

Utilización de técnicas de acceso a datos

DESARROLLO WEB EN ENTORNO SERVIDOR

6.1

CONECTAR CON UNA BASE DE DATOS

- **MySQL 4.1 y posteriores:**

```
$conector =  
mysqli_connect($host,$user,$password[,dbname]);
```

- **MySQL 4.0 y anteriores:**

```
$conector = mysql_connect($host,$user,$password);  
mysql_select_db($dbname);
```

EJECUTAR SENTENCIAS DE BASE DE DATOS

- **MySQL 4.1 y posteriores:**

```
$result =  
mysqli_query($conector,$sentencia[, $modo_resultado]);
```

- **MySQL 4.0 y anteriores:**

```
$result = mysql_query($sentencia[, $conector]);
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Crear una BD:

```
CREATE DATABASE nombre_db
```

- Borrar una BD:

```
DROP DATABASE nombre_db
```

- Crear una tabla:

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Borrar una tabla:

```
DROP TABLE nombre_tabla
```

- Borrar contenido de una tabla:

```
TRUNCATE TABLE nombre_tabla
```

- Modificar una tabla:

```
ALTER TABLE nombre_tabla opciones
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Las opciones que se pueden añadir:

- Añadir una columna a una tabla:

```
ADD COLUMN nombre_columna tipo
```

- Eliminar una columna de una tabla:

```
DROP COLUMN nombre_columna
```

- Modificar el tipo de una columna:

```
ALTER COLUMN nombre_columna tipo_nuevo
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Añadir una restricción:
 - NOT NULL (campo obligatorio y que no puede contener un valor nulo):

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- UNIQUE (campo cuyo valor debe ser único):

- **MySQL:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  UNIQUE(nombre_columna1)  
)
```

- **SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL UNIQUE,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN  
)
```


SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **MySQL/SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  CONSTRAINT nombre_restricción  
  UNIQUE(nombre_columna1,...)  
)
```

- **Otras:**

1. ALTER TABLE nombre_tabla ADD UNIQUE(nombre_columna1)
2. ALTER TABLE nombre_tabla ADD CONSTRAINT
 nombre_restricción UNIQUE(nombre_columna1,...)

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- PRIMARY KEY(clave primaria de una tabla):

- **MySQL:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  PRIMARY KEY(nombre_columna1)  
)
```

- **SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL PRIMARY KEY,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **MySQL/SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  CONSTRAINT nombre_restricción PRIMARY KEY  
    (nombre_columna1,...)  
)
```

- **Otras:**

1. ALTER TABLE nombre_tabla ADD PRIMARY KEY (nombre_columna1)
2. ALTER TABLE nombre_tabla ADD CONSTRAINT nombre_restricción
PRIMARY KEY (nombre_columna1,...)

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- FOREIGN KEY(clave ajena de una tabla):

- **MySQL:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  FOREIGN KEY(nombre_columna) REFERENCES nombre_tabla2(PK_tabla2)  
)
```

- **SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1,  
  nombre_columna2 tipo2 FOREIGN KEY REFERENCES  
  nombre_tabla2(PK_tabla2),  
  ...  
  nombre_columnaN tipoN  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **MySQL/SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
    nombre_columna1 tipo1 NOT NULL,  
    nombre_columna2 tipo2,  
    ...  
    nombre_columnaN tipoN,  
    CONSTRAINT nombre_restricción FOREIGN KEY  
        (nombre_columna) REFERENCES  
        nombre_tabla2(PK_tabla2,...))
```

- **Otras:**

1. ALTER TABLE nombre_tabla1 ADD FOREIGN KEY (nombre_columna) REFERENCES nombre_tabla2(PK_tabla2)
2. ALTER TABLE nombre_tabla ADD CONSTRAINT nombre_restricción FOREIGN KEY (nombre_columna) REFERENCES nombre_tabla2(PK_tabla2,...)

