

**Curso:**

**Base de datos**

**Practica Calificada 3**

**Docente:**

**Juan Ramirez Ticona**

**Integrantes:**

**Marcelo Mendoza Montes U21315939**

**Carlos Soriano Coaguila U22248178**

**Gerson Jesús Zevallos Valdivia 1626623**

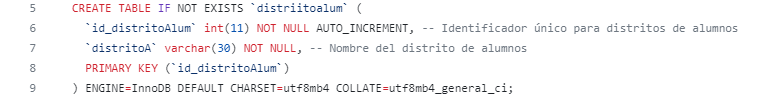
**Cristopher Ruben Torres Castillo u20214529**

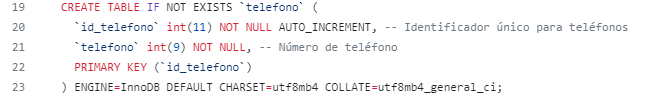
**Diego Sebastian Gonzales Gomez u22248606**

**2023**

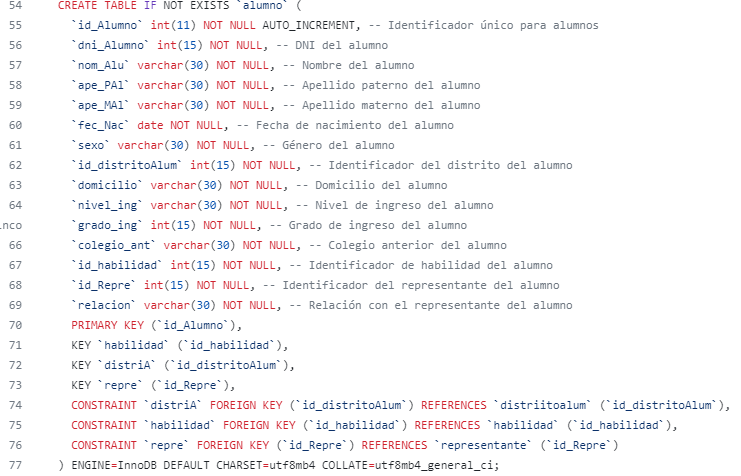
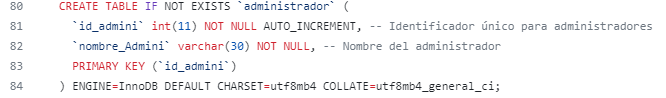
**PRIMERO:** creación de la base de datos denominada “matriculas”.  
- ‘IF NOT EXITS’ asegura que la base de datos solo se cree si no exista ya.  
**SEGUNDO:** Ejecutar la instrucción “USE”, cualquier operación de base de datos que se realice se aplicara en la base de datos “matriculas”.

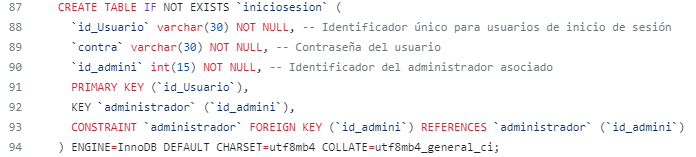
**TERCERO:** creación de las tablas

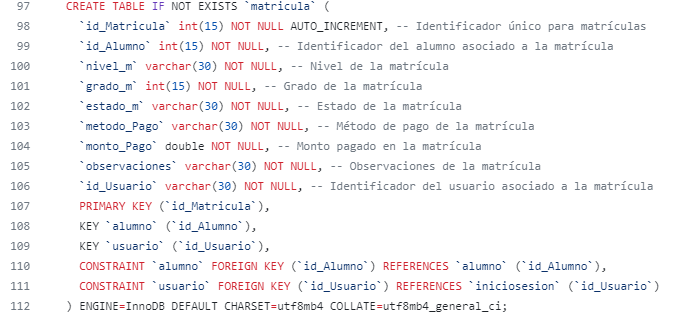
1. tabla: distritoalum
2. Interfaz de usuario gráfica, Texto, Aplicación, Word

