	Restricciones
Id_reservas	INT	PRIMARY KEY	NOT NULL AUTO_INCREMENT
Id_pasajero	INT	FOREIGN KEY	NOT NULL, REFERENCE pasajeros(id_pasajero)
Id_vuelo	INT	FOREIGN KEY	NOT NULL, REFERENCE vuelos(id_vuelo)
Fecha_reserva	DATE		NOT NULL
Asiento	INT	UNIQUE(compuesto)	NOT NULL, UNIQUE por combinación (id_vuelo, asiento)
Estado	ENUM('CONFIRMADA','CANCELADA', 'PENDIENTE')		DEFAULT 'PENDIENTE'

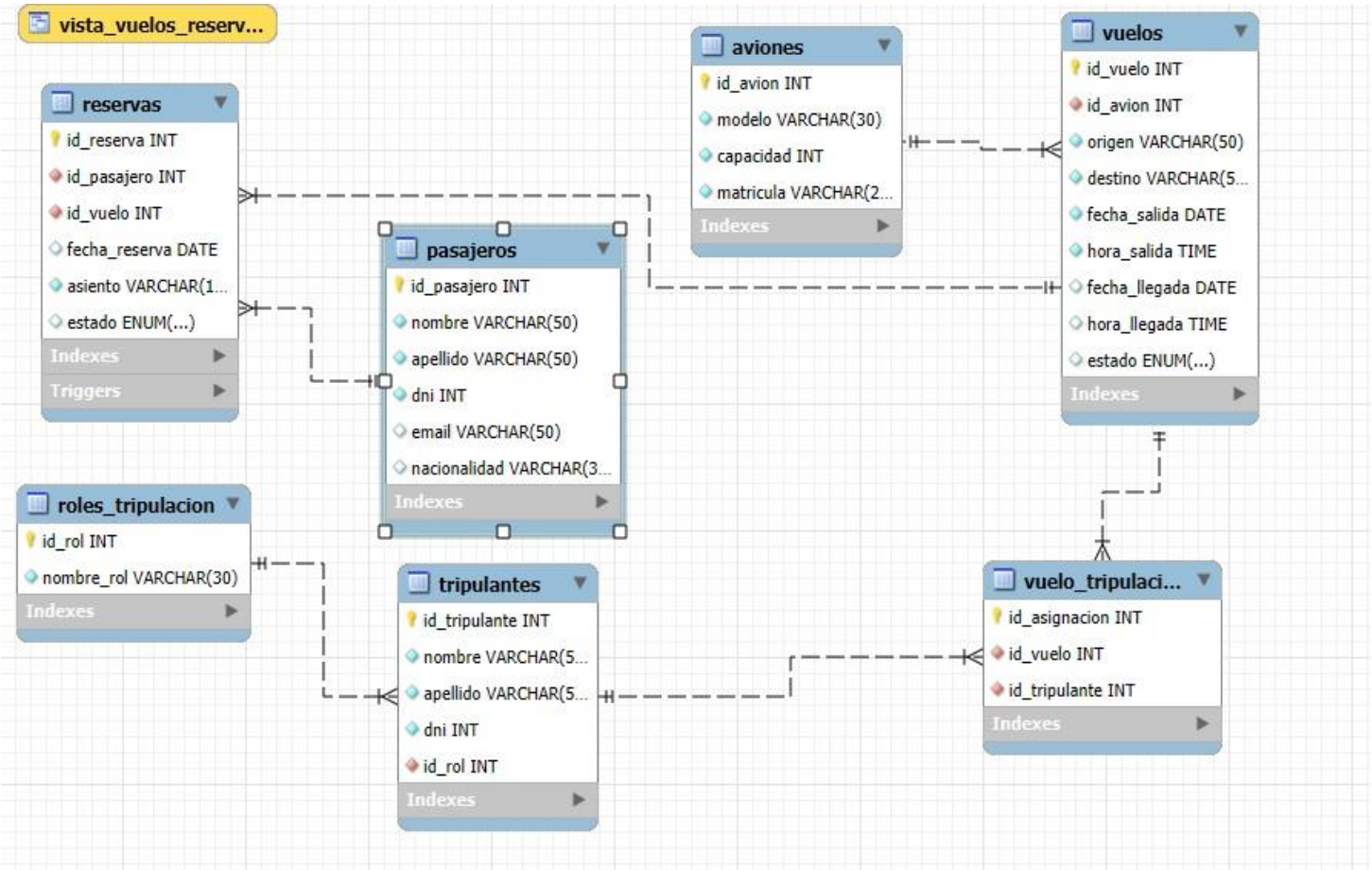
tripulantes

Campos	Tipo de dato	Clave	Restricciones
Id_tripulante	INT	PRIMARY KEY	AUTO_INCREMENT, NOT NULL
Nombre	VARCHAR(30)		NOT NULL
Apellido	VARCHAR(30)		NOT NULL
Dni	INT	UNIQUE	NOT NULL
Rol	INT	FOREIGN KEY	REFERENCES roles_tripulacion(id_rol), NOT NULL

Tripulación_por_vuelo

Campos	Tipo de dato	Clave	Restricciones
Id_vuelo	INT	PRIMARY KEY, FOREIGN KEY	REFERENCE vuelos(id_vuelo), NOT NULL
Id_tripulante	INT	PRIMARY KEY, FOREIGN KEY	REFERENCE tripulantes(id_tripulante), NOT NULL

Diagrama Entidad Relación



Vistas

VISTA 1: vista_pasajeros_confirmados

- Objetivo: Mostrar todos los pasajeros que tienen reservas confirmadas, junto con el vuelo y asiento asignado.
- Tablas involucradas: PASAJEROS, RESERVAS.
- Descripción: Esta vista permite identificar rápidamente quiénes son los pasajeros que ya tienen su reserva confirmada y en qué vuelo y asiento están. Es útil para reportes de embarque o control de ocupación.

VISTA 2: vista_tripulacion_vuelos

- Objetivo: Mostrar la tripulación asignada a cada vuelo, incluyendo nombre y rol de cada miembro.
- Tablas involucradas: vuelos, vuelo_tripulacion, tripulantes, roles_tripulacion.
- Descripción: Facilita la supervisión y planificación de vuelos, permitiendo conocer de forma clara qué tripulantes y roles están asignados a cada vuelo.

Vista 3: vista_vuelos_reservas

