VERİ TABANI YÖNETİM SİSTEMLERİ

Terminolojik Bilişim Sözlüğü ve Kavram Taksonomisi

Temel kavramlar • Veri modelleme • İlişkisel yapı • Veri tipleri • SQL • Araçlar • Kavram karışıklıkları

Kaynak: Veri Tabanı Yönetim Sistemleri-I ders dokümanı (46 sayfa)

Bu sözlük nasıl kullanılmalı?

Bu çalışma, ders dokümanındaki konu ve terminoloji esas alınarak hazırlanmıştır. Terimler, öğrencinin önce temel kavramları, ardından veri modelini, ilişkisel yapıyı ve SQL işlemlerini anlayacağı bir öğrenme sırasına göre sınıflandırılmıştır. İngilizce karşılıklar, bilişim literatüründe karşılaşılabilecek adlandırmaları tanımayı kolaylaştırmak amacıyla verilmiştir.

Önemli not: dokümanda MS Access, MySQL, Oracle ve SQL Server için verilen bazı veri tipi açıklamaları eski sürümlere aittir. Ürün sürümleri arasında uzunluk, destek ve davranış farklılıkları bulunabilir. Bu nedenle veri tipi tabloları, öncelikle ders terminolojisini tanıtma amacı taşır; güncel üretim sistemlerinde kullanılmadan önce ilgili VTYS sürümünün resmî dokümantasyonu kontrol edilmelidir.

Taksonomik genel görünüm

Ana dal	Alt dallar	Bu dalın cevapladığı temel soru
1. Veri tabanı temelleri	Veri, bilgi, veri tabanı, VTYS, veri bağımsızlığı	Ne saklanır ve kim yönetir?
2. Veri modelleme	Veri modeli, ER modeli, varlık, nitelik, ilişki, rol, kısıt	Gerçek dünya nasıl modellenir?
3. İlişkisel yapı	İlişki/tablolar, satır, sütun, şema, anahtarlar, kardinalite	Model tablolarda nasıl temsil edilir?
4. Veri tanımlama	Veri tipleri, uzunluk, kesinlik, NULL, bütünlük	Bir alan hangi tür değeri kabul eder?
5. SQL dili	DDL, DML, sorgulama, koşullar, sıralama, fonksiyonlar	Veri ve şema üzerinde nasıl işlem yapılır?
6. VTYS araç ekosistemi	Access, MySQL, Oracle, SQL Server, PostgreSQL; istemci ve yönetim araçları	Hangi yazılım ortamında çalışılır?

