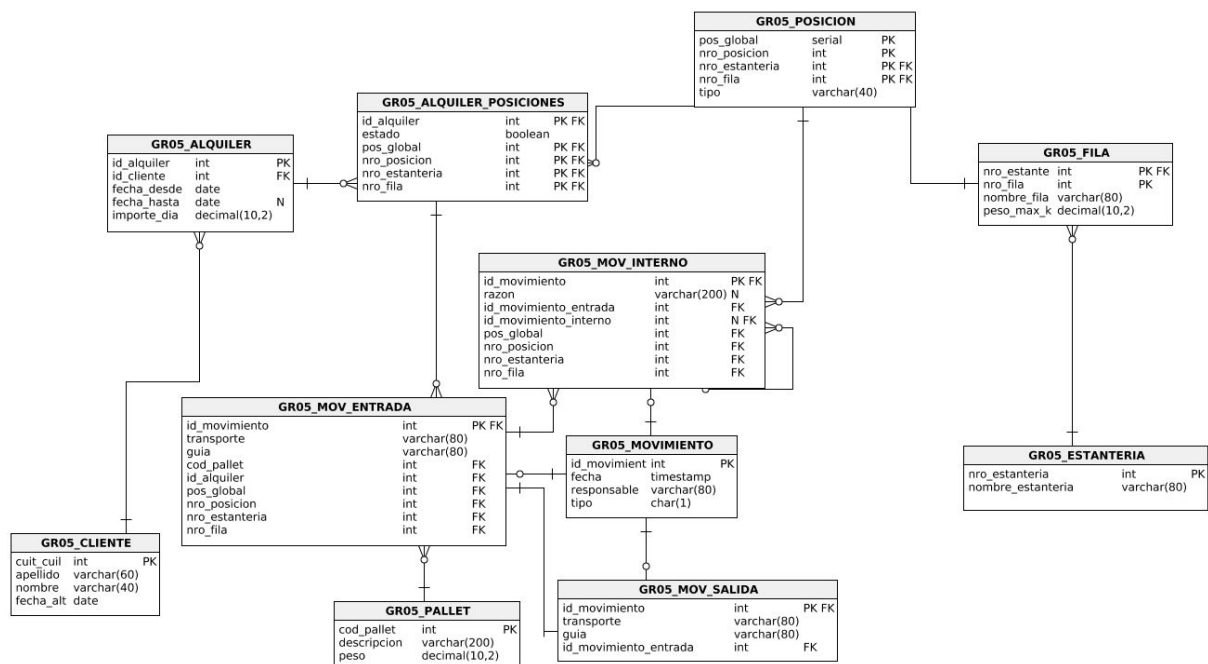


Decisiones de diseño

Se tomaron una serie de decisiones de diseño en cuanto a las tablas. Se añadió una PK en la tabla “GR05_POSICION” para manejar un único número de posición de tipo Serial, por lo cual nunca va a haber posiciones repetidas. Además de esto se agregaron una serie de FK para referenciar los movimientos. Para movimientos de egresos de mercadería se añadió: “FK_GR05_MOV_SALIDA_MOV_ENTRADA”. Luego en los movimientos internos añadimos una FK a si mismo denominada “FK_GR05_MOV_INTERNO_MOV_INTERNO” y otra denominada “FK_GR05_MOV_INTERNO_MOV_ENTRADA” haciendo referencia a un movimiento de entrada.

Nuestro esquema de tablas quedó así:



Restricciones

En cuanto a las restricciones incluidas en el inciso B decidimos que lo más conveniente en cada una fue:

- a) La tomamos como una restricción de tupla y creamos la CONSTRAINT “UQ_GR05_alquiler_fecha_valida”, la declaración fue:

```
ALTER TABLE Gr05_Alquiler
ADD CONSTRAINT UQ_GR05_alquiler_fecha_valida
CHECK (fecha desde < fecha hasta);
```

- b) En el caso de la restricción de peso en la fila la tomamos como una restricción global por lo cual hicimos una ASSERTION que compruebe el peso actual de una fila de la estantería y si el pallet que se desea ingresar a dicha fila supera al peso máximo soportado no permite ingresarlo en el depósito. Esta restricción fue creada en la tabla

