



Banco de germoplasma de lenteja (*Lens culinaris*)

Ing. Agr. Fernando Maglia

Año: 2022

ÍNDICE

1. INTRODUCCIÓN	3
2. OBJETIVO	4
3. INSTALACIÓN	4
4. ESTRUCTURA	4
4.1. DIAGRAMA DE ENTIDAD RELACIÓN	5
4.2 TABLES	5
4.3 VIEWS	15
4.4 STORED FUNCTIONS	17
4.5 STORED PROCEDURES	19
4.6 TRIGGERS	20

1. INTRODUCCIÓN

Se entiende a germoplasma como toda parte de un ser vivo que tiene capacidad para regenerar a un nuevo individuo, siendo la semilla el ejemplo mas claro, y a un banco de germoplasma como un reservorio activo de material genético cuyo objetivo principal es la conservación de la variabilidad genética de las especies.

Los bancos de germoplasma desempeñan un papel fundamental en la conservación, la disponibilidad y el uso de una amplia diversidad fitogenética para la mejora de los cultivos y con ello la seguridad alimentaria y nutricional. Sirven de puente entre el pasado y el futuro, asegurando la disponibilidad continua de los recursos fitogenéticos para la investigación, la reproducción y la mejora del suministro de semillas para un sistema agrícola sostenible.

Se debe velar para que la identidad de las accesiones (diferentes muestras) de germoplasma se mantenga a lo largo de los distintos procesos, desde la adquisición hasta el almacenamiento, regeneración y distribución; por lo que la correcta identificación de dichas accesiones cumple un papel fundamental en dichos bancos.

Con el fin de garantizar la integridad y comunicación de toda la información, es fundamental registrar todo en una base de datos electrónica, por lo que se plantea en el siguiente trabajo la creación de una base de datos tipo SQL (Structured Query Language) mediante el motor MySQL.

Una base de datos se define como una colección estructurada de datos. SQL tiene la característica de ser una base de datos relacional, siendo estas bases de datos en las cuales los datos se almacenan en tablas separadas en lugar de poner todo en una sola tabla plana, obteniéndose una optimización en la velocidad de procesamiento y en el espacio ocupado en memoria. SQL es el lenguaje estandarizado más común para acceder a bases de datos y se encuentra definido por los estándares ANSI (*American National Standards Institute*) a partir de 1986 y por ISO (*International Standard Organization*) a partir de 1987.

2. OBJETIVO

Crear una base de datos SQL mediante MySQL del banco de germoplasma de lenteja alojado físicamente en el Instituto de Investigaciones de Ciencias Agrarias de Rosario (IICAR) del Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), con localización en la Facultad de Ciencias Agrarias de la Universidad Nacional de Rosario, Zavalla, Santa Fe, Argentina.

3. INSTALACIÓN

La base de datos lens se encuentra disponible en (<https://github.com/ferandmag/lens>) y en el mismo se encuentran 3 archivos:

1) lens_schema.sql: contiene el código necesario para para crear la estructura de la base de datos lens, incluyendo tables, views, stored procedures y triggers.

2) lens_data.sql: contiene el código necesario para insertar datos en el schema creado con lens_schema.sql.

3) lens_der.mwb: contiene un modelo que se puede utilizar para examinar la estructura del schema mediante MySQL workbench.

4. ESTRUCTURA

4.1 Diagrama de entidad relación (DER)

Se presenta el DER (figura 1) realizado con reverse engineer a partir de la base de datos ya creada.