1. Veri tabanı temelleri ve yönetim kavramları

Terim	İngilizce karşılık	Tanım ve açıklama	Örnek / not
Veri	Data	İşlenmemiş, tek başına anlamı sınırlı olabilen ham değer veya gözlemdir.	95, 'Ayşe', 12.05.2026
Bilgi	Information	Verilerin anlamlı biçimde düzenlenmesi ve yorumlanmasıyla elde edilen, karar vermeyi destekleyen çıktıdır.	Öğrencinin not ortalamasının 85 olması.
Veri tabanı	Database	Birbiriyle ilişkili verilerin belirli bir amaç için düzenli, kalıcı ve erişilebilir biçimde saklandığı yapıdır.	Üniversitenin öğrenci, ders ve not kayıtları.
Veri tabanı yönetim sistemi (VTYS)	Database Management System (DBMS)	Veri tabanını tanımlama, oluşturma, sorgulama, güncelleme, koruma ve yönetme işlevlerini sağlayan yazılım sistemidir.	MySQL, Oracle, SQL Server.
İlişkisel VTYS	Relational DBMS (RDBMS)	Verileri tablolar ve tablolar arasındaki anahtar ilişkileriyle yöneten VTYS türüdür.	PostgreSQL, MySQL, Oracle, SQL Server.
Veri tabanı sistemi	Database System	Veri tabanı, VTYS, uygulamalar, kullanıcılar ve bunların birlikte çalışmasını sağlayan bileşenlerin tümüdür.	Yalnızca DBMS yazılımından ibaret değildir.
Veri tabanını tanımlama	Database Definition	Veri yapılarının, alanların, veri tiplerinin ve kısıtların belirlenmesidir.	ÖğrenciNo alanının INT olarak tanımlanması.
Veri tabanını oluşturma	Database Creation	Tanımlanan veri yapılarının fiziksel ortamda oluşturulması ve verilerin saklanabilir hâle getirilmesidir.	CREATE DATABASE / CREATE TABLE.
Veri tabanı üzerinde işlem	Database Operations	Veriyi ekleme, okuma, güncelleme, silme ve raporlama işlemleridir.	CRUD işlemleri.
Bakım ve süreklilik	Maintenance	Veri tabanının güncel, erişilebilir ve işletilebilir kalmasını sağlayan yönetim faaliyetleridir.	Yedekleme, onarma, geri yükleme.
Veri bağımsızlığı	Data Independence	Uygulamaların, verinin fiziksel saklanma ayrıntılarındaki değişikliklerden mümkün olduğunca etkilenmemesidir.	Diskteki saklama düzeni değişse de uygulama aynı mantıksal veriyi kullanabilir.
Veri tekrarı	Data Redundancy	Aynı bilginin gereksiz biçimde birden fazla yerde tutulmasıdır.	Aynı müşteri adresinin çok sayıda dosyada kopyalanması.
Veri tutarsızlığı	Data Inconsistency	Tekrarlanan verilerin farklı kopyalarında birbiriyle çelişen değerlerin bulunmasıdır.	Bir dosyada adres Ankara, diğerinde İzmir.
Veri paylaşımı	Data Sharing	Yetkili kullanıcıların ortak veri kümesine erişebilmesidir.	Öğrenci işleri ve bölüm yönetiminin yetkileri ölçüsünde aynı kayıtlara erişmesi.
Veri bütünlüğü	Data Integrity	Verilerin doğruluk, geçerlilik ve tutarlılık kurallarını korumasıdır.	Not alanının 0–100 aralığında olması.
Veri güvenliği	Data Security	Verilerin yetkisiz erişim, değiştirme, ifşa veya kayıptan korunmasıdır.	Kullanıcı yetkileri, parola ve erişim denetimi.
Gizlilik	Confidentiality / Privacy	Verinin yalnızca erişim yetkisi olan kişi veya süreçlerce görülmesini sağlamaya yönelik ilkedir.	Bir personelin yalnız kendi biriminin kayıtlarını görmesi.
Yedekleme	Backup	Verinin kayıp veya bozulma durumunda geri kazanılabilmesi için kopyasının alınmasıdır.	Günlük tam yedek.
Geri yükleme	Restore / Recovery	Yedek veya kurtarma kayıtlarından veri tabanını önceki çalışır duruma döndürme işlemidir.	Sunucu arızası sonrası yedekten dönüş.

2. Veri modelleri ve ER (Varlık-İlişki) modelleme

ER modeli, uygulama ve VTYS seçimine bağlı kalmadan gerçek dünyadaki nesneleri, özellikleri ve aralarındaki ilişkileri çözümlemek için kullanılan kavramsal modelleme yaklaşımıdır. Ders dokümanında ER diyagramlarının daha sonra ilişkisel şemaya dönüştürülmesi vurgulanır.