“gr05_mov_entrada” ya que nos pareció conveniente que no deje ingresar en esa fila al palet que se desea. La asserction es la siguiente:

```
ALTER TABLE gr05_mov_entrada
ADD CONSTRAINT TR_GR05_peso_valido
CHECK (NOT EXIST(SELECT 1
                from gr05_fila f inner join gr05_posicion p on
f.nro_estanteria=p.nro_estanteria and f.nro_fila = p.nro_fila
                inner join gr05_alquier_posiciones ap on
ap.nro_estanteria=p.nro_estanteria and ap.nro_fila =
p.nro_fila
                inner join gr05_mov_entrada me on
ap.nro_estanteria=me.nro_estanteria and ap.nro_fila =
me.nro_fila
                inner join gr05_pallet p on
me.cod_pallet=p.cod_pallet
                where me.nro_estanteria AND me.nro_fila
                group by me.nro_fila, me.nro_estanteria
                HAVING SUM(f.peso)< (SELECT peso
                FROM pallet p
                where cod_pallet=me.cod.pallet)
));
```

- c) Esta restricción fue tomada como una restricción de atributo en las cuales el “tipo” dentro de la tabla “gr05_mov_entrada” en : “general”, “vidrio”, “insecticidas”, “inflamable”. La declaración fue:

```
ALTER TABLE gr_05posicion
ADD CONSTRAINT UQ_GR05_posicion_tipo_posicion_valida
CHECK (tipo like 'general' or tipo like 'vidrio' or tipo like
'insecticidas' or tipo like 'inflamable')
```

Servicios

En cuanto a los servicios pedidos en el inciso C se crearon dos funciones que retornan tablas, una para cada servicio.

En el caso del punto 1 determinamos una función que ingresada una fecha de tipo date devuelve una tabla con:

- número de estantería
- número de fila
- número de posición

Para ello tomamos la tabla “gr05_posicion” con un full join a gr05_alquiler_posiciones y está a su vez unida con full join a “gr05_alquiler” para obtener acceso a las fechas de las posiciones alquiladas. En esta consultamos que la fecha ingresada sea igual o menor a la

fecha en la cual deberían retirar el pallet o que en esa posición no haya ningún alquiler. La función realizada fue la siguiente:

```
Create FUNCTION FN_GR05_pos_libres (fechaingresada date)
RETURNS table (nro_estanteria int, nro_fila int, nro_posicion int)
AS $$
BEGIN
    RETURN QUERY
        SELECT p.nro_estanteria, p.nro_fila, p.nro_posicion
        FROM gr05_posicion p full join gr05_alquiler_posiciones g05ap
        on p.pos_global = g05ap.pos_global and p.nro_posicion =
        g05ap.nro_posicion and p.nro_estanteria = g05ap.nro_estanteria and
        p.nro_fila = g05ap.nro_fila
        full join gr05_alquiler g05a on g05ap.id_alquiler =
        g05a.id_alquiler
        WHERE g05a.fecha_hasta <= fechaingresada or g05ap.id_alquiler
        is null;
END; $$
LANGUAGE plpgsql;
```

Por otra parte en caso del inciso dos tomamos la tabla “gr05_cliente” unida con inner join a “gr05_alquiler” y la tabla que retorna esta función da el id del cliente, y los días restantes para el vencimiento. Para ello debemos ingresar cuantos días necesitamos de antelación.

```
Create FUNCTION FN_GR05_vencimiento_alquiler (dias_ingresados int)
RETURNS table (id_cliente int, dias_restantes int) AS $$
BEGIN
    RETURN QUERY
        SELECT id_cliente, fecha_hasta - current_date
        FROM gr05_cliente join gr05_alquiler g05a on
        gr05_cliente.cuit_cuil = g05a.id_cliente
        WHERE fecha_hasta - current_date = dias_ingresados;
END; $$
LANGUAGE plpgsql;
```

Vistas

Luego de esto para el inciso D (definición de vistas) realizamos dos vistas, una para cada punto respectivamente.

En el punto 1 creamos una vista llamada “GR05_pos_libres” en las cuales se utilizan las tablas “gr05_alquiler”, “gr05_alquiler_posiciones”, “gr05_posicion”. Para cumplir con el objetivo de mostrar cada una de las posiciones y si están ocupadas mostrar cuantos días restan de alquiler, realizamos una subconsulta en la cual si hay un alquiler asociado a la posición devuelva la cantidad de días restantes de la misma. Gracias al uso del coalesce en caso de estar libres las posiciones vamos a ver reflejado un “-1” en la tabla resultante de esa vista.

```
Create VIEW GR05_pos_libres AS
```

```

Select p.nro_posicion, coalesce( ap.nro_fila, -1) AS
numero_fila, coalesce(ap.nro_estanteria, -1) AS nro_estanteria,
coalesce((SELECT fecha_hasta-current_date

FROM gr05_alquiler a1 JOIN gr05_alquiler_posiciones ap1 ON
a1.id_alquiler = ap1.id_alquiler

WHERE a.id_alquiler = a1.id_alquiler),-1) AS días que faltan
FROM gr05_posicion p FULL JOIN gr05_alquiler_posiciones ap
ON p.nro_posicion = ap.nro_posicion and p.nro_estanteria =
ap.nro_estanteria and p.nro_fila = ap.nro_fila
full join gr05_alquiler a on a.id_alquiler = ap.id_alquiler;

