MySQL Egitim Serisi 2

select (4+1)\*5; (25)

select sin(pi()/4); (0.7071067811865475)

select (4+1)\*5 as toplam; (toplam (kolon adi olur) 25)

select “ali” as string; (string (kolon adi) ali)

select version(); (version() (kolon adi) 5.6.34-log) (bizim versiyonumuz 5.6.34-log imis)

select now(); (su andaki tam tarihi saati ile birlikte dondurur) (2017-06-10 17:27:17)

show databases; (MySQL sunucusunda bulunan veritabanlarini goruruz)

create database bilkav; (bilkav adinda bir veritabani olusturur)

show databases;

- Yazacagimiz kodlar olusturdugumuz veritabaniyla ilgili olsun istiyorsaniz yani onun ustunde islem yapsin istiyorsaniz once o veritabanina baglanmaniz gereklidir.

use bilkav; (bu sekilde veritabanimiza baglaniriz.

create table musteri(isim varchar(20), soyisim varchar(20), tckimlik varchar(12), dogumtarihi date)

- varchar ile char arasindaki farki biliyorsun zaten ( dynamic allocation gibi hatirla ;) )

- data saat tutmaz sadece tarih datetime ise hem tarih hem saat tutar.

desc musteri; (describe’in kisaltmasidir tablonun bilgilerini aliyoruz)

- Atiyorum baska bir veritabaninda calisiyoruz biz mesela localhost’ta ama tablomuz bilkav da bu durumda direkt olarak desc musteri; dersek hata verir cunku aktif veritabaninda yani localhost’ta boyle bir tablo yok ama biz localhost’ta mesela desc bilkav.musteri; diye calistirirsak yine ayrintilari aliriz.

insert into bilkav.musteri (isim, soyisim, tckimlik, dogumtarihi) values ("ali", "demir", "123456789123", 2016-08-08);

select \* from musteri;

insert into musteri values ("ahmet", "yildiz", "012345678912", '2016-08-09');

select \* from musteri;

insert into musteri (isim, tckimlik) values ("cem", "546798132028");

select \* from musteri;

- insert into musteri (tckimlik, isim) values ("546798132028","cem"); (boyle ters de girebiliriz) (keyfim oyle istedi yani boyle giriyorum :D )

select \* from musteri where isim = "ali";

select isim, tckimlik from musteri;

select \* from musteri where isim = "ali" or isim = "ahmet";

- where sozcugu turkceye oyleki diye gecebilir.

select \* from musteri order by soyisim;

MySQL Egitim Serisi 3 Delete and Update

update musteri set soyisim = “yildiz” where tckimlik = “546798132028”;

select \* from musteri;

delete from musteri where “546798132028”

select \* from musteri;

update musteri set soyisim = 'yıldız' where soyisim = 'yildiz';

select \* from musteri;

MySQL Egitim Serisi 4 Alter and Drop

alter table musteri add column (opsiyonel – istersen kullanma) telefon varchar(20);

desc musteri;

alter table musteri drop telefon;

drop table musteri; (tablo gider bu kodu calistirma sadece bil)

MySQL Egitim Serisi 5 Primary Key

create table calisan ( tckno int(11), isim varchar(20), departman int(3) );

create table departman ( no int(3), isim varchar(10), butcesi int(10) );

alter table calisan add primary key (tckno);

desc calisan;

alter table calisan drop primary key; (tablodaki tum primary keyleri dusurur, siler eger varsa)

insert into calisan values (888, 'can yildiz', 456);

insert into calisan values (3423432, 'ali demir', 123);

insert into calisan values (888, 'ali veli', 43434); ( duplicate veri yapip hatayi gormeye calisacagiz ;) )

select \* from calisan;

alter table calisan add primary key (tckno); (Hatamizi aldik - Duplicate entry '888' for key 'PRIMARY' )

create table deneme ( id int, isim varchar(20), primary key (id, isim) );

MySQL Egitim Serisi 6 Join Kurulmasi

- En basta tum tablolarimizi ve veritabanizi sildik.

create database deneme;

create table calisan (id int, ismi varchar(20), maas int, dept int);

- innoDB foreign key’I destekler.