Terim	İngilizce karşılık	Tanım	Örnek / gösterim
Veri modeli	Data Model	Verinin hangi yapılar ve kurallarla temsil edileceğini belirleyen soyutlama.	Hiyerarşik, ilişkisel, ER modeli.
Hiyerarşik model	Hierarchical Model	Verileri ağaç biçiminde düzenler; bir alt düğümün tipik olarak tek üst düğümü bulunur.	Genel Müdür → Müdür Yardımcısı → Bölüm Şefi.
İlişkisel model	Relational Model	Verileri satır ve sütunlardan oluşan tablolarla temsil eder.	ÖĞRENCİ(ÖğrenciNo, Ad, Soyad).
ER modeli	Entity-Relationship Model	Varlıkları, nitelikleri ve varlıklar arasındaki ilişkileri kavramsal olarak gösterir.	ÖĞRENCİ — ALIR — DERS.
Varlık	Entity	Hakkında veri tutulabilen, diğerlerinden ayırt edilebilen nesne veya olaydır.	Bir öğrenci, kitap, satış veya sınav.
Varlık kümesi	Entity Set	Aynı türdeki varlıkların oluşturduğu kümedir.	Öğrenciler, dersler, personeller.
Nitelik	Attribute	Bir varlığın veya ilişkinin özellik bilgisidir.	Öğrenci için ad, soyad, doğum tarihi.
Etki alanı	Domain	Bir niteliğin alabileceği geçerli değerler kümesidir.	Not niteliği için 0–100.
Türetilen nitelik	Derived Attribute	Başka bir veya birkaç nitelikten hesaplanabilen niteliktir.	Doğum tarihinden hesaplanan yaş.
Birleşik nitelik	Composite Attribute	Alt bileşenlerine ayrılabilen niteliktir.	Adres: şehir, ilçe, mahalle, sokak.
Anahtar nitelik	Key Attribute	Varlıkları benzersiz biçimde ayırt etmek için kullanılan niteliktir.	ÖğrenciNo.
İlişki	Relationship	İki veya daha fazla varlık arasındaki anlamlı bağlantıdır.	Öğrenci dersi alır.
İlişki kümesi	Relationship Set	Aynı türdeki ilişkilerin oluşturduğu kümedir.	Öğrencilerin aldıkları ders ilişkileri.
İlişki rolü	Role	Bir varlığın ilişki içindeki işlevini belirtir.	Personel-yönetici ilişkisinde üst ve ast.
İlişki niteliği	Relationship Attribute	İlişkinin kendisine ait özellik bilgisidir.	Öğrenci-Ders ALIR ilişkisinde kayıt tarihi veya not.
İkili ilişki	Binary Relationship	İki varlık kümesini ilişkilendirir.	Öğrenci — Ders.
Üçlü ilişki	Ternary Relationship	Üç varlık kümesini aynı ilişki içinde birleştirir.	İşçi, ürün ve makine arasındaki üretim ilişkisi.
Soyutlama	Abstraction	Gereksiz ayrıntıları dışarıda bırakıp problem için önemli özelliklere odaklanmadır.	Öğrenci modelinde saç rengi gerekli değilse tutulmaması.
Kısıt	Constraint	Verinin alabileceği değerleri veya ilişkilerin geçerliliğini sınırlar.	Not 0–100; öğrenci numarası benzersiz.

2.1 ER diyagramı sembolleri

Sembol / gösterim	Anlamı	Öğrencinin hatırlaması gereken
Dikdörtgen	Varlık kümesi	ÖĞRENCİ, DERS gibi nesne kümeleri.
Çift çizgili dikdörtgen	Zayıf varlık kümesi	Kimliği tek başına yeterli anahtarla belirlenemeyen varlık kümesi.
Elips	Nitelik	Ad, soyad, doğum tarihi gibi özellik.
Altı çizili elips / nitelik	Anahtar nitelik	Varlığı benzersiz ayırt eden özellik.
Kesikli elips	Türetilen nitelik	Başka nitelikten hesaplanan değer; ör. yaş.
Baklava / eşkenar dörtgen	İlişki kümesi	ALIR, OKUR, YÖNETİR gibi bağlantı.
İlişki çizgisi üzerindeki 1, n, m	Kardinalite / çokluk	İlişkinin kaç varlığı eşleştirebildiği.

2.2 Varlık türleri ve anahtar mantığı

Terim	İngilizce	Tanım	Örnek
Güçlü varlık	Strong Entity	Kendi nitelikleriyle benzersiz bir anahtar oluşturabilen varlıktır.	Üniversite genelinde benzersiz ÖğrenciNo ile öğrenci.
Zayıf varlık	Weak Entity	Kendi nitelikleri tek başına benzersiz kimlik sağlamayan; tanımlayıcı sahibi varlığa ihtiyaç duyan varlıktır.	Farklı liselerde aynı öğrenci numarası bulunabilmesi.
Tanımlayıcı ilişki	Identifying Relationship	Zayıf varlığı sahibi/güçlü varlığa bağlayan ilişkidir.	ÖĞRENCİ, LİSE tarafından tanımlanır.
Bileşik anahtar	Composite Key	Birden fazla niteliğin birlikte benzersiz kimlik oluşturmasıdır.	(LiseKodu, ÖğrenciNo).

