INFORME FINAL DE BASE DE DATOS APP\_RAICES

PRESENTADO POR:

NENFAR ERNEY CHAPID CORDOBA

PRESENTADO A:

ING. BRAYAN IGNACIO ARCOS BURBANO

INSTITUTO TECNOLÓGICO DEL PUTUMAYO

DESARROLLO DE BASES DE DATOS

SEMESTRE V

06/12/2024

Tabla de contenido

[Resumen Ejecutivo 5](#_Toc184378222)

[Introducción 6](#_Toc184378223)

[Metodología 7](#_Toc184378224)

[Herramientas Utilizadas 7](#_Toc184378225)

[Procedimientos 7](#_Toc184378226)

[Desarrollo 9](#_Toc184378227)

[Descripción de la Base de Datos 9](#_Toc184378228)

[Esquema de la Base de Datos app\_raices 9](#_Toc184378229)

[Tablas y Atributos 9](#_Toc184378230)

[Consultas SQL 14](#_Toc184378231)

[Consultas Básicas 14](#_Toc184378232)

[Consultas Avanzadas. 17](#_Toc184378233)

[INNER JOIN 17](#_Toc184378234)

[LEFT JOIN 25](#_Toc184378235)

[RIGHT JOIN 32](#_Toc184378236)

[CROSS JOIN 37](#_Toc184378237)

[Subconsultas 41](#_Toc184378238)

[WHERE 41](#_Toc184378239)

[FROM 43](#_Toc184378240)

[SELECT 44](#_Toc184378241)

[IN 45](#_Toc184378242)

[EXISTS 46](#_Toc184378243)

[ANY y ALL 47](#_Toc184378244)

[VISTAS 48](#_Toc184378245)

[PROCEDIMIENTOS ALMACENADOS 54](#_Toc184378246)

[TRIGGERS 60](#_Toc184378247)

[ÍNDICES 64](#_Toc184378248)

[Índices Agrupados 64](#_Toc184378249)

[Índices No Agrupados 64](#_Toc184378250)

[Índices Únicos 66](#_Toc184378251)

[Índices Compuestos 68](#_Toc184378252)

[Diseño de Base de Datos 70](#_Toc184378253)

[Modelo de Datos: 70](#_Toc184378254)

[Consideraciones de Diseño 70](#_Toc184378255)

[Relaciones y Cardinalidad 70](#_Toc184378256)

[Normalización 72](#_Toc184378257)

[Procedimientos de Respaldo y Recuperación 73](#_Toc184378258)

[Respaldo en MySQL 73](#_Toc184378259)

[Respaldo en base de datos app\_raices 74](#_Toc184378260)

[Recuperación en base de datos app\_raices 75](#_Toc184378261)

[Análisis y Discusión 78](#_Toc184378262)

[Interpretación de Resultados 80](#_Toc184378263)

[Conclusiones 81](#_Toc184378264)

[Recomendaciones 82](#_Toc184378265)

[Referencias 83](#_Toc184378266)

# Resumen Ejecutivo

El presente informe documenta la base de datos diseñada para la aplicación de aprendizaje de lenguas indígenas del Putumayo, una herramienta que busca preservar y fomentar la enseñanza de idiomas nativos en riesgo de extinción. Este resumen ejecutivo ofrece una visión general de su estructura, diseño, consultas SQL implementadas y procedimientos de gestión.

# Introducción

La diversidad lingüística del Putumayo enfrenta retos significativos debido a la disminución del número de hablantes nativos. En respuesta, se desarrolló una base de datos como parte de una aplicación educativa destinada a ofrecer recursos interactivos para el aprendizaje de lenguas indígenas, promoviendo su preservación y fortalecimiento.

# Metodología

## Herramientas Utilizadas

* **Gestor de Base de Datos:** MySQL
* **Entorno de Desarrollo:** MySQL Workbench
* **Lenguaje:** SQL

## Procedimientos

1. **Métodos y pasos seguidos para llevar a cabo el análisis o el trabajo**
   1. **Análisis de Requisitos**

* **Objetivo:** Comprender las necesidades del sistema.
* **Actividades:**
  + Capacitación en el aula de clases sobre la creación de la base de datos con respecto a las necesidades.
  1. **Diseño Conceptual y Lógico**
* **Objetivo:** Crear un modelo entidad-relación (ERD) que represente la estructura de datos.
* **Actividades:**
  + - Diseño del esquema inicial.
    - Normalización hasta la tercera forma normal (3FN).
    - Identificación de relaciones y cardinalidades.
  1. **Implementación de la Base de Datos**
  + **Objetivo:** Traducir el diseño lógico a un esquema físico en MySQL.
  + **Actividades:**
    - Creación de tablas, índices y claves foráneas.
    - Configuración de restricciones para garantizar la integridad de los datos.
  1. **Desarrollo de Consultas SQL**
  + **Objetivo:** Diseñar consultas que respondan a necesidades específicas de información.
  + **Actividades:**
    - Redacción de consultas básicas y avanzadas.
    - Pruebas de ejecución y optimización de rendimiento.
  1. **Pruebas y Validación**
* **Objetivo:** Garantizar que el sistema funcione según lo esperado.
* **Actividades:**
  + - Pruebas unitarias en cada tabla de la base de datos.
    - Validación de resultados de consultas SQL.
  1. **Documentación**
* **Objetivo:** Proporcionar una guía clara y detallada sobre el diseño e implementación.
* **Actividades:**
* Redacción del manual de base de datos en Microsoft Word.

# Desarrollo

## Descripción de la Base de Datos

### Esquema de la Base de Datos app\_raices

Esta base de datos está diseñada para una aplicación que podría estar relacionada con un sistema de gestión de usuarios, lecciones, contenidos, suscripciones y pagos. A continuación, te doy una descripción de las tablas y sus atributos, así como las relaciones entre ellas, la cardinalidad y el nivel de normalización de cada una.

### **Tablas y Atributos**

1. **users**
   * **idUser** (INT): Identificador único del usuario.
   * **mail** (VARCHAR): Correo electrónico del usuario.
   * **password** (VARCHAR): Contraseña del usuario.
   * **createdAt** (DATETIME): Fecha y hora de creación del registro.
   * **updatedAt** (DATETIME): Fecha y hora de la última actualización del registro.
2. **gender**
   * **idGender** (INT): Identificador único de género.
   * **name** (VARCHAR): Nombre del género (por ejemplo, 'Masculino', 'Femenino').
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
3. **people**
   * **idPeople** (INT): Identificador único de la persona.
   * **name** (VARCHAR): Nombre de la persona.
   * **lastName** (VARCHAR): Apellido de la persona.
   * **birthdate** (DATE): Fecha de nacimiento de la persona.
   * **profilePicture** (VARCHAR): URL de la imagen de perfil de la persona.
   * **idUser** (INT): Relación con la tabla users (clave foránea).
   * **idGender** (INT): Relación con la tabla gender (clave foránea).
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
4. **roles**
   * **idRol** (INT): Identificador único del rol.
   * **name** (VARCHAR): Nombre del rol (por ejemplo, 'admin', 'student').
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
5. **userRoles**
   * **idUserRol** (INT): Identificador único de la relación entre usuario y rol.
   * **idUser** (INT): Relación con la tabla users (clave foránea).
   * **idRol** (INT): Relación con la tabla roles (clave foránea).
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
6. **languages**
   * **idLanguage** (INT): Identificador único del idioma.
   * **name** (VARCHAR): Nombre del idioma.
   * **community** (VARCHAR): Comunidad lingüística asociada.
   * **description** (TINYTEXT): Descripción del idioma.
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
7. **levels**
   * **idLevel** (INT): Identificador único del nivel.
   * **name** (VARCHAR): Nombre del nivel (por ejemplo, 'Principiante', 'Intermedio', 'Avanzado').
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
8. **lessons**
   * **idLesson** (INT): Identificador único de la lección.
   * **idLanguage** (INT): Relación con la tabla languages (clave foránea).
   * **idLevel** (INT): Relación con la tabla levels (clave foránea).
   * **title** (VARCHAR): Título de la lección.
   * **description** (TINYTEXT): Descripción de la lección.
   * **isPremium** (BOOLEAN): Si la lección es premium o no.
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
9. **userLesson**
   * **idUserLesson** (INT): Identificador único de la relación entre usuario y lección.
   * **idUser** (INT): Relación con la tabla users (clave foránea).
   * **idLesson** (INT): Relación con la tabla lessons (clave foránea).
   * **score** (INT): Calificación obtenida en la lección.
   * **lastAccess** (DATETIME): Última vez que el usuario accedió a la lección.
   * **createdAt** (DATETIME): Fecha de creación.
   * **updatedAt** (DATETIME): Fecha de actualización.
10. **mediaFiles**
    * **idMediaFile** (INT): Identificador único del archivo de medios.
    * **media** (VARCHAR): Ruta o nombre del archivo de medios (por ejemplo, imagen, video).
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
11. **content**
    * **idContent** (INT): Identificador único del contenido.
    * **idUser** (INT): Relación con la tabla users (clave foránea).
    * **idLesson** (INT): Relación con la tabla lessons (clave foránea).
    * **idMediaFile** (INT): Relación con la tabla mediaFiles (clave foránea).
    * **url** (VARCHAR): URL del contenido (por ejemplo, enlace a un archivo o video).
    * **description** (TINYTEXT): Descripción del contenido.
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
12. **state**
    * **idState** (INT): Identificador único del estado.
    * **name** (VARCHAR): Nombre del estado (por ejemplo, 'Completado', 'En progreso').
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
13. **learningProgress**
    * **idProgress** (INT): Identificador único del progreso de aprendizaje.
    * **idUser** (INT): Relación con la tabla users (clave foránea).
    * **idLesson** (INT): Relación con la tabla lessons (clave foránea).
    * **idState** (INT): Relación con la tabla state (clave foránea).
    * **initDate** (DATETIME): Fecha de inicio del progreso.
    * **endDate** (DATETIME): Fecha de finalización del progreso.
    * **score** (INT): Puntuación obtenida en la lección.
    * **attempCount** (INT): Número de intentos realizados.
    * **timeSpent** (TIME): Tiempo dedicado al aprendizaje.
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
14. **subscriptions**
    * **idSubscription** (INT): Identificador único de la suscripción.
    * **idUser** (INT): Relación con la tabla users (clave foránea).
    * **subscriptionType** (VARCHAR): Tipo de suscripción (por ejemplo, 'Premium', 'Gratis').
    * **startDate** (DATE): Fecha de inicio de la suscripción.
    * **endDate** (DATE): Fecha de finalización de la suscripción.
    * **status** (VARCHAR): Estado de la suscripción (por ejemplo, 'Activa', 'Cancelada').
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
15. **paymentsGateways**
    * **idPaymentsGateway** (INT): Identificador único de la pasarela de pago.
    * **gateway** (VARCHAR): Nombre de la pasarela de pago (por ejemplo, 'PayPal', 'Stripe').
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
16. **paymentMethods**
    * **idPaymentMethod** (INT): Identificador único del método de pago.
    * **methodName** (VARCHAR): Nombre del método de pago (por ejemplo, 'Tarjeta de crédito', 'Transferencia bancaria').
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.
17. **payments**
    * **idPayment** (INT): Identificador único del pago.
    * **paymentCode** (VARCHAR): Código de pago único.
    * **idUser** (INT): Relación con la tabla users (clave foránea).
    * **idSubscription** (INT): Relación con la tabla subscriptions (clave foránea).
    * **idPaymentMethod** (INT): Relación con la tabla paymentMethods (clave foránea).
    * **idPaymentsGateway** (INT): Relación con la tabla paymentsGateways (clave foránea).
    * **status** (VARCHAR): Estado del pago (por ejemplo, 'Completado', 'Pendiente').
    * **paymentDate** (DATETIME): Fecha de pago.
    * **amount** (DECIMAL): Monto del pago.
    * **createdAt** (DATETIME): Fecha de creación.
    * **updatedAt** (DATETIME): Fecha de actualización.

