

PHP ve MYSQL

MYSQL Veritabanı

- Veritabanı,içerisinde çeşitli verileri bulunduran birbirleriyle ilişkili tablolardan oluşmaktadır. MySQL veritabanı da SQL ve Access gibi farklı tablolardan ve bu tablo içerisindeki alanlardan meydana gelir.
- MySQL veritabanında sayısal bilgiler INT,string bilgiler VARCHAR türünde tanımlanır.

MySQL Kurulumu

- MySQL kurulumunun en son versiyonu www.mysql.com adresinden indirelebilir.Kurulum dosyasını çalıştırdıktan sonra ilk ekranda”Next”butonuna tıklayarak devam edilir.Kurulum tipi belirlenip “Install “butonuna basılır.Next butonlarıyla ilerlenip kurulum tamamlanır.
- MySQL sunucunun veritabanına erişimi için yapılandırılması gerekir. Standart ve detaylı olmak üzere iki şekilde yapılır. Yapılandırma işlemleri “Execute”butonuna basılarak aktif hale getirilir.
- “Finish”butonuna basılarak aktif hale getirilir.

PHPMyAdmin Kurulumu

- PHPMyAdmin,MySQL veri tabanına kolayca eriřimi saęlayarak veri tabanı işlemlerinin görsel yolla yapılabilmesi hazırlanmış arayüzdür.
- Kurulumu gerek yoktur.Sıkıştırılmış klasör içinde bulunan dosyaları açarak web sunucusu klasörüne(apache web sunucusu için htdocs klasörü)kaydetmek yeterlidir.

KURULUM AŞAMALARI

- Gerekli dosyalar www.phpmyadmin.net adresinden indirilebilir.
- İndirilen dosya açılarak htdocs klasörü altına kopyalanır.
- PhpMyAdmin yapılandırılması için aşağıdaki adımlar uygulanır:
- PhpMyAdmin klasörü içinde bulunan config.inc dosyasını açılarak;
- `$cfg ['PmaAbsoluteUri']='';` satırı
- `$cfg ['PmaAbsoluteUri']='http://localhost/phpmyadmin/';`
- olarak değiştirilir.

- `$cfg ['Servers'][$i]['User']='root';` satırında kullanıcı adının root olması gerekir.
- `$cfg ['Servers'][$i]['password']="";` root kullanıcı adı ile bağlantı yapılıyorsa şifre kısmının boş olması gerekir.
- Config.inc dosyası kaydedilir.
- İnternet tarayıcının adres çubuğuna <http://localhost/phpmyadmin/> yazdığınızda hata ile karşılaşılmıyorsa phpMyAdmin yazılımı başarıyla kurulmuş demektir.

MySQL Yazılım Kuralları

- **a) Veritabanı Oluşturma**
- MySQL'de veritabanı oluşturmak için **CREATE DATABASE** komutu verilir.
- **Kullanım Şekli;**
- `CREATE DATABASE veritabanı_adi;`
- **Örnek:** Aşağıdaki komutla okul adında bir veritabanı oluşturulmuştur.
- `CREATE DATABASE ogrenci;`

● b) Veritabanı Silme

- MySQL'de veritabanı silmek için **DROP DATABASE** komutu kullanılır. Komut çalıştırıldığında silinen veritabanının içerisindeki tüm tablolar ve kayıtlarda silinmiş olacaktır. Silinen kayıtlar geri getirilemez.
- **Kullanım Şekli;**
- DROP DATABASE veritabanı_adi;
- **Örnek:**Aşağıdaki komutla okul veritabanı silinecektir.
- DROP DATABASE okul;

c)MySQL Veri Türleri

Veri Türü	Açıklama
TINYINT	-128 ile 127 arasında tamsayı
SMALLINT	-32768 ile 32768 arasında tamsayı
MEDIUM	-8388608 ile 8388607 arasında tamsayı
INT	-2147483648 ile -2147483647 arasında tamsayı
BIGINT	-9223372036854775808-9223372036854775807 arasında
FLOAT	Virgülden sonra 4 yada 8 haneli ondalık sayı
TIMESTAMP	O andaki sistem zamanını tutar
TIME	HH:MM:SS,HHMMSS,HHMM,HH formatında zaman
DATETIME	YYYY-MM-DD HH:MM:SS formatında zaman
VARCHAR(n)	n sayısını geçmemek şartıyla değişen boyutta karakter
CHAR(n)	Kesinlikle n sayısı kadar karakter

d)Tablo Oluřturma

- Tablo oluřturmak için **CREATE TABLE** komutu kullanılır.Bu komut kullanılmadan önce tablonun oluřturulacađı veritabanı seřilmelidir.

