

VERİ TABANI VE SİSTEMLERİ BÖLÜM SONU SORULARI

Design Concept Chap 5 Soruları

12- Hangi temel özellikler arzu edilir olarak kabul edilir ?Her bir özelliğin neden arzu edilir olarak değerlendirildiğini açıklayın.

Cevap: 1. Veri Bütünlüğü (Data Integrity)

- **Neden arzu edilir?:** Veri bütünlüğü, verilerin doğru, tutarlı ve güvenilir olmasını sağlamak için önemlidir. Veritabanlarında yanlış veri girişi veya bozulmuş veriler, uygulamaların doğru çalışmamasına veya yanlış sonuçlar üretmesine yol açabilir. Veri bütünlüğü, kurallar (örneğin, birincil anahtar, yabancı anahtar kısıtlamaları) aracılığıyla sağlanır ve verilerin doğruluğunu garanti eder.

2. Veri Güvenliği (Data Security)

- **Neden arzu edilir?:** Veritabanı güvenliği, verilerin kötü niyetli erişim, kayıp veya tahribattan korunmasını sağlar. Güvenlik özellikleri, yetkisiz erişimleri engelleyerek verilerin gizliliğini ve bütünlüğünü korur. Bu, özellikle kişisel, finansal ve hassas verilerin bulunduğu sistemlerde büyük önem taşır.

3. Performans (Performance)

- **Neden arzu edilir?:** Yüksek performans, veritabanının hızlı bir şekilde veri sorgulama, ekleme, güncelleme ve silme işlemleri yapabilmesini sağlar. Performans, kullanıcı deneyimini doğrudan etkiler. Veri erişim hızının artırılması, büyük veri setlerinin işlenmesi ve işlem süresinin kısaltılması gereklidir.

4. Ölçeklenebilirlik (Scalability)

- **Neden arzu edilir?:** Ölçeklenebilirlik, veritabanının büyüyen veri taleplerine uygun şekilde genişletilebilmesini ifade eder. Veritabanı, hem yatay (daha fazla sunucu ekleyerek) hem de dikey (daha güçlü sunucular kullanarak) şekilde ölçeklendirilebilir olmalıdır. Bu özellik, veritabanının gelecekteki büyüme ihtiyaçlarına cevap verebilmesi için kritik öneme sahiptir.

5. Erişilebilirlik (Availability)

- **Neden arzu edilir?:** Veritabanlarının sürekli erişilebilir olması, kullanıcıların veya uygulamaların veritabanına her zaman ulaşabilmesi için önemlidir. Erişilebilirlik,

kesintisiz iş süreçlerini ve veritabanının sürekli çalışma durumunu garanti eder. Bu özellik, yüksek erişilebilirlikli sistemlerde (örneğin, finansal işlemler veya sağlık sistemleri) kritik öneme sahiptir.

6. Yedekleme ve Kurtarma (Backup and Recovery)

- **Neden arzu edilir?:** Veritabanlarında yedekleme ve kurtarma mekanizmalarının bulunması, verilerin kaybolması durumunda veritabanının hızlı bir şekilde eski haline getirilmesini sağlar. Bu özellik, veri kaybını önler ve veritabanı arızaları, doğal afetler veya hatalı işlemler sonucu veri kaybı yaşanmasının önüne geçer. Veritabanlarında bu temel özelliklerin bulunması, hem verilerin doğru ve güvenli bir şekilde saklanmasını sağlar hem de sistemin verimli bir şekilde çalışmasına yardımcı olur. Bu özellikler, kullanıcı memnuniyetini artırır, operasyonel verimliliği yükseltir ve organizasyonların uzun vadeli başarısını destekler.

13- Bileşik birincil anahtarlar hangi koşullar altında uygundur?

Cevap: **1. Tek Bir Sütun Benzersiz Değilse**

- **Koşul:** Bir tablodaki tek bir sütun, her satır için benzersiz değerler içermiyor veya doğal birincil anahtar olamayacak kadar fazla tekrar içeriyor.
- **Açıklama:** Örneğin, bir "Öğrenci" tablosunda, öğrencilerin "isim" veya "doğum tarihi" gibi sütunları, genellikle benzersiz olmayabilir çünkü aynı isimli veya aynı doğum tarihli birden fazla öğrenci olabilir. Bu durumda, hem "Öğrenci_ID" hem de "Kurs_ID" sütunları birleşerek bileşik birincil anahtar oluşturabilir.

2. Tabloya Ait Birden Fazla Önemli Özellik (Attribute) Varsa

- **Koşul:** Bir tablonun benzersiz kimliğini tanımlamak için iki veya daha fazla özellik (sütun) gereklidir.
- **Açıklama:** Örneğin, bir "Öğrenci_Kurs" ilişkisi tablosunda, her öğrenci birden fazla derse kayıtlı olabilir. Bu tablonun her kaydını benzersiz bir şekilde tanımlamak için **Öğrenci_ID** ve **Kurs_ID** sütunlarının birleşimi kullanılabilir. Tek başına bir sütun (Öğrenci_ID veya Kurs_ID) benzersiz olmayacaktır, ancak bu iki sütunun birleşimi her kaydı benzersiz kılacaktır.

