



## **ABP BASE DE DATOS**

## **INTEGRANTES**

Jesús David Montiel Vergara

Perez Arrieta Boris Yesid

## **PROFESOR**

Jhon Carlos Arrieta

CARTAGENA, BOLÍVAR

22/04/2021

## Exportar un archivo Excel a una base de datos MySQL

Para exportar un Archivo Excel a una base de datos MySQL, luego de hacer la respectiva limpieza y estructuración de los datos realizamos los siguientes pasos:

1. Se guarda el documento con el mismo nombre de la tabla en MySQL con el formato CSV (MS-DOS) como se evidencia en la *figura número 1*.
2. Vamos a MySQL y entramos a nuestra tabla donde insertamos los datos y en el banner superior se entra al apartado: "Importar" como se evidencia en la *figura número 2*.
3. Una vez dentro, se subirá el archivo .CSV que se guardó en el paso 1 haciendo clic en elegir archivo como se evidencia en la *figura 3*.
4. Seleccionamos el archivo .CSV creado en el paso 1.
5. Antes de dar a continuar, nos dirigimos a la parte inferior en las Opciones específicas al formato y en: "Columnas separadas por: "
6. Una vez de haber importados todos los datos del documento .CSV podemos visualizar que se han insertado de una manera satisfactoria

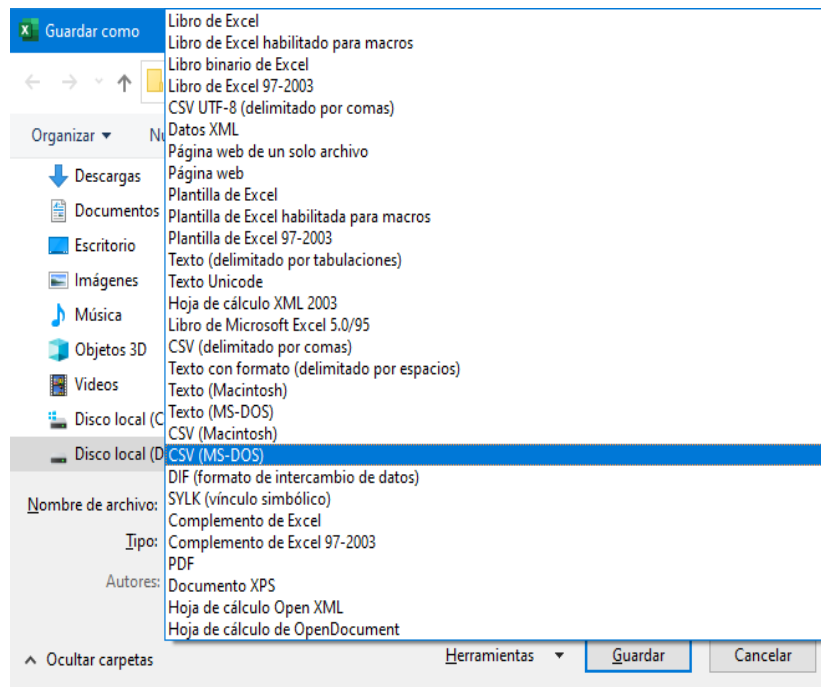


Figura 1.

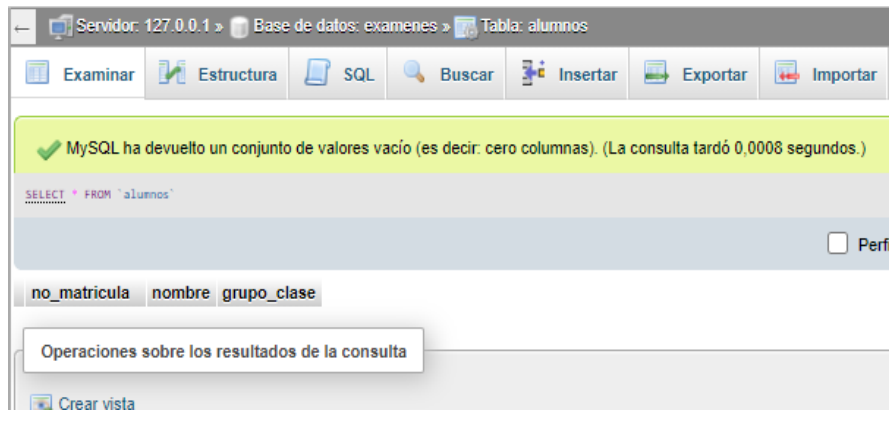


Figura 2.

