* Hareket (Transaction) Yöneticisi  
  Tutarlılık açısından, bir hareketi oluşturan işlemlerin ya tümünün gerçekleştirilmesi, ya da hiçbirinin gerçekleştirilmemesi gerekir. Bu özelliğe hareketin ACID (Atomicity, Consitency, Isolation, Durability) özelliği adı verilir.
* Veri tabanı yöneticisinin kullandığı başlıca olanak veri tanımlama dilidir (DDL). DML ve DDL dilleri sorgu amaçlı olmamalarına rağmen çoğu zaman SQL’in bir parçası olarak görülürler. Sorgu Dili Kullanıcıları DML komutlarını kullanabilir, fakat DDL’i sadece VT Yöneticisi kullanır.

metin, ekran görüntüsü, yazı tipi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, diyagram, ekran görüntüsü, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu**Ders 3: Veri Modelleri (Varlık-İlişki Modeli)**

**1. Varlık-İlişki Modeli Nedir?**

* **Tanım:** Varlık-İlişki (E-R) modeli, 1976 yılında P.P. Chen tarafından geliştirilen, VTYS'den bağımsız veri modelleme ve semantik veri analizinde kullanılan bir modeldir.
* **Kullanım Alanları:**
  + Verileri anlam ve özelliklere göre çözümlemek.
  + E-R çizelgeleri oluşturarak VTYS belirlemek.
  + Veritabanı şemalarına dönüştürmek.

**2. Temel Kavramlar**

**2.1. Varlık ve Varlık Kümesi**

* **Varlık (Entity):** Ayırt edilebilir her nesneye denir (Bir öğrenci, bir kitap vb.).
* **Varlık Kümesi:** Aynı türden benzer varlıkların oluşturduğu kümedir.
  + Örnek: "Bilgisayar Mühendisliği öğrencileri" veya "renkler".

**2.2. Nitelik (Attribute)**

* **Tanım:** Varlıkların özelliklerini ve farklılıklarını göstermek için kullanılır.
  + Gerçek dünyadaki niteliklerin hepsi modele dahil edilmez; sadece gerekli olanlar seçilir.
* **Etki Alanı (Domain):** Bir niteliğin alabileceği tüm değerler kümesidir.
  + Örnek: "Not" niteliğinin etki alanı 0-100 arası olabilir.
* **Türetilen Nitelik:** Diğer niteliklerden elde edilir.
  + Örnek: "Doğum tarihi" niteliğinden "yaş" hesaplanabilir.
* **Birleşik Nitelik:** Birden fazla niteliğin birleşimidir.
  + Örnek: "Adres" niteliği; şehir, cadde vb. bileşenlerden oluşur.

**2.3. İlişki ve İlişki Kümesi**

* **İlişki:** Varlıklar arasındaki bağlantılar.
  + Örnek: "Bir öğrencinin aldığı ders."
* **İlişki Kümesi:** Aynı türdeki ilişkilerin bir kümesidir.
  + Matematiksel tanım: .
* **Çoklu İlişki Kümesi:**
  + Birden fazla varlık kümesi arasında kurulan karmaşık ilişkiler.
  + Öneri: İkili ilişkilere bölünmesi tercih edilir.

**2.4. Rol**

* Varlıkların ilişkideki işlevi.
  + Örnek: Öğrenci ve ders ilişki kümesinde "öğrenci" alan, "ders" alınan rolündedir.

**2.5. 2.5. Anahtarlar**

* **Tanım: Bir varlığın veya ilişkinin diğerlerinden ayırt edilmesini sağlayan nitelik veya nitelik grubu.**
  + **Süper Anahtar (Superkey): Bir kümedeki varlıkları veya ilişkileri ayırt etmeyi sağlayan bir veya daha fazla niteliğin birleşimidir. Fazlalık içerebilir.**
  + **Aday Anahtar (Candidate Key): Süper anahtarın bir alt kümesidir. Gereksiz nitelikler içermez ve varlıkları ayırt etmek için yeterlidir.**
  + **Birincil Anahtar (Primary Key): Aday anahtarlar arasından seçilen ve veri tabanında her varlığı benzersiz şekilde tanımlayan anahtardır.**
  + **Yabancı Anahtar (Foreign Key): Bir başka tabloyu işaret eden anahtar niteliğidir. Varlıklar arasındaki ilişkileri belirtmekte kullanılır.**

**2.6. İlişki Türleri**

* **Birden Bire (1-1): Her varlık sadece bir başka varlıkla ilişkilidir.**
  + **Örnek: "Evlilik" ilişkisinde her bireyin bir eşi olabilir.**
* **Birden Çoğa (1-N): Bir varlık birden fazla başka varlıkla ilişkilidir.**
  + **Örnek: "Bölüm" ve "Öğrenci" arasındaki ilişki.**
* **Çoktan Bire (N-1): Birden fazla varlık bir varlıkla ilişkilidir.**
  + **Örnek: "Öğrenci" ve "Bölüm" arasındaki ilişki.**
* **Çoktan Çoğa (N-M): Birden fazla varlık birden fazla başka varlıkla ilişkilidir.**
  + **Örnek: "Öğrenci" ve "Ders" arasındaki ilişki.**

**3. Varlık-İlişki Çizelgeleri**

**3.1. Chen Notasyonu**

* Varlık Kümesi: Dikdörtgen.
* Nitelik: Elips.
* Anahtar Nitelik: Altı çizili.
* İlişki: Elmas.
* Zayıf Varlık: Çift dikdörtgen.
* Bağımlı Varlık: Çift elmas.