```

En el punto 2 el objetivo era hacer una lista de los diez cliente que más han invertido y ordenarlos de mayor a menor. Para esto realizamos una vista llamada “GR05_dinero_invertido” en la cual devuelve una tabla con el id del cliente y la cantidad invertida en el último año, tomando como referencia la fecha en la cual inició el alquiler un año hacia atrás de la fecha actual, para ello sumamos los importes por día multiplicados a la cantidad de días de dicho alquiler. En la consulta de esta vista utilizamos las tablas “gr05_cliente” unida con inner join a “gr05_alquiler”.

```

Create VIEW GR05_dinero_invertido AS
select a.id_cliente, SUM(importe_dia*(fecha_hasta-fecha_desde))
from gr05_cliente c inner join gr05_alquiler a on
c.cuit_cuil=a.id_cliente
where ((extract(year from now()))-1 < extract(year from
a.fecha_desde))
group by a.id_cliente
order by 2 desc
limit 10;

```

Sitio Web

Por último en el inciso E realizamos dos funciones que retornan una tabla cada una, cada función corresponde a un servicio del sitio.

En primer lugar para poder realizar una consulta para dada una fecha de un listado de posiciones libres realizamos una función que ingresada una fecha de tipo date retorna una tabla con los números de posiciones libres, la fila de dicha posición y la estantería. Para ello en la consulta necesaria utilizamos las tablas “gr05_posicion” unida con full join a “gr05_alquiler_posiciones” y está unida respectivamente con full join a “gr05_alquiler”. La función es la siguiente:

```

Create FUNCTION FN_GR05_pos_libres_fecha (fecha date)
RETURNS table (nro_posicion int, nro_fila int, nro_estanteria int)
AS $$
BEGIN

```

```

RETURN QUERY
    SELECT p.nro_posicion, p.nro_fila, p.nro_estanteria
    FROM gr05_posicion p full join gr05_alquiler_posiciones g05ap
on p.pos_global = g05ap.pos_global and p.nro_posicion =
g05ap.nro_posicion and p.nro_estanteria = g05ap.nro_estanteria and
p.nro_fila = g05ap.nro_fila
    full join gr05_alquiler g05a on g05ap.id_alquiler =
g05a.id_alquiler
    WHERE ((fecha > g05a.fecha_hasta) or (g05ap.id_alquiler is
null)) and p.nro_posicion is not null;
END; $$
LANGUAGE plpgsql;

```

En segundo lugar para realizar la consulta dado un número de cliente que posiciones tiene ocupadas actualmente realizamos una función que retorna el número de posiciones ocupadas del cliente, la estantería de dicha posición y la fila respectivamente. En la consulta de dicha función se utilizaron las siguientes tablas: “gr05_posicion” unida con inner join a “gr05_alquiler_posiciones” y está unida a “gr05_alquiler” respectivamente. La función es la siguiente:

```

Create FUNCTION FN_GR05_pos_ocupadas_cliente (cliente_ingresado
int)
RETURNS table (nro_posicion int, nro_estanteria int, nro_fila int)
AS $$
BEGIN
    RETURN QUERY
    SELECT p.nro_posicion, p.nro_estanteria, p.nro_fila
    FROM gr05_posicion p INNER join gr05_alquiler_posiciones
g05ap on p.pos_global = g05ap.pos_global and p.nro_posicion =
g05ap.nro_posicion and p.nro_estanteria = g05ap.nro_estanteria and
p.nro_fila = g05ap.nro_fila
    INNER join gr05_alquiler g05a on g05ap.id_alquiler =
g05a.id_alquiler
    WHERE ((now() < g05a.fecha_hasta) or (g05ap.id_alquiler is
null)) and g05a.id_cliente=cliente_ingresado;
END; $$
LANGUAGE plpgsql;

```