- **Kullanım Őekli;**

Create Table <tablo adı>

alan_adı veri_türü [varsayılan][kısıtlama],

...

...

[tablo kısıtlaması]);

“tablo Adı” tabloya verilecek isim(örnek:ogrenci).

“alan Adı” tabloda yer alacak alanların adı(örnek:ogr_no,adi,soyadi).

“veri Türü”girilecek veri türü (sayı,metin,tarih).

“varsayılan”deđer girilmezse bařlangıçta atanacak deđer.

- “kısıtlama” boş olmaması, sadece E veya K girilmesi kısıtlamalar.
- “tablo kısıtlaması” her kaydın belirli alanlara göre kısıtlanması (kayıtların tek olmasını sağlamak)

Örnek: Mezun olan öğrenciler için mezun bilgilerinin tutulacağı bir tablo oluşturalım.

Bu tablo için adi, soyadi ve ogrno alan adları kullanılacaktır.

```
CREATE TABLE ;kayit ad varchar(15) not null,  
                soyad varchar(15) not null,  
                ogrno smallint(5) unsigned,  
                constraint ogrno_pk primary key(ogrno));
```


e)Tablo Silme

- DROP TABLE ifadesiyle istenilen tablo silinebilir.
- **Kullanım Şekli;**
- DROP TABLE tablo_adi;
- **Örnek:**Müşteri tablosunu silmek için aşağıdaki ifade kullanılır.
- DROP TABLE musterisi;

f)Tabloya Kayıt Ekleme

- MySQL veritabanlarına yeni kayıt eklemek için **INSERT** komutu kullanılır.
- **Kullanım Şekli;**
- `INSERT INTO TABLO (alan1,alan2,.....,alan_n)`
- `VALUES(deger1,deger2,.....,deger_n)`

- **Örnek:** Aşağıdaki komutla okul veritabanında bulunan öğrenci tablosuna yeni kayıt eklenmiştir.
- `INSERT INTO ogrenci(no,ad,soyad,veli)`
- `VALUES (125,"Melda","Selda","Eda")`

● g)Tablodan Kayıt Alma

- MySQL veritabanlarında istenilen şartlara uyan bilgileri almak için **SELECT** komutu kullanılır.SELECT komutunun kullanımının birçok farklı seçeneği vardır .
- **Örnek;** Aşağıdaki kod belirli bir şarta bağlı olmaksızın istenilen verileri ogrenci tablosundan okumak için kullanılır.
- `SELECT*FROM ogrenci-> ogrenci tablosundaki tüm kayıtları gösterir.`
- `SELECT ad,no FROM ogrenci->ogrenci tablosundaki verilerin sadece ad ve no bilgilerini gösterir.`

- Belirli bir şarta uyan verilerin okunması için **WHERE** parametresi kullanılır .
- **Örnek:** Numarası 15 olan öğrencilerin tüm bilgilerinin görüntülenmesi için aşağıdaki ifade yazılır.
- `SELECT *FROM ogrenci WHERE no=15;`
- **Örnek:**Sınav notundan 50-70 arası not alan öğrencilerin ad ve numarasının görüntülenmesi için aşağıdaki ifade yazılır.
- `SELECT ad,no FROM ogrenci WHERE not_1 BETWEEN 50-70`
- **Örnek:** Adının içinde 'M'geçen öğrencilerin ad ve numarasının görüntülenmesi için aşağıdaki ifade yazılır.
- `SELECT ad,no FROM ogrenci WHERE ad LIKE '%M%'`

