

پایگاه داده (Database)

دیتابیس وردپرس از ۱۲ جدول تشکیل شده است که یکی از آنها دیگر کاربردی ندارد (wp_links) و بلااستفاده است. به طور کلی در پایگاه داده وردپرس نام هر جدول از دو بخش پیشوند (Prefix) و نام جدول تشکیل شده است. پیشوند جدول به صورت پیش فرض "wp_" است که می توانید آن را در فرآیند نصب وردپرس و یا از طریق کانفیگ وردپرس تغییر دهید.

کوئری (Query):

- کوئری وردپرس ابزاری برای انتقال اطلاعات با تعیین شرایط برای نمایش اطلاعات استفراخ شده از دیتابیس به کمک لوپ ها یا همان حلقه های وردپرس است. کوئری ها انواع مختلفی دارند که شامل موارد زیر است.
- کوئری ها، کلاس هایی تعریف شده در wp-includes هستند که برای بخش های مختلف وردپرس کوئری های مفتض خود را دارند.
- ✓ در صفحات هایدراکی وردپرس بصورت اتوماتیک کوئری وجود دارد و نیاز به سافت مچدر کوئری نمی باشد و تنها استفاده از حلقه کفایت می کند.
- ✓ هنگام استفاده از کوئری ها باید از حلقه (while) وردپرس جهت نمایش اطلاعات استفاده کرد.
- ✓ بعد از اتمام کوئری باید متما از متد (wp_reset_postdata()) پاکسازی کوئری استفاده کرد.

• WP_Query:

- یک کلاس جهت کوئری از پست تایپ ها می باشد و از جدول posts,users,terms,taxonomy,relationships,meta استفاده می کند.
- پارامترهای مورد استفاده شامل: author,category,tag,taxonomy,search,post & page,password,post type,status,comment,pagination,order & orderby,date, custom field,permission,mime type,caching,return field

• WP_Meta_Query:

- یک کلاس جهت کوئری از متا دیتا یا همان کاستوم فیلدها می باشد و از جدول postmeta,usermeta,termmeta استفاده می کند. (بصورت پیشفرض نمی توان استفاده کرد و بیشتر یک کوئری کمکی برای کوئری های پست تایپ ها، ترم ها، نظرات و کاربران می باشد)
- پارامترهای مورد استفاده شامل: key,value,type,compare,query,relation

• WP_Date_Query:

- یک کلاس جهت کوئری از تاریخ ها می باشد و از جدول posts,users,terms,taxonomy,relationships,meta استفاده می کند. (بصورت پیشفرض نمی توان استفاده کرد و بیشتر یک کوئری کمکی برای کوئری های پست تایپ ها، ترم ها، نظرات و کاربران می باشد)
- پارامترهای مورد استفاده شامل: key,value,type,compare,query,relation

• WP_Tax_Query:

- یک کلاس جهت کوئری از تاکسونومی ها می باشد و از جدول terms,taxonomy,relationships,meta استفاده می کند. (بصورت پیشفرض نمی توان استفاده کرد و بیشتر یک کوئری کمکی برای کوئری پست تایپ ها می باشد)
- پارامترهای مورد استفاده شامل: taxonomy,term,field,operator,children,relation

• WP_Term_Query:

- یک کلاس جهت کوئری از ترم ها می باشد و از جدول terms,termmeta,taxonomy استفاده می کند.
- پارامترهای مورد استفاده شامل: taxonomy,term,search,include & exlude,descendants,pagination,order,custom field,count,return field,cache,offset

• WP_Comment_Query:

- یک کلاس جهت کوئری از نظرات می باشد و از جدول comments,commentmeta استفاده می کند.
- پارامترهای مورد استفاده شامل: comment,post,user,search,order,pagination,custom field,count

• WP_User_Query:

- یک کلاس جهت کوئری از کاربران می باشد و از جدول users,usermeta استفاده می کند.
- پارامترهای مورد استفاده شامل: role,include & exclude,network,search,pagination,order,custom field,whocount,return

• WP_Site_Query:

- یک کلاس جهت کوئری از سایت ها می باشد.
- پارامترهای مورد استفاده شامل: site,network,domain,path,search,type,pagination,order,count,return,cache

• WP_Network_Query:

- یک کلاس جهت کوئری از شبکه می باشد.
- پارامترهای مورد استفاده شامل: network,domain,path,search,pagination,order,count,return,cache

```
$args_query = array(
    'order' => 'DESC'
);
$query = new WP_Query( $args_query );
if ( $query->have_posts() ) {
    while ( $query->have_posts() ) {
```

```

$query->the_post();
}
}
wp_reset_postdata();

```

توابع (Function):

