BMB202. Veritabanı Yönetimi

Ders 7.
MySQL Giriş
View(Görünüm), Index (İndeks), Constraints (Kısıtlayıcılar)

Erdinç Uzun
NKÜ Çorlu Mühendislik Fakültesi
Bilgisayar Mühendisliği Bölümü

Dersin Planı

- MySQL
  - MySQL Connectors
  - Tarihçe
  - my.ini / my.cnf
  - Command Line
  - Grafiksel Arayüzler
    - MySQL Workbench
    - PHPMyAdmin
  - MyISAM vs InnoDB
  - Performans İpuçları
- View (Görünüm) Kullanımı
- İndeksleme
- Kısıtlayıcılar

MySQL

- MySQL
  - çoklu iş parçacıklı (multi-threaded)
  - çok kullanıcılı (multi-user)
  - hızlı ve sağlam (robust)

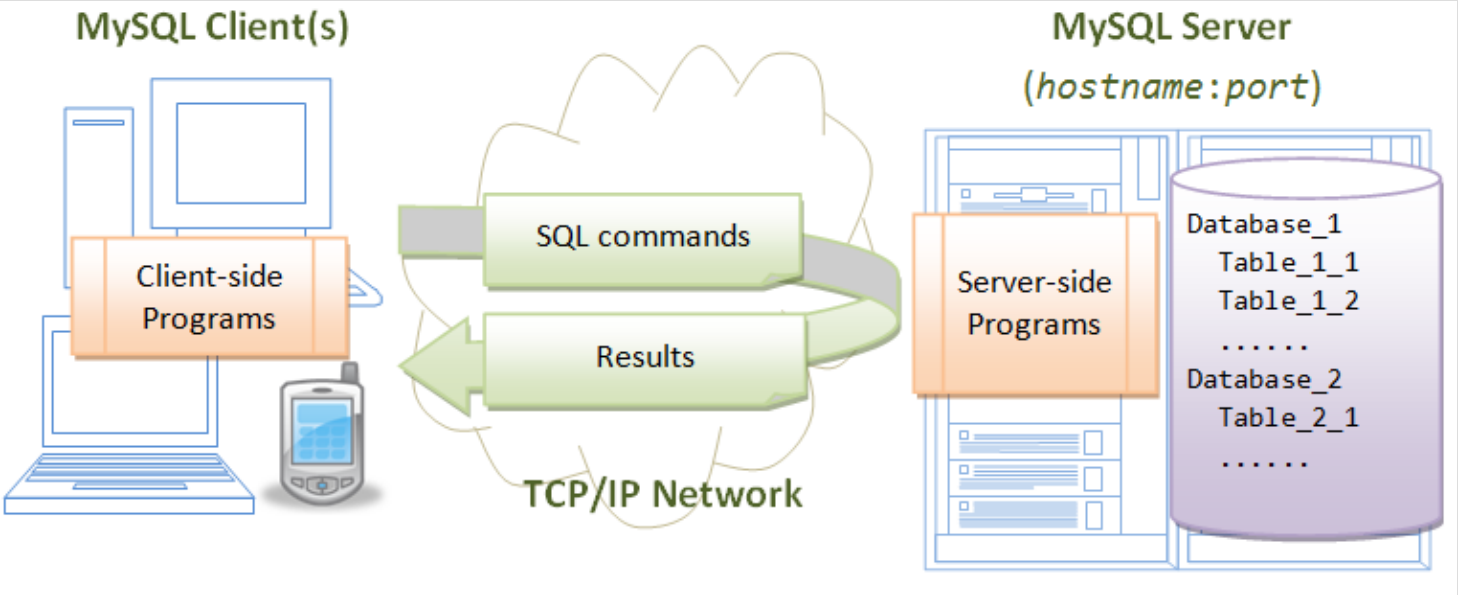
  bir veritabanı yönetim sistemidir.

MYSQL

- Multi-Platform
  - UNIX
  - OS/2
  - Windows
- OpenSource
  - Kaynak kodu açık olan MySQL'in pek çok platform için çalıştırılabilir ikilik kod halindeki indirilebilir sürümleri de mevcuttur.
- Ücretsiz (Ücretli araçları da var!)

MySQL - Programlama Dili

- Web sunucularında en çok kullanılan veritabanıdır, asp/asp.net, php gibi birçok web programlama dili ile kullanılabilir.
- Tüm programlama dilleri ile haberleşebilir.
  - ODBC sürücüleri de bulunduğu için birçok geliştirme platformunda rahatlıkla kullanılabilir.
  - Connector'larla daha hızlı bağlantı olanağı sunar.



ODBC

- ODBC(Open Database Connectivity) veritabanı yönetim sistemlerine erişmek için standart bir yazılım arayüzü sağlar.
- ODBC arayüzünden birçok VTYS'ye ve hatta excel'e programla dili tarafından bağlanmak ve işlem yaptırmak mümkündür.

MySQL Connector

- MySQL Connector, standart ODBC'ye çok daha hızlı bir veritabanı bağlantısı sağlar. Kendisine ait bir ODBC sürücüsü de vardır.
  - ODBC Driver for MySQL (Connector/ODBC)
- MySQL Connector programlama dillerine göre versiyonları vardır.
  - ADO.NET Driver for MySQL (Connector/NET)
  - JDBC Driver for MySQL (Connector/J)
  - Python Driver for MySQL (Connector/Python)
  - C++ Driver for MySQL (Connector/C++)
  - C Driver for MySQL (Connector/C)
  - C API for MySQL (mysqlclient)
- https://www.mysql.com/products/connector/

MySQL Tarihçe

- 1994, Michael Widenius “**M**onty” and David Axmark tarafından MySQL'in geliştirilmesine başlandı.
- 23 Mayıs 1995 İlk MySQL sürümü yayınlandı.
- 2001 yılında MySQL 3.23 yayınlandı.
- 2004 MySQL 4.1 beta sürümü, ve gerçek sürümü Ekim 2004 yılında yayınlandı. (R-trees and B-trees, subqueries, prepared statements)
- 2008: MySQL 5.1 sürümü yayınlandı. (event scheduler, partitioning, plugin API, row-based replication, server log tables)
- 2008 Ocak: Sun Microsystems, bir milyar dolara MySQL'i satın aldı.
- 2009 Nisan: Oracle, Sun Microsystems'i satın aldı. Oracle, MySQL'i geliştirmeye devam edeceğini duyurdu.

MySQL tercih edenler

- Wikipedia
- Google (Arama için değil!)
- Facebook
- Twitter
- Youtube
- Flickr
- ADYS :)

Kurulum

- MySQL
  - http://www.mysql.com/

- Eğer PHP,Perl, Apache gibi özellikleri de yüklemeyi düşünüyorsanız.
  - XAMPP
    - https://www.apachefriends.org
  - WampServer
    - http://www.wampserver.com/en/

MySQL Ayarları

- MySQL kurulu olduğu dizin içinde bin dizininde my.ini veya my.cnf dosyasından başlangıç ayarları yapılabilir.
- Bu dosyanın yeri bazen değişebilir. Kurulum dizininde veya Windows dizininde olabilir.