3. İlişki türleri ve kardinalite

Kardinalite, bir taraftaki tek bir varlığın diğer tarafta kaç varlıkla ilişkilendirilebildiğini anlatır. İlişki yönü belirtilmeden yalnızca 'bir-çok' demek bazen belirsizlik yaratır; bu nedenle her iki tarafın çokluğu birlikte okunmalıdır.

İlişki türü	İngilizce	Anlamı	Örnek	Tabloya dönüşümde temel yaklaşım
Bire bir (1:1)	One-to-One	A tarafındaki bir kayıt en fazla bir B kaydıyla; B tarafındaki bir kayıt da en fazla bir A kaydıyla eşleşir.	Bir bölümün tek yöneticisi ve yöneticinin tek bölüm yönetmesi varsayımı.	Uygun tablonun anahtarını diğer tabloya FK olarak eklemek; ilişki niteliğini de uygun tarafa taşımak.
Bire çok (1:N)	One-to-Many	A tarafındaki bir kayıt birçok B kaydıyla; B tarafındaki her kayıt bir A kaydıyla ilişkilidir.	Bir bölümde birçok öğrenci bulunur; öğrenci bir bölüme bağlıdır.	1 tarafının PK alanı N tarafındaki tabloya FK olarak eklenir.
Çoka bir (N:1)	Many-to-One	N tarafındaki birçok kayıt tek bir A kaydına bağlanır; aynı ilişkinin ters yönden görünümüdür.	Birçok öğrenci aynı bölüme bağlıdır.	N tarafındaki tabloda 1 tarafının PK'sı FK olarak bulunur.
Çoka çok (M:N)	Many-to-Many	Her iki taraftaki bir kayıt karşı tarafta birden fazla kayıtla eşleşebilir.	Bir öğrenci birçok ders alır; bir ders birçok öğrenci tarafından alınır.	Araya ilişki/köprü tablosu konur; iki tarafın PK alanları FK olur.

3.1 İlişkilerde kritik dönüşüm örnekleri

ER örneği	İlişkisel tasarım fikri	Dikkat
BÖLÜM (1) — ÖĞRENCİ (N)	ÖĞRENCİ tablosuna BölümNo yabancı anahtarı eklenir.	FK, çok olan tarafta yer alır.
ÖĞRENCİ (M) — DERS (N)	ÖĞRENCİ_DERS(ÖğrenciNo, DersKodu, …) ara tablosu kurulur.	Not, kayıt tarihi gibi ilişki nitelikleri ara tabloda tutulabilir.
PERSONEL (1) — BÖLÜM (1)	Uygun tasarım tercihine göre FK taraflardan birine eklenir.	Bire bir ilişki için benzersizlik ve NULL gereksinimi de düşünülmelidir.
ZAYIF VARLIK — SAHİP VARLIK	Sahip varlığın anahtarı zayıf varlığın anahtarına katılır.	Örnek: (LiseKodu, ÖğrenciNo).