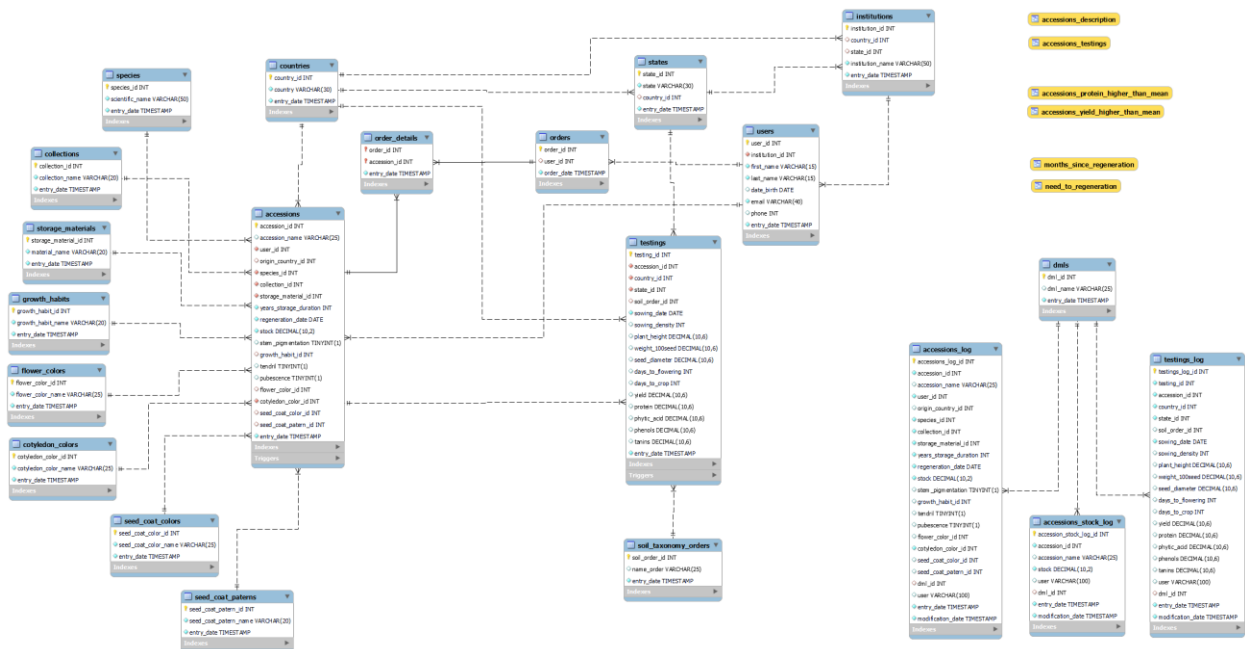


Figura 1: Diagrama de Entidad Relación (DER) del schema lens.

4.2 Tables

Las tablas son objetos de la base de datos que se comportan como contenedores de información las cuáles se encuentran lógicamente organizadas en formato de filas y columnas. Cada fila se considera una entidad que es descrita por las columnas que contienen los atributos o variables de la entidad.

Se define como normalización de una base de datos al proceso de organizar los datos en una base y uno de sus fundamentos principales son: eliminar la información redundante para optimizar el uso de memoria, y no cometer errores en la modificación de datos.

Las tablas se relacionan entre si mediante llaves primarias (PK) y llaveas foráneas (FK), creándose un enlace entre dos tablas donde la PK de una se asocia con la FK de la otra. De las relaciones surge el concepto de cardinalidad, que se define como una restricción en una relación que especifica el número de instancias de entidad que una entidad especifica puede estar relacionada a través de la relación. Las relaciones pueden ser de los siguientes tipos:

- Relación uno a uno (1:1): una entidad de la tabla A puede asociarse a lo sumo con una entidad de la tabla B y viceversa.
- Relación uno a muchos (1:N): una entidad de la tabla A puede asociarse con 2 o más entidades de la tabla B, sin embargo una entidad de la tabla B solo puede asociarse con una entidad de la tabla A.
- Relación muchos a muchos (N:N): una entidad de la tabla A puede asociarse con 2 o más entidades de la tabla B y una entidad de la tabla B puede asociarse con 2 o más entidades de la tabla A.

En cada tabla se especifica la información que contiene y las relaciones si las hubiese (a:b), siendo 'a' la relación de la tabla en cuestión y 'b' la relación de la tabla a la cuál se refiere.

Las tablas dimensionales almacenan información descriptiva y no redundante (que no se encuentra repetida), mientras que las tablas de hecho almacenan información redundante y se utilizan para crear informes a partir de las primeras.

Se define como índices de una tabla a una estructura de datos que contiene punteros a los contenidos de una tabla organizada en un orden específico y su objetivo es optimizar el tiempo de búsqueda.