## Consultas SQL

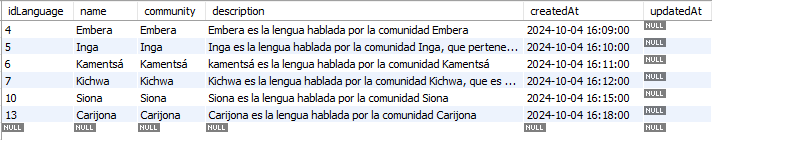
### Consultas Básicas

1. **Buscar lenguas que terminan con la letra 'a':**



Esta consulta busca todas las lenguas cuyo nombre termina con la letra 'a'. La cláusula LIKE utiliza el comodín % para coincidir con cualquier secuencia de caracteres que preceda a la letra 'a'.

**Resultados**:



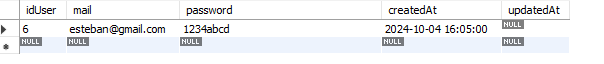
1. **Obtener los datos del usuario registrado con el identificador número 6:**



**Descripción:**

1. **SELECT \***: Esta parte de la consulta selecciona todos los campos o columnas de la tabla. El asterisco (\*) es un comodín que indica que se deben recuperar todas las columnas de los registros que coincidan con la condición especificada en la cláusula WHERE.
2. **FROM usuarios**: Especifica la tabla desde la que se realizarán las búsquedas. En este caso, la tabla es usuarios. Esta es la tabla que contiene la información de los usuarios.
3. **WHERE usuarios.id = 6**: La cláusula WHERE es utilizada para establecer una condición que los registros deben cumplir para ser seleccionados. En este caso, la condición es que el valor del campo id en la tabla usuarios debe ser igual a 6.
   * Esto significa que la consulta buscará el registro o los registros de la tabla usuarios cuyo campo id tenga el valor 6.

**Resultados**:





**Descripción:**

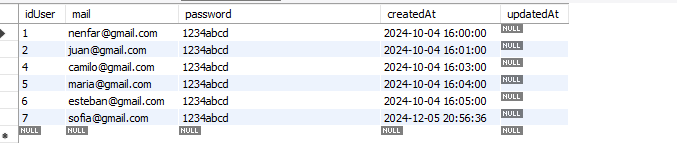
1. **Tabla consultada**:

SELECT \* FROM users

* + Recupera todas las columnas (\*) de la tabla users.

1. **Condiciones de filtro (WHERE)**:
   * **users.idUser <> 3**:
     + Excluye los registros donde el valor del campo idUser sea igual a 3.
     + Se seleccionan todos los demás usuarios excepto el que tenga el identificador 3.
   * **users.mail IS NOT NULL**:
     + Asegura que el campo mail no sea NULL.
     + Solo se seleccionan usuarios que tengan un correo electrónico definido.
   * **users.password IS NOT NULL**:
     + Asegura que el campo password no sea NULL.
     + Solo se seleccionan usuarios que tengan una contraseña definida.
2. **Combinación de condiciones**:
   * Las condiciones están unidas por el operador lógico AND, lo que significa que **todas las condiciones deben cumplirse simultáneamente** para que un registro sea incluido en el resultado.

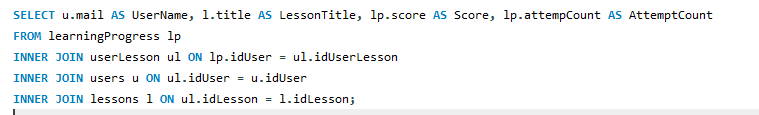
**Resultados**:



### Consultas Avanzadas.

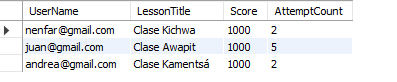
### INNER JOIN

1. **Consultar el progreso y número de intentos de usuarios por lección:**

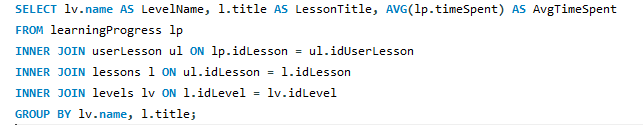
****

**Descripción**: Esta consulta obtiene el progreso de los usuarios en cada lección, incluyendo su puntaje (score) y el número de intentos (attempCount). Se hace un INNER JOIN entre las tablas learningProgress, userLesson, users y lessons para unir los datos relacionados con los usuarios, las lecciones y su progreso en ellas. La consulta muestra el nombre del usuario, el título de la lección, el puntaje obtenido y el número de intentos realizados por el usuario en cada lección.

**Resultados:**

****

1. **Tiempo promedio invertido por lección en cada nivel:**

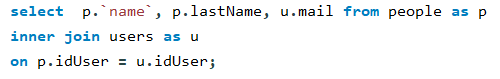
****

**Descripción**: Esta consulta calcula el tiempo promedio invertido por lección, agrupado por el nivel de la lección. Utiliza la función de agregación AVG() para calcular el tiempo promedio (timeSpent) que los usuarios han dedicado a cada lección. La consulta une las tablas learningProgress, userLesson, lessons y levels para obtener la información necesaria sobre el nivel, la lección y el tiempo invertido. Los resultados se agrupan por el nombre del nivel (lv.name) y el título de la lección (l.title).

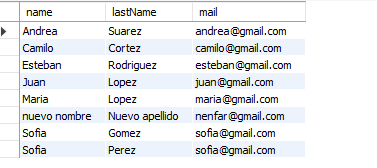
**Resultados:**

****

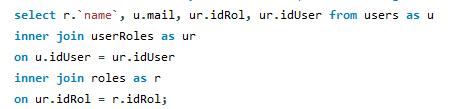
1. **Obtener los nombres y apellidos de los usuarios junto con sus correos electrónicos.**

**Descripción**: Esta consulta realiza un INNER JOIN entre las tablas people y users utilizando el campo idUser como clave de unión. Se obtienen los campos name y lastName de la tabla people, y el campo mail de la tabla users para cada usuario.

**Resultados**:

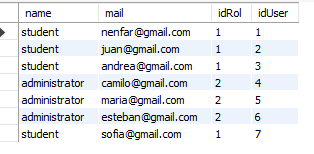


1. **Listar todos los roles y los usuarios que tienen asignados esos roles.**

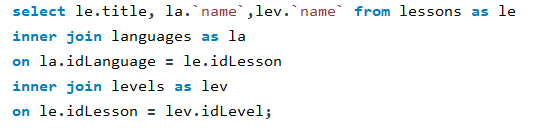
****

**Descripción**: Esta consulta hace una unión entre las tablas users, userRoles y roles. Relaciona los usuarios con los roles que tienen asignados a través de la tabla intermedia userRoles. Muestra los nombres de los roles, los correos electrónicos de los usuarios, y los identificadores de los roles y usuarios.

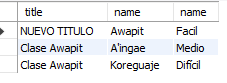
**Resultados**:



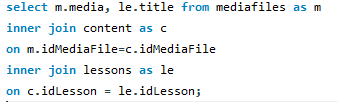
1. **Mostrar las lecciones junto con el nombre del idioma y el nivel.**

**Descripción**: Esta consulta une las tablas lessons, languages y levels para mostrar el título de cada lección, el nombre del idioma y el nombre del nivel asociado a la lección. La relación entre las tablas se hace a través de los campos idLanguage y idLevel.

**Resultados**:

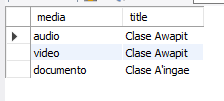


1. **Listar todos los contenidos junto con el título de la lección.**

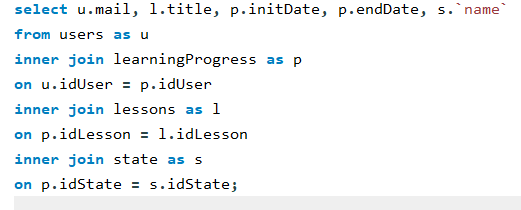
****

**Descripción**: Esta consulta obtiene una lista de archivos multimedia (media) y el título de la lección asociada a cada contenido. Relaciona las tablas mediaFiles, content y lessons usando las claves de unión correspondientes (idMediaFile y idLesson).

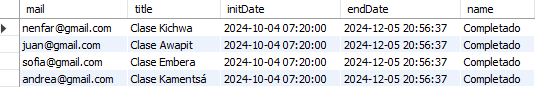
**Resultados**:



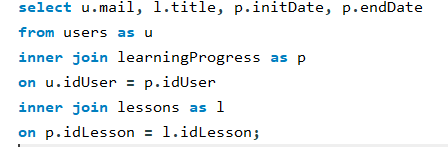
1. **Obtener los progresos de los usuarios en las lecciones, incluyendo el estado.**

**Descripción**: Esta consulta obtiene el progreso de los usuarios en las lecciones, incluyendo el estado del progreso. Hace un INNER JOIN entre las tablas users, learningProgress, lessons y state, utilizando los campos de identificación de usuario, lección y estado.

**Resultados**:

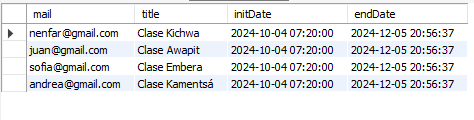


1. **Mostrar el progreso de los usuarios y la lección asociada.**

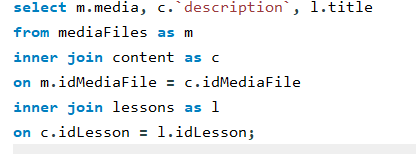


**Descripción**: Similar a la consulta anterior, pero sin incluir el estado. Muestra el progreso de los usuarios en las lecciones, obteniendo los campos mail de users, title de lessons y las fechas de inicio y fin del progreso, uniendo las tablas users, learningProgress y lessons.

**Resultados**:

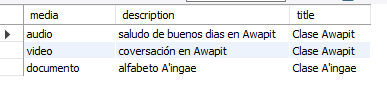


1. **Obtener los archivos multimedia junto con sus descripciones y la lección asociada.**

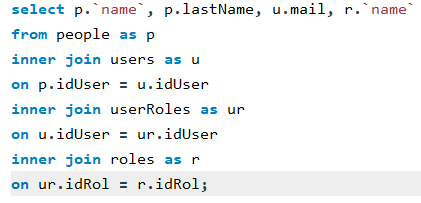


**Descripción**: Esta consulta relaciona las tablas mediaFiles, content y lessons para mostrar los archivos multimedia junto con su descripción y el título de la lección asociada.