MySQL Ayarları

- Yorum yazma
  - #comment, ;comment
- [group] özel bir gruba ayarlarını başlatmada kullanılır.
  - [client]
  - [mysqld]
  - [mysqladmin]

Örnek bir MySQL ayar dosyası

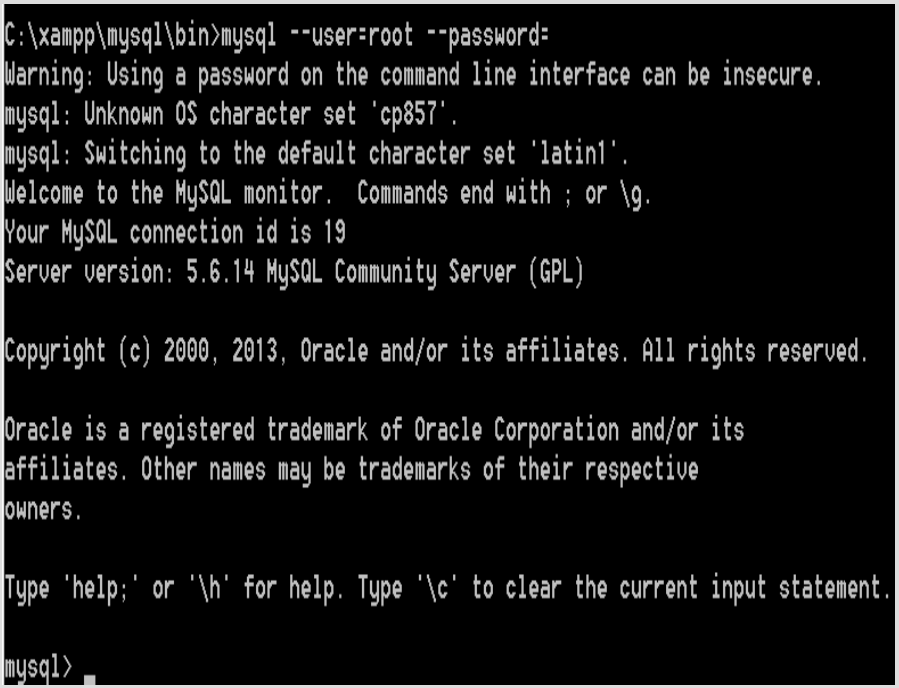
```
[client]
port=3306
socket=/tmp/mysql.sock

[mysqld]
port=3306
socket=/tmp/mysql.sock
key_buffer_size=16M
max_allowed_packet=8M
```

MySQL veritabanına bağlanma

- Mysql.exe bulunduğu dizinde command satırına
  - mysql --user=user_name --password=your_password db_name

  yazmanız yeterlidir.
- En genel kullanıcı “root”tur. Ve genelde başlangıç şifresi boştur.
- Veya kullanıcı ismi ve şifre size daha önceden verilen isim olabilir. (Kullanıcı Yönetimi konusunda anlatılacaktır.)



Veritabanını seçme ilk SQL sorgusu

- Seçim yapmak için
  - use Veritabanı İsmi

kullanılır.

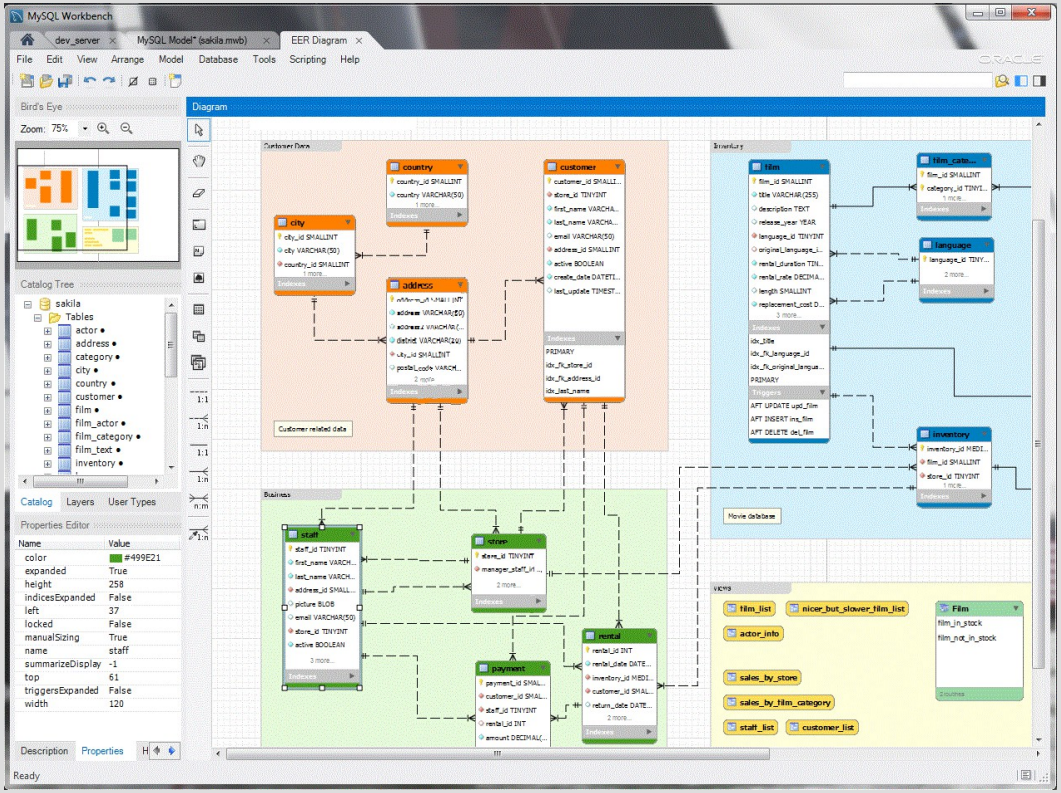
```
mysql> use adys
Database changed
mysql> select Student_Name from student limit 10;
+------------------+
| Student_Name     |
+------------------+
| Test Ö?rencisi 1 |
| Test Ö?rencisi 2 |
| Test Ö?rencisi 3 |
| BURCU BOLAT      |
| SEL?M POLAT      |
| SELÇUK SAL? EM?N |
| SERKAN GÜNDO?DU  |
| HASAN TAHA VAROL |
| BURHAN ÖZKAN     |
| HAZAL EZG? ÇAMA? |
+------------------+
10 rows in set (0.00 sec)
```

Grafiksel Arayüzler

- Command Line ekranından yönetim yapmak zordur. Bu sebepten dolayı yönetimi kolaylaştırıcı web ve windows tabanlı bir çok grafiksel arayüz geliştirilmiştir:
  - MySQL Workbench
  - Adminer
  - DBEdit
  - HeidiSQL
  - LibreOffice Base
  - Navicat
  - OpenOffice.org
  - phpMyAdmin
  - Webmin
  - SQLBuddy
  - SQLyog
  - Toad for MySQL