- Storage engine kismindan innoDB’yi seciyoruz. Tablolari aslinda arayuzden olusturduk ama ben kodlarini yazdim hatirlamak icin sen yine arayuzden olusturabilirsin.

create table departman (id int, ismi varchar(20), butce int, yonetici int);

- primary key ile unique arasindaki fark evet ikisi de bir column da duplicate (ayni) veri olmasini engelliyor ama mesela primary key’de tablomuz veritabanimiz bu birincil anahtar yardimiyla taranacagi icin bunun ustunde indexler olusturabiliyoruz ve bir diger fark ise mesela atiyorum 2 column icin unique sectik bu durumda bu column’lar kendi icinde unique olurlar yani iki column ayri ayri incelenip atiyorum id’si 1 olan iki veri olamaz bu ilk column icin mesela bir de iste salliyorum ismi ahmet olan iki kisi olamaz diye bunlar ayridir. Ama biz bunlari primary key olarak tanimlarsak bunlar birlikte incelenir yani hem id’si hem ismi ayni olan kisi olamaz ama atiyorum ismi ahmet olan iki kisi vardir ama id’leri farklidir mesela ;)

alter table departman add primary key (id);

alter table calisan add primary key (id);

insert into calisan values (15, 'ali', 5000, 1);

insert into calisan values (16, 'veli', 5500, 2);

select \* from calisan;

insert into departman values (1, 'muhasebe', 100000, 15);

insert into departman values (2, 'bilisim', 1000000, 16);

select \* from departman;

- iste atiyorum patronunuza bir rapor mu sunacaksiniz veya atiyorum musteriye vereceginiz belge sube kodunu mu yani departman kodunu (id’si) mu gostereceksiniz departman ismini gostermeyecek misiniz? Insanlar hangi id hangi departmana ait bilmek zorunda degil ve mantiksiz da zaten iste burda devreye joinlerimiz girecek.

select \* from calisan, departman; (inner join)

- Bu sekilde iki tablodan veri istedigimizde direkt olarak bu iki tablonun kartezyen carpimi gelir.

select \* from calisan, departman where calisan.dept = departman.id;

- asil dogru tablomuz bu cunku onceki durumda atiyorum ali muhasebe de calisiyordu ama bilisim kisminda da gozukuyordu veri cok complike olmustu.

insert into calisan values (17, 'ahmet', 8989, 1);

insert into calisan values (18, 'mehmet', 4324, 1);

select \* from calisan, departman where calisan.dept = departman.id;

- yukaridaki sorguda eger departmani once yazip sonra calisan’i yazsaydik o sirada veriler gelecekti deneyebilirsin ;)

MySQL Egitim Serisi 7 Foreing key ve 2’den fazla tablo arasinda Join

create table proje (id int, isim varchar(20), departman\_id int, foreign key (departman\_id) references departman (id));

- mesela atiyorum muhasebe departmanindan bir proje olusturabilmek icin proje tablosunda yani oncelikle muhasebe departmanin olmasi gerekir muhasebe diye bir departman yoksa bundan proje olusturmayi engellemis oluruz normal ve dogru olan haliyle ;) veya iste mesela muhasebe departmanimiz var ve bu departmanda da projelerimiz var bu durumda eger biz kalkip departman tablosundan muhasebeyi silmeye kalkarsak program bizi uyarip bak senin muhasebe departmaninin ustunde projeer var bunlarin sahipsiz kalmasi soz konusu bu yuzden ben bu tabloyu silemem ;)

insert into proje values (1, 'yatirim', 1);

insert into proje values (2, 'yatirim', 3); (olmayan departmana proje atamaya calistik hatamizi aldik)

select \* from departman, proje, calisan;

select \* from departman, calisan, proje where calisan.dept = departman.id and departman.id = proje.departman\_id;

MySQL Egitim Serisi 8 Outer Join (Left and Right)

- ust seylerde anlatilan aslinda inner join’di.

select \* from calisan, departman; (bu inner joindir ve esit oldugu durumda ortaya cikar yani foreignler)

- outer join ise null olmasi durumunda ortaya cikar.

insert into calisan values (3, 'fatma', 33424, 70);

- simdi mesela biz yine yukaridaki kod gibi inner join yapsak bu kayit gelmeyecek sonuclara cunku 70 diye bir departmanimiz yok iste bu durumda outer join yapmamiz gerek ki o kaydi da sonuclar arasinda gorebilelim.

select \* from calisan, departman; (inner join yaptik ve yukariyi dogruladik fatma yok sonuclarda)

select \* from calisan, departman where calisan.dept = departman.id; (fatma yok – inner join)

insert into departman values (95, 'depo', 231, 3);

select \* from calisan, departman where calisan.dept = departman.id; (departmanimiz ve fatma yok)

- yani burada sunu anlamaliiz mesela bir calisan var ama departmani yok bu durumda biz calisan ve departman arasinda inner join kurdugumuzde bu bilgi bize gelmez ayni sekilde mesela departman var ama calisani yok ;)

select \* from calisan left join departman on calisan.dept = departman.id; (outer join – fatma var)

- Burada sadece calisan’in null degerleri geldi yani mesela yeni olusturdugumuz departmanimiz hala yok cunku biz burada left join yaptik eger right join yapsaydik bu sefer de fatma olmayacakti depo departmanimiz olacakti.

select \* from calisan right join departman on calisan.dept = departman.id;

(select \* from calisan right join departman on calisan.dept = departman.id) union (select \* from calisan left join departman on calisan.dept = departman.id); (Bunu kullanirsak ikisinin de null degerleri gelir.)