4.2.1 Dimensionales

A continuación se describen todas las tablas dimensionales presentes en la base de datos. Entre paréntesis luego del nombre de la columna/atributo se menciona el tipo de dato que va a contener dicha columna.

4.2.1 accessions

La tabla accessions contiene información referida a las accesiones y referencia a las siguientes tablas: users (N:1), countries (N:1), species (N:1), collections (N:1), storage_materials (N:1), growth_habits (N:1), cotyledon_colors (N:1), seed_coat_colors (N:1) y seed_coat_patterns (N:1).

La tabla accessions también contiene dos tablas auxiliares denominadas accessions_log y accessions_stock_log, la primera se utiliza como historial frente a

modificaciones de tipo update y delete y la segunda se utiliza como historial frente a modificaciones de tipo update cuando el stock de la accession es menor a 2.5 unidades. Dichas tablas contienen columnas extra referidas a su propio id y otras que indican el usuario que realizó la modificación y su fecha y hora de ejecución.

Columns

- **accession_id** (int) – clave primaria única utilizada para identificar a cada accesión en la tabla.
- **accession_name** (varchar(40)) – identifica el nombre de la accesión.
- **user_id** (int) – clave foránea que referencia a la tabla users, identifica el usuario que introdujo el registro.
- **origin_country_id** (int) – clave foránea que referencia a la tabla countries, identifica el país donde se originó la accesión.
- **species_id** (int) – clave foránea que referencia a la tabla species, identifica la especie de la accesión.
- **collection_id** (int) – clave foránea que referencia a la tabla collections, identifica el tipo de colección de la accesión.
- **storage_material_id** (int) – clave foránea que referencia a la tabla storage_materials, identifica el tipo de material de almacenamiento.
- **years_storage_duration** (int) – identifica la cantidad de años que puede ser almacenada la accesión bajo condiciones ideales de temperatura, humedad y luz.
- **regeneration_date** (date) – identifica la fecha en la cuál fue regenerada la accesión.
- **stock** (decimal(10, 2)) – identifica la cantidad de gramos existentes de la accesión.
- **stem_pigmentation** (boolean) – identifica de forma booleana si el tallo de la accesión presenta ausencia de pigmentación antociánica (0) o presencia (1).
- **growth_habit_id** (int) – clave foránea que referencia a la tabla growth_habits, identifica el hábito de crecimiento de la accesión.
- **tendrils** (boolean) – identifica de forma booleana si la accesión presenta ausencia de zarcillos (0) o presencia (1).

- **pubescence** (boolean) – identifica de forma booleana si la accesión presenta ausencia de pubescencia (0) o presencia (1).
- **flower_color_id** (int) – clave foránea que referencia a la tabla flower_colors, identifica el color de flor de la accesión.
- **cotyledon_color_id** (int) - clave foránea que referencia a la tabla cotyledon_colors, identifica el color de cotiledón de la accesión.
- **seed_coat_color_id** (int) - clave foránea que referencia a la tabla seed_coat_colors, identifica el color del tegumento de la accesión.
- **seed_coat_patern_id** (int) - clave foránea que referencia a la tabla seed_coat_patern, identifica el patrón de tegumento de la accesión.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.2 users

La tabla users contiene información referida a los usuarios y referencia a la tabla institutions (N:1).

El field last_name es un índice.

Columns

- **user_id** (int) - clave primaria única utilizada para identificar a cada usuario en la tabla.
- **institution_id** (int) – clave foránea que referencia a la tabla institutions, identifica la institución a la cual pertenece el usuario.
- **first_name** (varchar(15)) – identifica el primer nombre del usuario.
- **last_name** (varchar(15)) – identifica el apellido del usuario.
- **date_birth** (date) – identifica la fecha de nacimiento del usuario.
- **email** (varchar(40)) – identifica la dirección de correo electrónico del usuario.
- **phone** (varchar(15)) – identifica el número de teléfono del usuario.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.3 countries

La tabla countries contiene información referida a los países.

Columns

- **country_id** (int) - clave primaria única utilizada para identificar a cada país en la tabla.
- **country_name** (varchar(30)) – identifica el nombre del país.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.4 states

La tabla states contiene información referida a los estados de los países y referencia a la tabla countries (N:1).