- Select ile gelen kayıtların sıralamasını deęiřtirmek için **ORDER BY** ifadesi kullanılır.**ORDER BY** ifadesi ile birlikte **ASC** kullanırsa artan sıralama,**DESC** kullanılırsa azalan sıralama yapar.Herhangi bir ifade yazılmazsa artan sıralama yapar
- **Örnek;** Adı”M” ile Bařlayan öęrencilerin bilgilerini soyadına göre artan sıralamada görüntülenmesi için ařaęıdaki ifade yazılır.(a.....z sıralama)
- `SELECT*FROM ogrenci WHERE adi='M%' ORDER BY soyadi`

h)Kayıt Silme

- Tablodan kayıt silmek için **DELETE** ifadesi kullanılır.
- **Kullanım Şekli;**
- DELETE FROM tablo_adi WHERE koşul
- **Örnek;**Aşağıdaki SQL komutu ogrenci tablosunda bolum bilgisi”bilisim”olan tüm kayıtları silecektir.
- DELETE *FROM ogrenci WHERE bolum='bilisim'

i)Kayıt Güncelleme

- Veritabanında bulunan kayıtlar üzerinde deęişiklik yapmak istendięinde UPDATE ifadesi kullanılır.Çoęunlukla where ifadesi ile kullanılır.Where ifadesi kullanılmaz ise tablodaki tüm kayıtlar güncellenecektir.
- **Kullanım Şekli;**
- UPDATE tablo_adi SET alan1=deęer1,alan2=deęer2,.....
- WHERE Koşul cümlesi ;
-
- **Örnek;** Öğrenci numarası 35 olan öğrencinin doğum tarihini 05/02/1991 olarak deęiştirecek SQL komutu aşıęıdaki gibidir.
- UPDATE ogrenci SET dogum_tarihi='1991/05/02' WHERE no=35

MySQL Fonksiyonları

- **a) Standart Fonksiyonlar**
- **AVG():** Parametre olarak verilen alanın aritmetik ortalamasını bulur.
- **SUM():** Parametre olarak verilen alanın toplamını alır.
- **MIN():** Parametre olarak verilen alanın en küçük değerini bulur.
- **MAX():** Parametre olarak verilen alanın en büyük değerini bulur.
- **COUNT():** Parametre olarak verilen alanın kayıt sayısını bulur.
- **ROUND():** Parametre olarak verilen alanın değerinin virgülden sonraki kısmını yuvarlar.

• b) Karşılaştırma Operatörleri ve Fonksiyonlar

• Eşittir “=”

• SELECT 1 =0; SONUÇ=0

• SELECT '0' =0; SONUÇ=1

• Eşit Değildir”<>” ,”!=“

• SELECT'.01'<> '0.01';SONUÇ=1

• Küçüktür veya Küçük eşittir “<” , “<=“

• SELECT 2<2; SONUÇ=1 ??

Büyüktür veya büyük eşittir “>” , “>=“

SELECT 2>=2;SONUÇ =1

GREATEST(değer1,değer2,.....)

Parametre olarak verilen değerlerden en büyük olan değeri bulur.

SELECT GREATEST (2,0) ; sonuç =2

LEAST (değer1,değer2,.....)

Parametre olarak verilen değerlerden en küçük olan değeri bulur.

SELECT LEAST (2,0) ; sonuç =0

• c)Matematiksel Fonksiyonlar

- **ABS():** Parametre olarak verilen değerin mutlak değerini bulur.
- SELECT ABS(2); Sonuç =2
- **MOD():** Parametre olarak verilen iki değerin bölümünden kalanını verir.
- SELECT MOD(234,10); Sonuç=4
- **SQRT():** Parametre olarak verilen değerin karekökünü verir.
- SELECT SQRT (4) ; Sonuç=2

PHP VE MySQL

MySQL Sunucusuna Bağlantı

- PHP dilinde veritabanı ile işlem yapılacağı zaman ilk olarak web sunucusu aracılığıyla veritabanı dosyası ile bağlantı kurulması gerekir.
- Bağlantı bir değişkene atanır ve MySQL sunucu kurulumunda şifre konulmamışsa şifre parametresi boş bırakılmalıdır.
- **Kullanım Şekli;**
- `$baglanti=mysql_connect(sunucu,kullanici_Adi,sifre;yeni_baglanti);`
- “sunucu” bağlantı kurulacak MySQL sunucusunun ismidir.
- **Örnek;** localhost sunucusunda root kullanıcı adı ve root şifresiyle bağlantı için aşağıdaki ifade kullanılır.
- `$baglanti=mysql_connect(“localhost” ,”root” ,”root”)`
- *Yerel sunucuda sunucu localhost ismini alır.*

Veritabanı Seçimi

- Sunucu ile bağlantı kurulduktan sonra kullanılacak veritabanının seçilmesi gerekir. Veritabanı seçim için **mysql_select_db()** Fonksiyonu kullanılır.
- Fonksiyon başarı ile çalışırsa TRUE, çalışmazsa FALSE değeri döndürür.
- **Kullanım Şekli;**
- `mysql_select_db("veritabanı","bağlantı_değişkeni")`
- ("veritabanı")=Bağlantı kurulacak veritabanı
- ("bağlantı_değişkeni")= mysql_connect fonksiyonunun atandığı bağlantı değişkenidir.