## Importando en la tabla "alumnos"

### Archivo a importar:

El archivo puede ser comprimido (gzip, bzip2, zip) o descomprimido.

A compressed file's name must end in `.[format].[compression]`. Example: `.sql.zip`

Buscar en su ordenador:  No se eligió ningún archivo (Máximo: 40MB)

También puede arrastrar un archivo en cualquier página.

Conjunto de caracteres del archivo:

### Importación parcial:

Figura 3.

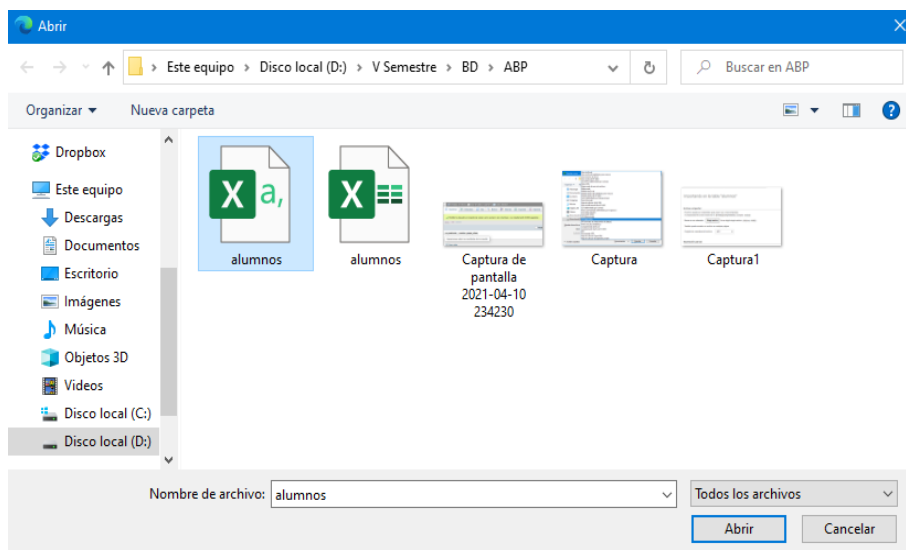


Figura 4.

Opciones específicas al formato:

☐ Actualizar datos cuando las llaves importadas están duplicadas (agregar ON DUPLICATE KEY UPDATE)

Columnas separadas por:

Columnas encerradas entre:

Caracter de escape de columnas:

Líneas terminadas en:

Importe este gran número de filas (opcional):

nombres de columna:

☐ No abortar si ocurre un error con INSERT

Continuar

Figura 5.

Servidor: 127.0.0.1 » Base de datos: exámenes » Tabla: alumnos

Ocultar panel

ExaminarEstructuraSQLBuscarInsertarExportarImportarPrivilegiosOperacionesMás

Importación ejecutada exitosamente, 24 consultas ejecutadas. (alumnos.csv)

1 fila insertada. (La consulta tardó 0.3475 segundos.)

INSERT INTO `alumnos` VALUES ('100070012', 'ADRIANA CAROLINA HERNANDEZ MONTERROZA', 'ING\_SIST 120CSIS-1')

[Editar en línea][Editar][Crear código PHP]

1 fila insertada. (La consulta tardó 0.0769 segundos.)

INSERT INTO `alumnos` VALUES ('100070013', 'ADRIANA MARCELA REY SANCHEZ', 'ING\_IND 120CSIS-2')

[Editar en línea][Editar][Crear código PHP]

1 fila insertada. (La consulta tardó 0.2144 segundos.)

INSERT INTO `alumnos` VALUES ('100070014', 'ALEJANDRO ARONIANO ACEVEDO', 'ING\_CVL 120CSIS-3')

[Editar en línea][Editar][Crear código PHP]

1 fila insertada. (La consulta tardó 0.0640 segundos.)

INSERT INTO `alumnos` VALUES ('100070015', 'ALEJANDRO ARONIANO ACEVEDO', 'ING\_SIST 120CSIS-1')

[Editar en línea][Editar][Crear código PHP]

1 fila insertada. (La consulta tardó 0.0709 segundos.)