- **wp_options** جدول
 - برای بدست آوردن اطلاعات جدول wp_options می توان از تابع `get_option` استفاده کرد همچنین برای آپدیت، افزودن و حذف کردن می توان از توابع `update_option` ، `add_option` ، `delete_option` بهره برد.
 - لیست توابع مهم: `get_option` و `wp_load_alloptions`
- **wp_users** جدول
 - برای بدست آوردن لیست کاربران با استفاده از تابع `get_users` که سبک تر و بهینه تر نسبت به کلاس `wp_user_query` می باشد. و از `wp_user_query` استفاده می کند.
 - لیست توابع مهم: `get_users` و `get_userdata` و `get_currentuserinfo` و `get_user_by` و `get_user_by_email`
- **wp_usermeta** جدول
 - برای بدست آوردن اطلاعات متای یک کاربر با استفاده از تابع `get_user_meta` استفاده می شود.
 - لیست توابع مهم: `get_user_meta` و `get_user_meta_values`
- **wp_posts** جدول
 - برای بدست آوردن لیست پست ها با استفاده از تابع `get_posts` که سبک تر و بهینه تر نسبت به کلاس `wp_query` می باشد. و از `wp_query` استفاده می کند.
 - برای بدست آوردن لیست پست ها با استفاده از تابع `get_pages` که مستقیماً از دیتابیس درخواست می کند.
 - لیست توابع مهم: `get_posts` و `get_pages` و `get_post` و `get_postdata`
- **wp_postmeta** جدول
 - برای بدست آوردن اطلاعات متای یک پست با استفاده از تابع `get_post_meta` استفاده می شود.
 - لیست توابع مهم: `get_post_meta`
- **wp_comments** جدول
 - برای بدست آوردن لیست نظرات با استفاده از تابع `get_comments` که سبک تر و بهینه تر نسبت به کلاس `wp_comment_query` می باشد. و از `wp_comment_query` استفاده می کند.
 - لیست توابع مهم: `get_comments` و `get_commentdata`
- **wp_commentmeta** جدول
 - برای بدست آوردن اطلاعات متای یک نظر با استفاده از تابع `get_comment_meta` استفاده می شود.
 - لیست توابع مهم: `get_comment_meta`
- **wp_terms** جدول
 - برای بدست آوردن لیست ترم های یک تاکسونومی با استفاده از تابع `get_terms` که سبک تر و بهینه تر نسبت به کلاس `wp_term_query` می باشد. و از `wp_term_query` استفاده می کند. و همچنین برای بدست آوردن اطلاعات یک ترم از `get_term_by` نیز می توان استفاده کرد.
 - لیست توابع مهم: `get_terms` و `get_term_by` و `wp_get_term_taxonomy_parent_id`
- **wp_termmeta** جدول
 - برای بدست آوردن اطلاعات متای یک ترم با استفاده از تابع `get_term_meta` استفاده می شود.
 - لیست توابع مهم: `get_term_meta`
- **wp_term_taxonomy** جدول
 - برای بدست آوردن لیست تاکسونومی ها می توان از ترکیب توابع `get_taxonomies` و `get_term_by` استفاده کرد.
 - لیست توابع مهم: `get_taxonomies` و `get_taxonomies_for_attachments` و `get_post_taxonomies` و `get_the_taxonomies` و `get_object_taxonomies`
- **wp_term_relationships** جدول
 - برای بدست آوردن لیست پست تایپ ها می توان از ترکیب توابع `get_post_types` و `get_term_by` استفاده کرد.
 - لیست توابع مهم: `get_post_types`
- **wp_links** جدول
 - کلا این جدول منسوخ شده است و توصیه می شود اصلاً استفاده نکنیم.

کلاس (Wpdb):

کلاس wpdb ابزاری ساده اما قدرتمند برای برقراری ارتباط مستقیم با پایگاه داده وردپرس است، از این کلاس در هر جایی می توانید استفاده کنید. wpdb امکان افزودن، بروزرسانی و حذف هرگونه اطلاعاتی را فراهم می کند، سافتار این کلاس در فایل wp-db.php دافل دایرکتوری wp-includes قراردارد. برای استفاده از کلاس wpdb ابتدا باید به آبجکت سراسری wpdb دسترسی پیدا کنید، این آبجکت شامل متدها و خصوصیات کار با پایگاه داده است.

در کلاس wpdb متدهای مختلفی برای فراخوانی و بازیابی داده ها از دیتابیس وردپرس وجود دارد:

- :get_var**

این متد یک پرس و جو(Query) به عنوان ورودی دریافت کرده و یک مقدار برگشت می دهد. به عنوان مثال تعداد پست های منتشر شده (Publish) یا تعداد کاربرانی که دارای نقش نویسنده هستند، پس در مواقعی که قرار است کوئری شما به عنوان خروجی فقط یک مقدار برگشت دهد از این متد استفاده کنید.
- :get_row**

گاهی مواقع نیاز دارید که فقط یک رکورد از پایگاه داده بازیابی کنید، به عنوان مثال نوشته ای با شناسه ۱۰ یا آخرین رکورد یک جدول، در چنین شرایطی این متد بهترین انتخاب خواهد بود.