   Descripción generada automáticamenteTabla: distritorepre
3. Tabla: teléfono
4. Interfaz de usuario gráfica, Texto, Aplicación

   Descripción generada automáticamenteTabla: habilidad
5. Texto

   Descripción generada automáticamenteTabla: representante
6. Tabla: alumno
7. Tabla: administrador
8. Tabla: iniciosesion



1. Tabla: matricula

**GROUP BY**

**Consulta para saber el número de alumnos por distrito**

SELECT da.distritoA, COUNT(\*) AS total\_estudiantes

FROM distriitoalum da

JOIN alumno a ON da.id\_distritoAlum = a.id\_distritoAlum

GROUP BY da.distritoA;

**Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente**

Esta consulta selecciona la columna **'distritoA'** de la tabla **'distriitoalum'** y cuenta el número total de estudiantes en cada distrito, combinando la información con la tabla 'alumno' mediante la condición de que los identificadores de distrito coincidan. El resultado se agrupa por distrito, mostrando el nombre del distrito y la cantidad correspondiente de estudiantes. El alias 'da' se utiliza para referenciar **'distriitoalum'** y 'a' para 'alumno', simplificando la sintaxis de la consulta. En esencia, la consulta proporciona un resumen claro del número de estudiantes por distrito en la base de datos.

**Consulta para saber la cantidad de representantes por distrito:**

SELECT dr.distritoR, COUNT(\*) AS total\_representantes

FROM distritorepre dr

JOIN representante r ON dr.id\_distritoR = r.id\_distritoR

GROUP BY dr.distritoR;

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

La consulta selecciona el nombre del distrito (**distritoR**) de la tabla **distritorepre** y cuenta el número total de representantes en cada distrito, utilizando la función de agregación COUNT(\*). La información se combina con la tabla representante a través de la condición de unión ON **dr.id\_distritoR** = **r.id\_distritoR**, asegurando que se cuenten solo los representantes asociados a un distrito específico. Los resultados se agrupan por distrito mediante GROUP BY **dr.distritoR**, proporcionando así un resumen claro que incluye el nombre del distrito y el total de representantes en cada uno. El uso de JOIN facilita la combinación de datos relevantes de ambas tablas antes de realizar el conteo y la agrupación, ofreciendo una visión detallada de la distribución de representantes por distrito en la base de datos.

**Consulta para obtener el promedio de monto de pago de matrícula por método de pago**

SELECT m.metodo\_Pago, AVG(m.monto\_Pago) AS promedio\_monto\_pago

FROM matricula m

GROUP BY m.metodo\_Pago;

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

En esta consulta, se busca entender los patrones de pago examinando el promedio del monto de pago de matrícula para cada método de pago registrado en la tabla matricula. La función de agregación AVG(**m.monto\_Pago**) calcula el promedio de los montos de pago agrupados por el método de pago correspondiente. Esto proporciona información sobre la distribución promedio de los pagos de matrícula para cada método registrado en la base de datos.

**Consulta para saber el numero de alumnos por nivel educativo**

SELECT a.nivel\_ing, COUNT(\*) AS total\_alumnos

FROM alumno a

GROUP BY a.nivel\_ing;

**Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente**

La consulta SELECT a.nivel\_ing, COUNT(\*) AS total\_alumnos FROM alumno a GROUP BY a.nivel\_ing; proporciona un análisis detallado de la distribución de alumnos según su nivel educativo. Se selecciona la columna nivel\_ing de la tabla alumno y se utiliza la función de agregación COUNT(\*) para contar el número total de alumnos para cada nivel educativo. Los resultados se agrupan por el nivel educativo, lo que resulta en una tabla que muestra los distintos niveles educativos y la cantidad total de alumnos asociados a cada uno. Esta consulta es valiosa para comprender la composición de la población estudiantil en términos de niveles educativos, permitiendo realizar análisis más profundos sobre la distribución y proporciones de alumnos en la base de datos.

