

# MANUAL DE PRACTICAS



Nombre de la práctica	Uperaciones CRUD			No.	3
Asignatura:	Taller de Bases de Datos	Carrera:	ISIC	Duración de la práctica (Hrs)	2hrs

Nombre: Jocelin Reyes Rodriguez

Grupo: 3501

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro): Aula

III. Material empleado: Computadora

IV. Desarrollo de la práctica:

```
Operaciones CRUD en
# Operaciones CRUD en MySQL
                                                               MySQL
Las operaciones *CRUD* son un conjunto de 4
operaciones fundamentales, en el manejo de
bases de datos y aplicaciones web. CRUD es un
                                                               Las operaciones CRUD son un conjunto de 4 operaciones
acrónimo que representa las siguiente
                                                               fundamentales, en el manejo de bases de datos y
operaciones:
- **C**REATE
                (Crear)
                                                               aplicaciones web. CRUD es un acrónimo que representa las
- **R**EAD
                (Leer)
                                                               siguiente operaciones:
- **U**pdate
                (Actualizar)
- **D**elete
                (Eliminar)
                                                                  • CREATE (Crear)
                                                                  • READ (Leer)
**Primero creamos una tabla:**

    Update (Actualizar)

```sql
  • Delete (Eliminar)
CREATE TABLE Usuarios(
   Primero creamos una tabla:
    id usuario INT PRIMARY KEY AUTO INCREMENT,
    email VARCHAR(100) UNIQUE NOT NULL CHECK
    (email "%_@_%._%"),
    password VARCHAR(15) NOT NULL CHECK(LENGTH
   CREATE TABLE Usuarios(
    (password) >=8 )
   id_usuario INT PRIMARY KEY
   AUTO_INCREMENT,
   email VARCHAR(100) UNIQUE NOT NULL
   CHECK(email "%_@_%._%"),
   password VARCHAR(15) NOT NULL
## Create
   CHECK(LENGTH(password) >=8 )
La operacion *crear* es responsable de crear
nueos datos en la base de datos en lenguaje
SQL, esto se realiza con la sentencia `INSERT
INTO` y en el caso de MySQL `INSERT` también
funciona. El propósito de la operación es
   Create
añadir el nuevo resgistro a una tabla
```

# GOBIERNO DEL ESTADO DE MÉXICO

# MANUAL DE PRÁCTICAS



```
operacionesCRUD.md > ••• # Operaciones CRUD en MySQL
      # Operaciones CRUD en MySQL
   Create
      ## Create
      La operacion *crear* es responsable de crear
   La operacion crear es responsable de crear nueos datos en
      nueos datos en la base de datos en lenguaje
   la base de datos en lenguaje SQL, esto se realiza con la
      SQL, esto se realiza con la sentencia `INSERT
   sentencia INSERT INTO y en el caso de MySQL INSERT
      INTO` y en el caso de MySQL `INSERT` también
   también funciona. El propósito de la operación es añadir el
      funciona. El propósito de la operación es
   nuevo resgistro a una tabla
      añadir el nuevo resgistro a una tabla
      -- Ejemplo de una insersión valida usando
  -- Ejemplo de una insersión valida usando
      todos los campos
  todos Los campos
      INSERT INTO Usuarios VALUES (1, "ejemplo@mail.
  INSERT INTO Usuarios VALUES (1,
      com", "12345678");
  "ejemplo@mail.com", "12345678");
      -- Ejemplo de una insersión valida usando el
  -- Ejemplo de una insersión valida usando
      comando DEFAULT
      INSERT INTO Usuarios VALUES (DEFAULT,
  INSERT INTO Usuarios VALUES (DEFAULT,
      "ejemplo2@gmail.com", "abcdefgh");
  "ejemplo2@gmail.com", "abcdefgh");
      -- Ejemplo de una insersión sin incluir un **
  -- Ejemplo de una insersión sin incluir un
      id usuario **
  ** id_usuario **
      INSERT Usuario(email, password) VALUES
  INSERT Usuario(email, password) VALUES
      ("email3@hotmail.com", "12345678");
  ("email3@hotmail.com", "12345678");
      ### Ejercicios
   Ejercicios
      - Identifica los tipos de errores que pueden
      salir en esta tabla
      ···sql
   • Identifica los tipos de errores que pueden salir en
           -- Violación de la unicidad del email:
   esta tabla
  Ejercicios
 ### Ejercicios
  - Identifica los tipos de errores que pueden
   • Identifica los tipos de errores que pueden salir en
 salir en esta tabla
   esta tabla
     -- Violación de la unicidad del email:
  -- Violación de la unicidad del email:
 en la tabla
  existe en la tabla
 INSERT INTO Usuarios (email, password)
  INSERT INTO Usuarios (email, password)
 VALUES ('usuario1@mail.com', 'nuevaPassword');
  VALUES ('usuario1@mail.com',
   'nuevaPassword');
     -- Violación de la longitud mínima del
     password
  -- Violación de la longitud mínima del
 INSERT INTO Usuarios (email, password)
 VALUES ('usuario5@mail.com', 'short');
  INSERT INTO Usuarios (email, password)
  VALUES ('usuario5@mail.com', 'short');
 INSERT INTO Usuarios (email, password)
   -- Violación del formato del email
 VALUES ('usuario6mail.com', 'validpassword');
  INSERT INTO Usuarios (email, password)
  VALUES ('usuario6mail.com',
   'validpassword');
 INSERT INTO Usuarios (email, password)
 VALUES (NULL, 'validpassword');
  -- Inserción de valores NULL en campos
  INSERT INTO Usuarios (email, password)
  VALUES (NULL, 'validpassword');
 - Inserta 4 registros nuevos en un solo INSERT
 INSERT INTO Usuarios (email, password)
```



de 8 caracteres

sal

# MANUAL DE PRÁCTICAS



