

Descripción de entrega – Juan Felipe Diaz

Cambios desde entrega anterior:

- Adición de campo sexo en la tabla residentes.
- Mentores normalizados de acuerdo con id en tabla mentores.
- Adición de campo semestre en la tabla contratos.
- Teléfonos cambiados a tipo de dato VARCHAR desde INT.
- Descripción de mesa normalizada de acuerdo con id en tabla mesa.
- Restricción de llave foránea eliminada en tabla contratos con respecto a tabla leads, y viceversa. Facilita inserción de estos e independencia lógica.
- Al campo red_social de tabla leads se eliminó restricción de unicidad para poder tener valor nulo en uno o más registros
- El campo lead_id en tabla contratos ya tiene la restricción de unicidad, en caso de que un residente generado por un lead anterior haga un segundo contrato.

Vistas:

- vbase_datos_lookat: Como los datos de los residentes se encuentran en una tabla diferente a la que tiene los datos de sus viviendas y mentores en el semestre, se juntan estas dos para tener la información de cada residente durante el actual semestre.
- vpuntaje_anotaciones: Como los residentes pueden tener anotaciones con puntuación negativa, se crea una vista que rápidamente permita traer la información de los residentes y sus respectivos puntajes. Para facilidad, esta vista depende de la anterior creada (vbase_datos_lookat).
- vacomodaciones_restantes: De acuerdo con el número de contratos activos en el periodo actual, se calcula en número de acomodaciones restantes de cada tipo. También se muestra la fecha más próxima en la que un contrato para tal o tal acomodación acaba.
- vprom_puntuacion: Promedio de la puntuación de encuestas de satisfacción dadas al final de semestre. Tiene en cuenta que las puntuaciones posibles están en el rango [1,10].
- vorigen_leads: Crea una vista que cuenta cuantos leads vinieron de cada proveniencia, además de mostrar el número total.
- vcontratos_vigentes_nivelduracion: Una última vista que simula el número de contratos vigentes aún para el siguiente semestre. Clasificados de acuerdo con nivel de educación y duración del contrato.

Funciones:

- fnum_aptos_x_gente: De acuerdo con el número de personas que deseen ingresar a la ciudadela, la función retorna el número de apartamentos totales

entre todas las acomodaciones para ese número de personas o un número mayor. Asume que el nombre de columna será uno válido.

- `fnum_gente_grave`: Retorna el número de residentes con un puntaje de gravedad mayor o igual al ingresado por parámetro.

Procedimientos almacenados:

- `sp_ordenar_residentes`: Estos SP se crean siguiendo las instrucciones de la asignación de la semana. Se pide que el primer SP reciba un parámetro de acuerdo con el cual ordenar una tabla y un segundo parámetro que indique si será ascendiente o descendiente. Asume que el nombre de columna ingresado será válido.
- `sp_insertar_anotacion`: Se solicita que el segundo SP pueda ingresar datos en una tabla. Se escoge la tabla de anotaciones para generar anotaciones desde el SP.

Triggers:

Estos triggers se crean siguiendo las instrucciones de la asignación de la semana. Se pide que se elijan dos tablas a partir de las cuales se crearan tablas de bitácora. Para cada tabla se generará un trigger before y uno after. La primera tabla log se hará en base a la tabla de contratos, siendo esta de las tablas más importantes de la base, para empezar a manejar registros por el lado económico.

- `trigger_verifica_fecha_contrato`: Este trigger con before lo que hace es verificar que la fecha de inicio del contrato o de fin de este no sean en temporalidades 2 años menores o mayores a las actuales (esto con objetivo de que no se agregue un cero de más o algo similar, se elige dos años porque es el límite para hacer reservas por adelantado y se permite de hace dos años para que no haya problema añadiendo contratos del año pasado o algún caso similar.
- `trigger_log_contrato_realizado`: Este trigger con after creará un log con el usuario que realizó la operación, la fecha de realización, y la hora en la que datos fueron añadidos a la tabla de contratos. Todo conectado con el id de los contratos insertados.

La segunda tabla log se hará en base a la tabla de residentes, siendo esta la tabla con información que puede ser vital en algún momento de la operación.

- `trigger_verifica_contacto`: Este trigger con before lo que hace es verificar que el número y correo ingresados sean válidos
- `trigger_log_residente_registrado`: Este trigger con after creará un log con el usuario que realizó la operación, la fecha de realización, y la hora en la que datos fueron añadidos a la tabla de contratos. Todo conectado con el id de los residentes registrados.