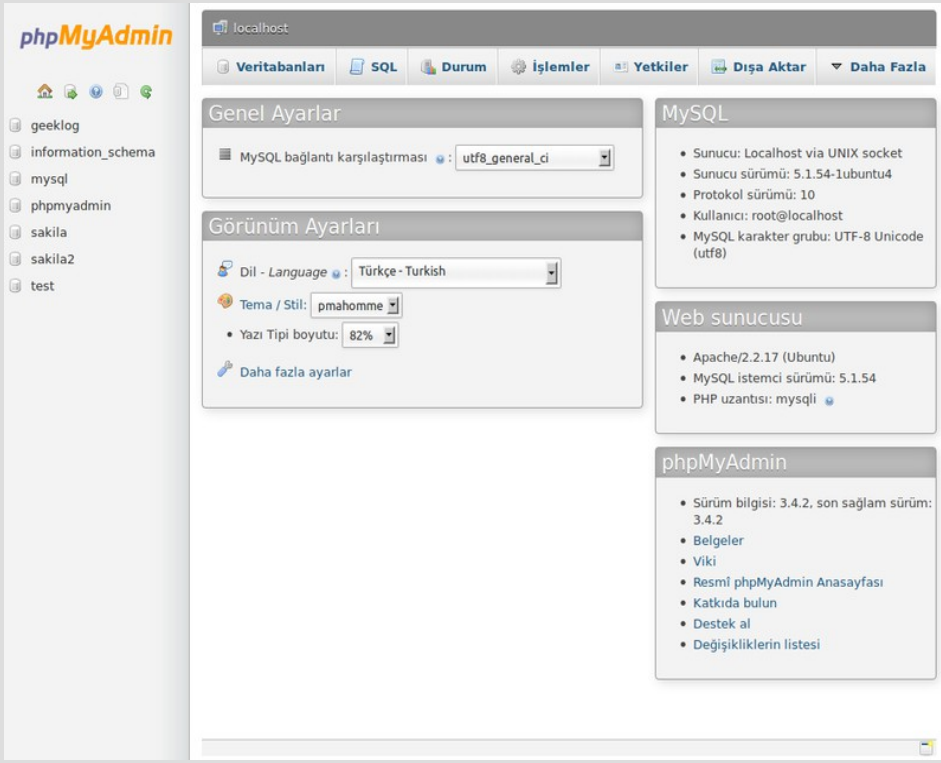
MySQL Workbench

- Bir MySQL veritabanı tasarım ve yönetim aracıdır.
  - RelationShip Model destekler.
  - Sürükle bırak özelliği
  - Reverse / Forward Engineering desteği
  - Query Browser
  - Rahat tablo tasarlama imkanı



PHPMyAdmin

- phpMyAdmin, PHP ile yazılmış açık kaynak kodlu bir araçtır.
- Başlıca kullanım amacı İnternet üzerinden MySQL veritabanı yönetimidir. Veritabanı oluşturma ve silme, tablo ekleme/değiştirme/silme, alan ekleme/değiştirme/silme, SQL sorguları çalıştırma, kullanıcıları, yetkileri ve alan anahtarlarını yönetme gibi işlevleri yapabilen bir bedava yazılımdır.
- Halihazırda 62 farklı dili destekleyen yazılım phpMyAdmin Projesi çatısı altında Olivier Müller, Marc Delisle, Alexander M. Turek, Michal Čihař ve Garvin Hicking tarafından yürütülmektedir.



Diğer VTYS sistemlerdeki benzer uygulamalar

- Çok benzer bir araç olan phpPgAdmin, PostgreSQL için benzer özellikler sunar. Başlangıçta phpMyAdmin'in bir kaynak ikizi olan bu araç şu anda tümüyle özgün bir yapıya sahiptir.
- Microsoft SQL Server yönetimi için geliştirilmiş araç ise phpMSAdmin'dir. Bu ürün, tasarım ilkeleri itibariyle phpMyAdmin'e benzese de tümüyle özgün bir kaynak koduna sahiptir.
- MySQL veritabanı yönetimi için bir araç olan phpMinAdmin, phpMyAdmin'in tüm önemli özelliklerine sahip olmakla birlikte yalnızca bir PHP dosyasından oluşur.

Reverse Engineering / Forward Engineering

- Reverse Engineering özelliği ile mevcut veritabanınızın şemasını çıkarma (InnoDB kullanıyorsanız foreign-key (yabancı anahtar) tanımlarını algılayıp diyagramda gösteriyor)
- Forward Engineering ile tasarladığınız şema ile veritabanı oluşturuyor veya yaptığınız değişiklikleri veritabanına uyguluyor. (Gerekli kontrolleri otomatik yapıp ALTER scriptini otomatik oluşturuyor)

MySQL'in desteklediği Tablo Yapıları (Table Structures)

- MySQL, iki farklı tür tablo yapısını destekler.
  - Transaction Tabloları :
    - InnoDB
    - Berkeley DB
  - Atomik İşlem Tabloları :
    - MyISAM
    - HEAP
    - MERGE
    - ISAM

Transaction Tablolar vs Atomik İşlem Tablolar

- Transaction'lı tablo ile atomik işlemle çalışan tablo arasındaki en büyük fark performans konusunda oluşur.
- Transactionlı tablolar çalışırken daha fazla bellek, daha fazla disk alanı ve daha fazla işlemci gücü harcar.

InnoDB vs MyISAM

Özellik	InnoDB	MyISAM	Açıklama
transaction	**var**	yok	sıralı olarak yaptığınız veritabanı sorgularında, işlemleri geri alma, gibi özellikler sunan bir sistem
fulltext	yok	**var**	kompleks aramalarda çok işe yarayan bir arama yöntemi
foreign key	**var**	yok	diğer tablolardan referans olarak anahtar sütunların alınması
relationship constraints	**var**	yok	diğer tablolardan referansla alınan sütunların yanlış işlemlerde kısıtlama yapması, uyarması vs.
row-level lock	**var**	yok	satırları update,delete gibi işlemlerden koruyabilme özelliği
table-level lock	**var**	**var**	tabloyu update,delete gibi işlemlerden koruyabilme özelliği
ram kullanımı	yüksek	**düşük**	
disk footprint	yüksek	**düşük**	çalışırken tablolarınızı daha etkin kullanabilmek için oluşturulan dosyaların boyutu
select,count	yavaş	**hızlı**	bazı sql komutları
insert,update	**hızlı**	yavaş	bazı sql komutları

MyISAM

- MYSQL'in standart tablo yapısıdır.
- Transaction yapısı olmadığı için çok hızlı INSERT eder fakat veri sayısı arttıkça çok yavaş SELECT yapar.
- 4GB'lık veri boyutundan sonra select sorgularında verim çok fazla düşer.
- Ayrıca metin alanlarını indexlemek için kullanılan full text index yapısını kullanmak için bu tablo yapısına ihtiyaç duyarız.
- Bu tablo yapısının en güzel tarafı da bizi LİKE komutunun hantallığından kurtarmasıdır.
- Mesela sürekli log tutmak istiyorsak da MYISAM tablo yapısına ihtiyaç duyarız.

