



ADMINISTRADOR DE DEPOSITO AUTONOMO

Base de datos

[Resumen](#)

Información total para depósitos de última generación

Gabriel Shulman

Gabriel.esteban.shulman@gmail.com

Problemática

Una fábrica quiere hacer su sector de envíos de piezas de repuesto, pero con la particularidad de que los encargados de reponer el stock y cumplir con los pedidos, son drones experimentales y se precisa un control de todas las partes para verificar su funcionamiento, su mantenimiento y brindar información, tanto en los pedidos que se hace como en las posibles incidencias con los drones.

Para eso se requiere una base de datos con la cual se pueda obtener información de los pedidos que se realizan, su estado, el estado de los drones e informes de posibles incidencias.

En este documento, se describen las vistas, procedimientos, triggers y funciones de la base de datos propuesta.

Listado vistas

vw_consultar_stock

tablas afectadas: ubicación.

Funcionalidad: obtener el stock presente del almacén, con su ubicación, pieza y cantidad.

vw_consultar_stock		
columna	variable	descripción
pieza	varchar(100)	nombre de la pieza
cantidad	int	cantidad de la pieza
ubicación_id	int	id de la ubicación
x	tinyint	vector de ubicación x
y	tinyint	vector de ubicación y
z	tinyint	vector de ubicación z

vw_incidente

tablas afectadas: droide, pedido.

Funcionalidad: brindar toda la información disponible al departamento técnico por incidentes con los droides, a fin de poder saber la falla del mismo, bajo qué circunstancias.

vw_incidente		
columna	variable	descripcion
droide_id	int	
droide_nombre	varchar(100)	nombre del droide
fecha_servicio	timestamp	fecha de alta
en_servicio	tinyint	esta en servicio actualmente?
falla_descripcion	varchar(255)	descripción de la falla del droide
falla_hora	timestamp	hora del incidente
pedido_id	int	pedido que realizaba
pieza	int	pieza que transportaba
cantidad	int	cantidad del pedido
tipo_movimiento	tinyint	ingresaba o restaba del stock

vw_reposicion_mensual

tablas afectadas: pedido.

Funcionalidad: Muestra los pedidos hechos en el corriente mes.

vw_reposicion_mensual		
columna	variable	descripción
remito	int	numero de remito
pieza	int	pieza seleccionada
qty	int	cantidad
patente	int	patente del transporte que la envía al deposito

vw_retiro_stock y vw_reposicion_stock

tablas afectadas: pedido.

Funcionalidad: Poseen la misma estructura, pero retiro muestra las restas del stock y reposición, las sumas del mismo.

vw_retiro_stock/vw_reposicion_stock		
columna	variable	descripcion
guia	int	numero de remito
pieza	int	pieza seleccionada
qty	int	cantidad

Listado procedimientos

Procedure: *crear_carga*

Tablas: carga

Funcionalidad: a partir de un pedido existente, anunciar un cargamento de productos.

crear_carga		
Columna	Variable	Descripción
patente:	[IN] INT	Patente del vehiculo
remito:	[IN] INT	nº de pedido
pieza:	[IN] INT	id de la pieza
qty:	[IN] INT	cantidad

Procedure:

crear_pedido_restar_sumar

Tablas: pedido y productos

Funcionalidad: crea los pedidos de repuestos y los resta o agrega al stock, según el caso.

crear_pedido_restar_sumar		
Columna	Variable	Descripción
pieza:	[IN] INT	nombre de la pieza
cantidad:	[IN] INT	cantidad de la pieza
droideAsignado:	[IN] INT	id del droide asignado
tipo_movimiento:	[IN] TINYINT	1 para sumar y 0 para restar

Procedure: *crear_ubicacion*

Tablas: ubicacion

Funcionalidad: crear del depósito donde se procede a guardar los repuestos.

crear_ubicacion		
Columna	Variable	Descripción
id	[IN] INT	id de la ubicación
x	[IN] TINYINT	vector x
y	[IN] TINYINT	vector y
z	[IN] TINYINT	vector z

Procedure: *crear_producto*

Tablas: producto y ubicación

Funcionalidad: permite crear un producto con su stock y su ubicación.

crear_producto		
Columna	Variable	Descripcion
name	VARCHAR (100)	nombre del producto
qty	[IN] INT	cantidad
ubicacion_id	[IN] INT	ubicación (id de ubicación)

Procedure: *editar_droide_falla*

Tablas: droide

Funcionalidad: anunciar el estado de falla de un droide en la tabla.

editar_droide_falla		
Columna	Variable	Descripcion
id	[IN] INT	id del droide
falla_descripcion	VARCHAR (255)	descripcion de la falla

Procedure: *obtener_productos*

Tablas: productos

Funcionalidad: obtener todos los productos existentes.

obtener_productos		
-------------------	--	--

Procedure: *obtener_pedido_por_id*

Tablas: pedido

Funcionalidad: filtra un pedido según su id.

obtener_pedido_por_id		
Columna	Variable	Descripcion
pedido_id:	[IN] INT	id del pedido

Procedure: *editar_producto*

Tablas: producto

Funcionalidad: modificar el nombre, la cantidad de productos y la ubicación del mismo.

editar_producto		
Columna	Variable	Descripcion
id	[IN] INT	id del producto
name	VARCHAR (255)	nuevo nombre del producto
qty	[IN] TINYINT	nueva cantidad
ubicación_id	[IN] TINYINT	nueva ubicación

Procedure: *obtener_ubicaciones*

Tablas: ubicacion

Funcionalidad: obtiene un listado de todas las ubicaciones de orden ascendente con sus vectores y descripciones.

obtener_ubicaciones

Listado triggers

Los 4 triggers descritos en la siguiente tabla, crean registros en la tabla log_movimientos de las tablas afectadas en sus movimientos (inserciones y modificaciones).

Nombre del trigger	Tablas afectadas
trg_droide_insert	droide
trg_droide_en_servicio_update	droide
trg_pedido_stock_update	pedido
trg_productos_ingresados	productos

Listado funciones

La función `mensaje_personalizado`, permite adecuar la respuesta de la base de datos al servicio que consume de la misma, personalizando la respuesta del servidor.

mensaje_personalizado	p_tipo VARCHAR (20)	p_texto VARCHAR (255)
	establecer mensaje de éxito, atención o error	descripción del mensaje