BUSTA n.1

Domande orali:

1. Descrivere le differenze sostanziali tra la versione 7 e 8 del Contente Management System Drupal.
2. Sicurezza nell'uso del linguaggio Javascript: descriva alcune differenze con l'uso di PHP.
3. Il candidato descriva le funzioni del Direttore della SISSA

Testo inglese:

There are two ways to access databases from PHP. One is to use a database-specific extension; the other is to use the database-independent PEAR DB library. There are advantages and disadvantages to each approach.

If you use a database-specific extension, your code is intimately tied to the database you're using. The MySQL extension's function names, parameters, error handling, and so on are completely different from those of the other database extensions. If you want to move your database from MySQL to PostgreSQL, it will involve significant changes to your code. The PEAR DB, on the other hand, hides the database-specific functions from you; moving between database systems can be as simple as changing one line of your program.

The portability of an abstraction layer like PEAR's DB library comes at a price. Features that are specific to a particular database (for example, finding the value of an automatically assigned unique row identifier) are unavailable. Code that uses the PEAR DB is also typically a little slower than code that uses a database-specific extension.

Keep in mind that an abstraction layer like PEAR DB does absolutely nothing when it comes to making sure your actual SQL queries are portable. If your application uses any sort of nongeneric SQL, you'll have to do significant work to convert your queries from one database to another. For large applications, you should consider writing a functional abstraction layer; that is, for each database your application needs to support, write a set of functions that perform various database actions, such as get_user_record( ), insert_user_record( ), and whatever else you need, then have a configuration option that sets the type of database to which your application is connected. This approach lets you use all the intricacies of each database you choose to support without the performance penalty and limitations of an abstraction layer.

Digitally signed by: ANTONIO LANZA
Date: 13/04/2021 09:36:15