**3.2. UML Notasyonu**

* Varlık Kümesi: Dikdörtgen.
* Nitelik: Dikdörtgen içinde.
* İlişki: İki varlık arasında çizgi.

**Ders 5: Varlık-İlişki Çizelgelerinin Tablolara Dönüştürülmesi**

**1. Genel Dönüştürme Yöntemi**

* **Amaç:** Varlık-İlişki modelini ilişkisel modele çevirmek.
* **Temel Kural:** Her varlık kümesi ve ilişki kümesi bir tabloya dönüştürülür.

**2. Varlık Kümesinin Dönüştürülmesi**

* **Güçlü Varlık:** Tabloda nitelikleri ve birincil anahtarı yer alır.
  + Örnek: Öğrenci tablosu (ÖğrNo, Ad, Soyad).
* **Zayıf Varlık:** Bağımlı olduğu güçlü varlığın anahtarı da eklenir.
  + Örnek: Lise ve Öğrenci tablosu (ÖğrNo, LiseNo).

**3. İlişki Kümesinin Dönüştürülmesi**

* **İkili Çoktan-Çoğa:** Ayrı bir tablo olarak dönüştürülür.
  + Örnek: Satış ilişkisi (MagazaNo, MalKod, Miktar).
* **İkili Birden-Çoğa:** Bağımlı varlık kümesinde tanımlanır.

**4. Genelleme ve Kümeleme**

* **Genelleme:** Alt ve üst varlık kümesinin nitelikleri tek tabloda birleştirilebilir.
* **Kümeleme:** Bütün-parça ilişkileri görüşüyle dönüştürülür.

metin, ekran görüntüsü, diyagram, çizgi içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazı tipi, sayı, numara içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, yazı tipi, ekran görüntüsü, beyaz içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, sayı, numara, diyagram, ekran görüntüsü içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldumetin, ekran görüntüsü, yazı tipi, diyagram içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

**Ders 6: İlişksel Cebir ve SQL**

**1. İlişksel Diller**

* **İlişksel Cebir:** Cebirsel işlemlerle sorguların belirlenmesi.
  + Kullanıcı işlemleri sıralı olarak yazar.
* **İlişksel Hesaplama:** Kısıtlarla sorguların belirlenmesi.
  + Kullanıcı nasıl yerine ne sorusunu yanıtlar.

**2. İlişksel Cebir İşlemleri**

**2.1. Temel İşlemler**

* **Seçme:** Belirli bir koşulu sağlayan satırları seçer.
  + Notasyonu: .
* **İzdüşüm:** Belirli sütunları seçer.
  + Notasyonu: .

**2.2. Birleşirme ve Kesişim**

* **Birleşirme:** Tabloların küme birleşimi.
  + Notasyonu: .
* **Kesişim:** Tabloların küme kesışimi.
  + Notasyonu: .

**2.3. Doğal Birleşme ve Kartezyen Çarpım**

* **Doğal Birleşme:** Ortak nitelikler üzerinden birleşir.
  + Notasyonu: .
* **Kartezyen Çarpım:** Tüm kombinasyonları üretir.
  + Notasyonu: .

**2.4. Diğer İşlemler**

* **Fark:** Bir tablodan diğerinin farkı.
  + Notasyonu: .
* **Bölme:** Karmaşık tabloların çözümü.
  + Notasyonu: .

**3. SQL ve Uygulamaları**

metin, ekran görüntüsü, yazı tipi içeren bir resim

Açıklama otomatik olarak oluşturuldu

DERS 7

Tabii ki, bu slaytı detaylı bir şekilde özetleyebilirim. İşte "Ders-7 Bütünlük Kısıtlamaları" slaytının kapsamlı özeti:

**1. Bütünlük Kısıtlamaları**

* **Tanım**: Veri tabanının tutarlı ve doğru olmasını sağlamak için tanımlanan kısıtlamalardır.
* **Özellikler**: Yanlış, eksik, çelişen veya tutarsız veri olmamalıdır.
* **Uygulama**: Ekleme, güncelleme ve silme işlemleri bütünlük kısıtlamalarına uymalıdır.

**2. Bütünlük Kısıtlaması Çeşitleri**

* **Veri tabanı yapısında yer alanlar**: Tanımlanmış kurallar.
* **İşletim aşamasında denetlenenler**: Uygulama programları içinde yer alır.
* **Veri tabanı tanımında olmayanlar**: Kullanıcıların uyması beklenir.

**3. Temel Bütünlük Kısıtlamaları**

* **Alan Kısıtlamaları**: Niteliklerin alabileceği değerlerin sınırlandırılması.
* **Alan kısıtlamaları:** Her niteliğe bir alan eşlenmesi ve niteliğin alabileceği değerlerin bu alandaki değerlerle sınırlandırılmasıdır.