4. İlişkisel veri tabanı terminolojisi

Terim	İngilizce karşılık	Tanım	Örnek / ipucu
İlişki (matematiksel)	Relation	İlişkisel modelde belirli niteliklerden oluşan tabloyu ifade eder.	DERS(DersKodu, DersAdi, Kredi).
Tablo	Table	İlişkisel verinin satır ve sütunlar hâlinde saklandığı yapıdır.	ÖĞRENCİ tablosu.
Satır / kayıt	Row / Record / Tuple	Tablodaki tek bir veri örneğidir.	2024001, Ayşe, Yılmaz.
Sütun / alan	Column / Field / Attribute	Her kayıtta aynı türden bilgiyi temsil eden tablonun dikey bileşenidir.	ÖğrenciNo, Ad, Soyad.
İlişkisel şema	Relational Schema	İlişki adları ve bunların niteliklerini gösteren mantıksal tanımdır.	OGRENCI(OgrenciNo, Ad, Soyad).
Veri tabanı şeması	Database Schema	Veri tabanındaki tabloların, alanların, ilişkilerin ve kısıtların yapısal tanımıdır.	Tablolar + anahtarlar + kısıtlar.
Kayıt örneği / örneklem	Instance	Şemanın belirli bir andaki gerçek veri içeriğidir.	Öğrenci tablosundaki mevcut satırlar.
Birincil anahtar	Primary Key (PK)	Tablodaki her satırı benzersiz tanımlayan seçilmiş anahtar alan veya alan grubudur.	ÖğrenciNo.
Yabancı anahtar	Foreign Key (FK)	Bir tablodaki alanın başka tablonun anahtarına (veya uygun benzersiz anahtarına) referans vermesidir.	ÖĞRENCİ.BölümNo → BÖLÜM.BölümNo.
Aday anahtar	Candidate Key	Bir kaydı benzersiz tanımlayan ve gereksiz nitelik içermeyen minimal anahtar seçeneğidir.	ÖğrenciNo veya benzersiz e-posta.
Süper anahtar	Superkey	Bir satırı benzersiz ayırt eden bir veya daha fazla niteliktir; gereksiz alan içerebilir.	ÖğrenciNo + Ad + Soyad.
Alternatif anahtar	Alternate Key	Aday anahtarlardan birincil anahtar seçilmeyen diğeridir.	PK ÖğrenciNo ise benzersiz e-posta alternatif anahtar olabilir.
Bileşik anahtar	Composite Key	Birden fazla sütunun birlikte anahtar olmasıdır.	ÖğrenciNo + DersKodu.
Otomatik artan anahtar	Auto-increment / Identity	Yeni kayıt eklenirken sistemin otomatik değer ürettiği kimlik alanıdır.	SQL Server IDENTITY; MySQL AUTO_INCREMENT.
NULL	NULL	Bir alan için bilinmeyen, mevcut olmayan veya uygulanamaz değer durumudur; sıfır ya da boş metinle aynı değildir.	Telefon bilgisi henüz girilmemiş.
Benzersizlik	Uniqueness	Bir alan veya alan grubundaki değerlerin tekrar etmemesi kuralıdır.	UNIQUE e-posta.
Referans bütünlüğü	Referential Integrity	Yabancı anahtarların geçerli bir referansa işaret etmesini sağlayan kuraldır.	Olmayan BölümNo ile öğrenci kaydı açılmaması.

5. Veri tipleri ve alan tanımlama

Veri tipi, bir sütunda hangi tür değerlerin saklanabileceğini ve bu değerlerin nasıl yorumlanacağını belirler. Kaynak doküman Access, MySQL, Oracle ve SQL Server veri tiplerini ayrı başlıklarda ele alır. Aşağıdaki tablolar, bu sınıflandırmayı kavramsal olarak bir araya getirir.

Veri tipi ailesi	Ne saklar?	Yaygın örnekler	Dikkat edilmesi gereken
Tam sayı	Kesir içermeyen sayılar	TINYINT, SMALLINT, INT, BIGINT	Değer aralığı ve kapladığı alan seçilmelidir.
Kesin ondalıklı sayı	Belirli basamak hassasiyetindeki sayılar	DECIMAL(p,s), NUMERIC(p,s)	Para ve hassas ölçümlerde genellikle uygun seçimdir.
Yaklaşık ondalıklı sayı	Kayan noktalı yaklaşık değerler	FLOAT, REAL, DOUBLE	İkili gösterim nedeniyle bazı ondalık değerler tam temsil edilmeyebilir.
Karakter - sabit uzunluk	Uzunluğu sabit veya sabite yakın metin	CHAR(n), NCHAR(n)	Kısa kodlar gibi uzunluğu belli değerler.
Karakter - değişken uzunluk	Uzunluğu değişen metin	VARCHAR(n), NVARCHAR(n), TEXT	İhtiyaç duyulan uzunluk ve Unicode desteği dikkate alınır.
Tarih ve saat	Takvim ve zaman bilgileri	DATE, TIME, DATETIME, TIMESTAMP	VTYS'ye göre kapsam ve davranış farklı olabilir.
Mantıksal	İki durumlu değer	BIT, BOOLEAN, YES/NO	Evet/hayır, aktif/pasif gibi.
Parasal	Para değerleri	MONEY, SMALLMONEY, DECIMAL	Para birimi ve hassasiyet tasarımı ayrıca düşünülmelidir.
İkili veri	Metin olmayan bayt dizileri	BINARY, VARBINARY, BLOB	Görsel, ses, belge gibi içerikler.
Özel / kimlik	Özel amaçlı değerler	UNIQUEIDENTIFIER, GUID, ROWID, ENUM, SET	VTYS'ye özgü türlerdir.