Columns

- **state_id** (int) - clave primaria única utilizada para identificar a cada estado en la tabla.
- **state_name** (varchar(30)) – identifica el nombre del país.
- **country_id** (int) - clave foránea que referencia a la tabla countries, identifica el país al cuál pertenece el estado.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.5 species

La tabla species contiene información referida al nombre científico de la especie.

Columns

- **species_id** (int) - clave primaria única utilizada para identificar a cada especie en la tabla.
- **scientific_name** (varchar(50)) – identifica el nombre científico de la especie, conteniendo sub especie si la hubiese.

- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.6 testings

La tabla testings contiene información del área donde se realizó la evaluación y sus resultados. Referencia a las tablas: accessions (1:N), countries (N:1), states (N:1) y soil_taxonomy (N:1).

La tabla testings también contiene una tabla auxiliar denominada testing_log que se utiliza como historial frente a modificaciones de tipo update y delete. Dichas tablas contienen columnas extra referidas a su propio id y otras que indican el usuario que realizó la modificación y su fecha y hora de ejecución.

Los fields weight_100seed y yield son índice.

Columns

- **testing_id** (int) - clave primaria única utilizada para identificar a cada especie en la tabla.
- **accession_id** (int) - clave foránea que referencia a la tabla accessions, identifica la accesión evaluada.
- **country_id** (int) - clave foránea que referencia a la tabla countries, identifica el país en donde se realizó la evaluación.
- **state_id** (int) - clave foránea que referencia a la tabla states, identifica el estado donde se realizó la evaluación.
- **soil_taxonomy_id** (int) - clave foránea que referencia a la tabla soil_taxonomy, identifica el orden de suelo donde se realizó la evaluación.
- **sowing_date** (date) – identifica la fecha de siembra.
- **sowing_density** (int) – identifica la densidad de siembra en plantas/m².
- **plant_height** (decimal(10, 6)) – identifica la altura de planta medida en centímetros.
- **weight_100_seed** (decimal(10, 6)) – identifica el peso de las 100 semillas promedio expresado en unidad de gramos.

- **seed_diameter** (decimal(10, 6)) – identifica el diámetro de semilla promedio expresado en unidades de milímetros.
- **days_to_flowering** (int) – identifica los días a floración desde la siembra, expresado en días.
- **days_to_crop** (int) – identifica los días a cosecha desde la siembra, expresado en días.
- **yield** (decimal(10, 6)) – identifica el rendimiento de la accesión expresado en unidades de kg/ha.
- **protein** (decimal(10, 6)) – identifica el porcentaje de proteína de la accesión.
- **phytic_acid** (decimal(10, 6)) – identifica el porcentaje de ácido fítico de la accesión.
- **phenols** (decimal(10, 6)) – identifica el porcentaje de fenoles de la accesión.
- **tanins** (decimal(10, 6)) – identifica el porcentaje de taninos de la accesión.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.7 soil_taxonomy

La tabla soil_taxonomy contiene información de la clasificación taxonómica de suelos, propuesta por Guy Smith (1975) en cuanto al nivel de orden.

Columns

- **soil_order_id** (int) - clave primaria única utilizada para identificar a cada orden de suelo en la tabla.
- **name_order** (varchar(50)) – identifica el nombre del orden del suelo.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.8 collections

La tabla collections contiene información del tipo de colección almacenada.

Columns

- **collection_id** (int) - clave primaria única utilizada para identificar a cada colección en la tabla.
- **name_collection** (varchar(25)) – identifica el nombre de la colección.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.9 storage_materials

La tabla storage_materials contiene información referida al tipo de material almacenado.

Columns

- **storage_material_id** (int) - clave primaria única utilizada para identificar a cada material de almacenamiento en la tabla.
- **material_name** (varchar(25)) – identifica el nombre del material de almacenamiento.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.10 growth_habits

La tabla growth_habits contiene información referida al hábito de crecimiento.

