**Veritabanı Kullanırken Aklınızda Bulundurun!**

Projelerde veritabanı kullanırken -henüz proje büyümeye başlamadan- aşağıdaki önerileri dikkate alırsanız eliniz oldukça hafifleyecektir. Veritabanına not defteri muamelesi yapmayın.

1. İşin ölçeğine göre veritabanı seçilmelidir.
2. Az veri, anlaşılır isimlendirme olmalıdır.
3. Arama gerektiren veriler sonradan birleştirilecek şekilde ayrı ayrı tablolarda tutulmamalı, mümkünse aynı tabloda tutulmalıdır.
4. Unique ID kullanımı mümkünken kafanıza göre ID türevleri olmamalıdır. Olacaksa da düzenli ve sistemli olmalıdır.
5. İndex’i unutma.
6. Veritabanının ve kullandığın sürümün sana sunduğu özellikleri incele, basit işler için kodlarla çözüm arama.
7. Bir sonra ki sürümde yapılabilecekleri düşün, veritabanını ona göre düzenle ama kod kısmını sonraya bırak.
8. Foreign Key, Stored Procedure, Trigger, Function, Event gibi naneleri bil, öğren.
9. SQL Maliyeti denilen kavramı öğrenmeden kodlamaya başlama.
10. Yapacağın her şeyi ilk önce kâğıt üzerinde çalış, sonra kodlamaya geç.
11. Default değerleri NULL olarak verme.
12. Tablo isimlerini anlamlı ver. Field Name olarak tabloadi\_ veya tbl\_ gibi prefixler (ön isim) kullan. Ayrıca Camel Case de kullan. Mesela tbl\_UyeBilgileri ya da tbl\_FirmaListesi.
13. Veritabanında silme işlemi yerine aktif-pasif değerlerini kullan. Mümkünse bu sütunların özelliğini de ENUM (0/1) olarak atamalısın. Daha sonra ilişkili tablolarda sorun yaşamak istemezsin.
14. Tüm field ve tablolarında Comment özelliği mevcuttur. Bunları mutlaka kullan.
15. Her şey veritabanında tutulmaz. Çok basic bilgileri bir config dosyasında tutabiliyorsan tut. Yok, veritabanında tutacağım diyorsan da Cache kullan.
16. View özelliğini es geçme.
17. Sadece integer (sayı) kullanacağın bir alana varchar veya türevi bir şey verme. Maksimum 4 haneli bir karakter sınırın varsa bunu da belirt. Verdiğin değerleri de unutup uzun rakamlar girme, sonra neden bu sayıyı yazıyorum da o başka yazıyor deme.
18. Boolean modunda sorgular yapacaksan önce abdest al.
19. SQL sorgusu yazarken SELECT, FROM, WHERE, AND, ORDER gibi SQL in kendine has dalgalarını büyük, kendi tanımlarını küçük yaz. Mesela SELECT ad, soyad FROM mallar WHERE gibi.
20. Çok sayıda join yapılmış bir query’ye sub Query atacaksan, Unique kayıtların olduğu tabloya at.
21. SELECT \* FROM gibi kazmaca bir hareketten uzak dur. Duramıyorsan da kodlama bittiğinde mutlaka ihtiyaçların kadar field yaz. Kodlama esnasında bu seni yorabilir. Düzeltmezsen de sistemi yorabilir.
22. MySQL de hangi tablo işi ne işe yarar bunu bil. Bir kazmalık edip bak bu hızı diye memory tipini seçme. MySQL kapandığında veri meri kalmaz orda.
23. Sen sen ol, yedek al…
24. Connection açtığın gibi kapatmayı da unutma.
25. Cluster, Mirror, Master-Slave gibi kavramları öğren, sunucudan az da olsa anla. Optimizasyon yapabiliyorsan yap.
26. Dışarıdan test query’leri atacaksan “Begin Tran”, “Commit Tran”, “Rollback” gibi şeyleri ve türevlerini öğren.
27. Zihin açık değilken veritabanı düzenleme. Alkol, uykusuzluk, Extacy alıp “hadi canım sevişelim biraz” diye veritabanı düzenlenmez.
28. Eğer tek bir kayıt alacaksan “LIMIT 1” i hatırla. Her seferinde sunucuyu şişirme.
29. Veritabanı dandik bir proje gördünüz, revize falan dediler, hemen kaçın. Koda bakmadan önce buna bakın…
30. Veritabanı veri saklamak içindir. Mümkünse veritabanında sık değişmeyen verileri Cache mekanizması ile başka bir yerde tutun yada 15 de anlatıldığı gibi bir config içerisine gömün.
31. Access bir veritabanı değildir. Size bunu yutturmaya kalkanı en yakın polis karakoluna şikayet edin.
32. Localhost’da denedim oldu diye hata anında savunmaya geçmeyin, demek ki olmamış. Veritabanları 1 kişi’ye bir kaç milisaniye de cevap verirken, 1000 kişi aynı anda birşey isteyince milyon kat daha yavaşlar.
33. Her SQL sorgusunun daha az maliyetlisi olabilir. İyice kontrol edin. Bir sorguda ki değil, milyon sorguda ki maliyeti hesaplayın.