MyISAM

- Fakat bu kadar güzel olmasının yanısıra bir çok kötü özelliğe sahiptir.
- Bunladan biride yapılan her UPDATE komutu yapının sırasını bozacaktır ve SELECT sorgularında performans düşecektir.
- MYISAM ile UPDATE yapmaktan kaçınmak gerekir.
- Eğer sürekli UPDATE yapmamız gerekiyorsa INNODB kullanmamız gerekir.
- Eğer MYISAM tablo yapısında ısrar ediyorsak tabloyu sürekli REPEAR etmemiz gerekir.
- Kötü özelliklerden bir diğeri de windows sunucuda çalışıyorsa çok güvenli olmayışıdır.
- MyISAM, text bazlı yapıya sahip olduğu için server üzerinde oluşabilecek aksaklıklardan dolayı tablo yapısında bozulma olasılığı yüksektir.

INNODB

- Kayıt girilirken MyISAM gibi bütün tabloyu kitlemezler.
- Bu tip veritabanları özel transaction fonksiyonlarınıda (basit olarak locking, begin ve commit olayları) çalıştırmanızı sağlar.
- Sürekli UPDATE işlemleri yapılacak projelerde ve veri sayısının fazla olduğu yerlerde performans artışı için kullanılır.
- Bu durumlarda BERKELEYDB tablo yapısıda kullanılabilir.
- Bu iki tablo yapısının da transaction desteği vardır.
- İyi ve performanslı tablo yapıları olduğu için bu iki tablo yapısını da ORACLE firması satın almış ve kendi bünyesinde geliştirmektedir.

MEMORY

- Bazı durumlarda verinin çok hızlı gelmesi gerekir.
- Mesela session bilgilerini veri abanında istiyorsak tablo yapısını muhakkak MEMORY tanımlamamız gerekir.
- Ancak unutmamamız gereken bir nokta daha var ;MEMEORY tablo yapısı veriyi RAM de sakladığından ,fiziksel bellekle doğrudan bağı olmadığından hızlı bir şekilde çalışır.
- Dikkat edilecek diğer bir hususta oluşturacağımız RAM miktarıdır.
- Geçerli RAM boyutu 16MB'dır.Ama RAM miktarı my.ini dosyasından ayarlanabilir.

Diğer Tablo Yapıları

- ARCHIVE
  - Adındanda anlaşılacağı gibi Arşiv niteliği taşıyan bilgilerimizi tutmak için kullanılır.
  - index desteği yoktur.
  - MYISAM ve INNODB gibi standart gelen bir tablo yapısı değildir.
- CSV
  - excel kullanımı ile ilgili bir tablo yapısıdır.
- 10'dan fazla tablo yapısı vardır. Bu bölümde en önemlileri anlatılmıştır.

MySQL Veri Tipleri

TINYINT	Çok küçük integer değerler içindir	Signed tanımlı durumda iken alabileceği değerler –128 ile 127 arasındadır.Unsigned tanımlı aralık 0 ile 255 arasındadır.
SMALLINT	Küçük integer değerler içindir	Signed tanımlı durumda iken alabileceği değerler –32768 ile 32767 arasındadır.Unsigned tanımlı aralık 0 ile 65535 arasındadır.
MEDIUMINT	Orta büyüklükteki integer değerler içindir.	Signed tanımlı durumda iken alabileceği değerler –8388608 ile 8388607 arasındadır.Unsigned tanımlı aralık 0 ile 16777215 arasındadır.
INT or INTEGER	Normal büyüklükteki integer değerler içindir.	Signed tanımlı durumda iken alabileceği değerler –2147483648 ile 2147483647 arasındadır.Unsigned tanımlı aralık 0 ile 4294967295 arasındadır.
BIGINT	Büyük integer değerler içindir.	The signed range is –9223372036854775808 to 9223372036854775807. The unsigned range is 0 to 18446744073709551615
FLOAT	Küçük (single-precision) floating-point değerler içindir.Unsigned olarak çalışmazlar.	Değer aralıkları; –3.402823466E+38 ile –1.175494351E-38, 0 arası ve 1.175494351E-38 ile 3.402823466E+38 arasındadır. Eğer sayının onluk kısmı(virgül öncesi) atanmamışsa ya da 24 basamaktan küçük veya eşitse <= 24 bu bir single-precision floating point değeridir. (single-precision / floating-point)
DOUBLE, DOUBLE PRECISION, REAL	A normal-size (double-precision) floating-point number. Cannot be unsigned	Değer aralıkları; -1.7976931348623157E+308 ile -2.2250738585072014E-308, 0 arası ve 2.2250738585072014E-308 ile 1.7976931348623157E+308 arası. Eğer sayının onluk kısmı(virgül öncesi) atanmamışsa ya da onluk basamak sayısı 25 ve 53 arasında ya da 25 veya 53'e eşitse bu bir double-precision floating point değeridir.

MySQL Veri Tipleri

DATE	Tarih	Desteklenen aralık '1000-01-01' ile '9999-12-31' arasıdır. MySQL tarihleri YYYY-AA-GG biçiminde gösterir.
DATETIME	Tarih ve zaman kombinasyonu	Desteklenen aralık '1000-01-01 00:00:00' ile '9999-12-31 23:59:59' arasıdır. MySQL DATETIME değerlerini 'YYYY-MM-DD HH:MM:SS' biçiminde gösterir.
TIMESTAMP	Zaman damgasıı	Desteklenen aralık '1970-01-01 00:00:00' ile 2037 yılında herhangi bir zaman arasındadır.MySQL TIMESTAMP değerlerini YYYYAAGGSaSaDkDkSnSn, YYAAGGSaSaDkDkSnSn, YYYYAAGG veya YYGGAA biçimlerinde gösterir.TIMESTAMP insert ve update işlemlerinde kullanışlıdır çünkü değer verilmesi bile işlemin yapıldığı andaki tarih zaman bilgisinin timestamp karşılığı otomatik olarak kaydedilir.
TIME	Zaman	Desteklenen aralık '-838:59:59' ile '838:59:59' arasındadır. MySQL TIME değerlerini 'HH:MM:SS' biçiminde gösterir.
YEAR	2 ya da 4 basamaklı yıl bilgisi (öntanımlı 4 basamaklı)	4 basamaklı yıl bilgisinde değer aralığı 1901 ile 2155 arasındadır. 2 basamaklı yıl bilgisinde değer aralığı 1970-2069 için 70 ile 69 arasındadır.MySQL YEAR değerlerini YYYY formatında gösterir.

MySQL Veri Tipleri

CHAR	A fixed-length string that is always right-padded with spaces to the specified length when stored	Maksimum 255 karakter barındırabilir. Tekrar eden boşluklar değer alındığı zaman silinir.Öntanımlı karakter seti BINARY tanımlanmamışsa CHAR değerler büyük-küçük harf duyarlılığı olmadan sıralanır veya karşılaştırılırlar.
VARCHAR	A variable-length string. Note: Trailing spaces are removed when the value is stored (this differs from the ANSI SQL specification)	Maksimum 255 karakter barındırabilir. VARCHAR değerleri BINARY anahtar sözcüğü verilmediği takdirde büyük-küçük harf duyarlılığında sıralanır veya karşılaştırılırlar.
TINYBLOB, TINYTEXT		255 karakterli bir BLOB ya da TEXT verisi
BLOB, TEXT		65535 karakterli bir BLOB ya da TEXT verisi
MEDIUMBLOB, MEDIUMTEXT		16777215 karakterli bir BLOB ya da TEXT verisi
LONGBLOB, LONGTEXT		4294967295 karakterli bir BLOB ya da TEXT verisi
ENUM	An enumeration	Değer listesinden seçilebilecek her biri maksimum 65535 karakterli string değerler tutabilir.
SET	A set	Maksimum 64 elemanlı string değerler kümesi.