Columns

- **growth_habit_id** (int) - clave primaria única utilizada para identificar a hábito de crecimiento en la tabla.
- **growth_habit_name** (varchar(25)) – identifica el nombre del hábito de crecimiento.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.11 flower_colors

La tabla flower_colors contiene información referida al color de flor.

Columns

- **flower_color_id** (int) - clave primaria única utilizada para identificar el color de flor en la tabla.
- **flower_color_name** (varchar(25)) – identifica el color de flor.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.12 cotyledon_colors

La tabla cotyledon_colors contiene información referida al color de cotiledón.

Columns

- **cotyledon_color_id** (int) - clave primaria única utilizada para identificar el color de cotiledon en la tabla.
- **cotyledon_color_name** (varchar(25)) – identifica el color del cotiledón.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.13 seed_coat_colors

La tabla seed_coat_colors contiene información referida al color de tegumento de semilla.

Columns

- **seed_coat_color_id** (int) - clave primaria única utilizada para identificar el color del tegumento de la semilla en la tabla.
- **seed_coat_color_name** (varchar(25)) – identifica el color del tegumento de la semilla.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.14 seed_coat_patterns

La tabla seed_coat_patterns contiene información referida al patrón de tegumento de semilla.

Columns

- **seed_coat_patern_id** (int) - clave primaria única utilizada para identificar el patrón del tegumento de la semilla en la tabla.
- **seed_coat_color_name** (varchar(25)) – identifica el patrón del tegumento de la semilla.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.15 institutions

La tabla institutions contiene información referida a las instituciones y referencia a las tablas countries (N:1) y states (N:1).

El field institution_name es un índice.

Columns

- **institution_id** (int) - clave primaria única utilizada para identificar la institución en la tabla.
- **country_id** (int) - clave foránea que referencia a la tabla countries, identifica el país en donde se encuentra la casa matriz de la institución.
- **state_id** (int) - clave foránea que referencia a la tabla states, identifica el estado en donde se encuentra la casa matriz de la institución.
- **institution_name** (varchar(50)) – nombre de la institución.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.16 orders

La tabla orders contiene información referida a las ordenes o peticiones de accesiones y referencia a las tabla users (1:N).

Columns

- **order_id** (int) - clave primaria única utilizada para identificar la orden en la tabla.
- **user_id** (int) - clave foránea que referencia a la tabla users, identifica al usuario que realiza la orden.

- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.17 order_details

La tabla order_details es usada para dar soporte a la relación muchos a muchos entre las tablas accessions y orders. Para cada orden, habrá un registro en la tabla order_detail que enumere el orden y la accesión.

Las claves foráneas **order_id** y **accesion_id** conforman la clave primaria compuesta de la tabla order_detail.

Columns

- **order_id** (int) – clave foránea que referencia a la tabla orders, identifica a la orden.
- **accesion_id** (int) – clave foránea que referencia a la tabla accessions identifica a la accesión a la cuál pertenece la orden.
- **quantity** (int) – cantidad expresada en gramos de la accession.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.2.18 dmls

La tabla dmls contiene información referida a las acciones de *Data Manipulation Language* (DML) y se referencia en las tablas accessions_log, accessions_stock_log y testings_log.

- **dml_id** (int) – clave primaria única utilizada para identificar el dml en la tabla.
- **dml_name** (int) – nombre del tipo de acción de manipulación de tablas.
- **entry_date** (timestamp) – identifica la fecha y hora en la cual el registro fue creado o se modificó.

4.3 Views

Las vistas son tablas que no existen físicamente sino que se forman mediante selección y/o filtrado de los componentes de otras tablas y son una forma lógica de ver

los datos físicos ubicados en las tablas. El fundamento de las vistas es no tener que realizar la codificación de consultas de uso frecuente.

A su vez las vistas también permiten ocultar información a usuarios que no tienen acceso a ciertos campos ya que en muchos casos se puede almacenar información confidencial.