```
### Ejercicios
 - Inserta 4 registros nuevos en un solo INSERT
 INSERT INTO Usuarios (email, password)
  VALUES
 VALUES
     ('usuario1@mail.com', 'contraseña1'),
     ('usuario2@mail.com', 'password1234'),
('usuario3@mail.com', 'abcdefghi'),
     ('usuario4@mail.com', 'mypassword');
   Read
## Read
 La operacion *leer* es utilizada para
 consultar o recuperar datos de la base de
 datos. Esto no modifica los datos, simplemente
 los extrae. En MySQL est operación se realiza
 con la sentencia select
   con la sentencia select
 ···sql
 -- Ejmeplo de una consulta para todos lo datos
 de una tabla
  datos de una tabla
 SELECT * FROM Usuarios;
 -- Ejemplo de consulta para un registro en
 especifico a través del id_usuario
SELECT * FROM Usuarios WHERE id usuario=1;
···sql
  ENSEMBLE
   SELECT * FROM Usuarios:
-- Ejmeplo de una consulta para todos lo datos
de una tabla
SELECT * FROM Usuarios:
   especifico a través del id_usuario
-- Ejemplo de consulta para un registro en
especifico a través del id usuario
SELECT * FROM Usuarios WHERE id_usuario=1;
   SELECT * FROM Usuarios WHERE
-- Ejemplo de consulta para un rgeistro con un
   email="ejemplo@mail.com";
email en específico
SELECT * FROM Usuarios WHERE
email="ejemplo@mail.com";
SELECT email, password FROM Usuarios;
   SELECT * FROM Usuarios WHERE
-- Ejemplo de consulta con un condicional
   LENGTH(password)>9;
SELECT * FROM Usuarios WHERE LENGTH(password)
   Ejercicio
### Ejercicio
- Realiza una consulta que solo muestre el id,
pero que coincida con una constraseña de mas
```

· Inserta 4 registros nuevos en un solo INSERT

```
INSERT INTO Usuarios (email, password)
    ('usuario1@mail.com', 'contraseña1'),
    ('usuario2@mail.com', 'password1234'),
    ('usuario3@mail.com', 'abcdefghi'),
    ('usuario4@mail.com', 'mypassword');
```

La operacion *leer* es utilizada para consultar o recuperar datos de la base de datos. Esto no modifica los datos, simplemente los extrae. En MySQL est operación se realiza

```
-- Eimeplo de una consulta para todos lo
SELECT * FROM Usuarios;
-- Ejemplo de consulta para un registro en
especifico a través del id_usuario
SELECT * FROM Usuarios WHERE id_usuario=1;
```

```
-- Ejemplo de consulta para un registro en
SELECT * FROM Usuarios WHERE id_usuario=1;
-- Ejemplo de consulta para un rgeistro con
SELECT email, password FROM Usuarios;
```

• Realiza una consulta que solo muestre el id, pero que coincida con una constraseña de mas de 8 caracteres



## MANUAL DE PRÁCTICAS