PHP ile MySQL Veritabanı Oluşturma

- Mysql istemci yazılımı veya mysqladmin kullanmadan MySQL veritabanı oluşturmak için `mysql_create_db()` fonksiyonu kullanılır .
- **Örnek;** MySQL sunucu üzerinde personel isminde bir veritabanı oluşturmak için aşağıdaki ifade kullanılabilir.
- **<?PHP**
- `$bg=@mysql_connect("localhost","root","mysql")`
- `or die ("MySQL e Bağlanılamıyor!");`
- `mysql_query("CREATE DATABASE personel",$bg)`
- `or die ("Veritabanı oluşturulamadı".mysql_error());`
- **?>**

SQL İfadelerinin Kullanımı

- Bağlantı kurulan veritabanı üzerine SQL ifadelerini kullanmak için **mysql_query** ifadesi kullanılır.
- **Kullanım Şekli;**
- `Sonuc=mysql_query("sql_ifadesi","bağlantı_değişkeni");`
- `sql_ifadesi`=Çalıştırılmak istenen SQL ifadesidir.
- `bağlantı_değişkeni`=mysql_connect fonksiyonunun atandığı bağlantı değişkenidir.

MySQL Sunucu İşlemleri

- Sunucu üzerinde çalıştırılan UPDATE,INSERT ve DELETE işlemleri geriye değer döndürmez.Sadece SELECT ifadelerinde geriye sonuç döner.Bu sonuçlar üzerinde işlem yapmak için aşağıdaki fonksiyonlar kullanılır.
- **a) mysql_num_field(“sorgu_değişkeni”)**
- Bu komut SQL ifadesinde seçilen alanların sayısını bulmak için kullanılır.
- **b) mysql_num_rows (“sorgu_değişkeni”)**
- SELECT sorgusunun çalışması sonucu elde edilen kayıt sayısını bulmak için kullanılır.INSERT,UPDATE veya DELETE gibi tablo üzerinde değişiklik yapan SQL ifadeleri için **mysql_affected_rows()**fonksiyonu kullanılır.
- **c)mysql_fetch_row(“sorgu_değişkeni”)**
- SELECT ifadesi sonucu dönen kayıtları almak için kullanılır.Sorgudan dönen alanlar mysql_fetch_row()fonksiyonun atandığı değişkene dizi şeklinde atanacaktır.Daha fazla döndürecek kayıt kalmadığında “false” değeri döner

- **d) mysql_result(“sorgu_değişkeni”,kayıt”,”alan”)**
- Sorgu sonucu dönen değerlerden istenilen kaydın alan bilgisi alınabilir .
- “kayıt” parametresi sayısal bir değerdir.”alan” parametresi sayısal olabileceği gibi alan ismide olabilir.Örneğin,sorgudan dönen değerlerden 5.satırdaki kaydın ad bilgisi için mysql_result(\$sorgu,5,”ad”)veya mysql_result(\$sorgu,5,50)
- **d) mysql_error() ve mysql_errno()**
- MySQL ifadelerinin çalışması esnasında oluşan hata mesajlarını yazılı olarak almak için **mysql_error();**hata kodunu almak için **mysql_errno ()** fonksiyonları kullanılır.Parametre olarak bağlantı değişkeninin adını alır.

Sunucu Bağlantısını Kapatma

- Veritabanına bağlantı sayısı sınırlı olabileceği için veritabanıyla ilgili işlemler bittiğinde bağlantının kapatılması gerekir .Aksi takdirde diğer kullanıcılar bağlanmak istediğinde sorun oluşabilir.Bağlantı kapatmak için mysql_close fonksiyonu kullanılır
- **Kullanım Şekli;**
- Mysql_close(baglanti_degiskeni);
- **Örnek :** Aşağıdaki kullanım \$baglanti değişkenine atanan MySQL bağlantısını kapatmaktadır.
- Mysql_close(\$connect)