**Consulta para saber la cantidad de representantes por género:**

SELECT r.sexo, COUNT(\*) AS total\_representantes

FROM representante r

GROUP BY r.sexo;

Tabla

Descripción generada automáticamente

La consulta SELECT **r.sexo**, COUNT(\*) AS **total\_representantes** FROM representante r GROUP BY **r.sexo**; realiza un análisis detallado de la distribución de representantes según su género en la base de datos. Se selecciona la columna sexo de la tabla representante, y la función de agregación **COUNT(\*)** cuenta el número total de representantes para cada género. Los resultados se agrupan por género, generando una tabla que muestra los distintos géneros y la cantidad total de representantes asociados a cada uno. Este análisis es valioso para evaluar la diversidad de género en roles de representación y comprender la distribución de representantes en términos de género en la base de datos.

**Consulta para saber la edad promedio de los alumnos por nivel de ingreso:**

SELECT a.nivel\_ing, AVG(YEAR(CURRENT\_DATE) - YEAR(a.fec\_Nac)) AS edad\_promedio

FROM alumno a

GROUP BY a.nivel\_ing;

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

La consulta SELECT **a.nivel\_ing**, AVG(YEAR(CURRENT\_DATE) - **YEAR(a.fec\_Nac))** AS **edad\_promedio** FROM alumno a GROUP BY **a.nivel\_ing**; se enfoca en determinar la edad promedio de los alumnos según su nivel de ingreso. Se selecciona la columna de nivel de ingreso de la tabla de alumnos, y la función de agregación AVG se emplea para calcular la edad promedio. Esto se logra restando el año de nacimiento de cada alumno al año actual y obteniendo el promedio de estas edades para cada nivel de ingreso. La cláusula GROUP BY se utiliza para agrupar los resultados por nivel de ingreso, ofreciendo una visión detallada de la edad promedio en cada categoría educativa. Este análisis resulta valioso para comprender la distribución de edades entre los diferentes niveles educativos y puede proporcionar **insights** relevantes para la planificación educativa y la atención personalizada de los estudiantes en función de sus edades.

**CLAUSULA HAVING**

**Consulta acerca de la cantidad de representantes en cada distrito, pero solo mostrar distritos con más de 1 representante.**

SELECT id\_distritoR, COUNT(\*) AS cantidad\_representantes

FROM representante

GROUP BY id\_distritoR

HAVING cantidad\_representantes > 1;

**Interfaz de usuario gráfica, Aplicación, Teams

Descripción generada automáticamente**

La consulta proporcionada busca obtener información detallada sobre los distritos que cuentan con más de un representante en la base de datos. En la cláusula SELECT, se selecciona el ID del distrito **(id\_distritoR**) y se cuenta el número de representantes para cada distrito, nombrando esta cantidad como **cantidad\_representantes**. La consulta se realiza desde la tabla representante. Luego, en la cláusula GROUP BY, se agrupan los resultados según el ID del distrito, asegurando que la cuenta se realice para cada distrito individualmente. Posteriormente, la cláusula HAVING filtra los resultados agrupados, incluyendo únicamente aquellos distritos que tienen más de un representante. En resumen, la consulta identifica y devuelve los distritos que cuentan con una representación plural en términos de representantes registrados en la base de datos.

**Consulta de estudiantes en cada distrito, mostrando solo los distritos con mas de 1 estudiante.**

SELECT id\_distritoAlum, COUNT(\*) AS cantidad\_estudiantes

FROM alumno

GROUP BY id\_distritoAlum

HAVING cantidad\_estudiantes > 1;

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

La consulta en cuestión busca obtener información detallada acerca de los distritos que cuentan con más de un estudiante matriculado en la base de datos. Utilizando la tabla alumno, se selecciona el ID del distrito **(id\_distritoAlum**) y se cuenta el número de estudiantes para cada distrito, denominando esta cantidad como **cantidad\_estudiantes**. La cláusula GROUP BY se utiliza para agrupar los resultados según el ID del distrito, asegurando que la función de agregación (COUNT) se aplique a cada distrito individualmente. Posteriormente, la cláusula HAVING filtra los resultados agrupados, incluyendo solo aquellos distritos que tienen más de un estudiante matriculado. En resumen, la consulta identifica y devuelve los distritos que tienen una pluralidad de estudiantes matriculados en la base de datos.