INSERT INTO `alumnos` VALUES ('100070016', 'ANDREA CATALINA ACERO CARDI', 'ING\_SIST 120CSIS-1')

[Editar en línea][Editar][Crear código PHP]

Figura 5.

	no_matricula	nombre	grupo_clase
<input type="checkbox"/>	100070012	ADRIANA CAROLINA HERNANDEZ MONTERROZA	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070013	ADRIANA MARCELA REY SANCHEZ	ING_IND 128CSIS-2
<input type="checkbox"/>	100070014	ALEJANDRO ABONDANO ACEVEDO	ING_CVL 128CSIS-3
<input type="checkbox"/>	100070015	ALEJANDRO ABONDANO ACEVEDO	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070016	ANDREA CATALINA ACERO CARO	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070017	ANDREA LILIANA CRUZ GARCIA	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070018	ANDRES FELIPE VILLA MONROY	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070019	ANGELA PATRICIA MAHECHA PIÑEROS	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070020	ANGELICA LISSETH BLANCO CONCHA	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070021	ANGELICA MARIA ROCHA GARCIA	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070022	ANGIE TATIANA FERNÁNDEZ MARTÍNEZ	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070023	TATIANA ANGIE MARTÍNEZ FERNÁNDEZ	ING_SIST 128CSIS-1
<input type="checkbox"/>	100070024	CAMILO VILLAMIZAR ARISTIZABAL	ING_IND 128CSIS-2
<input type="checkbox"/>	100070025	CAMILO RODRÍGUEZ BOTERO	ING_IND 128CSIS-2
<input type="checkbox"/>	100070026	CAMILO ALBERTO CORTÉS MONTEJO	ING_IND 128CSIS-2
<input type="checkbox"/>	100070027	CAMILO ENRIQUE GÓMEZ RODRÍGUEZ	ING_IND 128CSIS-2
<input type="checkbox"/>	100070028	CARLOS ANDRÉS POLO CASTELLANOS	ING_IND 128CSIS-2
<input type="checkbox"/>	100070029	CARLOS DIDIER CASTAÑO CONTRERAS	ING_CVL 128CSIS-3

Figura 6.

## Sentencias para consultas utilizadas

Seleccionamos todo lo que hay en la tabla *alumnos* donde el valor de la columna *nombre* sea todos *nombre* que comienzan con la inicial A.

```
select * from alumnos where nombre like "a%";
```

Seleccionamos la columna *nombre* de la tabla *acceso* donde la columna *codigo* sea igual al valor de: *100070013* y se obtendrá como resultado el nombre del usuario.

```
SELECT nombre FROM acceso WHERE codigo = "100070013";
```

Seleccionamos la columna *rol* de la tabla *acceso* donde la columna *codigo* sea igual al valor de *100070013* y se obtendrá como resultado el rol del usuario.

```
SELECT rol FROM acceso WHERE codigo = "100070013";
```

Seleccionamos las columnas *codigo* y *password* de la tabla *acceso* donde *codigo* sea igual a: *100070013* y *password* sea igual a: *123456* y se obtendrán sus credenciales de acceso para ser validadas.

```
SELECT codigo, password FROM acceso WHERE codigo = "100070013" && password = "123456" ;
```

Seleccionamos todos los datos que tiene la tabla *alumnos* incluyendo la tabla donde está siendo referenciada (llave foránea) pero al visualizar las tablas sale dos veces la columna *no\_matricula*

```
SELECT * FROM alumnos INNER JOIN alumnos_controles_escritos ON alumnos.no_matricula = alumnos_controles_escritos.alumnos_no_matricula
```

Seleccionamos todos los datos de la comuna *alumnos* pero un poco más ordenas de la consulta mostrada anteriormente

```
SELECT no_matricula,nombre,grupo_clase,fecha,calificacion,controles_escritos_no_control FROM alumnos INNER JOIN alumnos_controles_escritos ON alumnos.no_matricula = alumnos_controles_escritos.alumnos_no_matricula
```

Actualizamos en la tabla *alumnos\_controles\_escritos* la columna *calificacion* con el valor de: *4.5* donde la columna *alumnos\_no\_matricula* sea igual al valor de: *100070013* para editar la antigua calificación del estudiante.

```
UPDATE alumnos_controles_escritos SET calificacion= "4.5" WHERE alumnos_no_matricula= "100070013";
```