La base de datos cuenta con las siguientes vistas:

- **accessions_description:** resume en una tabla las características cualitativas de las accesiones. Las columnas presentes son: accession_name, origin_country, collection_name, storage_material, stem_pigmentation, tendril, growth_habit_name, pubescence, flower_color_name, cotyledon_color_name, seed_coat_color_name y seed_coat_pattern_name.
- **months_since_regeneration:** resume en una tabla los meses que transcurrieron desde la fecha de regeneración hasta la fecha actual de todas las accesiones presentes. Las columnas presentes son: accession_name, storage_material, years_storage_duration y months_since_regeneration; la misma se presenta ordenada en forma decreciente en función de la columna months_since_regeneration.
- **need_to_regeneration:** resume en una tabla las accesiones que hay que regenerar debido a que la diferencia entre la fecha actual y la de regeneración es superior a la cantidad de años que la accesión se puede almacenar. Las columnas presentes son: accession_name, storage_material, years_storage_duration y years_since_regeneration; la misma se presenta ordenada en forma decreciente en función de la columna years_since_regeneration.
- **accessions_testings:** resume en una tabla los datos de evaluación de todas las accesiones. Las columnas presentes son: accession_name, country, state, soil_order, sowing_date, sowing_density, plant_height, weight_100seed, seed_diameter, days_to_flowering, days_to_crop, yield, protein, phytic_acid y phenols, tanins.

- **accessions_yield_higher_than_mean:** resume en una tabla las accesiones que tuvieron un rendimiento superior al rendimiento promedio de todas las evaluaciones. Las columnas presentes son: accession_name y yield; la misma se presenta ordenada en forma decreciente en función de la columna yield.
- **accessions_protein_higher_than_mean:** resume en una tabla las accesiones que tuvieron un % de proteína superior al porcentaje promedio de todas las evaluaciones. Las columnas presentes son: accession_name y protein; la misma se presenta ordenada en forma decreciente en función de la columna protein.

4.4 Stored functions

La base de datos cuenta con las siguientes funciones creadas:

- **fn_minMaxStandardization:** se puede aplicar a las columnas yield, phytic_acid, phenols y tanins. Toma dos parámetros, primero el nombre de la columna a la cual se desea aplicar la función y luego el mismo nombre entre comillas simples y retorna el valor estandarizado según la metodología scaling:

$$X' = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Siendo:

- X': valor estandarizado.
- X: valor i de X.
- X_{min}: mínimo.
- X_{max}: máximo

Ejemplo de uso:

```
select accession_name, yield, fn_minMaxStandardization(yield, 'yield')
from testings
```

```
inner join    accessions
on           testings.accession_id = accessions.accession_id
;
```

- **fn_normalization:** se puede aplicar a las columnas yield, phytic_acid, phenols y tanins. Toma dos parámetros, primero el nombre de la columna a la cual se desea aplicar la función y luego el mismo nombre de la columna entre comillas simples y retorna el valor normalizado según la metodología puntuación estándar:

$$X' = \frac{X - \mu}{\sigma}$$

Siendo:

- X': valor normalizado.
- X: valor i de X.
- μ : media.
- σ : desvío estándar.

Ejemplo de uso:

```
select accession_name, phenols, fn_normalization(phenols, 'phenols')
from   testings
inner join    accessions
on           testings.accession_id = accessions.accession_id
;
```

- **fn_sowingDensity:** toma como parámetro la columna weight_100seed de la tabla testings, la densidad de siembra objetivo en plantas/m² y el coeficiente de logro en porcentaje y retorna un int que representa los kg de semilla a sembrar sobre hectárea (1 hectárea equivale a 10000 m²):

$$\text{Kg semilla/hectárea} = (\delta m^2 * p1000) / CL$$

Siendo:

- Kg semilla/hectárea: cantidad de kg de semillas necesarias para lograr la densidad objetivo.

- δm^2 : plantas objetivo sobre metro cuadrado.
- p1000: peso de las 1000 semillas en gramos.
- CL: coeficiente de logro, siendo un valor entre 0 y 100.

