Imperative Programming: programın nasıl yürütüldüğünü açıklayan bir tür programlama paradigmasıdır. Geliştiriciler daha çok adım adım nasıl yanıt alınacağıyla ilgileniyor. Imperative komutlar dizisi içerir. Bu paradigmada yürütme sırası çok önemlidir ve hem mutable hem de immutable verileri kullanır. Fortran, Java, C, C++ programlama dilleri Imperative Programming örnekleridir.

Declarative Programming: hangi programların yürütüleceğini açıklayan bir tür programlama paradigmasıdır. Geliştiriciler, alınan yanıtla daha fazla ilgilenir. Ne tür sonuçlar istediğimizi bildirir ve programlama dilini bir kenara bırakıp, sadece bunların nasıl üretileceğini bulmaya odaklanır. Basit bir deyişle, esas olarak nihai sonuca odaklanır. Hesaplamanın mantığını ifade eder. Miranda, Erlang, Haskell, Prolog, Declarative Programming'in birkaç örneğidir.

Veri tabanlarının sorgu optimizasyonlarında index: Indexleme sayesinde veritabanlarında veri düzenlemesi belli bir sıraya göre yapılır. Bu sayede yeni bir sordu oluşturulduğunda, db'de nereye gidileceği en başta bilinir. Bu sayede sorgu sürei ciddi anlamda kısalır. Az veri olan Dbde bu çok büyük bir avantaj olmayablir ama data büyüdükçe ciddi performans artışı olacaktır. Fakat bu yöntemde sürekli ve heryerde kullanılmamalıdır.

İlişkisel veritabanları için normalizasyon: dataların minimum bağımlı değişkenleri ile tek bir tabloda tutulması demektir. Bu sayede aslında tekrardan arındırılmış daha fazla sayıda ama daha az satırlık (genelde) tablolar ile veri saklanır. Mesela bir IMDB örneği düşünelim. Filmlerin database'in MovieID, Movie Name, Country, Language, ActorID gibi fieldleri barındırdığını düşünelim. 1. Normalizasyn aslında hepsinin olduğu şekldir. 2.Normal formda MovieID, Country, Language ayrılır, çünkü Language Country'e, Country de MovıeID'ye bağlıdır. 3. Normal formda ise direkt olarak Country ve sadece Language tutulabilir çünkü Country'den language türer sadece ona bağlıdır.

ORM: DRY : Veri modelinizi yalnızca bir yere yazarsınız ve kodu güncellemek, sürdürmek ve yeniden kullanmak kolaydır. Veritabanı işlemeden I18N 'ye kadar pek çok işlem otomatik olarak yapılır. Sizi yazmaya zorlar MVC kod, sonunda kodunuzu biraz temizler. Kötü biçimlendirilmiş bir SQL yazmak zorunda değilsiniz Hijyenik; Hazırlanan ifadeleri veya işlemleri kullanmak, bir yöntemi çağırmak kadar kolaydır. Doğal kodlama biçiminize uyuyor. DB sistemini soyutlar, böylece istediğiniz zaman değiştirebilirsiniz. Model uygulamanın geri kalan kısmına zayıf şekilde bağlı olduğundan, onu değiştirebilir veya başka bir yerde kullanabilirsiniz.Baş ağrısız veri kalıtımı gibi OOP iyiliğini kullanmanıza izin verir.Performans her zamanki sorgular için uygundur, ancak bir SQL master büyük projeler için kendi SQL'i ile her zaman daha iyisini yapar.

Domain Specific Language (DSL): Programlama dilleri, özel(Domain Specific Language) ve genel (General Purpose Language) kapsamda kullanılan diller olarak ikiye ayrılabilir. DSL (Domain Specific Language), özel bir uygulama alanı için kullanılan dildir. DSL Örneği olarak web uygulaması geliştireceksiniz ve uygulamanın özel bir kapsamı olan tasarım konusunda CSS'i verebiliriz.Örneğin java ile uygulama geliştiriyorsunuz ve veritabanı işlerini SQL ile yaparken, build ve deploy işlemleri için ANT kullanıyorsunuz.

Long lived transaction: Uzun süren transactionlar, databasede arama bulma gibi İşlemlerde kilitlenirse diğer operasyonlara devam edemeyeceği için Long lived transaction deriz. Lock, timeout gibi sorunlara yol açabilir.

Thread pool: Işletim sisteminin kaynaklarını kullandıkları için kontrolsüz bir şekilde kullanıldığında bu kaynakları hızlıca tüketmiş olurlar. örnek vermek gerekirse, donanımsal olarak 2 işlemci çekirdeğimiz mevcut olduğu bir ortamda 4 thread çalıştırmak gerekse bile önce ilk 2 sinin bitmesini bekleyip daha sonra diğer 2 sini oluşturup çalıştırmalıyız. Bu olayı bu şekilde takip etmek yerine Java nın bize sağladığı Thread havuzlarını kullanabiliriz. Thread sayısının ve çalışmasının düzenli ve kontrollü bir şekilde gerçekleştirilmesi için Java bize Executor adında bir sınıf sunmaktadır. Böylece aynı anda en fazla kaç adet thread çalışacağı belirlenmiş olur.

Scalability: Dikey ölçeklenebilirlik varolan makinenizin CPU, RAM gibi özelliklerini artırarak daha güçlü hale getirilmesidir. Yatay ölçeklenebilirlik ise varolan kaynak havuzunuza daha fazla makine eklenmesidir. Yatay ölçeklenebilirlikte load balancer kullanımıyla sistemin yükü tüm makinelere eşit şekilde dağıtılır.Cassandra , MongoDB yatay ölçeklenebilirliğe iyi birer örnektir. MySQL – Amazon RDS ise dikey ölçeklenebilirlike iyi birer örnektir.

Data replication ve sharding: Data replication datanın yatay ölçekleme ile aynı datanın farklı nodelara dağıtılmasıdır. Sharding ise datanın ufak parçalara bölünüp farklı nodelarda saklanmasıdır.