Seleccionamos *nombre* y *apellido* de la tabla *profesores* y esto nos regresa el nombre y apellido de todos los profesores registrados en la tabla profesores.

```
SELECT nombre,apellido FROM profesores;
```

Seleccionamos el nombre y apellido, utilizando AS le podemos cambiar por cualquier otro parámetro donde no afectará esos cambios en la base de datos solo se mostrará el cambio cuando se ejecuta la consulta

```
SELECT nombre AS name ,apellido AS surname FROM profesores;
```

Seleccionamos todos los datos distintos de la columna *grupo\_clase* de la tabla *alumnos*.

```
SELECT DISTINCT grupo_clase FROM alumnos;
```

Selecciona todos los datos de la tabla *acceso* y nos ordena los nombre de forma ascendente (A-Z)

```
SELECT * FROM acceso ORDER BY nombre ASC
```

Seleccionamos todos los datos de la tabla *acceso* y nos ordena los nombres de forma descendente (Z-A)

```
SELECT * FROM acceso ORDER BY nombre DESC
```

Seleccionamos todos los campos de la tabla *alumnos* y agrupamos a todos los estudiantes que pertenezcan al grupo de clase *ING\_SIST 128CSIS-1*

```
SELECT  
no_matricula,nombre,grupo_clase,fecha,calificacion,controles_escritos_no_co  
ntrol from alumnos INNER JOIN alumnos_controles_escritos ON  
alumnos.no_matricula = alumnos_controles_escritos.alumnos_no_matricula  
WHERE grupo_clase = 'ING_SIST 128CSIS-1'
```

Seleccionamos todo de la tabla *acceso* donde el valor de la columna *nombre* sea todos los nombres que inician con la letra *a* y el rol sea *alumno*.

```
SELECT * FROM acceso WHERE nombre LIKE "a%" AND rol = "alumno"
```

Seleccionamos todo de la tabla *acceso* donde el valor de la columna *nombre* sea todos los nombres que inician con la letra *Y* y el rol sea *profesor*.

```
SELECT * FROM acceso WHERE nombre LIKE "y%" AND rol = "profesor"
```

Seleccionamos la columna *nombre* de la tabla *alumnos* y lo unimos con lo que hay en la tabla *nombre* de la tabla *profesores*.

```
SELECT nombre FROM alumnos UNION ALL SELECT nombre FROM profesores
```

Seleccionamos todas las calificaciones de los alumnos y con la función SUM sumamos todas las calificaciones que se encuentran en la columna calificación

```
SELECT SUM(calificacion) FROM alumnos INNER JOIN alumnos_controles_escritos  
ON alumnos.no_matricula = alumnos_controles_escritos.alumnos_no_matricula
```

Seleccionamos todas las calificaciones de los alumnos y con la función AVG sacamos el promedio de las calificaciones de los alumnos y con la función ROUND redondeamos el resultado del promedio

```
SELECT ROUND(AVG(calificacion),2) FROM alumnos INNER JOIN  
alumnos_controles_escritos ON alumnos.no_matricula =  
alumnos_controles_escritos.alumnos_no_matricula
```

Seleccionamos el nombre y con la función MAX obtenemos la calificación máxima que un alumno obtuvo

```
SELECT nombre, MAX(calificacion) FROM alumnos INNER JOIN  
alumnos_controles_escritos ON alumnos.no_matricula =  
alumnos_controles_escritos.alumnos_no_matricula
```

Seleccionamos todos los datos y con la función COUNT sabemos cuantos datos están registrados en la base de datos

```
SELECT COUNT(codigo) FROM acceso
```

Seleccionamos todos los datos del alumnos y con HAVING sacamos todos los alumnos que tenga una calificación superior a 3.20

```
SELECT  
no_matricula,nombre,grupo_clase,fecha,calificacion,controles_escritos_no_co  
ntrol FROM alumnos INNER JOIN alumnos_controles_escritos ON  
alumnos.no_matricula = alumnos_controles_escritos.alumnos_no_matricula  
HAVING calificacion > 3.20
```

Seleccionamos la columna que queremos que salga todas en mayúscula con la función UPPER de la tabla acceso

```
SELECT UPPER(nombre) from acceso
```

Seleccionamos la columna que queremos que salga todas en minúsculas con la función LOWER de la



tabla acceso

```
SELECT LOWER(nombre) from acceso
```