3. Birden Fazla Yabancı Anahtar İlişkisi Varsa

- **Koşul:** Bileşik birincil anahtar, iki veya daha fazla tablonun ilişkilendirilmesi sonucu, yabancı anahtarların birleşiminden türetilir.
- **Açıklama:** Bileşik birincil anahtarlar, bazen bir ilişkisel tablonun birden fazla yabancı anahtar içerdiği durumlarda kullanılır. Örneğin, bir "Sipariş_Detay" tablosunda hem "Sipariş_ID" hem de "Ürün_ID" yabancı anahtar olabilir. Her iki sütun da kendi başlarına benzersiz olmayabilir, ancak bunların birleşimi her sipariş detayını benzersiz kılacaktır.

Özetle:

Bileşik birincil anahtarlar, **benzersizliği sağlamak için iki veya daha fazla sütunun birleşimiyle** oluşturulurlar ve genellikle **birden fazla sütun gerektiren ve çoktan çoğa ilişkilerin bulunduğu** tablolarda uygundur. Bu tür anahtarlar, veritabanı tasarımında özellikle **veri bütünlüğünü sağlamak, ilişkileri modellemek ve tekrarı engellemek** amacıyla kullanılır.

14- Vekil birinci anahtar nedir ve ne zaman kullanırsınız?

Cevap: **Vekil birincil anahtar** (veya **yapay birincil anahtar**), veritabanındaki bir tablonun birincil anahtarı olarak kullanılan, gerçek dünyada anlam taşımayan ve genellikle veritabanı yönetim sistemleri tarafından (DBMS) otomatik olarak oluşturulan bir değerdir. Bu tür bir anahtar, genellikle **arttırılan sayılar (auto-increment)**, **GUID'ler (Global Unique Identifier)** veya **UUID'ler (Universally Unique Identifier)** gibi değerler olabilir.

Vekil Birincil Anahtar Ne Zaman Kullanılır?

1. **Doğal Anahtar Bulunamadığında veya Karmaşık Olduğunda:**
 - a. **Koşul:** Bir tablodaki doğal anahtar (gerçek dünyadaki benzersiz bir değer) karmaşık, çok parçalı veya değişken olabilir.
 - b. **Örnek:** Bir "Öğrenci" tablosunda öğrenci numarasının benzersiz olması gerektiği düşünülse de, öğrenci numarası ilerleyen zamanlarda değişebilir veya çok uzun olabilir. Bu durumda, öğrenci tablosunda "Öğrenci_ID" gibi kısa ve anlam taşımayan bir vekil anahtar kullanılabilir.

2. Veri Bütünlüğü ve Yönetimi Kolaylaştırmak İçin:

- a. **Koşul:** Doğal anahtarların değiştirilmesi, güncellenmesi veya yönetilmesi zor olduğunda.
- b. **Açıklama:** Vekil anahtar, doğrudan ilişkili olmayan ama her zaman benzersiz bir değer sunduğu için veri güncellemelerini kolaylaştırır. Doğal anahtarlar değiştirilebilir, ancak vekil anahtarlar değişmez ve bu, veritabanı bütünlüğünü korumak için önemlidir.

3. Veritabanı Performansını İyileştirmek İçin:

- a. **Koşul:** Karmaşık ve çok sütunlu doğal anahtarlar veritabanı sorgularını yavaşlatıyorsa.
- b. **Açıklama:** Vekil birincil anahtarlar genellikle daha küçük veri türlerine sahiptir (örneğin, bir tam sayı veya GUID), bu da sorguları hızlandırır. Çok büyük ve karmaşık doğal anahtarlar sorguları yavaşlatabilirken, vekil anahtarlar daha verimli erişim sağlar.

4. Birleşik Anahtarları Basitleştirmek İçin:

- a. **Koşul:** Bileşik anahtarlar kullanılıyorsa, bu anahtarları sadeleştirmek ve yönetimi kolaylaştırmak için.
- b. **Açıklama:** Eğer bir tablodaki birincil anahtar, birden fazla sütunun birleşiminden oluşuyorsa, bu durum veritabanı tasarımını karmaşıklıştırabilir. Vekil birincil anahtar kullanarak, bileşik anahtarı tek bir sütunla değiştirebilir ve veritabanı yönetimini kolaylaştırabilirsiniz.

Vekil birincil anahtarlar, genellikle karmaşık veri yapılarında, değişken veri ve ilişkilerde, performans gereksinimlerinde ve veri bütünlüğü sağlamak için kullanılır. Bu anahtarlar, veritabanı tasarımını basitleştirir, performansı artırır ve veri bütünlüğünü korur. Ancak, kullanımı gerektiğinde dikkatli düşünülmeli ve gereksiz karmaşıklığı önlemek için doğru yerde kullanılmalıdır.