```
Ejercicio
  ### Ejercicio
  - Realiza una consulta que solo muestre el id,
   • Realiza una consulta que solo muestre el id, pero
  pero que coincida con una constraseña de mas
  que coincida con una constraseña de mas de 8
  de 8 caracteres
  caracteres
  ···sql
  SELECT id_usuario
  SELECT id_usuario
  FROM Usuarios
  WHERE LENGTH(password) > 8;
  FROM Usuarios
  WHERE LENGTH(password) > 8;
   - Otra que realice una consulta a los id´s
   • Otra que realice una consulta a los id's pares
    ``sql
  SELECT id usuario
  FROM Usuarios
  SELECT id usuario
  WHERE id_usuario % 2 = 0; -- o MOD
  FROM Usuarios
  WHERE id_usuario % 2 = 0; -- o MOD
  ## Update
  La operación *actualizar* se utiliza para
  Update
  modificar registros existentes en la base de
  datos. Esto se hace con la sentencia `UPDATE`
  ···sql
  La operación actualizar se utiliza para modificar registros
  -- Ejmeplo para actualizar la contraseña por
  existentes en la base de datos. Esto se hace con la
  sentencia UPDATE
  UPDATE Usuario SET password = "a1b2c3d4" WHERE
  id_usuario = 1;
   -- Ejmeplo para actualizar la contraseña
  -- Ejemplo para actualizar el email y password
  por su id
## Update
   La operación actualizar se utiliza para modificar registros
  existentes en la base de datos. Esto se hace con la
   sentencia UPDATE
```

```
La operación *actualizar* se utiliza para
modificar registros existentes en la base de
datos. Esto se hace con la sentencia `UPDATE`
 ``sal
UPDATE Usuario SET password = "a1b2c3d4" WHERE
id_usuario = 1;
-- Ejemplo para actualizar el email y password
UPDATE Usuario SET password = "a1b2c3d4",
email = "luciohdz3012@gmail.com" WHERE
id_usuario = 1;
### Fiercicio
- Intenta actualizar registros con valores que
violen las restricciones (minimo 3)
  `sal
UPDATE Usuarios
SET email = 'usuario2@mail.com'
WHERE id_usuario = 1;
    -- Actualización que viola la longitud
    mínima del password
```

```
-- Ejmeplo para actualizar la contraseña
por su id
UPDATE Usuario SET password = "a1b2c3d4"
WHERE id_usuario = 1;
```

-- Ejemplo para actualizar el email y password de un usaurio en específico UPDATE Usuario SET password = "a1b2c3d4", email = "luciohdz3012@gmail.com" WHERE id\_usuario = 1;

#### **Ejercicio**

 Intenta actualizar registros con valores que violen las restricciones (minimo 3)

```
-- Actualización que viola la unicidad
del email:
-- Asumiendo que los emails
'usuario1@mail.com' y 'usuario2@mail.com'
ya existen
UPDATE Usuarios
```



cuidadosos con esta operación, ya que una vez

que los datos son eliminados, NO pueden ser

**RECUPERADOS\*\*** 

## MANUAL DE PRÁCTICAS



Debemos ser muy cuidadosos con esta operación, ya

que una vez que los datos son eliminados, NO pueden

```
-- Asumiendo que los emails 'usuario1@mail.
com' y 'usuario2@mail.com' ya existen
   UPDATE Usuarios
UPDATE Usuarios
   SET email = 'usuario2@mail.com'
SET email = 'usuario2@mail.com'
   WHERE id_usuario = 1;
WHERE id_usuario = 1;
  -- Actualización que viola la longitud
    -- Actualización que viola la longitud
   mínima del password
    mínima del password
   UPDATE Usuarios
UPDATE Usuarios
   SET password = 'short'
SET password = 'short'
   WHERE id_usuario = 2;
WHERE id_usuario = 2;
    -- Actualización que viola el formato del
   del email:
   UPDATE Usuarios
    email:
UPDATE Usuarios
   SET email = 'invalidemail.com'
SET email = 'invalidemail.com'
   WHERE id_usuario = 3;
WHERE id_usuario = 3;
   Delete
## Delete
La operacón *elimianr* se usa para borrar
registros de la base de datos. Esto se realiza
   La operacón elimianr se usa para borrar registros de la
con la sentEncia `DELETE`. **Debemos ser muy
   base de datos. Esto se realiza con la sentEncia DELETE.
```



## MANUAL DE PRÁCTICAS



