--ROLES DE USUARIO

Necesitamos organizar el tema de los roles de usuario. Deben existir 3, Administrador que en db='admin', Supervisor que en db='supus' y Usuario que en db='usuba'.  
  
Por defecto cualquier usuario creado debe crearse con el rol 'usnor' que en vista debe decir Usuario básico y ese usuario solo podrá acceder a crear albaranes de ruta o a editar los que hubieran sido creados por él, no eliminar.

El Administrador ‘admin’ puede crear, editar y eliminar tanto usuarios como albaranes, pero el rol de los usuarios que puede elegir el ‘admin’ solo puede ser ‘admin’ y el por defecto ‘usuba’, también podrá consultar los dashboard o consultar cualquier dato referente a los módulos de Data analytics.

El Supervisor ‘supus’ debe cambiar su nombre en la vista web a Supremo y no tendrá restricciones de ninguna clase, es decir, podrá acceder a cualquier cosa del sistema.

--LÍNEAS DE ALBARÁN

antes de seguir y poner el prompt necesito agregar un campo más a la tabla lineas\_albaran\_ruta y es fecha\_alb que hace referencia a la fecha de albaran de la consulta sql y no a la fecha de albaran de la cabecera en postgresql me entiendes? este campo debe ir antes de serie\_alb y numero\_alb pero no se debe diligenciar sino que también debe ser resultado de la búsqueda.

vas a generar las líneas de albarán con toda la información que te pase en el archivo donde está la imagen de cómo debe de quedar. El objetivo es que existan dos campos input en el primero el usuario deberá ingresar serie albarán que en la tabla lineas\_albaran\_ruta equivale a ‘serie\_alb’ y en el segundo Número albarán que en la tabla lineas\_albaran\_ruta equivale a ‘numero\_alb’, al dar enter con estos datos y utilizando la siguiente consulta se tienen que llenar los siguientes datos de la tabla que tú ya conoces porque te compartí la imagen de una tabla con las columnas CAMPOS|DESCRIPCIONES pero igualmente te la vuelvo a compartir:

SELECT CONCAT(a.SerieAlbaran, a.NumeroAlbaran) AS ALBARAN, a.RazonSocial, a.CodigoCliente, a.MunicipioEnvios, a.ImporteLiquido, a.FechaAlbaran, a.MargenBeneficio, a.PorMargenBeneficio, t.Transportista, a.BaseImponible, a.NumeroLineas, deuda\_table.deuda FROM [MMARKET].[dbo].[CabeceraAlbaranCliente] a LEFT JOIN Transportistas t ON t.CodigoEmpresa = a.CodigoEmpresa AND t.CodigoTransportista = a.CodigoTransportistaEnvios LEFT JOIN ( SELECT CodigoClienteProveedor, CONCAT(FORMAT(SUM(ImporteEfecto), 'N2', 'es-ES'), '€') AS deuda FROM CarteraEfectos WHERE CodigoEmpresa = 1 AND Prevision = 'C' AND StatusBorrado = 0 GROUP BY CodigoClienteProveedor ) AS deuda\_table ON deuda\_table.CodigoClienteProveedor = a.CodigoCliente WHERE a.CodigoEmpresa = 1 AND a.Ejercicioalbaran > 2023

|  |  |
| --- | --- |
| CAMPOS | DESCRIPCIONES |
| id | pk |
| cabecera\_id | fk |
| fecha\_albaran | fecha seleccionada en la cabecera |
| num\_alb\_ruta | numero correspondiente a la cabecera |
| serie\_alb | viene de la consulta sql debe ser tipo texto |
| fecha\_alb | viene de la consulta sql |
| numero\_alb | viene de la consulta sql debe ser tipo texto |
| nombre\_cliente | viene de la consulta sql debe ser tipo texto |
| cod\_cliente | viene de la consulta sql debe ser tipo texto |
| municipio\_envío | viene de la consulta sql debe ser tipo texto |
| importe | viene de la consulta sql |
| beneficio | viene de la consulta sql |
| imp\_liqu | viene de la consulta sql |
| porcentaje | viene de la consulta sql |
| conductor | viene de la consulta sql |
| ilpa | campo calculado obtenido del campo 'importe' multiplicado por el campo 'porcentaje\_pactado' ejemplo: 'importe'=100 'porcentaje pactado'=4 entonces 100\*0.04 |
| ben \_post\_ log | campo calculado de la resta de 'beneficio' menos 'ilpa' |
| por\_ben\_real | campo calculado dividiendo campo 'ben\_post\_log' sobre 'importe' |
| deuda | viene de la consulta sql |
| lineas | viene de la consulta sql |
| fecha\_registro | fecha y hora en la que se registra el dato en la base de datos por un usuario (inmodificable) |

Una vez los campos estén ya con la información correspondiente, deben ser calculados como te lo enuncie en la tabla aquellos que requieran calculo, los demás solo poner la información que viene de la consulta.

Cada línea debe tener tanto al principio como al final de la línea dos botones uno de chulito que indique ok y el otro que indique más, el ok servirá para pasar a otra línea y el más para agregarla, la diferencia es que el ok solo termina la línea pero no genera otra, en cambio el más servirá tanto para terminar la línea en la que está como para agregar la siguiente línea.

Los campos que se vienen de la consulta sql y se pueden modificar solo son serie albarán, número albarán y el campo importe, sin embargo, este último cada que se vaya a modificar por el usuario deberá recibir una alerta para indicar que está seguro de modificarlo o no.

Luego la cabecera albarán ‘cabecera\_albaran\_ruta’ podrá terminar de calcular los campos una vez que todas las líneas estén y el usuario de el ok, sin embargo, a medida que ponga las líneas el sistema debe ir calculando esos campos. Si prefieres primero hacemos líneas y luego terminamos cabecera.