**Resultados**:

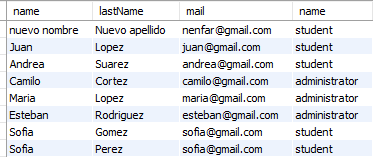


1. **Listar los usuarios y los roles que tienen.**

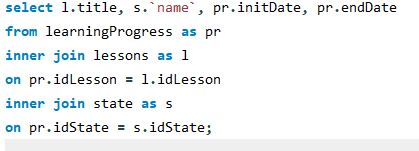


**Descripción**: Esta consulta obtiene los usuarios y los roles que tienen asignados, al hacer un INNER JOIN entre las tablas people, users, userRoles y roles. Muestra los nombres de los usuarios, sus correos electrónicos y los roles asignados.

**Resultados**:

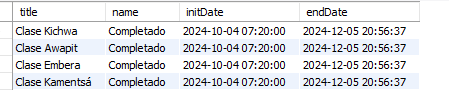


1. **Mostrar el estado de progreso de las lecciones.**

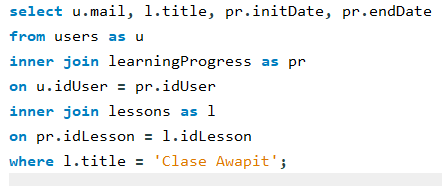


**Descripción**: Muestra el progreso de las lecciones y su estado. Realiza un INNER JOIN entre las tablas learningProgress, lessons y state para obtener el título de la lección, el estado y las fechas de progreso.

**Resultados**:

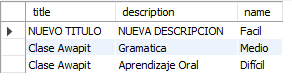


1. **Obtener la lista de usuarios y el progreso en lecciones específicas.**



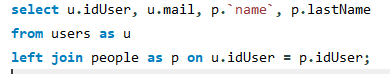
**Descripción**: Muestra los usuarios y su progreso en una lección específica llamada "Clase Awapit". Se relacionan las tablas users, learningProgress y lessons, filtrando por el título de la lección.

**Resultados**:

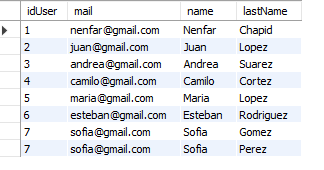


### LEFT JOIN

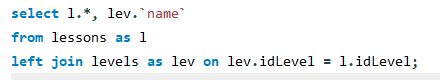
1. **Usuarios y sus datos personales, incluso si no tienen un registro en la tabla** people:



**Descripción**:  
Esta consulta obtiene todos los usuarios y sus datos personales (nombre y apellido) de la tabla people. Si un usuario no tiene un registro en people, igualmente será incluido, pero los campos correspondientes a name y lastName aparecerán como NULL.

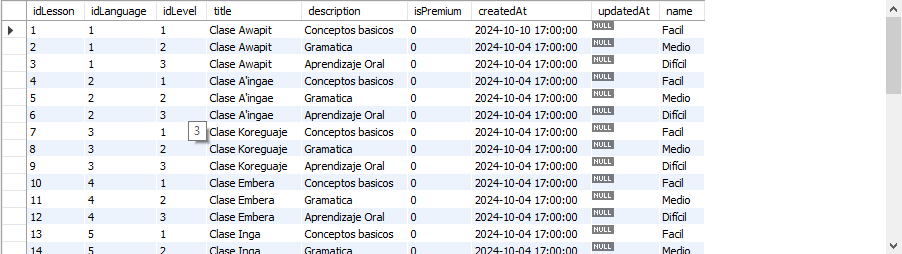
**Resultado**:  


**2.** **Lecciones y el nivel correspondiente, incluso si no tienen un nivel asociado**:

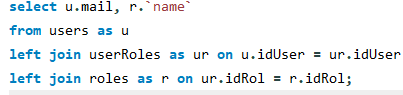


**Descripción**:  
Esta consulta muestra todas las lecciones con sus respectivos niveles. Si una lección no está asociada a ningún nivel, aparecerá con los campos de nivel como NULL.

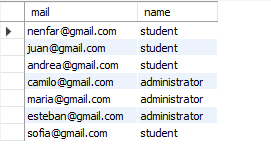
**Resultado**:



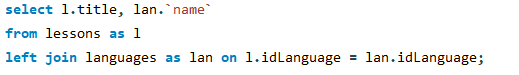
#### 3. ****Usuarios y sus roles, incluso si no tienen un rol asignado****:



**Descripción**:  
La consulta lista todos los usuarios y los roles que tienen asignados. Si un usuario no tiene roles, será incluido igualmente, pero el campo name del rol aparecerá como NULL.

Resultado:  


1. **Lecciones y el idioma correspondiente, incluso si no están asociadas a un idioma**:

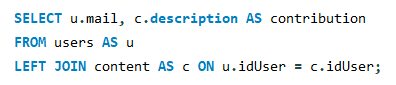


**Descripción**:  
Se obtiene el título de cada lección junto con el nombre del idioma correspondiente. Si una lección no está vinculada a ningún idioma, el campo del idioma será NULL.

**Resultado**:

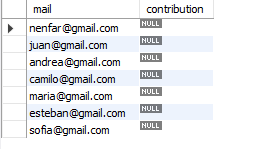


1. **Listar usuarios y sus contribuciones, incluso si no han realizado ninguna:**

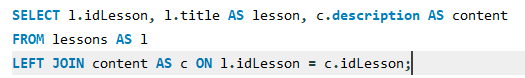


**Descripción**: Lista los usuarios con sus correos y las contribuciones que han realizado (ahora almacenadas en content). Si un usuario no tiene contribuciones, el campo de descripción aparecerá como NULL.

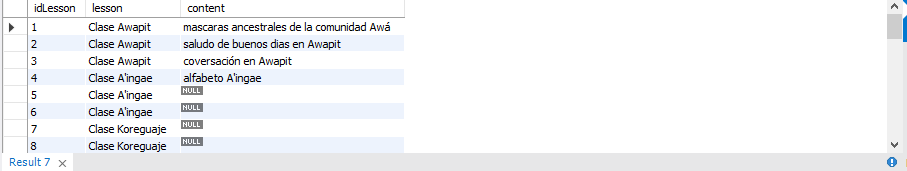
**Resultados**:



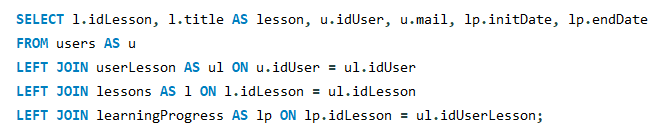
1. **Lecciones y sus contenidos, incluso si no tienen contenido asociado**



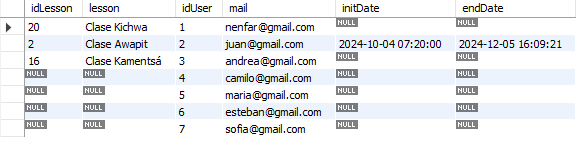
**Descripción**: Devuelve todas las lecciones con sus títulos y los contenidos asociados. Si una lección no tiene contenido, el campo de contenido aparecerá como NULL.

**Resultados**:  


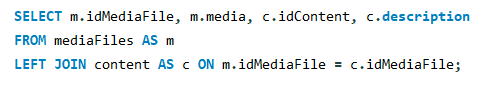
1. **Lecciones y el progreso del usuario, aunque no hayan iniciado ninguna lección.**

**Descripción**: Lista todas las lecciones y muestra el progreso de los usuarios, incluso si no han iniciado ninguna. Si un usuario no tiene progreso registrado, los campos de fechas aparecerán como NULL.

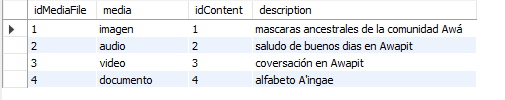
**Resultados**:



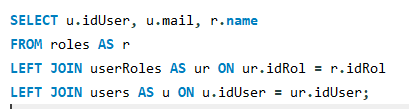
1. **Archivos multimedia y el contenido asociado, incluso si no están en uso**



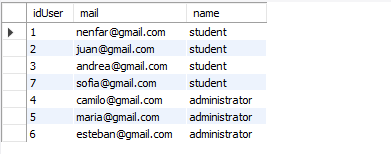
**Descripción**: Devuelve todos los archivos multimedia y el contenido con el que están asociados. Si un archivo multimedia no está en uso, los campos de contenido aparecerán como NULL.

**Resultados**:  


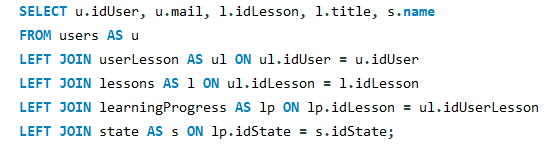
1. **Roles y los usuarios que los tienen asignados, aunque no todos los roles estén asignados**

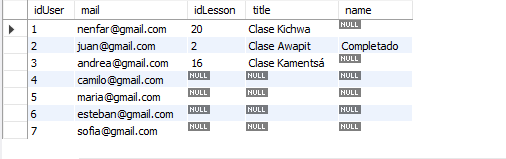


**Descripción**: Lista todos los roles y los usuarios que tienen asignados estos roles. Si un rol no está asignado a ningún usuario, los campos de usuario aparecerán como NULL.

**Resultados**:  


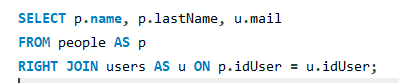
1. **Lecciones y el estado de progreso del usuario, aunque no haya progreso registrado**

**Descripción**: Devuelve una lista de lecciones con su estado de progreso para cada usuario. Si no hay progreso registrado, los campos del estado aparecerán como NULL.

**Resultados**:  


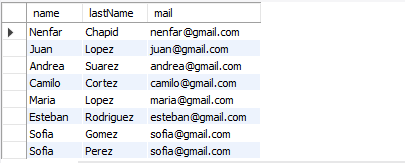
### RIGHT JOIN

1. Datos de las personas y los usuarios asociados, incluso si no están registrados como usuarios

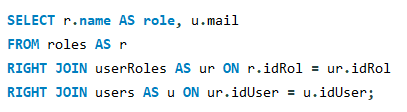


**Descripción**:  
Esta consulta obtiene los nombres y apellidos de las personas junto con los correos electrónicos de los usuarios asociados. Si una persona no está asociada con un usuario, sus campos aparecerán como NULL.

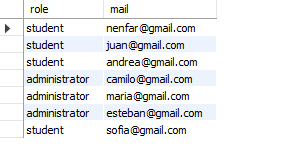
**Resultados**:



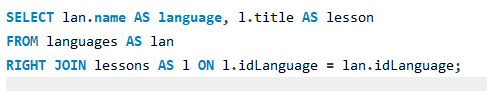
1. **Roles y los usuarios asignados, incluso si no hay usuarios para un rol**



**Descripción**:  
Esta consulta lista todos los roles y los usuarios asociados. Si un rol no tiene usuarios asignados, sus campos de rol aparecerán como NULL.