**Consulta acerca del promedio del monto de pago por matricula, mostrando solo promedios superiores a 350 soles.**

SELECT nivel\_m, AVG(monto\_Pago) AS promedio\_pago

FROM matricula

GROUP BY nivel\_m

HAVING promedio\_pago > 350;

Tabla

Descripción generada automáticamente

La consulta en cuestión busca obtener información detallada acerca de los niveles educativos que tienen un promedio de pago de matrícula superior a 350. Utilizando la tabla matricula, se selecciona el nivel educativo (**nivel\_m)** y se calcula el promedio de los montos de pago de matrícula para cada nivel, denominando este valor como **promedio\_pago**. La cláusula GROUP BY se emplea para agrupar los resultados según el nivel educativo, asegurando que la función de agregación (AVG) se aplique individualmente a cada nivel. Posteriormente, la cláusula HAVING filtra los resultados agrupados, incluyendo únicamente aquellos niveles educativos cuyo promedio de pago de matrícula sea superior a 350. En resumen, la consulta identifica y devuelve los niveles educativos con un promedio de pago de matrícula más elevado que el umbral establecido de 350 en la base de datos.

**Consulta acerca de la cantidad de matrículas por administrador, mostrar solo administradores con más de 1 matricula.**

SELECT i.id\_admini, a.nombre\_Admini, COUNT(\*) AS cantidad\_matriculas

FROM iniciosesion i

JOIN administrador a ON i.id\_admini = a.id\_admini

JOIN matricula m ON i.id\_Usuario = m.id\_Usuario

GROUP BY i.id\_admini

HAVING cantidad\_matriculas > 1;



La consulta en cuestión busca identificar y proporcionar información detallada sobre los administradores que han realizado más de una matrícula en la base de datos. Utilizando las tablas **iniciosesion**, administrador, y matricula, se selecciona el ID y nombre del administrador, y se cuenta el número de matrículas realizadas por cada administrador, denominando este valor como **cantidad\_matriculas**. La cláusula JOIN se utiliza para relacionar las sesiones de inicio de sesión con la información del administrador y las matrículas. Posteriormente, se agrupan los resultados por el ID del administrador mediante la cláusula GROUP BY, asegurando que la función de agregación (COUNT) se aplique individualmente a cada administrador. La cláusula HAVING filtra los resultados agrupados, incluyendo únicamente aquellos administradores que han realizado más de una matrícula. En resumen, la consulta identifica y presenta los administradores activos que han tenido un impacto significativo en el proceso de matriculación al realizar múltiples operaciones de este tipo en la base de datos.

**FUNCIONES**

**Consulta para saber el número de representantes por consulta**

SELECT id\_distritoR, COUNT(\*) AS cantidad\_representantes

FROM representante

GROUP BY id\_distritoR;

**Tabla

Descripción generada automáticamente**

La consulta realiza un recuento de representantes por distrito en la tabla representante. Utiliza la función de agregación **COUNT**(\*) para contar la cantidad de filas correspondientes a cada distrito, agrupando los resultados mediante la cláusula GROUP BY **id\_distritoR**. El resultado incluye una lista de distritos **(id\_distritoR**) junto con la cantidad de representantes en cada uno (**cantidad\_representantes**). Es una manera eficiente de obtener una visión resumida de la distribución de representantes en la base de datos, agrupada por distrito.