- Objetivo: Mostrar cada vuelo con la cantidad de reservas confirmadas que posee.
- Tablas involucradas: vuelos, reservas.
- Descripción: Permite evaluar la ocupación de los vuelos de manera rápida, identificando los vuelos con más o menos reservas confirmadas. Útil para análisis de capacidad y planificación logística.

Vista 4: vista_tripulacion_por_rol

- Objetivo: Mostrar para cada vuelo qué tripulantes están asignados agrupados por rol.
- Tablas involucradas: vuelos, tripulacion_por_vuelo, tripulantes, roles_tripulacion.
- Descripción: Permite analizar la composición de la tripulación por rol en cada vuelo, facilitando la asignación correcta de pilotos, copilotos y azafatas, y evitando conflictos de planificación.

Vista 5: vista_pasajeros_frecuentes

- Objetivo: Mostrar pasajeros con más de una reserva, incluyendo los vuelos en los que están registrados.
- Tablas involucradas: pasajeros, reservas, vuelos.
- Descripción: Identifica clientes frecuentes y sus patrones de viaje, útil para generar estrategias comerciales, promociones o análisis de fidelización de clientes.

Funciones

Función 1: Cantidad de reservas de un pasajero

- Descripción: esta función nos devuelve el total de reservas realizadas por un pasajero en particular, identificándolo por su id_pasajero.
- Objetivo: facilitar un análisis individual sobre los pasajeros, sabiendo cuantas reservas lleva realizadas (confirmadas, canceladas o pendientes). También nos permite llevar un control de clientes frecuentes, y de comportamientos en las compras realizadas por este mismo.
- Tablas que manipula: se utiliza la tabla RESERVAS, y se consulta la columna de id_pasajero, para contar la cantidad re reservas realizadas por el pasajero.

Función 2: Calcular ocupación de un vuelo en porcentaje.