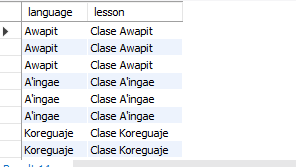
**Resultados**:  


1. **Idiomas y las lecciones disponibles, incluso si no hay lecciones para un idioma**

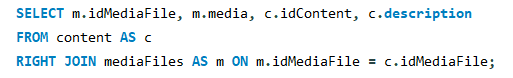


**Descripción**:  
Devuelve una lista de idiomas y sus lecciones asociadas. Si un idioma no tiene lecciones asociadas, el campo de idioma aparecerá como NULL.

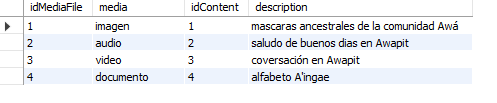
**Resultados**:



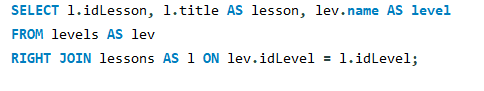
1. **Archivos multimedia y los contenidos relacionados, incluso si no están asociados a contenido**



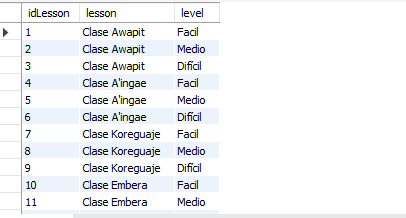
**Descripción**:  
Obtiene una lista de archivos multimedia y los contenidos relacionados. Si un archivo multimedia no está asociado con contenido, los campos de contenido aparecerán como NULL.

**Resultados**:  


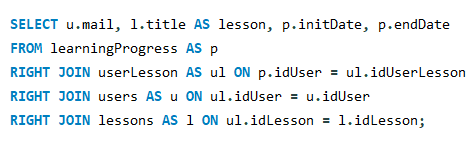
1. **Lecciones y sus niveles asociados, incluso si no tienen nivel asignado**



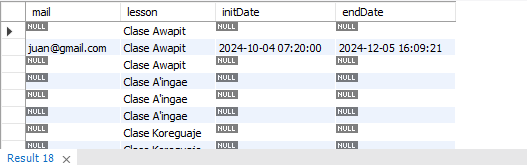
**Descripción**:  
Devuelve una lista de lecciones con sus niveles asociados. Si una lección no tiene un nivel asignado, el campo del nivel aparecerá como NULL.

**Resultados**:  


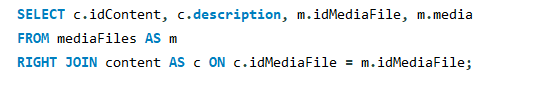
1. **Usuarios y sus progresos en las lecciones, incluso si no tienen progreso registrado**



**Descripción**:  
Esta consulta lista todos los usuarios con sus correos, las lecciones en las que están inscritos y su progreso. Si un usuario no tiene progreso registrado, los campos de progreso aparecerán como NULL.

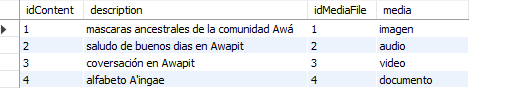
**Resultados**:  


1. **Contenidos y los archivos multimedia asociados, incluso si no tienen archivos multimedia vinculados**



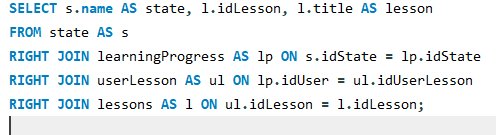
**Descripción**:  
Devuelve una lista de contenidos y los archivos multimedia asociados. Si un contenido no tiene archivos multimedia vinculados, los campos de archivo aparecerán como NULL.

**Resultados**:

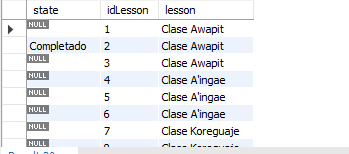


1. **Estados de progreso y las lecciones relacionadas, incluso si no tienen un estado asociado**

**Consulta Original**:

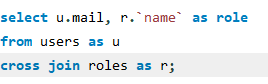


**Descripción**:  
Lista todos los estados de progreso con las lecciones relacionadas. Si una lección no tiene un estado de progreso asociado, los campos del estado aparecerán como NULL.

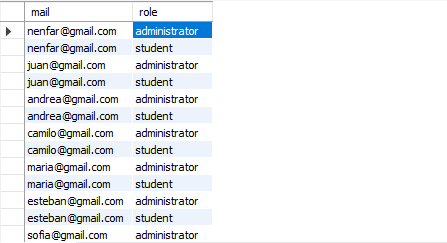
**Resultados**:  


### CROSS JOIN

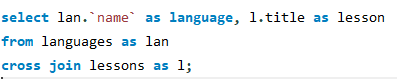
1. **Combinación de todos los usuarios con todos los roles posibles**:



**Descripción**:  
Genera un conjunto combinado de todos los usuarios y todos los roles. Cada usuario aparece combinado con cada rol, sin considerar relaciones previas.

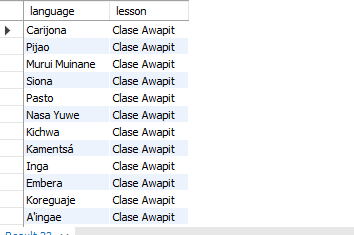
**Resultados**:  


1. **Combinación de todos los idiomas con todas las lecciones posibles**:

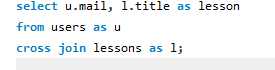


**Descripción**:  
Crea una combinación de todos los idiomas con todas las lecciones, sin importar si están relacionadas.

**Resultados**:

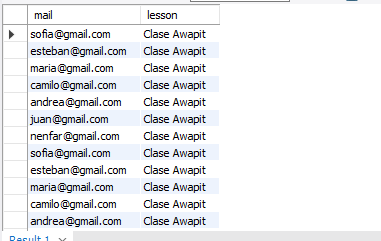


1. **Combinación de todos los usuarios con todas las lecciones posibles**:

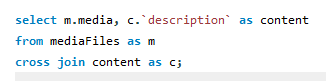


**Descripción**:  
Lista cada usuario con cada lección disponible, generando todas las combinaciones posibles sin considerar relaciones existentes.

**Resultados:**

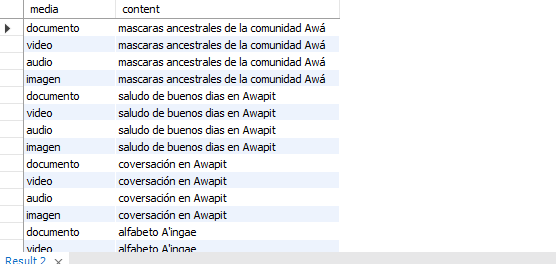


1. **Combinación de todos los archivos multimedia con todos los contenidos posibles**:

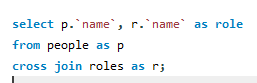


**Descripción**:  
Genera una lista de cada archivo multimedia combinado con cada contenido, sin importar si están asociados.

**Resultados**:

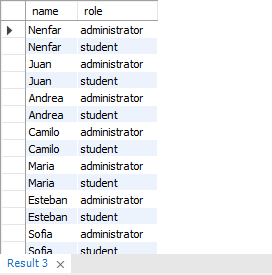


1. **Combinación de todas las personas con todos los roles posibles**:



**Descripción**:  
Crea una combinación de todas las personas con todos los roles disponibles, sin importar si hay relaciones entre ellos.

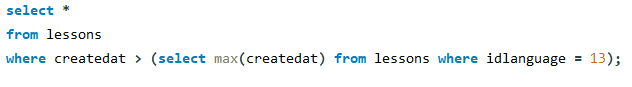
**Resultados**:



Subconsultas:

### WHERE

1. **Filtrar por comparación de fechas**  
   **Consulta**:

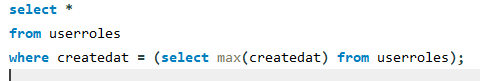


**Descripción**:  
Esta consulta obtiene todas las lecciones cuya fecha de creación (createdAt) es posterior a la fecha de creación más reciente de las lecciones pertenecientes al idioma con idLanguage = 13.

**Resultados**:



1. **Filtrar registros basados en el valor mínimo de otra columna**  
   **Consulta**:

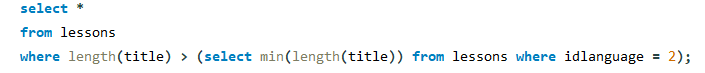


**Descripción**:  
Devuelve todos los registros de la tabla userRoles cuya fecha de creación (createdAt) sea la más reciente.

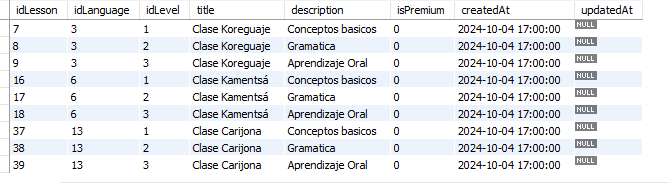
**Resultados**:



1. **Filtrar registros por longitud de texto**  
   **Consulta**:

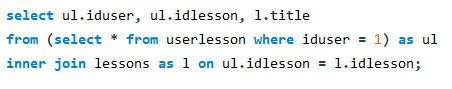
**Descripción**:  
Esta consulta selecciona todas las lecciones cuyo título tiene una longitud mayor que el título más corto entre las lecciones del idioma con idLanguage = 2.

**Resultados**:



### FROM

1. **Subconsulta para usuarios específicos**  
   **Consulta**:

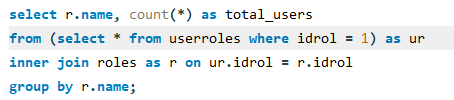


**Descripción**:  
Obtiene el ID del usuario, las lecciones asociadas y los títulos de las lecciones para el usuario con idUser = 1.

**Resultados**:

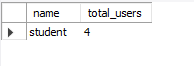


1. **Contar usuarios en un rol específico**  
   **Consulta**:



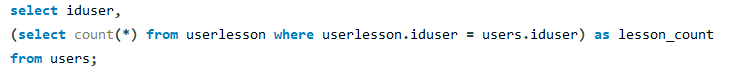
**Descripción**:  
Devuelve el nombre del rol (name) y la cantidad total de usuarios asociados al rol con idRole = 1.

**Resultados**:



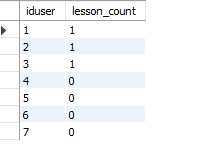
### SELECT

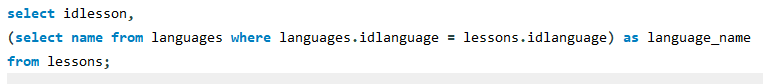
1. **Contar lecciones asociadas a cada usuario**:



**Descripción**:  
Para cada usuario, esta consulta cuenta la cantidad de lecciones asociadas a él en la tabla userLesson.