Performans İpuçları
Where

- Where Bölümündeki Sıralama
  - SELECT sorgusu kullanılırken birden fazla WHERE şartı kullanılacaksa bunların sırası çok önemlidir. Mesela toplam öğrenci sayısının 300 olduğu bir tablo varsayalım. Bu tabloda Erkeklerin sayısı 150 olsun.1.sınıfta ki öğrenci sayısı ise 60 olsun. Şimdi bu öğrenci kayıtlarının arasından cinsiyeti erkek ve 1.sınıfta okuyan öğrencileri isteyelim. İlk WHERE şartında 1.sınıf öğrencilerini,ikinci WHERE şartında ise cinsiyetini istememiz daha performanslı olur. Neden derseniz ilk önce 300 kayıt içinde 60 kayıt bulacak daha sonra 60 veri içinden erkekleri seçecek. Yani(önce 300 kayıt içinden ,sonra 60 kayıt içinden seçecek). Diğer türlü 300 kayıt içinden önce erkekleri seçecek,daha sonra erkek olan 150 kayıt içinden 1.sınıfları seçecek. Yani(önce 300 kayıt içinden seçecek,sonra 150 kayıt içinden seçecek). İlk sorgu daha performanslı olur.

Performans İpuçları
Inner Join

- Inner Join: sadece eşleşen kayıtlar işinize yarar.
  - Mesela,JOIN komutu kullanılacak tabloların, tablolar arası bağlantıyı kuran sütunların veri tipleri ve veri uzunlukları birbirinden farklıysa JOIN kullanmak sorguyu çok uzatır ve performans kaybı yaşarsınız. Ne istediğinizi bilmelisiniz. Asla işe yaramayacak verileri istemeyin bilin ki en yaygın kullanılan JOIN tipi olan INNER JOIN iki tabloyu birleştirir ve sadece iki tabloda da eşleşen kayıtlar varsa getirir. Eğer sadece eşleşen kayıtlar işinize yarayacaksa INNER JOIN kullanın.

Performans İpuçları
Limit

- Belirli durumlarda tek bir özgün sonuç arıyor olabilirsiniz. Ya da sadece WHERE koşullarına uygun kayıtlar olup olmadığını kontrol ediyor olabilirsiniz. Her iki durumda da LIMIT 1 kullanmak sorgu hızını arttırabilir.
  - SELECT * FROM user WHERE sehir = 'İstanbul'
  - SELECT * FROM user WHERE sehir = 'Istanbul' LIMIT 1

Performans İpuçları
MySQL Partition

- MySQL PARTITION özelliği ne işe yarar derseniz şu şekilde açıklayalım.
- Mesela elinizde 1 milyon satır veri var diyelim ve bu verilerin yıllar bazında oluşturulma tarihleri olsun. Biz sadece 2013 yılına ait verileri istediğimizde normalde 1 milyon veri arasından 2013 yılına ait verileri getirir. Ancak MYSQL PARTITION özelliğini kullanırsak yıllara göre sıralayıp 2013 yılında 100 bin kayıt varsa sadece 100 bin kayıt arasından SELECT yaparız.
- Bu da performans ve hız açısından iyidir. Ancak bu özelliği kullanırken çok iyi bir hakimiyetinizin olması gerekir.

Performans İpuçları
PROCEDURE ANALYSE ( )

- MySQL'deki Procedure Anlayse fonksiyonu ile bu alanların bazı betimsel istatistiklerini ve önerilen türünü öğrenebilirsiniz.
  - SELECT *

    FROM student

    PROCEDURE ANALYSE ( )

Performans İpuçları
View Kullanımı

- Sıklıkla kullandığımız sorguları VIEW kullanarak yaparsak daha performanslı sonuçlar elde edebiliriz.
- View Nedir?
  - Sorguları basitleştirmek ,
  - sorgu sürelerini kısaltmak ve
  - sistemin daha performanslı çalışmasını sağlamak için kullanılan sanal tablodur.
- VIEW sanal bir tablo olarak işlem yapar,ancak bu sanal tabloyu daha önceden oluşturmamız gerekir.

Sık kullandığımız bir sorgu

- //sık kullandığımız bir sorgu
  - SELECT ogrenciler.ogrenci_adi,
    ogrenciler.ogrenci_no,
    fakulteler.fakulte_adi,
    fakulteler.fakulte_id
    FROM ogrenciler
    LEFT JOIN fakulteler
    ON fakulteler.fakulte_id=ogrenciler.fakulte_id

View oluşturma

- CREATE VIEW fakulte_ogrenci
  AS SELECT ogrenciler.ogrenci_adi,
  ogrenciler.ogrenci_no,
  fakulteler.fakulte_adi,
  fakulteler.fakulte_id
  FROM ogrenciler
  LEFT JOIN fakulteler
  ON fakulteler.fakulte_id=ogrenciler.fakulte_id

View’den bilgi çekme ve View’i silme

- View’den bilgi çekme
  - SELECT * FROM fakulte_ogrenci;
- View’i Silme
  - DROP VIEW fakulte_ogrenci;

İndeks (Index)

- Bir indeks, veri tabanı ortamında bir tablo ya da bir view gibi bir nesnedir ve ilişkili olarak kullanıldığı tablo ya da view'deki satırların, indeksleme alanı (key field (anahtar alan)) olarak kullanılan kolondaki verilere göre sıralanmış biçimde işleme sokulmasını (listeleme ya da arama işlemi) sağlar.
- Bir tablo, indekslenmiş ise, bu tablo içinde gerçekleştirilecek bir arama (search) ya da koşullu listeleme (SELECT komutu ile) işlemi çok daha hızlı biçimde gerçekleştirilebilecektir.

Niye indeksleme

- Genel olarak veritabanlarında veri miktarı arttıkça, tüm veriye ulaşımın maliyeti de benzer oranda artar. Mesela aynı tablo için 100 kayıtla çalışmak ile 1 milyon kayıtla çalışmak arasında aradığınız veriye ulaşım açısından çok buyuk maliyet farkı vardır. Bu maliyet genel olarak CPU ve disk erişimi olarak karşımıza çıkacaktır. Eğer veritabanınız başka bir makine üzerinde ise buna ilaveten bir de network kullanımı maliyeti eklenecektir.

