**MOBİL BULUT PROGRAMLAMA CEVAPLAR**

**C-1)**Dynamo DB’de,klasik veritabanında olduğu gibi ilişkisel model yoktur.Tablolar arası ilişkiler developer tarafından yönetilir, bundan dolayı developer’ın tablolara hakim olması gerekir.Tablolarda oluşturulan ögeleri almak için kod yazılır.Tabloları sorgulamak için primary key; hash veya hash and range tipinde olmalıdır.ProductCatalog tablosunda sadece hash varken,Forum,Reply ve Thread tablolarında hash and range vardır.

Klasik ilişkisel Database de read-write capacity önemli değildi ancak Kay value pair database de bu önemli bir noktadır.İlişkisel database de sadece ömodeli önemsiyoruz, işlevsel gereksinimleri karşılayabilmek için. Read ve write capacity işlevsel bir gereksinim değildir,performans related bişeydir.

Veritabanı yazma ve okuma kapasiteleri arasında fark var.Forum tablosunda çok fazla ekleme silme yapılacak bir tablo değil yani çok sık güncelleme omaz.ProductCatalog tablosunda ise Forum’ a göre daha fazla okuma ve yazma yapılır.Birşey yazıldığında hemen reply eder.

ÖR:ÖSYM örneğinde sistem ilişkisel veritabanıyla yapıldığı için performans kaybı yaşanır.Çünkü aynı anda birden fazla talep gelir ve veritabanı buna cevap veremez.Burada kimse bir şey yazmaz sürekli value’lar okunur.Bundan dolayı burda okuma kapasitesi çok yüksek olmalıdır.

**C-2) İlişkisel DB:**

* Fiziksel tablolar ve ilişkiler var.
* Develeoper modele hakimdir ve metadata bunu destekler.
* İlişkiler joinlerle sağlanır.
* Sorgulama yapılır.

**Key value pair DB:**

* Tablolar arası ilişkiler fiziksel olarak değildir bunu developer yönetir.
* Bu yüzden developer’ın modele hakim olması gerek ancak bunu destekleyen bir metadata yok.
* Join işlemleri yok.
* Sorgulama Nerdeyse yok.Bunun yerine ögeler hash and range kullanılarak programlamayla çekiliyor.

Key value pair DB’de Developer’ın ilişkileri yönetmesi developer’a büyük yük getirir.(Dezavantaj)

Join işleminin olmaması karmaşıklığı azaltır.Key value pair DB’de hash and range alanlarına göre sorgulama yapılması performansı artırır.

**C-3)**İlişkisel veritabanında bir transaction olması için ACID ilkelerine uyması gerekir.

**ACID; Atomicity,Consistency,Isolation ve Durability(roll back)** kavramlarının açılımıdır.

**Atomicity :** Transaction içindeki tüm işlemcikler başarılı olmalıdır .

**Consistency :** Yapılan işlemler sonucunda oluşan çıktıların tutarlılığı anlamına gelir.

**Isolation :** Transaction gerçekleşirken dışardan müdahaleye izin verilmemilidir.

**Durability(Rollback) :**Eğer transaction içinde bir hata olursa sistem en başa döner.

Dağıtık sistemlerde hata toleransı yüksektir.Birkaç düğümde meydana gelen hata tüm sistemi etkilemez.

**BASE;** ACID’e alternatif olarak gelmiştir.NoSQL DB’lerde kullanılır.

* **Basic Availability**
* **Soft-state**
* **Eventual consistency**

**BASE’** de hata toleransı **ACID**’e göre daha yüksektir ancak gelişmesi güçtür.

* ACID yani ilişkisel veritabanında strongly consistent ön plandadır.
* Strongly consistent tutarlıdır çünkü ilişkisel yapıdadır.
* Tek lokasyonda olduğu için faz farkı ve gecikmeler olmaz.
* Eventually consistent’ta ise tutarlılık var ancak farklı lokasyonlarda olduğu için biraz gecikmeler meydana gelir(Latency).Ancak bu gecikmeler saniye bazındadır.
* Farklı lokasyonlarda olduğu için faz farkı olur.
* Arka planda dağıtık yapı olduğundan saniyede 1 milyon transaction işlenir.

**C-4) Saas:** Bir hizmet sağlayıcı tarafından merkezi olarak sunucu üzerinde barındırılan bir yazılım uygulamasının; birden fazla kişi/kurumun kullanımına sunulması işe dağıtımını sağlayan modeldir. Tamamen SaaS, Servis sağlayıcılar tarafından sunulmaktadır. Bu şu demektir: müşteri olarak işletme ve operasyonel maliyetlerden kurtulmak ve sadece ihtiyacınız kadar olanı için düşük maliyetli ama kaliteli hizmet almak. SaaS modeli, şirketlerin ticari olarak lisanslama yolu ile kendi iç bünyelerinde barındırdıkları ve işlettikleri yazılımların yüksek ön maliyetli ve kompleks yapılarına alternatif olarak düşük maliyetli bir şekilde aynı avantajlara ulaşmayı sağlar. Herhangi bir web sayfasına nasıl ulaşıyorsanız bu yazılıma da web üzerinden bir istemci olarak yine aynı şekilde ulaşmamız mümkündür. Sahip olmanız gereken tek şey ise yalnızca o servisi almanızı sağlayacak bir hesap.Google Docs bir örnektir.

**Paas:** PaaS müşterilere kendi servislerini geliştirebileceğini platform hizmetini sunuyor. PaaS tedarikçilerin servis üretimi, sürdürümü, servis entegrasyonu ve işletmesini içine alan hizmetleri üstlenir, müşterilerin bu platformlar üzerinde kendi servislerini düşük maliyetle üretmesi fikrine dayanan yeni bir iş modelini hayata geçirir.Google App Engine Microsoft Azure örnek verilebilir.

**Iaas:** Iaas, bulut sağlayıcısının imkan ve kabiliyetleri doğrultusunda işlem gücü, depolama alanı, ağ ve diğer temel bilişim kaynaklarına erişebilmekte olup, bu kaynaklar üzerinde işletim sistemi ve uygulamalar da dahil olmak üzere her türlü yazılımı istediği gibi kullanabilmektedir. Kullanıcıların bulut alt yapısı ile ilgi ve alakaları olmaksızın sadece işletim sistemi, depolama ve dağıtımı yapılmış yazılımlar ile kullanılacak olan ağ bileşenlerinin seçimi gibi kısıtlı yapılandırma alanları üzerinde işlem yapma hakkına sahiptir.Ödeme şeklinde ise “kullandığın kadar öde” biçimiyle hizmet veren bulut bilişim ürünüdür.Bu servis türüne Amazon ECS ve RackSpace Cloud ortamları örnek verilebilir.

**C-5)**AWS’nin başarısının arkasındaki gerekçeler:

* Sanallaştırma
* API sunması
* Gereksinim odaklı(İhtiyaca göre geliştirilmiş ve herkes kullanmış)
* Esneklik
* Performans
* Ölçeklenebilirlik
* Hata Toleransı yüksek