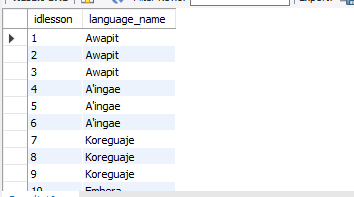
**Resultados**:



1. **Obtener el idioma de cada lección**  
   ****

**Descripción**:  
Devuelve el ID de cada lección y el nombre del idioma al que pertenece.

**Resultados**:



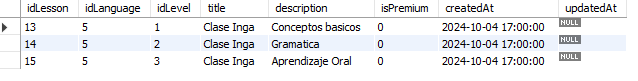
### IN

1. **Filtrar lecciones por comunidades específicas  
   Consulta:**



**Descripción**:  
Selecciona todas las lecciones cuyos idiomas están asociados a la comunidad 'Inga'.

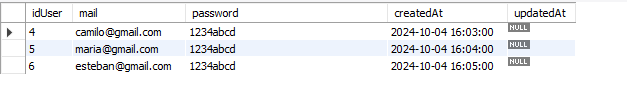
**Resultados**:



1. **Filtrar usuarios por roles específicos**  
   ****

**Descripción**:  
Devuelve todos los usuarios que tienen asignado el rol con idRole = 2.

**Resultados**:

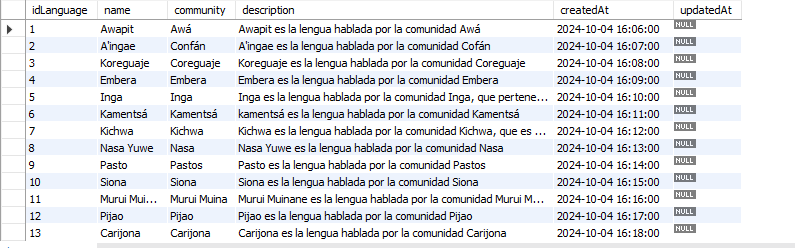


### EXISTS

1. **Idiomas con lecciones asociadas**

**Descripción**:  
Devuelve todos los idiomas que tienen al menos una lección asociada.

**Resultados**:

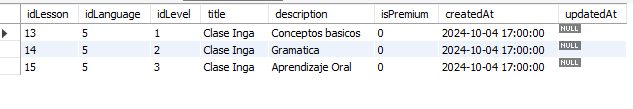


### ANY y ALL

1. **Filtrar lecciones con idiomas de una comunidad específica**  
   ****

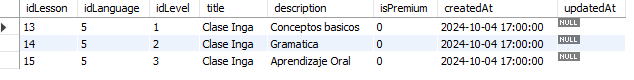
**Descripción**:  
Selecciona todas las lecciones asociadas a cualquier idioma que pertenezca a la comunidad 'Inga'.

**Resultados**:



1. **Filtrar progreso con fecha de inicio posterior a todas las de un estado específico**  
   **Descripción**:  
   Devuelve todos los registros de progreso cuya fecha de inicio es posterior a todas las fechas de inicio de los registros con idState = 1.

**Resultados**:

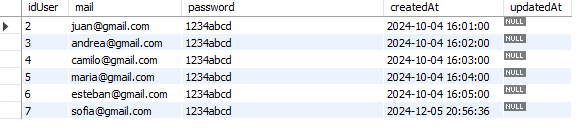


1. **Filtrar usuarios por fecha de creación en relación con correos de Gmail**

****

**Descripción**:  
Devuelve todos los usuarios cuya fecha de creación es posterior a la de al menos un usuario con un correo de Gmail.

**Resultados**:

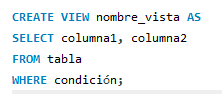


### VISTAS

En MySQL, las vistas son consultas guardadas que actúan como tablas virtuales. Aquí se explica cómo trabajar con ellas:

**1. Crear una Vista**

Se usa el comando CREATE VIEW para crear una vista. Ejemplo:



Esto crea una vista llamada nombre\_vista que contiene los datos definidos por la consulta.

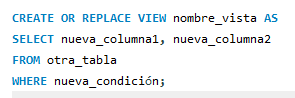
**2. Ejecutar una Vista**

Para usar una vista, simplemente se consulta como una tabla:



**3. Actualizar una Vista**

Si necesitas modificar una vista existente, usa el comando CREATE OR REPLACE VIEW:



Esto reemplaza la definición actual de la vista.

**4. Eliminar una Vista**

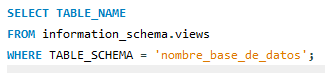
Para eliminar una vista, usa el comando DROP VIEW:



Esto elimina la vista de la base de datos.

**5. Ver Todas las Vistas Creadas**

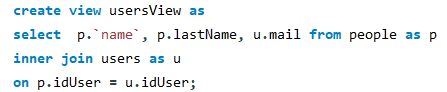
Para listar todas las vistas en la base de datos, consulta la tabla information\_schema.views:



Esto muestra todas las vistas creadas en una base de datos específica.

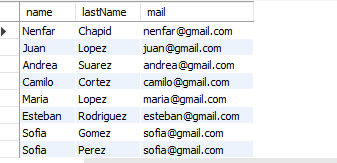
**Ejemplos**:

1. **Vista usersView**

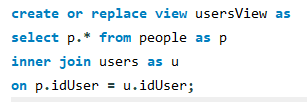


**Descripción**: esta vista trae los nombre y apellidos de los usuarios desde la tabla people y su correo de usuario desde la tabla users.

**Resultados**: para ver el resultado de la vista se utiliza el código “select \* from usersView;”

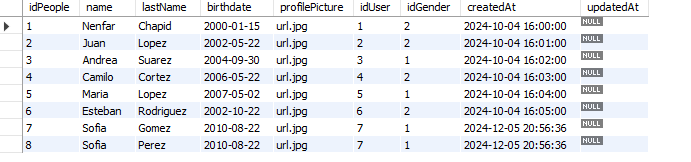


**Reemplazo de Vista: para actualizar o reemplazar la vista se utiliza el comando** CREATE OR REPLACE VIEW usersView y posteriormente se ejecuta el comando select \* from usersView;” para observar los resultados de la vista actualizada.

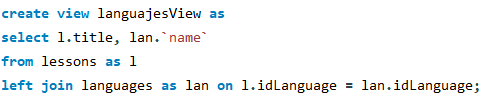


**Descripción**:  
Esta vista combina la tabla people con users, mostrando todos los campos de people para las personas que están registradas como usuarios (users).

**Resultados**:

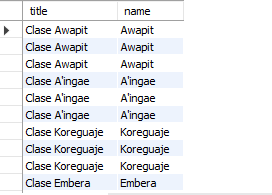


1. **Vista languajesView**

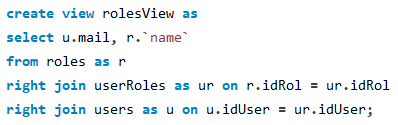
****

**Descripción**: Muestra los títulos de las lecciones (l.title) y los nombres de los idiomas (lan.name) asociados. Incluye todas las lecciones, incluso si no están vinculadas a un idioma.

**Resultados**:

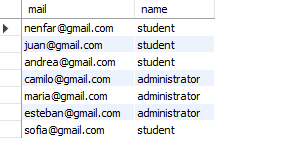


1. **Vista rolesView**

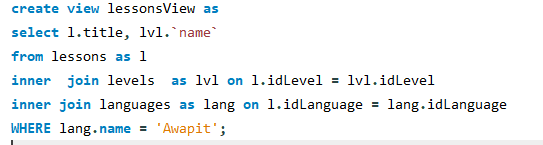
****

**Descripción**:  
Esta vista muestra el correo electrónico de los usuarios (u.mail) junto con los nombres de los roles (r.name). Utiliza un RIGHT JOIN para incluir a todos los usuarios, incluso si no tienen un rol asociado.

**Resultados**:

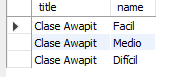


1. **Vista lessonsView**

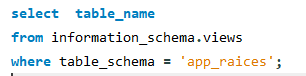
****

**Descripción**:  
Esta vista muestra los títulos de las lecciones (l.title) junto con el nombre del nivel (lvl.name) para las lecciones que están asociadas al idioma 'Awapit'.

**Resultados**:

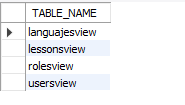


1. **Consulta para listar vistas en la base de datos:**

****

**Descripción**:  
Esta consulta devuelve la lista de todas las vistas definidas en la base de datos app\_raices.

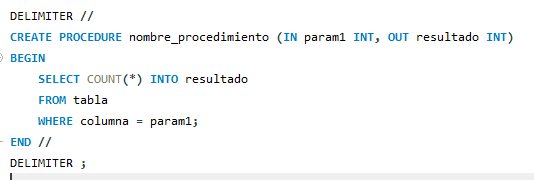
**Resultados**:



### PROCEDIMIENTOS ALMACENADOS

**1. Crear un Procedimiento Almacenado**

Se utiliza el comando CREATE PROCEDURE para definir un procedimiento. Ejemplo:



* **IN**: Parámetro de entrada (el usuario debe proporcionarlo).
* **OUT**: Parámetro de salida (devuelve un valor al usuario).
* **INOUT**: Parámetro que puede ser usado como entrada y salida.

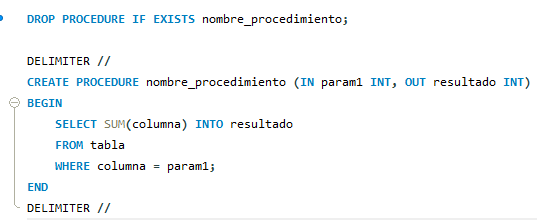
**2. Ejecutar un Procedimiento**

Usa el comando CALL para ejecutar un procedimiento. Ejemplo:



**3. Actualizar un Procedimiento**

Para modificar un procedimiento, primero debes eliminarlo y luego crearlo nuevamente. Ejemplo:



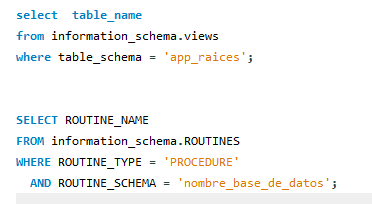
**4. Eliminar un Procedimiento**

Usa el comando DROP PROCEDURE para eliminar un procedimiento. Ejemplo:



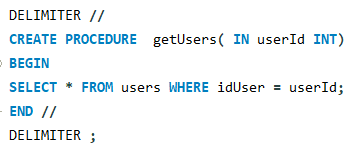
**5. Ver Todos los Procedimientos Almacenados**

Consulta la tabla information\_schema.ROUTINES para listar todos los procedimientos:



**Ejemplos**:

1. **getUsers**

****.

**Descripción**:  
Este procedimiento devuelve toda la información de un usuario específico basado en su idUser.

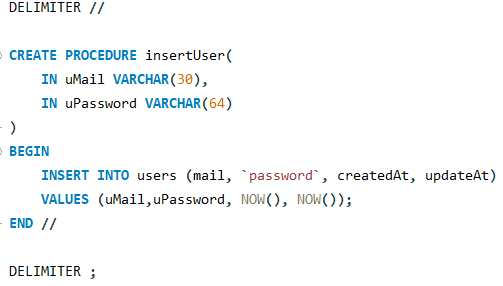
* **Parámetro de entrada**: userId (INT), que corresponde al identificador único del usuario.