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- DEFAULT (campo con valor por defecto):

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1,  
  nombre_columna2 tipo2 DEFAULT valor,  
  ...  
  nombre_columnaN tipoN  
)
```

- Otras:

- **MySQL:**

```
ALTER TABLE nombre_tabla  
  ALTER nombre_columna SET DEFAULT valor
```

- **SQL Server/Oracle/MS Access:**

```
ALTER TABLE nombre_tabla  
  ALTER COLUMN nombre_columna SET DEFAULT valor
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- CHECK (establece la condición que debe cumplir un campo):

- **MySQL:**

```
CREATE TABLE nombre_tabla(  
    nombre_columna1 tipo1,  
    nombre_columna2 tipo2,  
    ...  
    nombre_columnaN tipoN,  
    CHECK(condición)  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **SQL Server:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 CHECK(condición),  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN  
)
```

- **MySQL/SQL Server/Oracle/MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  CONSTRAINT nombre_restricción CHECK(condición1 AND ...)  
)
```


SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **Otras:**

1. ALTER TABLE nombre_tabla ADD CHECK(condición)
2. ALTER TABLE nombre_tabla ADD CONSTRAINT nombre_restricción CHECK(condición1 AND ...)

- AUTO_INCREMENT (crear índice que se autoincrementa):

- **MySQL:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 NOT NULL AUTO_INCREMENT,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
  PRIMARY KEY(nombre_columna1)  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **SQL Server:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 PRIMARY KEY IDENTITY,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
)
```

- **MS Access:**

```
CREATE TABLE nombre_tabla(  
  nombre_columna1 tipo1 PRIMARY KEY AUTOINCREMENT,  
  nombre_columna2 tipo2,  
  ...  
  nombre_columnaN tipoN,  
)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- **Oracle:**

```
CREATE SEQUENCE nombre_seq MINVALUE 1  
START WITH 1  
INCREMENT BY 1  
CACHE 10
```

```
INSERT INTO nombre_tabla (nombre_columna1,...)VALUES  
(nombre_secuencia.nextval,valor2,...)
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Eliminar una restricción:
 - UNIQUE:
 - **MySQL:**
`ALTER TABLE nombre_tabla DROP INDEX nombre_columna`
 - **SQL Server/Oracle/MS Access:**
`ALTER TABLE nombre_tabla DROP CONSTRAINT nombre_columna`
 - PRIMARY KEY:
 - **MySQL:**
`ALTER TABLE nombre_tabla DROP PRIMARY KEY`
 - **SQL Server/Oracle/MS Access:**
`ALTER TABLE nombre_tabla DROP CONSTRAINT nombre_columna`

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- FOREIGN KEY:

- **MySQL:**

- ```
ALTER TABLE nombre_tabla DROP FOREIGN KEY nombre_columna
```

- **SQL Server/Oracle/MS Access:**

- ```
ALTER TABLE nombre_tabla DROP CONSTRAINT nombre_columna
```

- DEFAULT:

- **MySQL:**

- ```
ALTER TABLE nombre_tabla ALTER nombre_columna DROP DEFAULT
```

- **SQL Server/Oracle/MS Access:**

- ```
ALTER TABLE nombre_tabla ALTER COLUMN nombre_columna DROP  
DEFAULT
```

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- CHECK:
 - **SQL Server/Oracle/MS Access:**
ALTER TABLE nombre_tabla DROP CONSTRAINT nombre_restricción
- Crear una vista:
CREATE VIEW nombre_vista AS SELECT nombre_columna1, ...
FROM nombre_tabla WHERE condición
- Eliminar una vista:
DROP VIEW nombre_vista

SENTENCIAS DE DEFINICIÓN DE DATOS (DDL)