**Consulta del promedio de edad de los alumnos por nivel educativo**

SELECT nivel\_ing, AVG(YEAR(CURDATE()) - YEAR(fec\_Nac)) AS promedio\_edad

FROM alumno

GROUP BY nivel\_ing;

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

La consulta realiza un análisis detallado de la edad promedio de los alumnos en la tabla alumno agrupados por el nivel de ingreso (**nivel\_ing**). Utiliza la función de agregación AVG junto con la expresión YEAR(**CURDATE**()) - YEAR(**fec\_Nac**) para calcular la diferencia en años entre el año actual y el año de nacimiento de cada alumno, obteniendo así la edad. La consulta finalmente devuelve una lista que muestra el nivel de ingreso y el correspondiente promedio de edad de los alumnos en cada grupo, proporcionando una visión resumida de la distribución de edades según los niveles de ingreso en la base de datos.

**Consulta con representante con más hijos matriculados**

SELECT r.id\_Repre, r.nom\_Repre, COUNT(\*) AS cantidad\_hijos\_matriculados

FROM representante r

JOIN alumno a ON r.id\_Repre = a.id\_Repre

JOIN matricula m ON a.id\_Alumno = m.id\_Alumno

GROUP BY r.id\_Repre, r.nom\_Repre

ORDER BY cantidad\_hijos\_matriculados DESC

LIMIT 1;

****

La consulta SQL proporcionada busca identificar al representante con la mayor cantidad de hijos matriculados en la base de datos. Utiliza las tablas representante, alumno, y matricula, realizando **joins** basados en los identificadores correspondientes. Luego, agrupa los resultados por el identificador y nombre del representante, contando la cantidad de hijos matriculados con **COUNT**(\*) y asignando este resultado a la columna **cantidad\_hijos\_matriculados**. Posteriormente, ordena los resultados de manera descendente según la cantidad de hijos matriculados y limita el resultado a una sola fila con LIMIT 1. En resumen, la consulta proporciona información sobre el representante con la mayor cantidad de hijos matriculados en la base de datos.

**FUNCIONES CADENA**

SELECT id\_Repre, CONCAT(nom\_Repre, ' ', ape\_PRe, ' ', ape\_MRe) AS nombre\_completo

FROM representante;

Tabla

Descripción generada automáticamente

La consulta selecciona la columna **id\_Repre** de la tabla representante y utiliza la función CONCAT de MySQL para crear un nuevo campo llamado **nombre\_completo**. Este campo combina los valores de las columnas **nom\_Repre** (nombre del representante), **ape\_PRe** (apellido paterno del representante) y **ape\_MRe** (apellido materno del representante), separados por espacios. Dando así una columna **nombre\_completo.** En esencia, la consulta genera una lista que muestra los identificadores únicos de los representantes junto con sus nombres completos, facilitando la presentación de información más legible y comprensible.

**consulta para mostrar la habilidad mas popular entre los alumnos**

SELECT h.habilidad, COUNT(\*) AS cantidad\_alumnos

FROM alumno a

JOIN habilidad h ON a.id\_habilidad = h.id\_habilidad

GROUP BY h.habilidad

ORDER BY cantidad\_alumnos DESC

LIMIT 1;



La consulta determinar la habilidad más común entre los alumnos de la base de datos. Utiliza una operación de unión entre las tablas alumno y habilidad basada en la correspondencia de los identificadores de habilidad. La instrucción SELECT elige la columna de habilidad y cuenta la cantidad de alumnos asociados a cada habilidad mediante la función de agregación **COUNT**(\*), almacenando este recuento en una columna llamada **cantidad\_alumnos.** La cláusula GROUP BY organiza los resultados por habilidad, y la cláusula ORDER BY los dispone en orden descendente según la cantidad de alumnos. Finalmente, con LIMIT 1, la consulta presenta solo la habilidad más común entre los alumnos, destacando así la información relevante sobre las preferencias o inclinaciones en habilidades dentro de la base de datos.