İndekslemenin dezavantajı

- İndeksleme, sabit diskte ekstra yer tutar. Verdiği performans artışının yanında karşılaştırılabilecek kadar bile değildir ancak veritabanınızdaki her tablonun her sütununda indeks oluşturursanız, sabit diskinizin umduğunuzdan çok daha çabuk dolduğuna şahit olabilirsiniz!
- İndeksleme INSERT, UPDATE ve DELETE komutlarının çalıştırılma sürelerinde yavaşlamaya neden olur. Yine bu yavaşlamanın gözle görülür bir fark olduğu söylenemez, ama MySQL'in tablolarda INSERT, UPDATE ve DELETE ile güncelleme yaparken indeks alanlarını da uygun şekilde güncellediğini unutmamak gerekir.
  - Yani her tablonun her sütununu indekslememek gerekir!

İndeksleme kararı nasıl verilir?

- Hangi sütunda indeksleme yapacağınıza aslında sizin ihtiyaçlarınız yön veriyor. Sık sık yazdığınız sorgu cümleleri, aramalarda kullanılacak alanlar genelde indeks için uygun alanlar olurlar.
- İndeksleme için en güzel aday sorgu cümlelerinde “WHERE” kelimesinden sonra yazdığınız alanlardır. Yazdığınız sorgu cümlelerinde bu şekilde öne çıkan bir sütun varsa o sütun indeks oluşturmak akıllıca olacaktır.

İndeksleme kararı nasıl verilir?

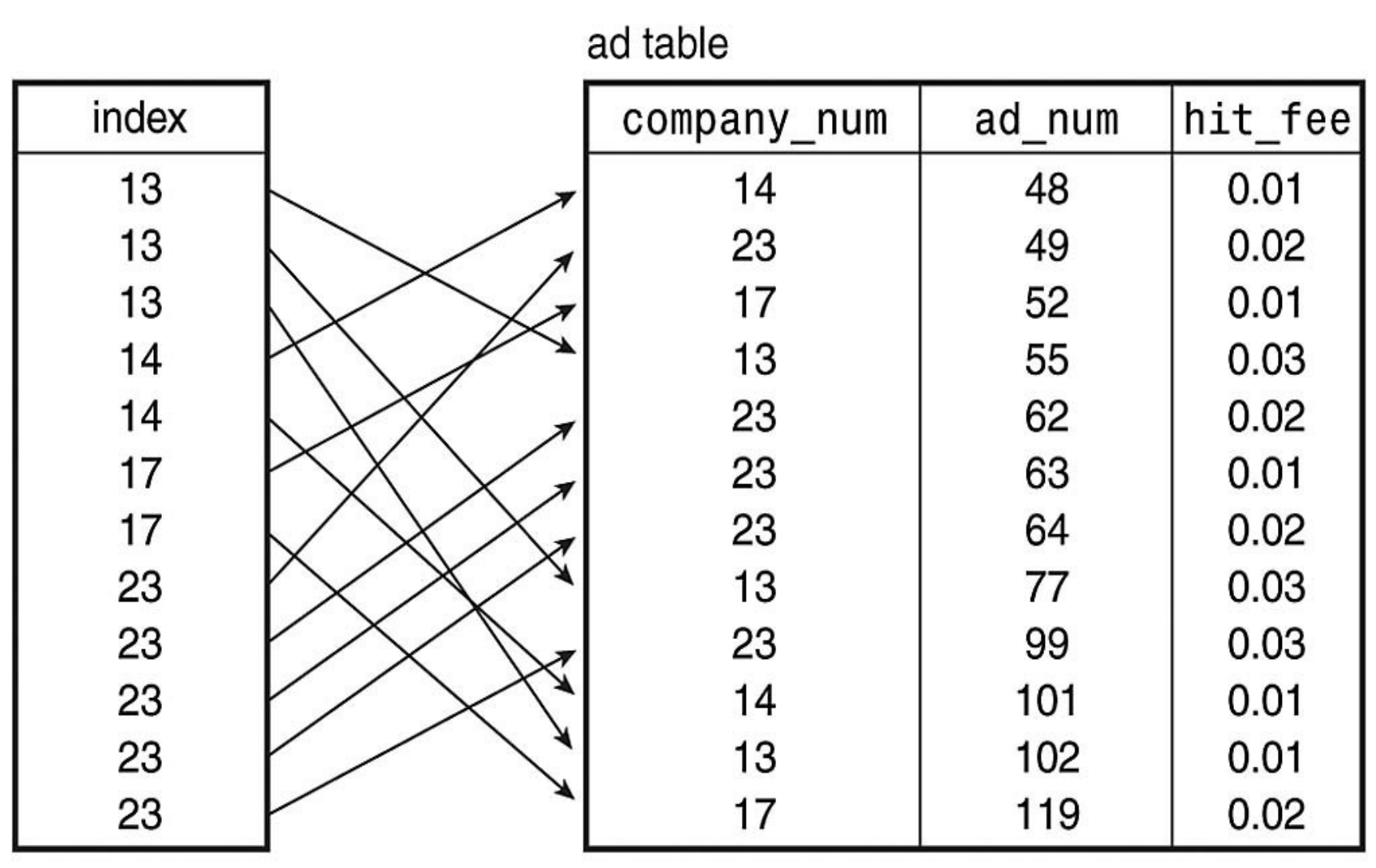
- İndeksleme için bir diğer aday sütunlar ise, içlerinde farklı değerler tutan alanlardır. Yukarıdaki tablomuzu düşünürsek, “cinsiyet” alanı indeksleme için pek de uygun sayılmaz, çünkü tutabileceği iki farklı değer vardır, “E” veya “K”. Bu alanda bir indeks oluşturulsa bile MySQL yine de toplam satırların yarısını taramak zorunda kalacaktır, bize pek performans getirisi sağlamaz. Bunun yerine hemen hemen her satırda farklı değerlerin tutulduğu sütunlar indeksleme için kullanışlı olabilirler.
- İndekslemede arama maliyeti ile DML işlemleri sonrası indeksin güncellenme maliyeti arasında kararı veritabanı yöneticisi verir.
- Veritabanı yöneticisinin deneyimi ve tecrübesi bu noktada önemli rol oynar.

İndekslenmemiş durum

ad table

company_num	ad_num	hit_fee
14	48	0.01
23	49	0.02
17	52	0.01
13	55	0.03
23	62	0.02
23	63	0.01
23	64	0.02
13	77	0.03
23	99	0.03
14	101	0.01
13	102	0.01
17	119	0.02

İndekslenmiş durum ve ek maliyet



İndeksleme Örneği

- Tablo(isim, adres, telefon)
  - SELECT * FROM tablo WHERE isim='osman'
- isim alanı çok sık şekilde sorgulanan bir alan olsun. Bu durumda bu alan indekslenebilir.
  - ALTER TABLE tablo_ismi

    ADD INDEX index_ismi (index_sütunu_ismi);
- Yukarıdaki sorgu için indeksleme
  - ALTER TABLE tablo ADD INDEX isim_indexi (isim);
  - ALTER TABLE tablo ADD INDEX isim_soyisim_yas (isim, soyisim, yas);

Kısıtlayıcı (Constraint)

- Veri üzerindeki mantıksal sınırlamalara kısıt adı verilir.
- Kısıtların genel olması tercih edilen bir durumdur.
- Kısıtlar, veri modellerinde bütünlük sağlamak için kullanılır.
- Kısıtlamalar, tabloların tanımlanmasıyla beraber oluşan öğelerdir.
- Kısıtlamalar ile Rule (kural) ve Default'ların (varsayılan) yapabileceği işler yapılabilir.
- Constraint'ler tablo oluştururken yani CREATE TABLE komutuyla tanımlanabilir. Tablo oluşturulmuşsa ALTER TABLE komutuyla bu işlem gerçekleşir. ALTER TABLE komutuyla kullanıldığında sütunlara girilen bilgilerin dikkate alınması gerekir.

