

Capítulo 2



Perl DBI



Base de Datos Relacionales

- Comenzaron a principios de los años 70
- Base de datos
 - Conjunto de tablas “rectangulares”
 - Cada fila de la tabla es un registro de alguna cosa
 - El registro contiene varias piezas de información, llamados campos
 - Ejemplo:

LASTNAME	FIRSTNAME	ID	POSTAL_CODE	AGE	SEX
Gauss	Karl	119	19107	30	M
Smith	Mark	3	T2V 3V4	53	M
Noether	Emmy	118	19107	31	F
Smith	Jeff	28	K2G 5J9	19	M
Hamilton	William	247	10139	2	M

Base de Datos Relacionales



- En el ejemplo
 - Los nombres de los campos son lastname, firstname, id, postal_code, age y sex
 - Cada línea de una tabla es llamada registro, fila o tupla
- SQL
 - Structured Query Language
 - Inventado por IBM en los 70
 - Lenguaje para escribir búsquedas y modificaciones en una base de datos relacional
 - Enorme éxito: simple de usar
 - Los sistemas de bases de datos lo soportan



Bases de Datos Relacionales

- Operaciones que se pueden hacer con una tabla:
 - SELECT
 - Busca todos los registros que cumplan cierta propiedad
 - INSERT
 - Agrega nuevos registros
 - DELETE
 - Elimina registros existentes
 - UPDATE
 - Modifica registros existentes
- Son los 4 comandos SQL más importantes
 - También se llaman sentencias o *queries*



Bases de Datos Relacionales

- Ejemplos

- Obtener el nombre de todas las personas con apellido “Smith”

```
SELECT firstname FROM people WHERE lastname = 'Smith'
```

- Borrar 'Mark Smith' de la tabla

```
DELETE FROM people WHERE id = 3
```

- Agregar un año a 'William'

```
UPDATE people SET age = age+1 WHERE id = 247
```



Bases de Datos Relacionales

- Ejemplo
 - Insertar nuevo registro

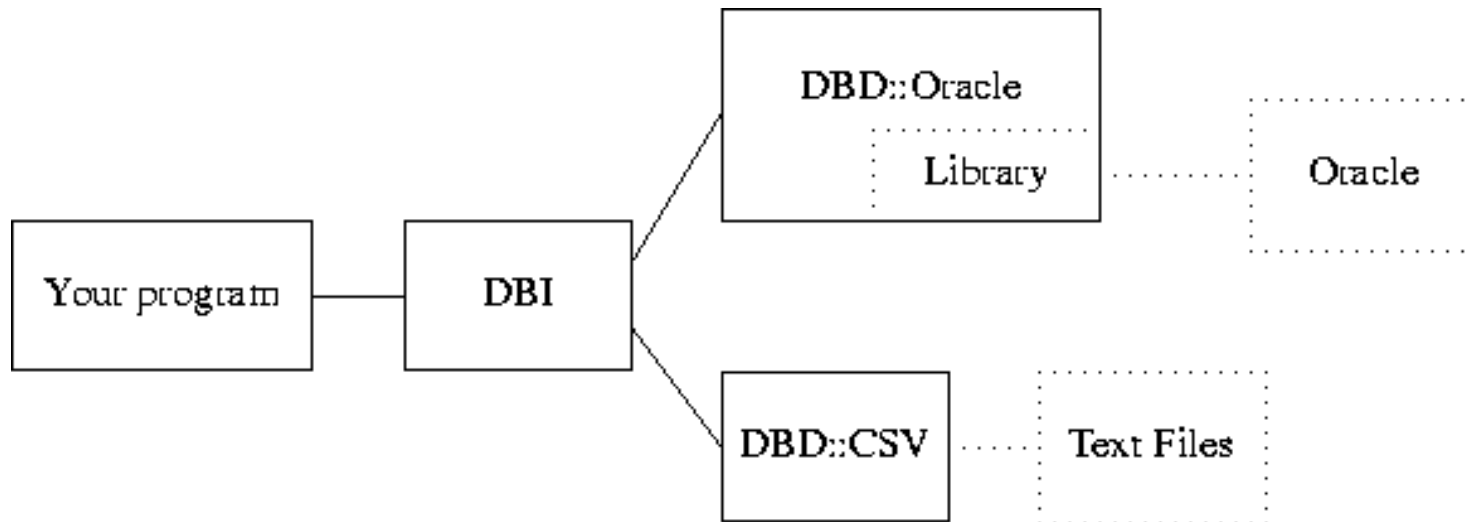
```
INSERT INTO people VALUES ('Euler', 'Leonhard', 248,  
NULL, 58, 'M')
```