- Crear un índice en una tabla:
 1. `CREATE INDEX nombre_índice ON TABLE nombre_tabla (nombre_columna)`
 2. `CREATE UNIQUE INDEX nombre_índice ON TABLE nombre_tabla (nombre_columna)`
- Eliminar un índice en una tabla:
 - **MS Access:**
`DROP INDEX nombre_índice ON nombre_tabla`
 - **SQL Server:**
`DROP INDEX nombre_tabla.nombre_índice`
 - **DB2/Oracle:**
`DROP INDEX nombre_índice`
 - **MySQL:**
`ALTER TABLE nombre_tabla DROP INDEX nombre_índice`

SENTENCIAS DE CONTROL DE DATOS (DCL)

- Añadir permisos:

```
GRANT privilegio1,... [(nombre_columna,...)] ON  
nombre_objeto TO {nombre_usuario| PUBLIC |  
nombre_rol}
```

- Eliminar permisos:

```
REVOKE privilegio1,... [(nombre_columna,...)] ON  
nombre_objeto FROM {nombre_usuario/s| PUBLIC |  
nombre_rol/es}
```


SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- Insertar datos en una tabla:

1. `INSERT INTO nombre_tabla VALUES(valor1, valor2,...)`
2. `INSERT INTO nombre_tabla (nombre_columna1, nombre_columna2,...) VALUES(valor1, valor2,...)`

- Borrar datos de una tabla:

`DELETE FROM nombre_tabla WHERE nombre_columna operador valor`

- Borrar todos los datos de una tabla:

1. `DELETE FROM nombre_tabla`
2. `DELETE * FROM nombre_tabla`

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- Actualizar campos de una tabla:

```
UPDATE nombre_tabla SET nombre_columna1=valor1,  
nombre_columna2=valor2,... WHERE nombre_columna  
operador valor
```

- Consultas:

- SELECT FROM: consulta el valor de determinadas columnas de una tabla.

1. SELECT nombre_columna1,... FROM nombre_tabla
2. SELECT * FROM nombre_tabla

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- DISTINCT: consulta sólo los valores de las columnas especificadas de una tabla, cuyo valor no se repite.

```
SELECT DISTINCT nombre_columna1, ... FROM  
nombre_tabla
```

- WHERE: consulta los campos de una columna que cumplan cierta condición.

```
SELECT nombre_columna1, ... FROM nombre_tabla WHERE  
nombre_columna operador valor
```

- ORDER BY: consulta los campos especificados de una tabla devolviendo el resultado ordenado.

```
SELECT nombre_columna1, ... FROM nombre_tabla ORDER  
BY nombre_columna1, ... [ASC | DESC]
```

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- UNION: une el resultado de realizar varias consultas de selección.

```
SELECT nombre_columna1,... FROM nombre_tabla1  
UNION SELECT nombre_columna1,... FROM  
nombre_tabla2
```

- INNER JOIN: devuelve las filas cuando al menos existe una coincidencia en ambas tablas.

```
SELECT nombre_columna1,... FROM nombre_tabla1  
INNER JOIN nombre_tabla2 ON  
nombre_tabla1.nombre_columna=  
nombre_tabla2.nombre_columna
```

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- LEFT JOIN: devuelve todas las filas de la tabla de la izquierda (nombre_tabla1), e incluso cuando no coincide con la tabla de la derecha (nombre_tabla2). No se mostrarán las columnas de la derecha que no coincidan con las de la izquierda.

```
SELECT nombre_columna1,... FROM nombre_tabla1 LEFT  
JOIN nombre_tabla2 ON  
nombre_tabla1.nombre_columna=nombre_tabla2.nombre_  
columna
```

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- RIGHT JOIN: devuelve todas las filas de la tabla de la derecha (nombre_tabla2), e incluso cuando no coincide con la tabla de la izquierda (nombre_tabla1). No se mostrarán las columnas de la izquierda que no coincidan con las de la derecha.

```
SELECT nombre_columna1,... FROM nombre_tabla1  
RIGHT JOIN nombre_tabla2 ON  
nombre_tabla1.nombre_columna=nombre_tabla2.nombre_  
columna
```