Constraint (Kısıtlayıcı)Türleri

- Primary Key Constraint
- Unique Constraint
- Not Null Kısıtlayıcısı
- Foreign Key Constraint
- Default Constraint
- Check Constraint

Primary Key Kısıtlayıcı

- Birincil anahtar kısıtlayıcı anlamındadır. Aynı olmayan değerler girilmesini sağlar. Bu da her kaydın farklı olması demektir. Her tablonun en fazla 1 adet Primary Key Constraint'i olabilir.

MySQL

```
1  CREAT TABLE Ogrenciler
2  (
3  ogr_no int NOT NULL,
4  ad varchar(255) NOT NULL,
5  soyad varchar(255) NOT NULL,
6  bolum varchar(255,
7  PRIMARY KEY(ogr_no)
8  );
```

MS SQL Server,Oracle,MS Access

```
1  CREAT TABLE Ogrenciler
2  (
3  ogr_no int NOT NULL PRIMARY KEY,
4  ad varchar(255) NOT NULL,
5  soyad varchar(255) NOT NULL,
6  bolum varchar(255,
7  );
```

Primary Key kısıtlayıcısını isimlendirmek ve birden fazla sütunu Primary Key olarak tanımlamak istersek,aşağıdaki SQL cümleciğimizi yazıyoruz.

```
1  CREAT TABLE Ogrenciler
2  (
3  ogr_no int NOT NULL,
4  ad varchar(255) NOT NULL,
5  soyad varchar(255) NOT NULL,
6  bolum varchar(255,
7  CONSTRAINT pk_ogr_no PRIMARY KEY(ogr_no,ad)
8  );
```

Alter Table Cümleciğinde Primary Key Kısıtlayıcısı

Tabloyu oluşturduktan sonra bir alanı Primary Key olarak tanımlamak istersek:

```
1 | ALTER TABLE Ogrenciler
2 | ADD PRIMARY KEY(ogr_no)
```

Yine aynı şekilde,birden fazla sütunu Primary Key olarak belirtmek istediğinizde:

```
1 | ALTER TABLE Ogrenciler
2 | ADD CONSTRAINT pk_ogr_no PRIMARY KEY(ogr_no,ad)
```

Primary Key Kısıtlayıcısını Silme(İptal Etme)

MySQL

```
1 | ALTER TABLE Ogrenciler
2 | DROP PRIMARY KEY
```

MS SQL Server,Oracle ve MS Access

```
1 | ALTER TABLE Ogrenciler
2 | DROP CONSTRAINT pk_ogr_no
```

Unique Constraint

- Tekil alan kısıtlayıcı anlamındadır. Birincil anahtar olan ve tablodaki diğer alanlar içinde aynı içeriğe sahip verilerin olmaması için Unique Constraint tanımlanır.
- T.C.Kimlik Nu. primary key ve Okul Nu. Unique şeklinde bir tanımlama Unique Constraint'e bir örnektir.

Unique Kısıtlayıcı

- UNIQUE ve PRIMARY KEY kısıtlayıcılarının her ikisi de benzer kayıtların girilmesinin engellenmesini sağlar.
- Bir PRIMARY KEY kısıtlayıcısı tanımladığınız zaman otomatik olarak UNIQUE kısıtlayıcısını da tanımlamış sayılırsınız.
- Dikkat edilecek husus,her tablo için birçok UNIQUE kısıtlayıcısı olabilir fakat bir tablonun sadece bir tane PRIMARY KEY kısıtlayıcısı olabilir.

Unique Kısıtlayıcı

MySQL

```
CREAT TABLE Ogrenciler
(
ogr_no int NOT NULL,
ad varchar(255) NOT NULL,
soyad varchar(255) NOT NULL,
bolum varchar(255,
UNIQUE(ogr_no)
);
```

MS SQL Server , Oracle ve MS Access

```
CREAT TABLE Ogrenciler
(
ogr_no int NOT NULL UNIQUE,
ad varchar(255) NOT NULL,
soyad varchar(255) NOT NULL,
bolum varchar(255,
);
```

```
CREAT TABLE Ogrenciler
(
ogr_no int NOT NULL,
ad varchar(255) NOT NULL,
soyad varchar(255) NOT NULL,
bolum varchar(255,
CONSTRAINT kisitlayici UNIQUE(ogr_no,ad)
);
```

Alter Table Cümleciğinde Unique Kısıtlayıcısı

```
ALTER TABLE Ogrenciler
ADD UNIQUE (ogr_no)
```

```
ALTER TABLE Ogrenciler
ADD CONSTRAINT kisitlayici UNIQUE(ogr_no,ad)
```

Unique Kısıtlayıcısını Silme(İptal Etme)

MySQL

```
ALTER TABLE Ogrenciler
DROP INDEX kisitlayici
```

MS SQL Server, Oracle ve MS Access

```
ALTER TABLE Ogrenciler
DROP CONSTRAINT kisitlayici
```

Not Null Kısıtlayıcısı

- NOT Null kısıtlayıcı bir kolondaki verilerin boş olmamasını sağlar.

NOT NULL ÖRNEK

```
1  CREATE TABLE Ogrenciler
2  (
3  ogr_no int NOT NULL,
4  ad varchar(255) NOT NULL,
5  soyad varchar(255) NOT NULL,
6  bolum varchar(255)
7  );
```

Foreign Key(Yabancı Anahtar) Kısıtlayıcısı

- Yabancıl anahtar kısıtlayıcı anlamındadır. Bir tablodaki bir sütuna ait verilerin başka bir tablonun belirli bir sütunundan gelmesini denetler.

kisiler tablosu:

id	ad	soyad	adres	il_kodu
1	Süleyman	Akgül	Yeni Mahalle, Şirin Sokak ,Saray Önü Apt No:13/4	23
2	Harun	Kolyiğit	Yeni Mahalle Şirin, Sokak Pınar ,Apt No:11/2	23
3	Eser	Gürbüz	Cingen Mah., Kimsesizler Sok. ,Huzur Apt. Bodrum Kat	42
4	Sinan	Tunç	Yeşil Mahalle,Tuna Sok.,Beyzade Apt.N0:11/1	70

iller tablosu:

il_kodu	il_adi
1	Adana
23	Elazığ
34	İstanbul
42	Konya
70	Karaman

Foreign Key(Yabancı Anahtar) Kısıtlayıcısı

MySQL

```
1  CREATE TABLE kisiler
2  (
3  id int NOT NULL ,
4  ad varchar(255)NOT NULL,
5  soyad varchar(255)NOT NULL,
6  adres varchar(255)NOT NULL,
7  il_kodu int NOT NULL,
8  PRIMARY KEY(id),
9  FOREIGN KEY(il_kodu)REFERENCES iller(il_kodu)
10 )
```