- Existe un montón de otros comandos SQL
 - Crear, eliminar tablas
 - Dar, quitar permisos de acceso
 - Para confirmar o abandonar transacciones
 - Mostrar todas las tablas
 - Interactuar con tablas especiales
 - etc...



- DBI: Database Interface
 - Escrito por Tim Bunce
 - Pensado para separar los detalles de implementación de la base de datos de la aplicación
 - Interfaz simple para indicar las sentencias SQL que se desean ejecutar
 - Y como obtener los resultados de vuelta
 - DBI no sabe como comunicarse con cualquier BD
 - Pero sabe ubicar y cargar módulos DBD (database driver)
 - DBD tiene las bibliotecas de proveedores de BD
 - Hay un módulo DBD para cada proveedor
 - DBI recibe la sentencia SQL de la aplicación y la envía al módulo DBD adecuado

Perl DBI





- Usar DBI

```
use DBI;  
use strict;
```

```
$dbh = DBI->connect($dsn, $user, $password,  
                    { AutoCommit => 0 });
```

- Con autocommit se indica cuando se envían los cambios a la base de datos
 - Cuando se ejecuta la sentencia o cuando se gatilla el commit
- Módulo DBI permite preparar sentencias para una ejecución posterior



- Típica invocación de un método SELECT

```
my $sth = $dbh->prepare("SELECT foo, bar FROM table  
                        WHERE baz=?");
```

```
$sth->execute( $baz );
```

```
while ( my @row = $sth->fetchrow_array ) {  
    print "@row\n";  
}
```



- Típica invocación método no SELECT

```
my $sth = $dbh->prepare("INSERT INTO table
                        (foo,bar,baz) VALUES
                        (?, ?, ?)");

while(<CSV>) {
    chomp;
    my ($foo,$bar,$baz) = split /,/;
    $sth->execute( $foo, $bar, $baz );
}
```

- Se puede utilizar “do” para métodos no SELECT
 - Commit para ejecutar las sentencias
 - Disconnect para desconectarse de la BD

```
$rows_affected = $dbh->do("UPDATE your_table SET foo = foo + 1");
```

```
$dbh->commit; # or call $dbh->rollback; to undo changes
$dbh->disconnect;
```



- Valores NULL

```
$sth = $dbh->prepare(qq{
    INSERT INTO people (fullname, age) VALUES (?, ?)
});
$sth->execute("Joe Bloggs", undef);
```

```
$sth = $dbh->prepare(qq{
    UPDATE people SET age = ? WHERE fullname = ?
});
$sth->execute(undef, "Joe Bloggs");
```

- Excepto en cláusula WHERE

```
$sql_clause = defined $age? "age = ?" : "age IS NULL";
$sth = $dbh->prepare(qq{
    SELECT fullname FROM people WHERE $sql_clause
});
$sth->execute(defined $age ? $age : ());
```



- MySQL
 - Base de datos relacional Open Source (todavía)
 - Más popular
 - Famosa en arquitectura LAMP
 - Linux, Apache, Mysql, PHP / Perl / Python
 - Funciona en más de 20 plataformas, incluyendo
 - Linux, windows, OS/X, HP-UX, etc
 - Descarga
 - <http://dev.mysql.com/downloads/>
 - Documentación
 - <http://dev.mysql.com/doc/>