**Ejemplo de Uso**:





1. **insertUser**

****

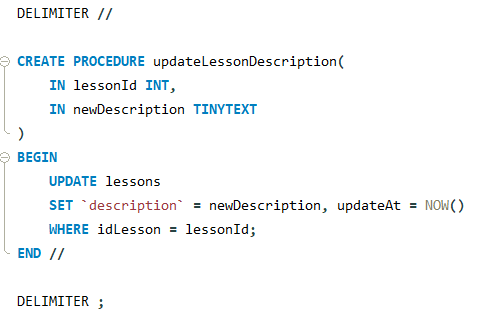
**Descripción**:  
Este procedimiento permite insertar un nuevo usuario en la tabla users, incluyendo su correo, contraseña y las marcas de tiempo.

* **Parámetros de entrada**:
  + uMail (VARCHAR): Correo electrónico del usuario.
  + uPassword (VARCHAR): Contraseña del usuario.

**Ejemplo de Uso**:



1. updateLessonDescription

****

**Descripción**:  
Este procedimiento actualiza la descripción de una lección específica y actualiza el campo updateAt con la marca de tiempo actual.

* **Parámetros de entrada**:
  + lessonId (INT): Identificador único de la lección.
  + newDescription (TEXT): Nueva descripción para la lección.

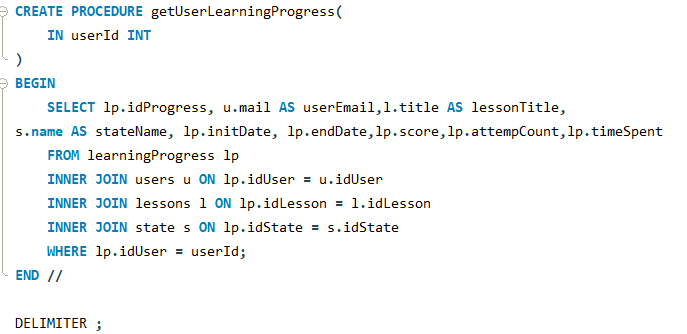
**Ejemplo de Uso**:

sql

Copiar código

CALL updateLessonDescription(2, 'Nueva descripción');

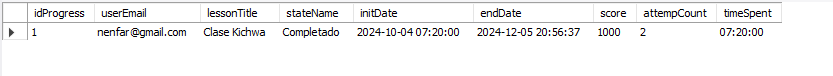
1. getUserLearningProgress

**Descripción**:  
Este procedimiento devuelve el progreso de aprendizaje de un usuario específico, incluyendo detalles como las lecciones, estado, fechas, puntaje, intentos y tiempo dedicado.

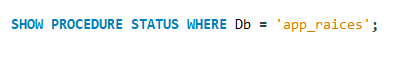
* **Parámetro de entrada**:
  + userId (INT): Identificador único del usuario.

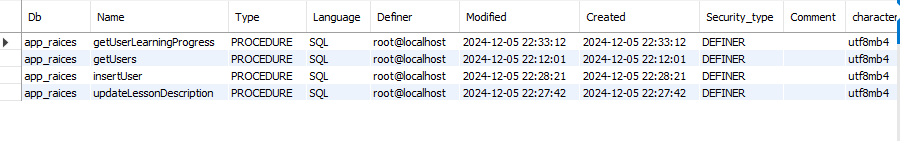
**Ejemplo de Uso**:





1. Mostrar Procedimientos Existentes

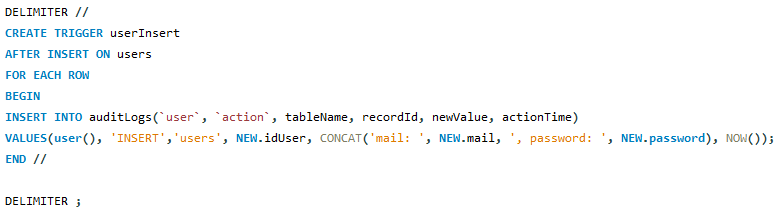
****

****

**Descripción**:  
Esta consulta lista todos los procedimientos almacenados en la base de datos app\_raices, mostrando detalles como el nombre del procedimiento, la fecha de creación y las últimas modificaciones.

### TRIGGERS

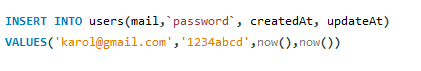
1. **userInsert**



**Descripción**:  
Este trigger se ejecuta después de cada **inserción** en la tabla users. Registra la acción en la tabla auditLogs, almacenando el usuario que ejecutó la acción, los datos insertados y el momento en que ocurrió.

* **Acción**: INSERT en users.
* **Campos registrados**: mail, password.

**Ejemplo de Inserción**:



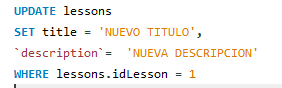
1. **lessonUpdate**

****

**Descripción**:  
Este trigger se ejecuta después de cada **actualización** en la tabla lessons. Guarda en la tabla auditLogs la información previa (OLD) y posterior (NEW) de los campos title y description, junto con el momento de la acción.

* **Acción**: UPDATE en lessons.
* **Campos registrados**: title, description.

**Ejemplo de Actualización**:



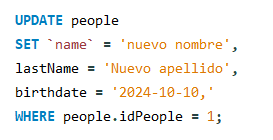
1. **peopleUpdate**

****

**Descripción**:  
Este trigger se ejecuta antes de cada **actualización** en la tabla people. Valida que el campo birthdate no sea una fecha futura y registra los cambios en la tabla auditLogs.

* **Acción**: UPDATE en people.
* **Validación**: birthdate debe ser menor o igual a la fecha actual.

**Ejemplo de Actualización**:



1. contentDelete



**Descripción**:  
Este trigger se ejecuta antes de cada **eliminación** en la tabla content. Guarda la información del registro eliminado en la tabla auditLogs.

* **Acción**: DELETE en content.
* **Campos registrados**: url, description.

**Ejemplo de Eliminación**:

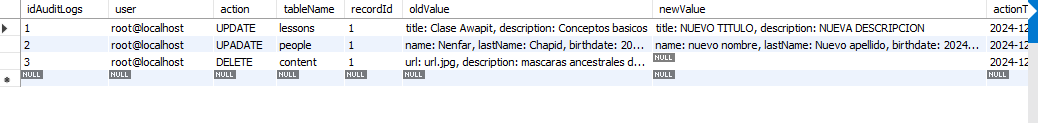


1. **Consulta de la Tabla auditLogs**



**Descripción**:  
Muestra todas las acciones registradas por los triggers en la tabla auditLogs.

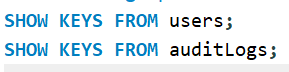
**Resultados**:



### ÍNDICES

### Índices Agrupados

1. **Mostrar Índices Existentes**



**Descripción**:  
Estas consultas muestran los índices existentes en las tablas users y auditLogs. Incluyen claves primarias y otros índices previamente definidos.

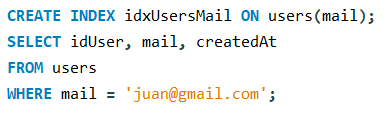
**Resultados**:





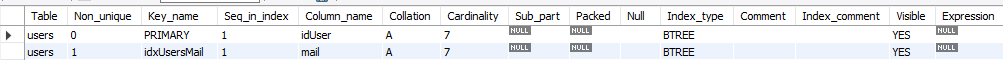
### Índices No Agrupados

1. **Índice en users.mail**

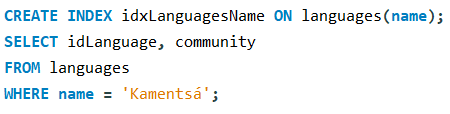


**Descripción**:  
Crea un índice no agrupado en el campo mail de la tabla users. Mejora el rendimiento de consultas que usan mail en condiciones WHERE.

**Resultados**:

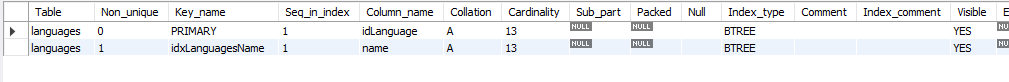


1. **Índice en languages.name**

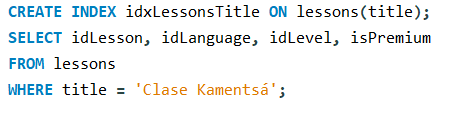


**Descripción**:  
Crea un índice no agrupado en el campo name de la tabla languages. Facilita búsquedas rápidas basadas en name.

**Resultados**:

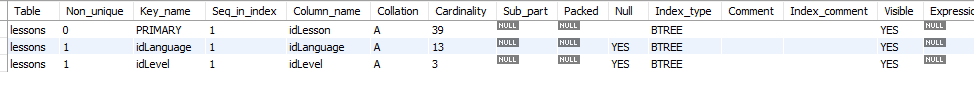


1. **Índice en lessons.title**



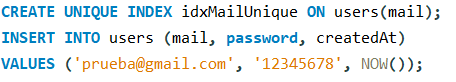
**Descripción**:  
Crea un índice no agrupado en el campo title de la tabla lessons. Optimiza consultas basadas en title.

**Resultados**:



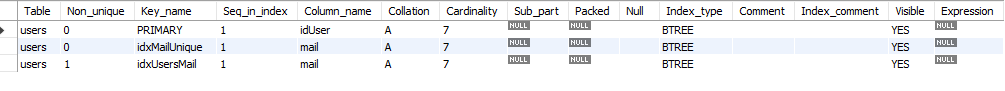
### Índices Únicos

1. **Índice Único en users.mail**

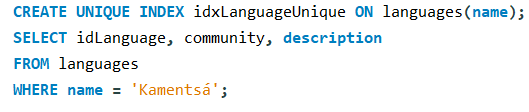


**Descripción**:  
Este índice garantiza que no haya duplicados en el campo mail de la tabla users. Intenta insertar un correo electrónico ya existente lanzará un error.

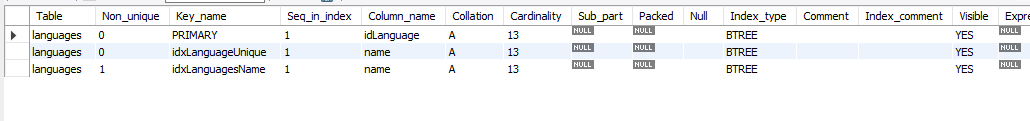
**Resultados**:



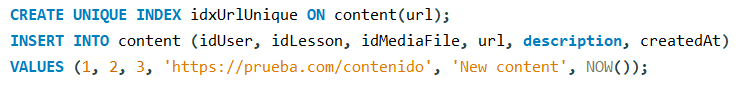
1. **Índice Único en languages.name**

**Descripción**:  
Garantiza que los valores en el campo name de la tabla languages sean únicos. Mejora la búsqueda en consultas con name.