34. Eğer sürekli count yapmanız gereken birşeyler varsa, o count rakamını arttıran ne varsa, oraya bir trigger koyup, bu rakamı cache yapmanız hayat kurtarır. Follower sayısı, Cevap sayısı, Soru sayısı gibi sonuçlar için idealdir. Canlı sorgu yerine Cache sorgu sonucu hayat kurtarır.
35. Eksi değerler size lazım değilse, sayısal alanlarsa UNSIGNED kullanın ki veri saklama kapasitesi 2 katına çıksın.
36. DUPLICATE KEY UPDATE kavramını öğrenin, bu mail daha önce varmı diye yüzlerce maili tüm veritabanında tek tek aramak gibi armutluklar yapmayın.
37. MySQL için INSERT INTO tablo SET field =’deger’, field2 = ‘3’, field3=’2011-03-11′; yazabildiğinizi, rakam tarih farketmez ‘ koyabildiğinizi bilin. Sonra bu SQL’i UPDATE haline almak çocuk oyuncağı. (field,field2,field3) VALUES (‘deger’,3,’2011-03-11) dan daha hızlı yazarsınız. Performans farkı yoktur.
38. MySQL’i imkan varsa Web sunucu ile içiçe kullanmayın. Yerel ağdan erişilen bir veritabanı gibisi yoktur. Hele o ağdaki trafiği inceleyin, hatalı mısın anlamak kolay olur. Megabitlerce trafik varsa ya siteniz Google’dir ya da sizde hata vardır.
39. Slow Query Log denen zatla tanışın, ihmal etmeyin
40. phpMyAdmin’i sallayın, Navicat gibi adam gibi adam uygulamalar kullanın.
41. Hem GROUP BY hem de DISTINCT kullanmayın, az araştırın daha kolayı hep vardır. O veritabanını sen tasarlamadın mı derler adama.
42. çoklu ver girişlerinde 100 tane insert çalıştırmayın, düzgünce döngüye sokup tek insert ile işi halledin. “INSERT INTO table (field1,field2) VALUES (‘val1′,’val2’),(‘foo1′,’foo2’),(‘bar1′,’bar2’)” gibi.
43. Sakın ama sakın ORDER BY RAND() denilen embesilliği sık kullanmayın. Hatta mümkünse kodla randomize eden bir yapı düşünün.
44. Resim cisimi veritabanına gömeceğim mallığına düşmeyin, tabi masaüstü ve az kullanımlı bir işiniz yoksa.
45. genel olarak ORDER, WHERE parçalarda RAND() gibi hesaplanması gereken olaylara girmeyin. SELECT ile FROM arasında kullanabilirsiniz.
46. Proje büyükse kesinlikle Enterprise sürümlere bakın, community sürümle bir yere kadar.
47. MySQL için gerekmedikçe max\_heap\_table\_size ayarını büyük rakamlar ayarlamayın. Görürsünüz sonra anyayı konyayı.
48. Artık devir çoklu dil devri, projenizde de veritabanınızda da UTF8 kullanın.
49. UPDATE cümlesi yazarken WHERE yapmadan ve kontrol etmeden execute etmeyin. Sonra her üyenizin soyadı ‘YILMAZ’ olabilir
50. SQL lerin içerisinde hazır fonksiyonlar vardır, bunları öğrenmeden SQL tasarlamaya kalkmayın.
51. Tarih için DATE, DATETIME gibi tipler varken VarChar verip kafanızca bir haltlar etmeyin. Tüm türler için geçerlidir bu.
52. İşinize işten anlamayan kimseyi karıştırmayın, değerlendireceğim deyin bırakın.
53. Mümkün mertebe az veri ve sql sorgusu ile çok iş yapmak önemlidir. Ben fantezi yapacağım diye onlarca join, group by vs ile kasıntı sql cümleleri kurmayın.
54. MySQL 5.1 ile gelen PARTITION özelliğini es geçmeyin
55. Sadece kayıt sayısı lazım ise sqlden bütün kayıtları isteyip sonra kaç kayıt var diye saymayın. “SELECT COUNT(id) toplam FROM tablo” sorgusunu kullanın. Bunu yaparken de 34’ü hatim etmeyi unutma!
56. Kullanıcıların yetkilerini düzgün belirleyin, sadece okuması gerekene silme yetkisi vermeyin.
57. Bazı işleri SQL’den çözmeyeceksin, maliyeti artırıyor. Buna iyi karar ver. Kağıt kalem kullanmayı unutma.
58. Ne zaman join ne zaman sub query kullanılır öğren tecrübe et.
59. Sadece google, kitap, dökümana bakma işin ehli, damdan düşen birisini tanı, sormaktan çekinme.
60. Veritabanına 10 TL, 20 USD, 25 EUR gibi şeyler yazma, onlar iki ayrı fieldde tutulacak farklı şeylerdir. Birisi miktar birisi birim. Daha sonra para birimi değişirse uğraşmak istemezsin.
61. MEMORY tipi veritabanı çok hızlıdır, analiz için kullanmaktan çekinmeyin. Ama asıl tabloyu yedekleyin de kullanın.

Kaynak: http://www.burakdursun.com/details/8456/mysql-acemi-kullanici-bildirgesi.html