 - پارامتر دوم این متد نوع مقدار برگشتی را مشخص می کند، مقداری که می توانید ست کنید بدین صورت است:
 - OBJECT: خروجی به صورت آبجکت (مقدار پیش فرض)
 - ARRAY_A: آرایه انهمنی (Associative Array)
 - ARRAY_N: آرایه با ایندکس عددی (Numeric Indexed Array)
- :get_col**

به کمک این متد می توانید یک ستون یا فیلد مشخصی را برگردانید، مقدار برگشتی این متد آرایه ای یک بعدی خواهد بود.
- :get_results**

این متد هیپکونه مفروودیتی در تعداد ستون ها و ردیف های خروجی اعمال نمی کند و می توانید هر نوع کوئری که نیاز دارید از چند عدد جدول متفاوت با آن پیاده سازی کنید.
- :query**

ممکن است در بعضی سناریوها آنقدر کوئری شما پیچیده باشد که هیپکدام از متدهای بالا جوابگوی کار شما نباشد، در چنین شرایطی می توان از این متد استفاده کرد. کلیه متدهایی که تا اینجا بررسی کردیم به راحتی با این متد قابل پیاده سازی هستند.
- :prepare**

برای اینکه کوئری های ما در برابر حملات SQL Injection مقاوم باشد باید از این متد استفاده کنیم. همچنین می توان با ترکیب متد query هم نوشت.
- :insert**

به کمک این متد می توانید اطلاعات مدنظرمان را به جدول اختصاصی خود اضافه کنیم، این متد سه پارامتر دارد:

 - عنوان جدول
 - آرایه ای از مقادیر
 - آرایه ای از فرمت و نوع مقادیر: %s داده رشته ای یا String - d% داده صمیح یا Integer - f% داده اعشاری یا Float
- :update**

به کمک این متد می توانید اطلاعات مدنظرمان را بروزرسانی کنیم، این متد سه پارامتر دارد:

 - عنوان جدول
 - آرایه ای از مقادیر
 - آرایه ای از شرط ها
- :delete**

این متد برای حذف اطلاعات از جدول با تعیین شرط استفاده می شود.

 - global \$wpdb;
 - \$result = \$wpdb->get_results('SELECT * FROM ' . \$wpdb->prefix . 'posts WHERE LENGTH(post_excerpt) > 0');

نکات:

- اکشن pre_get_posts: هر نوع داده ای که به هر طریقی وارد وب سایت تان می کنیم در دیتابیس وردپرس ذخیره می شود، وردپرس هم برای نمایش این اطلاعات بسته به اینکه در چه صفحه ای حضور داریم طبق سافتار هایلکی، کوئری های مختلفی را اجرا می کند. وردپرس این اکشن را قبل از فراخوانی یا Fetch اطلاعات از دیتابیس اجرا می کند و می توان کوئری پیشفرض را ویرایش نمود. استفاده از Conditional Tags و متد is_main_query و متد set از لزومات می باشد.
- تفاوت بین get های جمع با get های مفرد (برای مثال get_users یا get_commentes یا get_user یا get_comment) در این است که از get های جمع برای بدست آوردن یک لیست از کاربران، پست ها، نظرات، ترم ها و غیره استفاده می شود ولی برای بدست آوردن اطلاعات فقط یک کاربر، پست، نظر، ترم و غیره get های مفرد به کار می رود.
- تفاوت اصلی مابین کوئری کلاس ها با متدهای get در این موارد می باشد:
 - در تابع get_posts اگر پارامتر post_type ست نشود به طور پیش فرض از پست تایپ post استفاده خواهد شد، اما در کلاس WP_Query هتما باید post_type مشخص شود.
 - به جای حلقه while از foreach استفاده کردیم.
 - این امکان وجود دارد که در حلقه foreach به جای فراخوانی توابعی مثل the_content، get_the_ID و ... مستقیما از نام فیلدهای جدول wp_posts استفاده کنید.
 - مثلا \$post->post_content\$, post->ID
 - در حلقه foreach برخی توابع مثل the_content در دسترس نیستند، از این جهت با فراخوانی setup_postdata می توانید به اینگونه توابع دسترسی داشته باشید.

○ و در آشر باید از تابع wp_reset_postdata استفاده کنیم.

- شباهت اصلی مابین کوئری کلاس ها با مترهای get در این موارد می باشد که هر دو اطلاعات پردازش اصلی و برون اطلاعات متا را نمایش می دهند و برای بدست آوردن این اطلاعات متا باید از توابع the_meta برای کلاس کوئری ها و get_*_meta برای فانکشن کوئری ها استفاده کرد.