```
La operacon elimiani se usa para porrar registros de la
             .c. rilliralli . Sc nsa hala nollai
registros de la base de datos. Esto se realiza
   base de datos. Esto se realiza con la sentEncia DELETE.
con la sentEncia `DELETE`. **Debemos ser muy
   Debemos ser muy cuidadosos con esta operación, ya
cuidadosos con esta operación, ya que una vez
   que una vez que los datos son eliminados, NO pueden
que los datos son eliminados, NO pueden ser
   ser RECUPERADOS
RECUPERADOS**
···sql
  -- Eliminar el usuario por el id
-- Eliminar el usuario por el id
   DELETE FROM Usuarios WHERE id_usuario = 4;
DELETE FROM Usuarios WHERE id_usuario = 4;
  -- Eliminar los usuarios con el email
-- Eliminar los usuarios con el email
   específico
   DELETE FROM Usuarios WHERE
DELETE FROM Usuarios WHERE
   email="luciohdz3012@gmail.com";
email="luciohdz3012@gmail.com";
### Eercicios
   Eercicios
- Eliminar usuarios cuyo email contenga 1 o
mas "5"
- Eliminar usuarios que tengan una contraseña
  • Eliminar usuarios cuyo email contenga 1 o mas "5"
que contengan letras mayusculas usando
   • Eliminar usuarios que tengan una contraseña que
expresiones regulares (REGEX)
   contengan letras mayusculas usando expresiones
- Elminar usuarios con contraseña que
   regulares (REGEX)
contengan solo numeros
   • Elminar usuarios con contraseña que contengan solo
- Eliminar usuarios con correos que no
contengan el dominio "gmail"
   • Eliminar usuarios con correos que no contengan el
```sql
                                                                       dominio "gmail"
 -- Eliminar usuarios cuyo email contenga uno
 o más "5"
DELETE FROM Usuarios
                                                                    uno o más "5"
```



## MANUAL DE PRÁCTICAS



```
### Eercicios
contengan el dominio "gmail"
                                                                 -- Eliminar usuarios cuyo email contenga
                                                                uno o más "5"
                                                                 DELETE FROM Usuarios
 -- Eliminar usuarios cuyo email contenga uno
                                                                WHERE email REGEXP '5';
o más "5"
DELETE FROM Usuarios
WHERE email REGEXP '5';
                                                                DELETE FROM Usuarios
-- Eliminar usuarios que tengan una contraseña
                                                                WHERE password REGEXP '[A-Z]';
que contenga letras mayúsculas
DELETE FROM Usuarios
                                                                -- Eliminar usuarios cuya contraseña
WHERE password REGEXP '[A-Z]';
                                                                contenga solo números
                                                                DELETE FROM Usuarios
-- Eliminar usuarios cuya contraseña contenga
                                                                WHERE password REGEXP '^[0-9]+$';
solo números
DELETE FROM Usuarios
                                                                 -- Eliminar usuarios cuya contraseña
WHERE password REGEXP '^[0-9]+$';
                                                                contenga solo números
                                                                DELETE FROM Usuarios
-- Eliminar usuarios cuya contraseña contenga
                                                                WHERE password REGEXP '^[0-9]+$';
solo números
DELETE FROM Usuarios
                                                                 -- Eliminar usuarios con correos que no
WHERE password REGEXP '^[0-9]+$';
                                                                DELETE FROM Usuarios
-- Eliminar usuarios con correos que no
                                                                WHERE email NOT LIKE '%@gmail.com';
contengan el dominio "gmail"
DELETE FROM Usuarios
WHERE email NOT LIKE '%@gmail.com';
```

#### V. Conclusiones:

Las operaciones CRUD (Crear, Leer, Actualizar y Eliminar) son fundamentales en la gestión de bases de datos SQL. Las operaciones CRUD permiten estructurar la interacción con la base de datos de manera clara y organizada, facilitando el desarrollo y mantenimiento de aplicaciones.

Al realizar estas operaciones, es crucial aplicar restricciones y validaciones para mantener la integridad de los datos. Esto incluye el uso de claves primarias y foráneas, así como restricciones de unicidad y no nulidad. Las operaciones de lectura pueden optimizarse mediante el uso de índices, lo que mejora el rendimiento de las consultas y hace que la recuperación de datos sea más eficiente.

En resumen, las operaciones CRUD son la base para interactuar con bases de datos SQL, y su correcta implementación y gestión son cruciales para el éxito de cualquier aplicación que dependa de la persistencia de datos.