SQL Server,Oracle ve MS Access

```
1  CREATE TABLE kisiler
2  (
3  id int NOT NULL PRIMARY KEY,
4  ad varchar(255)NOT NULL,
5  soyad varchar(255)NOT NULL,
6  adres varchar(255)NOT NULL,
7  sehir int FOREIGN KEY REFERENCES iller(il_kodu)
8  )
```

Birden fazla sütunu FOREIGN KEY olarak tanımlamak ve yabancı anahtara bir isim vermek istersek aşağıdaki SQL ifadesini yazabiliriz.

```
1  CREATE TABLE kisiler
2  (
3  id int NOT NULL,
4  ad varchar(255)NOT NULL,
5  soyad varchar(255)NOT NULL,
6  adres varchar(255),
7  il_kodu int,
8  PRIMARY KEY (id),
9  CONSTRAINT fk_Kisiİl FOREIGN KEY (il_kodu)
10 REFERENCES iller(il_kodu)
11 )
```

Foreign Key(Yabancı Anahtar) Kısıtlayıcısı

Alter Table Cümleciğinde Foreign Key Kısıtlayıcısı

```
1  ALTER TABLE kisiler
2  ADD FOREIGN KEY (il_kodu)
3  REFERENCES Persons(il_kodu)
```

```
1  ALTER TABLE kisiler
2  ADD CONSTRAINT fk_kisi_il
3  FOREIGN KEY (il_kodu)
4  REFERENCES iller(il_kodu)
```

Foreign Key Kısıtlayıcısını Silme(İptal Etme)

MySQL

```
1  ALTER TABLE kisiler
2  DROP FOREIGN KEY fk_kisi_il
```

MS SQL Server,Oracle ve MS Access

```
1  ALTER TABLE kisiler
2  DROP CONSTRAINT fk_kisi_il
```

Check Kısıtlayıcısı

- Kontrol kısıtlayıcı anlamındadır. Belirtilen formata göre verilerin girilmesini sağlar. Örneğin, T.C.Kimlik Nu. alanına 11 karakterin girilmesi Check Constraint ile sağlanabilir.
- Server/Client veya Web tabanlı uygulamalarda bu kontrolü client tarafında yapılması tavsiye edilir. (Javascript ve JQuery)

Check Kısıtlayıcısı

Create Table Cümleciğinde Check Kısıtlayıcısı

MySQL

```
CREATE TABLE kisiler
(
id int NOT NULL ,
ad varchar(255)NOT NULL,
soyad varchar(255)NOT NULL,
adres varchar(255),
il_kodu int NOT NULL,
CHECK (il_kodu>0)
)
```

MS SQL Server,Oracle ve MS Access

```
CREATE TABLE kisiler
(
id int NOT NULL ,
ad varchar(255)NOT NULL,
soyad varchar(255)NOT NULL,
adres varchar(255),
il_kodu int NOT NULL CHECK(il_kodu>0)
)
```

Birden fazla CHECK kısıtlayıcısı tanımlamak ve isimlendirmek için aşağıdaki SQL cümleciğini deneyebilirsiniz.

```
CREATE TABLE kisiler
(
id int NOT NULL ,
ad varchar(255)NOT NULL,
soyad varchar(255)NOT NULL,
adres varchar(255),
il_kodu int NOT NULL ,
CONSTRAINT chk_kisi CHECK(il_kodu>0 AND soyad LIKE 'A%')
)
```

Check Kısıtlayıcısı

Alter Table Cümleciğinde Check Kısıtlayıcısı

```
1 | ALTER TABLE kisiler
2 | ADD CHECK (il_kodu>0)
```

Eğer birden fazla CHECK kısıtlayıcı ekleyeceksiniz;

```
1 | ALTER TABLE kisiler
2 | ADD CONSTRAINT chk_kisi CHECK (il_kodu>0 AND soyad LIKE 'A%')
```

Check Kısıtlayıcısını Silme(İptal Etme)

MySQL

```
1 | ALTER TABLE kisiler
2 | DROP CHECK chk_kisi
```

MS SQL Server,Oracle ve MS Access

```
1 | ALTER TABLE kisiler
2 | DROP CONSTRAINT chk_kisi
```

Default Kısıtlayıcısı

- Varsayılan kısıtlayıcı anlamındadır. Tablodaki herhangi bir alan için girilmesi gereken bir değerin atanmasıdır.
- INSERT komutu için geçerlidir. Örneğin, kişi bilgilerinin alındığı bir tabloda kişinin uyruğunun girilmesi işleminde varsayılan değer olarak “T.C.” atanabilir.

Default Kısıtlayıcısı

Create Table Cümleciğinde Default Kısıtlayıcısı

```sql
CREATE TABLE kisiler
(
id int NOT NULL ,
ad varchar(255)NOT NULL,
soyad varchar(255)NOT NULL,
adres varchar(255)DEFAULT 'Adres Belirtilmedi',
il_kodu int
)
```

DEFAULT kısıtlayıcıyısıyla beraber bazı sistem verilerini de kullanabilirsiniz.Mesela müşterinin sipariş tarihini varsayılan olarak bugünün tarihi yapmak için GETDATE() fonksiyonu kullanılabilir.

```sql
CREATE TABLE Siparisler
(
s_id int NOT NULL,
musteri_id int,
s_tarihi date DEFAULT GETDATE()
)
```

Default Kısıtlayıcısı

Alter Table Cümleciğinde Default Kısıtlayıcısı

Oluşturduğunuz tabloda sonradan varsayılan olarak değerler eklemek isterseniz ALTER TABLE cümleciğinin içerisinde DEFAULT kısıtlayıcısını kullanabilirsiniz.

MySQL

```
1 | ALTER TABLE kisiler
2 | ALTER adres SET DEFAULT 'Adres Belirtilmedi'
```

MS SQL Server ve MS Access

```
1 | ALTER TABLE kisiler
2 | ALTER COLUMN adres SET DEFAULT 'Adres Belirtilmedi'
```

Oracle

```
1 | ALTER TABLE kisiler
2 | MODIFY adres DEFAULT 'Adres Belirtilmedi'
```

Default Kısıtlayıcısını Silme(İptal Etme)

Default kısıtlayıcısını silmek için aşağıdaki SQL sorguları kullanılır.

MySQL

```
1 | ALTER TABLE kisiler
2 | ALTER adres DROP DEFAULT
```

MS SQL Server,Oracle ve MS Access

```
1 | ALTER TABLE kisiler
2 | ALTER COLUMN adres DROP DEFAULT
```

Kaynaklar

- http://www.aydinmahmut.com/mysql-programlama-icin-performans-ipuclari/
- http://bilal.im/blog/web/mysql/innodb-ile-myisam-arasindaki-farklar/
- http://en.wikipedia.org/wiki/MySQL
- http://www.dijitalders.com/icerik/13/990/sql_komutlarindan_index_olusturma_ve_faydalari_myadminde_gosterimi.html#.U15tAPI_sa4
- http://yazilimdunyam.com/category/veritabani-tasarimi-ve-yonetimi/sql