Ejemplo de uso:

```
select accessions.accession_name, fn_sowingDensity(weight_100seed, 200, 75)
as kgsemilla_hectarea
from testings
inner join accessions
on testings.accession_id = accessions.accession_id
;
```

4.5 Stored procedures

La base de datos cuenta con los siguientes stored procedures:

- **sp_addAccession:** es útil para agregar una accession a la tabla accessions sin tener la necesidad de usar un insert into. Tiene la ventaja que, si la accession se encuentra presente, no la agrega y aparece un mensaje de advertencia indicando que ya se encuentra en la tabla.
No presenta parámetros de salida y los de entrada son: p_accession_name varchar(25), p_user_id int, p_origin_country_id int, p_species int, p_collection_id int, p_storage_material_id int, p_years_storage_duration int, p_regeneration_date date, p_stock decimal(10, 2), p_stem_pigmentation boolean, p_growth_habit_id int, p_tendrill boolean, p_pubescence boolean, p_flower_color_id int, p_cotyledon_color_id int, p_seed_coat_color_id int, p_seed_coat_pattern_id int.
- **sp_addTesting:** es útil para agregar un testing a la tabla testings sin tener la necesidad de usar un insert into. Tiene la ventaja que, si la accession a la cuál se le quiere agregar el testing no se encuentra presente en la tabla accessions, no lo agrega y aparece un mensaje de advertencia indicando que la accession no se encuentra en la tabla accessions.
No presenta parámetros de salida y los de entrada son: p_accession_id int, p_country_id int, p_state_id int, p_soil_order_id int, p_sowing_date date,

p_sowing_density int, p_plant_height decimal(10, 6), p_weight_100seed decimal(10, 6), p_seed_diameter decimal(10, 6), p_days_to_flowering int, p_days_to_crop int, p_yield decimal(10, 6), protein decimal(10, 6), p_phytic_acid decimal(10, 6), p_phenols decimal(10, 6), p_tanins decimal(10, 6).

- **sp_addOrder:** es útil para agregar una order con su respectiva order_detail a las tablas orders y order_detail sin tener la necesidad de usar un insert into. Tiene las siguientes ventajas: 1) verifica que todas las accessions solicitadas existan en la tabla accessions y si alguna no esta presente, se genera un mensaje de advertencia indicando la primera accession ausente; 2) verifica que todas las accessions solicitadas tengan un stock superior a 2.5 y si alguna no cumple la condición se genera un mensaje de advertencia indicando la primera accession que no la cumple y 3) verifica que no haya accessions solicitadas repetidas y si alguna se encuentra mas de una vez, se genera un mensaje de advertencia indicando la primera ocurrencia repetida. Si alguna de las condiciones anteriores no se cumple, no se agregan registros a las tablas orders ni order_details.

No presenta parámetros de salida y los de entrada son: p_user_id int, p_accessions varchar(500), siendo este último un varchar que consiste en los accession_id que se deseen solicitar separados por comas (ej: '1, 21, 30').

El stock que se otorga es siempre de 1 gr del material almacenado en cuestión.

4.6 Triggers

La base de datos cuenta con los siguientes triggers:

- **accessions_update_log:** cuando se actualiza un registro de la tabla accessions, su estado previo a la actualización se almacena en la tabla accessions_log, que aparte de contener con todas las columnas de la tabla accessions, cuenta con: accessions_log_id, dml_id, user y modification_date.

- **accessions_delete_log:** cuando se elimina un registro de la tabla accessions, su estado previo al borrado se almacena en la tabla accessions_log, que aparte de contener con todas las columnas de la tabla accessions, cuenta con: accessions_log_id, dml_id, user y modification_date.
- **testings_update_log:** cuando se actualiza un registro de la tabla testings, su estado previo a la actualización se almacena en la tabla testings_log, que aparte de contener con todas las columnas de la tabla testings, cuenta con: testings_log_id, dml_id, user y modification_date.
- **testings_delete_log:** cuando se elimina un registro de la tabla testings, su estado previo al borrado se almacena en la tabla testings_log, que aparte de contener con todas las columnas de la tabla testings, cuenta con: testings_log_id, dml_id, user y modification_date.
- **testings_update_log:** cuando se actualiza el stock de un registro de la tabla accessions y el mismo es menor a 2.5 unidades, su estado posterior a la actualización se almacena en la tabla accessions_stock_log, que contiene las columnas accesión_stock_log_id, accession_id, accession_name, stock, user, dml_id, entry_date y modification_date.