15-1 :1 ilişkisini uygularken,bir taraf zorunlu ve bir taraf isteğe bağlıysa yabancı anahtarı nereye yerleştirmelisiniz?Yabancı anahtar zorunlu mu yoksa isteğe bağlı mı olmalıdır?

Cevap: **Zorunlu ve İsteğe Bağlı Taraflar:**

- **Zorunlu taraf:** Bu tarafın her kaydının, diğer tablodaki bir kayda karşılık gelmesi gerekir. Yani, bu tarafın kaydının olması **zorunludur**.
- **İsteğe bağlı taraf:** Bu tarafın kaydının, diğer tablodaki kayda karşılık gelmesi **isteğe bağlıdır**. Yani, bir kaydın olması durumunda karşılık gelen kaydın var olması zorunlu değildir.

Yabancı Anahtarın Nerede Bulunacağı?

Yabancı anahtar, isteğe bağlı olan tarafın tablosuna yerleştirilmelidir.

- **Açıklama:** Yabancı anahtar, bir tablonun başka bir tablodaki bir kayda referans verdiği sütundur. Eğer bir tarafın kaydının diğer tarafın kaydına karşılık gelmesi zorunluyorsa, yabancı anahtarın bulunduğu tablonun her kaydının mutlaka karşılık geldiği bir başka kayda sahip olması gerekir. Ancak, isteğe bağlı tarafta bu durum her zaman geçerli olmadığı için, yabancı anahtar burada olmalı ve bu yabancı anahtar **isteğe bağlı** olmalıdır.

Yabancı Anahtarın Zorunlu veya İsteğe Bağlı Olması:

- **Zorunlu Yabancı Anahtar:** Yabancı anahtarın zorunlu olması, karşılık gelen kayıt her zaman bulunması gereken taraf için geçerlidir. Bu durumda, zorunlu tarafın tablosunda, yabancı anahtarın **NULL** değeri alması **yasaktır**. Yani, her kaydın ilişkili olduğu bir diğer kaydın olması zorunludur.
- **İsteğe Bağlı Yabancı Anahtar:** İsteğe bağlı tarafın tablosunda, yabancı anahtar **NULL** olabilir. Bu, isteğe bağlı tarafın kaydının her zaman diğer tabloya bağlanmasının gerekmediği durumdur. Yani, bazen bir kaydın diğer tablodaki kayda karşılık gelmemesi mümkündür.

Özetle:

- **Yabancı anahtar, isteğe bağlı tarafın tablosunda yer almalıdır.**
- Yabancı anahtar **isteğe bağlı** olmalıdır, çünkü isteğe bağlı tarafın kaydının diğer tablodaki kayda karşılık gelmesi her zaman zorunlu değildir.

16- Zamanla değişen veri nedir ve veritabanı tasarımı açısından bu tür verilerle nasıl başa çıkarsınız?

Cevap: **Zamanla değişen veri**, bir veritabanında zaman içinde değişebilen, güncellenebilen veya geçmişi izlenmesi gereken veriyi ifade eder. Bu tür veriler genellikle, bir nesne, olay veya durumun geçmişteki, mevcut ve gelecekteki halleri arasında izlenebilir olmalıdır. Örneğin, bir müşterinin adresi, bir ürünün fiyatı, bir çalışanın maaşı veya bir öğrencinin notları zamanla değişebilen verilere örnek teşkil edebilir.

Veritabanı tasarımında **zamanla değişen verilerle başa çıkmak**, bu verilerin doğru bir şekilde izlenmesi, saklanması ve gerektiğinde geçmiş verilerine erişilmesi için dikkatli bir tasarım gerektirir. Aksi takdirde, veri kayıpları, doğruluk sorunları ve performans zorlukları ortaya çıkabilir.

17- En yaygın tasarım tuzağı nedir ve nasıl ortaya çıkar?

Cevap: **En yaygın tasarım tuzağı**, genellikle "**aşırı normalizasyon**" veya "**yetersiz normalizasyon**" olarak tanımlanabilir ve her iki durum da veritabanı tasarımında sık karşılaşılan problemler oluşturur. Bu tasarım tuzakları, veritabanının etkinliğini, verimliliğini ve esnekliğini olumsuz etkileyebilir.

Aşırı Normalizasyonun Ortaya Çıkma Sebepleri:

- **Normalizasyon Kurallarına Aşırı Bağlılık:** Veritabanı tasarımcıları, veritabanını 3NF veya daha yüksek normal formlara ulaştırmaya çalışırken, veri modelini gereksiz yere böler ve sorgulama süreçleri karmaşıklaşır.
- **Gereksiz Tabloların Oluşturulması:** Bu durum, küçük veri gruplarının bile ayrı tablolarda tutulmasına yol açabilir, bu da sorgu performansını düşürür.