**Consulta para contar el número de matrículas realizadas por cada administrador**

SELECT i.id\_admini, a.nombre\_Admini, COUNT(\*) AS cantidad\_matriculas

FROM iniciosesion i

JOIN administrador a ON i.id\_admini = a.id\_admini

JOIN matricula m ON i.id\_Usuario = m.id\_Usuario

GROUP BY i.id\_admini, a.nombre\_Admini;

**Tabla

Descripción generada automáticamente**

La consulta realiza un análisis detallado de la actividad de matriculación en la base de datos. A través de las cláusulas JOIN, la consulta une las tablas **iniciosesion** (**i**), administrador (**a**), y matricula (**m**) utilizando los campos correspondientes. La cláusula SELECT selecciona el identificador del administrador **(i.id\_admini),** el nombre del administrador (**a.nombre\_Admini**), y utiliza la función de agregación COUNT(\*) para determinar la cantidad de matrículas realizadas por cada administrador, etiquetándolo como **cantidad\_matriculas**. La cláusula GROUP BY agrupa los resultados por el identificador del administrador y su nombre. En última instancia, la consulta proporciona una lista de administradores junto con la cantidad total de matrículas realizadas por cada uno, lo que facilita el análisis de la contribución individual de los administradores al proceso de matriculación en la institución educativa.

**FUNCIONES ESCALARES**

**Obtener nombre de una habilidad a partir de su ID**

DELIMITER //

CREATE FUNCTION GetHabilidadName(habilidad\_id INT)

RETURNS VARCHAR(255)

DETERMINISTIC

READS SQL DATA

BEGIN

DECLARE habilidad\_name VARCHAR(255);

SELECT habilidad INTO habilidad\_name

FROM habilidad

WHERE id\_habilidad = habilidad\_id;

RETURN habilidad\_name;

END //

DELIMITER ;

SELECT id\_habilidad, GetHabilidadName(id\_habilidad) AS nombre\_habilidad

FROM habilidad;

**Tabla

Descripción generada automáticamente**

La función escalar **GetHabilidadName** creada se utiliza para obtener el nombre de una habilidad a partir de su ID. Esta función, definida con **DETERMINISTIC** y **READS SQL DATA**, declara una variable **habilidad\_name** de tipo **VARCHAR**(255). Luego, realiza una consulta SQL para seleccionar el nombre de la habilidad correspondiente al **habilidad\_id** proporcionado y almacena el resultado en la variable. Finalmente, la función devuelve el nombre de la habilidad. Esto permite obtener nombres legibles en consultas en lugar de trabajar directamente con **IDs**, como se muestra en el ejemplo que devuelve una lista de **IDs** de habilidades junto con sus nombres correspondientes.

**Función el cual obtiene el nombre de un distrito a partir de su ID**

DELIMITER //

CREATE FUNCTION GetDistrictName(distrito\_id INT)

RETURNS VARCHAR(255)

DETERMINISTIC

READS SQL DATA

BEGIN

DECLARE distrito\_name VARCHAR(255);

SELECT distritoA INTO distrito\_name

FROM distriitoalum

WHERE id\_distritoAlum = distrito\_id;

RETURN distrito\_name;

END //

DELIMITER ;

SELECT id\_distritoAlum, GetDistrictName(id\_distritoAlum) AS nombre\_distrito

FROM distriitoalum;

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

***Búsqueda 10***

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

La función escalar llamada **GetAdminName** se crea con el objetivo de obtener el nombre de un administrador a partir de su ID. El bloque de código utiliza un delimitador temporal // para definir la función, establece que la función es determinística y realiza operaciones de lectura en la base de datos. Dentro del bloque, se declara una variable local para almacenar el nombre del administrador y se utiliza una consulta para seleccionar el nombre correspondiente a partir del ID proporcionado. Finalmente, la función devuelve el nombre como resultado. Este enfoque modulariza la lógica de obtener el nombre del administrador, permitiendo su reutilización en consultas para mejorar la legibilidad y mantenibilidad del código.