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- FULL JOIN: devuelve todas las filas de las dos tablas.

```
SELECT nombre_columna1,... FROM nombre_tabla1 FULL  
JOIN nombre_tabla2 ON  
nombre_tabla1.nombre_columna=  
nombre_tabla2.nombre_columna
```

- SELECT...INTO: se utiliza para crear copias de seguridad (*backup*) de las tablas.

```
SELECT nombre_columna1,... INTO nombre_tabla_copia  
[IN nombre_bd_externa] FROM nombre_tabla_original
```

SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- BETWEEN: se utilizan en la condición de las consultas de selección con la cláusula WHERE para seleccionar sólo aquellos campos que estén dentro de un rango de valores.

```
SELECT nombre_columna1,... FROM nombre_tabla WHERE  
nombre_columna BETWEEN valor1 AND valor2
```

- IN: el operador IN se utiliza en la condición de las consultas de selección con la cláusula WHERE. Este operador hace que la consulta devuelva el resultado, sólo si la columna especificada en la condición toma un valor de los múltiples valores especificados con el operador IN.

- ```
SELECT nombre_columna1,... FROM nombre_tabla WHERE
nombre_columna IN (valor1, valor2,...)
```



# SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- TOP: especifica el número de filas que debe devolver una consulta.

- **SQL Server/MS Access:**

```
SELECT TOP número [PERCENT] nombre_columna1,... FROM
nombre_tabla
```

- **MySQL:**

```
SELECT nombre_columna1,... FROM nombre_tabla LIMIT
número
```

- **Oracle:**

```
SELECT nombre_columna1,... FROM nombre_tabla WHERE
ROWNUM <= número
```

# SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- ALIAS: son nombres abreviados que se le pueden asignar a columnas o tablas para utilizarlos dentro de la misma consulta.

```
SELECT nombre_columna...FROM nombre_tabla AS
alias_tabla
```

```
SELECT nombre_columna AS alias_columna FROM
nombre_tabla
```

- LIKE: este operador se utiliza para buscar determinados patrones en las columnas especificadas en una tabla.

```
SELECT nombre_columna1,... FROM nombre_tabla WHERE
nombre_columna LIKE patrón
```

# SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

- PATRONES: se utilizan para sustituir uno o más caracteres cuando realizamos búsquedas dentro de la base de datos. Es decir, que se utilizan cuando queremos que una consulta nos devuelva sólo ciertas filas en las que el valor que toma un campo de una columna cumpla una cierta estructura.

# SENTENCIAS DE MANIPULACIÓN DE DATOS (DML)

| Patrón                                    | Descripción                                       | Ejemplo                                                                                                       |
|-------------------------------------------|---------------------------------------------------|---------------------------------------------------------------------------------------------------------------|
| %                                         | Sustituye a cero o más caracteres.                | Campos que empiecen por es:<br>'es%'.                                                                         |
| _                                         | Sustituye sólo a un carácter.                     | Campos que empiecen por "ho"<br>tengan un carácter y después "a".<br>Lo cumpliría la cadena 'hola'<br>'ho_a'. |
| [lista_caracteres]                        | Sólo un carácter de los que estén en la lista.    | '%[nl]%' lo cumplirían cadenas como 'mano' o 'palo'.                                                          |
| [^lista_caracteres] o [!lista_caracteres] | Sólo un carácter de los que no estén en la lista. | '%[^nl]%' o '%[!nl]%' lo cumplen cadenas como 'paso'.                                                         |

# UTILIZACIÓN DEL CONJUNTO RESULTADO

- En PHP existen métodos para recorrer la estructura que devuelve la ejecución de una sentencia SQL.
- Destacar los siguientes: `mysql_fetch_row()`, `mysql_fetch_array()`, `mysql_result()` y `mysql_free_result()`.

# CERRAR CONEXIÓN DE LA BASE DE DATOS

- `mysql_free_result($result);`
- `mysql_close($conector);`