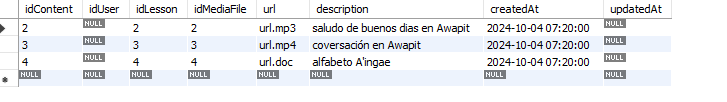
**Resultados**:



1. **Índice Único en content.url**

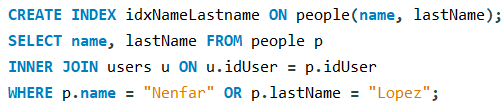
**Descripción**:  
Asegura que el campo url en la tabla content no tenga valores duplicados. Útil para manejar recursos únicos como enlaces o rutas.

**Resultados**:

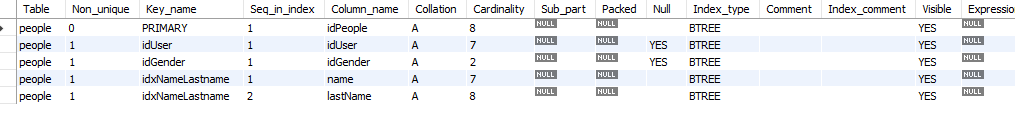


### Índices Compuestos

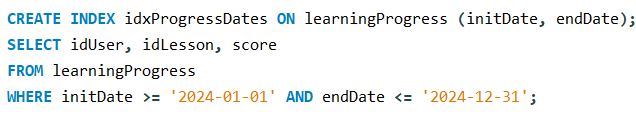
1. **Índice en people(name, lastName)**

**Descripción**:  
Crea un índice compuesto que mejora el rendimiento de consultas donde name o lastName se usan en filtros.

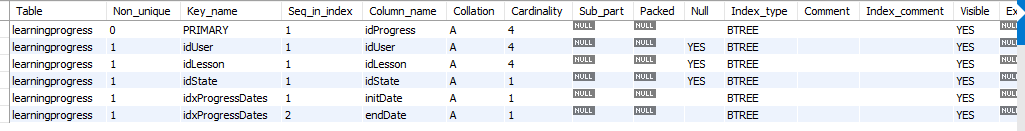
Resultados:



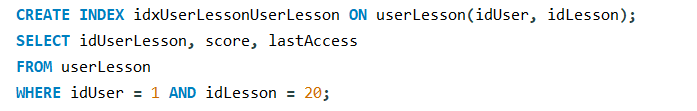
1. **Índice en learningProgress(initDate, endDate)**

**Descripción**:  
Crea un índice compuesto en los campos initDate y endDate. Mejora el rendimiento de consultas sobre rangos de fechas.

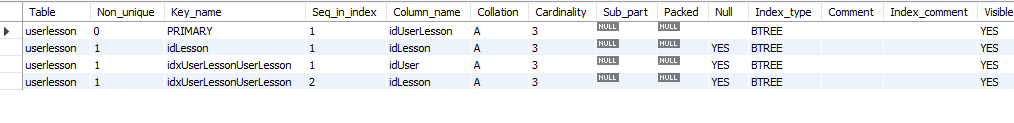
**Resultados**:



1. **Índice en userLesson(idUser, idLesson)**

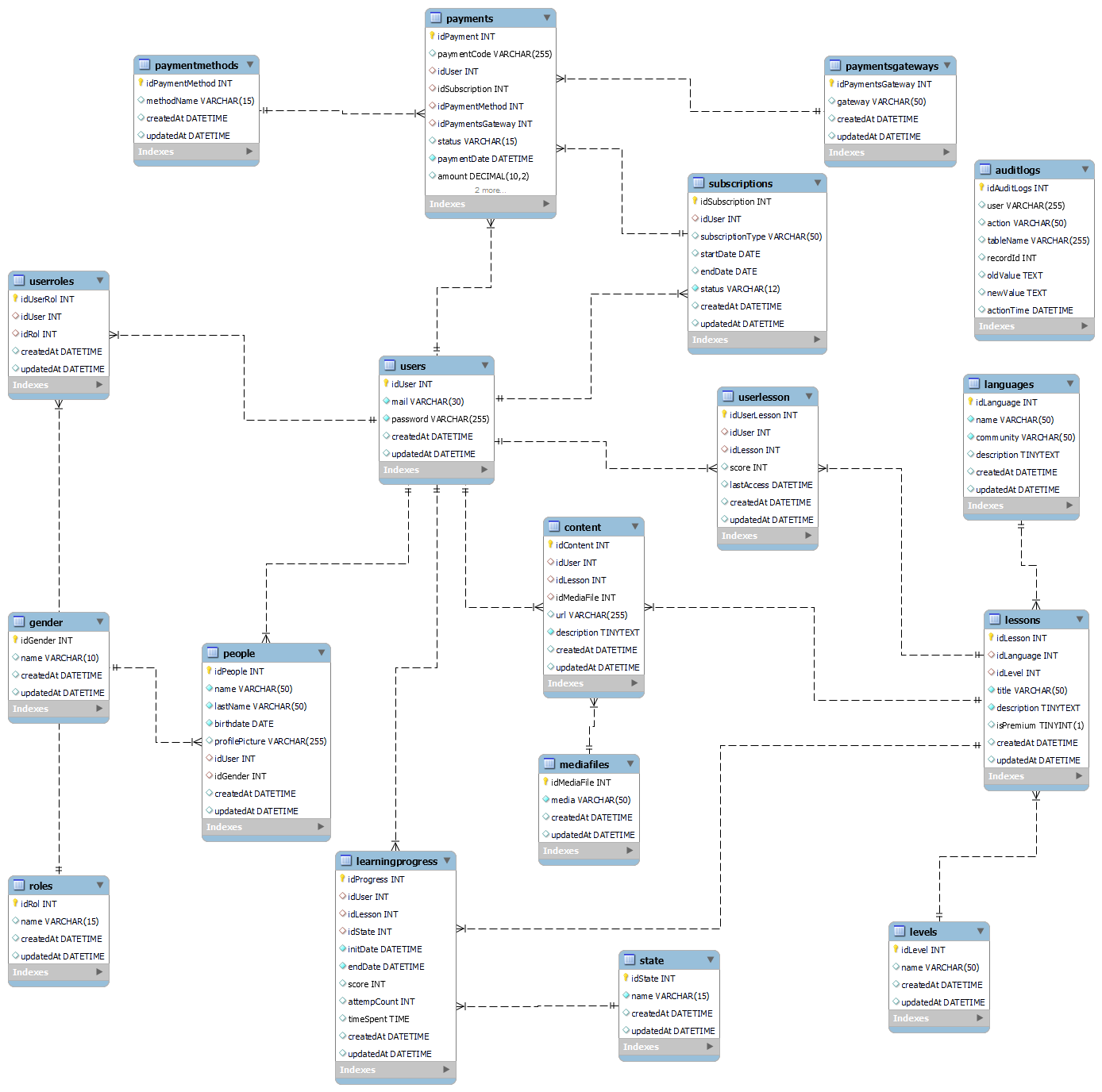
**Descripción**:  
Este índice compuesto acelera las búsquedas cuando ambas columnas (idUser y idLesson) se utilizan en condiciones WHERE.

**Resultados**:



## Diseño de Base de Datos

### Modelo de Datos:



### Consideraciones de Diseño

### **Relaciones y Cardinalidad**

1. **users - people**:
   * Un usuario puede tener una sola persona asociada (uno a uno).
   * Un registro de users puede estar relacionado con un único registro en people.
2. **users - userRoles**:
   * Un usuario puede tener varios roles (uno a muchos).
   * La relación entre users y roles es muchos a muchos a través de la tabla intermedia userRoles.
3. **roles - userRoles**:
   * Un rol puede estar asignado a muchos usuarios (uno a muchos).
   * La relación entre roles y users es muchos a muchos a través de la tabla userRoles.
4. **languages - lessons**:
   * Un idioma puede tener muchas lecciones (uno a muchos).
   * Cada lección se asocia a un único idioma.
5. **levels - lessons**:
   * Un nivel puede tener muchas lecciones (uno a muchos).
   * Cada lección se asocia a un único nivel.
6. **lessons - userLesson**:
   * Una lección puede ser tomada por muchos usuarios (uno a muchos).
   * Un usuario puede tomar muchas lecciones.
7. **lessons - content**:
   * Una lección puede tener múltiples contenidos asociados (uno a muchos).
   * Un contenido puede pertenecer a una sola lección.
8. **users - learningProgress**:
   * Un usuario puede tener muchos registros de progreso de aprendizaje (uno a muchos).
   * Cada registro de progreso pertenece a un único usuario.
9. **lessons - learningProgress**:
   * Una lección puede tener muchos registros de progreso de aprendizaje (uno a muchos).
   * Cada registro de progreso pertenece a una sola lección.
10. **users - subscriptions**:
    * Un usuario puede tener una suscripción activa a la vez (uno a uno).
    * Un usuario puede tener múltiples suscripciones a lo largo del tiempo, pero una sola activa.
11. **subscriptions - payments**:
    * Una suscripción puede estar asociada a muchos pagos (uno a muchos).
    * Un pago está relacionado a una única suscripción.
12. **payments - paymentMethods, paymentsGateways**:
    * Un pago puede usar un solo método de pago y una sola pasarela de pago (uno a uno).

### **Normalización**

1. **Primera Forma Normal (1FN)**:
   * Las tablas están en 1FN, ya que cada atributo contiene valores atómicos y no se repiten grupos de atributos.
2. **Segunda Forma Normal (2FN)**:
   * Las tablas también están en 2FN, ya que cada tabla está libre de dependencias parciales (es decir, todos los atributos no clave dependen completamente de la clave primaria).
3. **Tercera Forma Normal (3FN)**:
   * Las tablas están en 3FN, ya que no hay dependencias transitivas (es decir, ningún atributo no clave depende de otro atributo no clave).

## Procedimientos de Respaldo y Recuperación

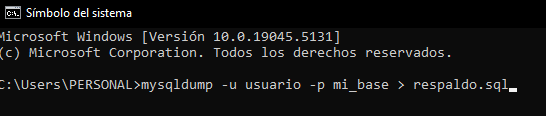
El respaldo (backup) y la recuperación (restore) son procesos esenciales para proteger los datos de una base de datos MySQL frente a posibles fallos, pérdidas accidentales, o corrupción de datos. Estos procesos aseguran que la información pueda ser restaurada a un estado funcional en caso de un desastre.