*"Alan kısıtlaması tür tanımı ve değerlerle ilgili kimi kısıtlamalarla gerçekleştirilir."*

* + **Örnek**: alan tür :Tamsayı, kesirli sayı, karakter, tarih gibi alan türleri.
  + **Değer sınırları** : alt üst sınır değeri vb.
  + **Boş null değer** :nitelik değerinin eksik olup olmayacağı
* **Referans Kısıtlamaları**: Bir ilişkideki niteliklerin başka bir ilişkideki niteliklerle sınırlandırılması. **Referans kısıtlamaları:** Bir ilişkideki bazı niteliklerin alabileceği değerlerin, başka bir ilişkideki mevcut değerlerle sınırlandırılmasıdır. Başka bir deyişle, bir ilişkideki kayıtların varlığı, başka bir ilişkideki belirli kayıtların varlığına bağlıdır.
* *"Başka bir deyişle bir ilişkideki çokluların varlığının bir başka ilişkideki belirli çokluların varlığına bağımlı olmasıdır."*
  + **Örnek**: SATIŞ ilişkisinde SÜKODU ve SMNO nitelikleri, ÜRÜN ve MAĞAZA ilişkilerindeki ÜKODU ve MNO değerlerine eşit olmalıdır.
* **Nitelikler Arası Bağımlılıklar**: Niteliklerin birbirine bağımlı olması.
  + **Örnek**: SATICI ilişkisinde ÜKODU, FNO, FADI, FADRESİ, SFİYATI nitelikleri.

**4. İşlevsel Bağımlılık Türleri**

* **İşlevsel bağımlılık: Bir nitelik kümesindeki (X) değerler, başka bir nitelik kümesindeki (Y) değerleri tek bir şekilde belirliyorsa, Y, X'e işlevsel bağımlıdır denir.**

***"Eğer X nitelik değerleri aynı olan tüm çoklularda Y nitelik değerlerinin de aynı olması gerekiyorsa X, Y’ yi işlevsel belirler ya da Y, X’e işlevsel bağımlıdır denir."***

* **Kısmi işlevsel bağımlılık: X, A'yı belirliyor ve X'in en az bir öz altkümesi de A'yı belirliyorsa, X → A işlevsel bağımlılığına kısmi işlevsel bağımlılık denir.**
* **Tam işlevsel bağımlılık: X, A'yı belirliyor ve X'in hiçbir öz altkümesi A'yı belirlemiyorsa, X → A işlevsel bağımlılığına tam işlevsel bağımlılık denir.**
* **Önemsiz işlevsel bağımlılık: X, A'yı belirliyor ve A, X'in bir altkümesi ise, X → A işlevsel bağımlılığına önemsiz işlevsel bağımlılık denir.**
* **Önemli işlevsel bağımlılık: X, A'yı belirliyor ve A, X'in bir altkümesi değilse, X → A işlevsel bağımlılığına önemli işlevsel bağımlılık denir.**
* **Geçişli işlevsel bağımlılık: X, Y'yi ve Y de Z'yi belirliyorsa, X → Z işlevsel bağımlılığına geçişli işlevsel bağımlılık denir.**

***"Eğer X, Y’yi Y de Z’yi belirliyorsa; (X → Y ve Y → Z) X→ Z işlevsel bağımlılığına geçişli işlevsel bağımlılık denir."***

**5. İşlevsel Bağımlılıkların Kapanışı**

* **Tanım**: Bir işlevsel bağımlılık kümesinin kapanışı, tüm önemsiz, kısmi ve geçişli bağımlılıkları içerir.
* **Örnek**: R(A,B,C,D) ve F = {A → B , BC → D} verildiğinde F+ kümesi.

**6. İşlevsel Bağımlılıkları Türetme Kuralları**

* **Dönüşlülük Kuralı**: Y ⊆ X ise X → Y.
* **Artırma Kuralı**: X → Y ise XZ → Y.
* **Geçişlilik Kuralı**: X → Y ve Y → Z ise X → Z.
* **Birleşim Kuralı**: X → Y ve X → Z ise X → YZ.
* **Ayrışma Kuralı**: X → YZ ise X → Y ve X → Z.
* **Sözde Geçişlilik Kuralı**: X → Y ve YZ → W ise XZ → W.
* Veri yinelemesi, güncelleme ve silme anomalileri, ilişki şemalarında nitelikler arası bağımlılıkların dikkate alınmamasından kaynaklanabilir.
* İşlevsel bağımlılıklar, veritabanı tasarımında normalleştirme teknikleri kullanarak bu sorunları çözmek için kullanılabilir.
* **Değersiz** işlevsel bağımlılıklar kısmi ve geçişli işlevsel bağımlılıklardır.
* **Değerli** işlevsel bağımlılıklar ise önemli, tam ve geçişli olmayan işlevsel bağımlılıklardır.

Bu özet, slayttaki tüm kavramları ve örnekleri kapsamaktadır. Başarılar dilerim! Sormak istediğin başka bir şey var mı?