- Descripción: Calcula el porcentaje de ocupación de un vuelo en particular, teniendo en cuenta la cantidad de asientos confirmados respecto de la capacidad total del avión asignado a ese vuelo.
- Objetivo: Brindar un indicador clave de gestión (KPI) para medir la eficiencia de cada vuelo en cuanto a su nivel de ocupación. Esto permite identificar vuelos con baja demanda, optimizar rutas y evaluar la rentabilidad de las operaciones.
- Tablas que manipula:

    - aviones: se usa la columna capacidad para obtener el total de asientos disponibles.
    - vuelos: se relaciona el id_avion con el vuelo correspondiente (id_vuelo).
    - reservas: se cuenta la cantidad de asientos confirmados (estado = 'CONFIRMADA') para calcular el porcentaje.

Stored Procedures

Stored Procedures 1: registrar nueva reserva.

- Descripción: Inserta una nueva reserva en el sistema con estado inicial *PENDIENTE*.
- Objetivo: Agilizar la carga de reservas de pasajeros en un vuelo de forma controlada.
- Tablas que manipula: utiliza la tabla reservas (inserta registros con datos de pasajeros, vuelo, fecha, asiento y estado.

Stored Procedures 2: Actualizar estado de un vuelo

- Descripción: permite la modificación del estado de un vuelo en base a su id_vuelo.
- Objetivo: Dar seguimiento a los vuelos y reflejar en la base de datos el progreso real, por ejemplo, cuando despega o cuando aterriza.
- Tablas que utiliza: vuelos, columna estado.

Stored procedures 3: Listar reservas confirmadas por vuelo

- Descripcion: devuelve todas las reservasconfirmadas de un vuelo en particular, junto con el nombre y apellido del pasajero.
- Objetivo: Facilitar el acceso rápido al listado de pasajeros confirmados en un vuelo, utili para el check-in o listado de embarque.
- Tablas que utiliza: Reservas y pasajeros.

Trigger

Trigger 1: evitar doble reserva

- Descripcion: evita que dos pasajeros ocupen el mismo asiento en un mismo vuelo.
- Objetivo: Garantizar que no se dupliquen los asientos en estado CONFIRMADA o PENDIENTE.
- Tablas que utiliza: RESERVAS.

Trigger 2: Asignar estado por defecto.

- Descripción: Si al insertar una reserva no se especifica el estado, se asigna automáticamente *PENDIENTE*.
- Objetivo: Evitar errores de inserción por valores nulos.
- Tablas que utiliza: reservas.

Trigger 3: actualizar el estado de vuelo.

- Descripcion: Cuando un vuelo alcanza el 100% de ocupación confirmada, su estado cambia automáticamente a *COMPLETADO*.
- Objetivo: Automatizar la gestión del estado de los vuelos sin necesidad de hacerlo manualmente.
- Tablas que utiliza: reservas, vuelos y aviones.

Scripts SQL y orden de ejecución

Para la correcta implementación del proyecto, se utilizaron tres scripts SQL distintos. A continuación se detalla su función y el orden recomendado de ejecución:

1. Script de creación de tablas

- Nombre del archivo: AEROTECK-ECKERS.sql
- Contenido: Creación de todas las tablas y sus relaciones (llaves primarias y foráneas).
- Orden de ejecución: Primero, para generar toda la estructura básica de la base de datos.

2. Script de vistas, funciones, triggers y datos

- Nombre del archivo: CreacionDeObjetos-AEROTECK_ECKERS.sql
- Contenido: Creación de vistas, funciones, stored procedures, triggers y la inserción de registros de prueba en todas las tablas.
- Orden de ejecución: Segundo, debe ejecutarse después del script de creación de tablas.

3. Script de pruebas / consultas

- Nombre del archivo: Pruebas_AEROTECK-ECKERS.sql

- Contenido: Contiene consultas de prueba y ejemplos de reportes sobre la base de datos.

- Orden de ejecución: Tercero, se ejecuta después de los scripts anteriores para validar resultados y generar informes.

REPOSITORIO EN GIT HUB.

https://github.com/patricioeckersfai-3548/CODERHOUSE