Respaldo en MySQL

El respaldo consiste en crear una copia de seguridad de los datos y la estructura de la base de datos. Hay varios métodos para realizar respaldos en MySQL:

**Respaldo con mysqldump**

mysqldump es una herramienta nativa de MySQL que genera un archivo SQL con los comandos necesarios para reconstruir la base de datos.  
Ejemplo para respaldar una base de datos llamada mi\_base:



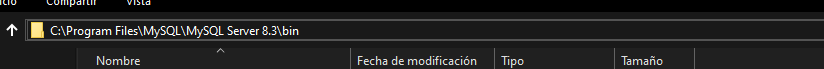
* **Parámetros importantes:**
  + -u usuario: Usuario de MySQL.
  + -p: Solicita la contraseña del usuario.
  + mi\_base: Nombre de la base de datos a respaldar.
  + > respaldo.sql: Archivo de salida que contiene el respaldo.

### Respaldo en base de datos app\_raices

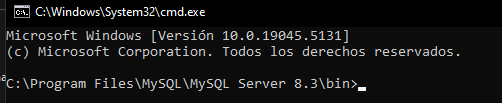
Copiar la ruta donde se quiere guardar el archivo de respaldo sql. Por ejemplo:



Ubicarse en la ruta donde está instalado MySQL e ingresar a la carpeta de la versión de MySQL Server y luego en la carpeta “bin”.



Desde allí abrir la terminal

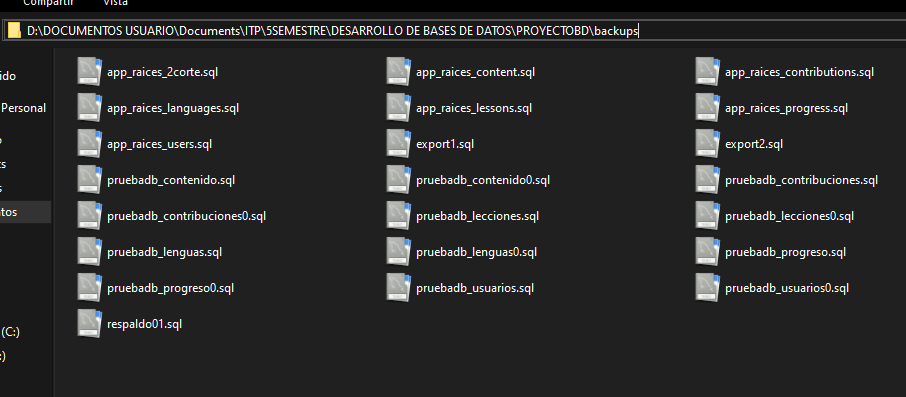


Escribir el comando de respaldo y, entre comillas, pegar la ruta donde se lo va a guardar 

A continuación se solicitará la contraseña de la conexión de MySQL. Ingresar la contraseña y ejecutar:



Verificamos que se haya creado el respaldo en la ruta especificada:



### Recuperación en base de datos app\_raices

La recuperación consiste en restaurar los datos respaldados en un servidor MySQL. Esto se logra ejecutando el archivo SQL generado durante el respaldo.

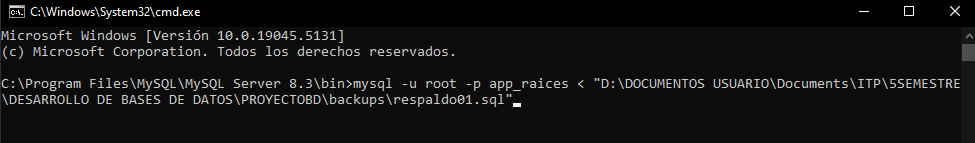
**Restaurar una Base de Datos**

Si el archivo de respaldo contiene una base de datos completa se ejecuta este comando:

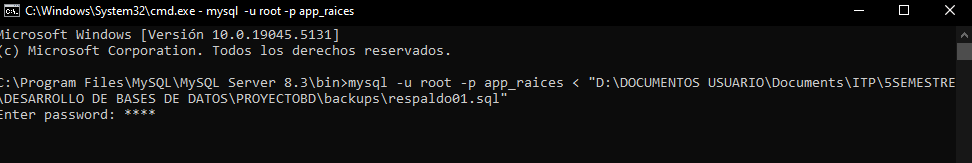
**mysql -u usuario -p mi\_base < respaldo.sql**

Ejemplo de recuperación de un respaldo de la base de datos app\_raices:

1. **Se accede a la ruta del bin** de MySQL y se abre la consola. Desde ahí se ejecuta el anterior comando junto con la ruta donde se aloja el respaldo**:**



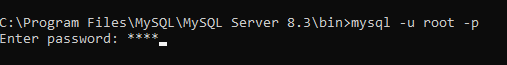
1. Se ingresa la contraseña:



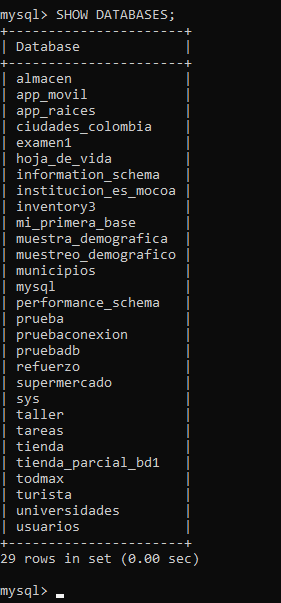
1. Ahora hay que ingresar al cliente de MySQL para comprobar que se haya hecho la recuperación:



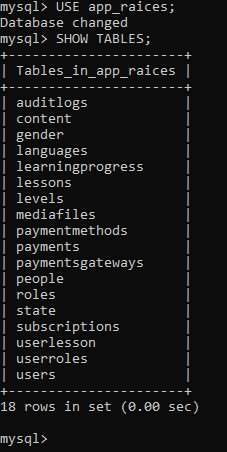
1. Ingresamos la contraseña:



1. Ejecutamos código sql para ver las bases de datos que tenemos en la conexión:



1. Ejecutamos código sql para ingresar a la base de datos restaurada, si no ocurren errores es porque la restauración fue exitosa:



# Análisis y Discusión

En esta sección, se analizan los resultados obtenidos tras la implementación de la base de datos app\_raices, observando cómo la estructura y las consultas se alinean con los objetivos de la aplicación. La base de datos fue diseñada para gestionar de manera eficiente el aprendizaje de las lenguas indígenas del Putumayo, permitiendo el manejo de usuarios, lecciones, contenidos, suscripciones y pagos.

Uno de los resultados más destacados fue la integración de las tablas que gestionan el progreso de los estudiantes, que incluye la tabla learningProgress con detalles sobre el puntaje, intentos y tiempo invertido, lo que permite a los administradores realizar un seguimiento adecuado de los avances del usuario. Esta funcionalidad se relaciona directamente con los objetivos de ofrecer una experiencia de aprendizaje personalizada y que permita un seguimiento adecuado.

La normalización hasta la tercera forma normal (3FN) aseguró que la base de datos esté optimizada y libre de redundancias, facilitando la integridad de los datos. Además, las relaciones entre las tablas de users, languages, lessons, content y progress están bien definidas, garantizando un flujo de información eficiente y coherente entre los diferentes módulos del sistema.

El uso de procedimientos almacenados, vistas e índices también permitió mejorar el rendimiento de las consultas, especialmente aquellas que involucran grandes volúmenes de datos, como la obtención de información sobre el progreso de los usuarios o las estadísticas de suscripciones.

En cuanto a las consultas SQL implementadas, se diseñaron para proporcionar respuestas rápidas y precisas a las necesidades de los usuarios, como obtener el progreso de un estudiante, sus lecciones completadas, o las lenguas que están aprendiendo. Las pruebas realizadas confirmaron que las consultas, procedimientos y funciones fueron exitosas, cumpliendo con los requerimientos del sistema.

# Interpretación de Resultados

Los resultados obtenidos a partir de las pruebas realizadas validan que la base de datos está cumpliendo con los objetivos definidos para la aplicación. Las relaciones entre tablas, como users y languages, y el seguimiento detallado del progreso de los estudiantes, demuestran que la base de datos es capaz de gestionar eficazmente los datos de los usuarios y sus interacciones con los contenidos educativos.

Además, la capacidad de realizar consultas complejas y obtener resultados precisos en tiempo real, como el historial de lecciones de los usuarios o la evolución de su progreso en las lenguas, es un aspecto crucial que contribuye al éxito de la aplicación. Este enfoque permite a los educadores realizar intervenciones más específicas y orientadas a mejorar el rendimiento de los estudiantes.

Las implementaciones de las vistas y procedimientos almacenados han sido útiles para asegurar la consistencia de los datos y la eficiencia en las consultas, mejorando el rendimiento general del sistema. El diseño de la base de datos, junto con las pruebas de validación, demuestra que la solución es escalable y flexible para adaptarse a futuros requisitos, como la inclusión de nuevas lenguas o funcionalidades.

# Conclusiones

La base de datos app\_raices ha sido diseñada y desarrollada con éxito para cumplir con los objetivos de la aplicación, aunque con la posibilidad de mejorar, proporcionando una solución sólida para la gestión de las lenguas indígenas del Putumayo. La estructura de la base de datos, junto con sus consultas optimizadas y el uso de procedimientos almacenados, asegura que la aplicación funcione de manera eficiente y confiable.

La normalización y la integridad referencial son componentes clave que contribuyen a la consistencia de los datos, mientras que las funcionalidades de seguimiento del progreso de los estudiantes ofrecen un valor agregado para mejorar la experiencia educativa. Además, la capacidad de realizar respaldos y recuperación de datos proporciona una capa adicional de seguridad y fiabilidad.

# Recomendaciones

1. **Monitoreo Continuo del Rendimiento**:

Aunque el sistema funciona de manera eficiente en este momento, se recomienda realizar un monitoreo continuo del rendimiento de las consultas y procesos almacenados, especialmente a medida que el número de usuarios crezca. Se podría considerar la implementación de un sistema de auditoría para detectar posibles cuellos de botella.

1. **Expansión de Funcionalidades**:

Dado que el sistema está diseñado de manera flexible, es recomendable considerar la expansión de las funcionalidades, como la inclusión de herramientas de gamificación o la integración con plataformas externas de pago, para ofrecer una experiencia más interactiva.

1. **Mejoras en la Gestión de Suscripciones**:

Se puede optimizar aún más la gestión de suscripciones y pagos, implementando notificaciones automáticas y renovaciones periódicas para mantener a los usuarios informados y facilitar el proceso de pago.

1. **Actualización Periódica de Contenidos**:

Es importante mantener los contenidos actualizados, incorporando nuevos materiales y lecciones, así como ampliando el número de lenguas indígenas en la plataforma para fomentar su preservación y enseñanza.

# Referencias

* MySQL Documentación (2024). Guía de MySQL: Gestión de bases de datos, optimización y seguridad.

<https://dev.mysql.com/doc/>

* García, M. (2022). Base de datos relacionales: Diseño, optimización y buenas prácticas. Editorial Universitaria.

<https://www.editorialelearning.com/catalogo/media/iverve/uploadpdf/1526034556_UF2175_demo.pdf>

* López, A., & Rodríguez, F. (2021). SQL avanzado y gestión de bases de datos en aplicaciones educativas. Tecnología y Educación, 34(2), 156-172.

<https://dialnet.unirioja.es/descarga/articulo/8073007.pdf>