5.1 Kaynakta geçen VTYS ve veri tipi örnekleri

VTYS / ortam	Kaynakta geçen türler	Kavramsal karşılığı
MS Access	Text, Memo, Number, Date/Time, Currency, AutoNumber, Yes/No, OLE Object, Hyperlink, Lookup Wizard	Metin, uzun metin, sayı, tarih-saat, para, otomatik kimlik, mantıksal, nesne ve bağlantı alanları.
MySQL	CHAR, VARCHAR, TINYTEXT, TEXT, BLOB, MEDIUMTEXT, LONGTEXT, TINYINT, SMALLINT, INT, BIGINT, FLOAT, DOUBLE, DECIMAL, DATE, DATETIME, TIMESTAMP, TIME, ENUM, SET	Metin/ikili veri, sayısal, tarih-saat ve özel liste/küme türleri.
Oracle	CHAR, DATE, NUMBER, VARCHAR2, LONG, LONG RAW, RAW, ROWID	Sabit/değişken metin, tarih, sayısal, uzun metin/ikili ve satır tanımlayıcı türleri.
MS SQL Server	BIGINT, INT, SMALLINT, TINYINT, DECIMAL, NUMERIC, FLOAT, REAL, MONEY, SMALLMONEY, DATETIME, SMALLDATETIME, CHAR, NCHAR, VARCHAR, NVARCHAR, BINARY, VARBINARY, BIT, UNIQUEIDENTIFIER, XML	Kesin/yaklaşık sayısal, parasal, tarih-saat, karakter, ikili ve özel amaçlı türler.

5.2 Sık geçen veri tipi parametreleri

Gösterim	Anlamı	Örnek yorum
CHAR(n)	Sabit uzunluklu karakter alanı; n uzunluğu belirtir.	CHAR(5) kod alanı için 5 karakterlik sabit uzunluk.
VARCHAR(n)	En fazla n karakterlik değişken uzunluklu metin.	VARCHAR(80) en fazla 80 karakterlik ad/adres alanı.
DECIMAL(p,s)	p toplam basamak, s ondalık basamak sayısıdır.	DECIMAL(7,2): toplam 7 basamak, 2 ondalık basamak.
FLOAT / REAL	Yaklaşık kayan noktalı sayı türleri.	Bilimsel ölçüm ve yaklaşık hesaplamalar.
NCHAR / NVARCHAR	Unicode karakterleri destekleyen karakter türleri.	Çok dilli metinlerde karakter kümesi desteği önemlidir.

6. SQL: dil, komut aileleri ve sorgulama

SQL (Structured Query Language), ilişkisel veri tabanlarında şema tanımlamak, veri eklemek/değiştirmek/silmek ve sorgulamak için kullanılan standartlaştırılmış sorgulama dilidir. Kaynakta SELECT, INSERT, UPDATE, DELETE ile CREATE, ALTER ve DROP komutları ayrıntılı örneklerle ele alınır.

Terim / komut	Sınıf / İngilizce	İşlev	Örnek
SQL	Structured Query Language	İlişkisel veri tabanlarında veri ve şema işlemlerini ifade eden dildir.	SELECT * FROM Ogrenci;
DDL	Data Definition Language	Veri tabanı nesnelerinin yapısını tanımlar veya değiştirir.	CREATE, ALTER, DROP.

Terim / komut	Sınıf / İngilizce	İşlev	Örnek
DML	Data Manipulation Language	Tablolardaki verileri ekler, değiştirir veya siler.	INSERT, UPDATE, DELETE.
DQL	Data Query Language	Veri sorgulama için kullanılan sınıflandırma adıdır.	SELECT (bazı kaynaklarda DML kapsamında da ele alınır).
CREATE	DDL	Veri tabanı veya tablo gibi nesneleri oluşturur.	CREATE TABLE Ogrenci (...);
ALTER	DDL	Mevcut nesnenin yapısını değiştirir.	ALTER TABLE Ogrenci ADD Eposta VARCHAR(100);
DROP	DDL	Veri tabanı nesnesini kaldırır.	DROP TABLE Ogrenci;
SELECT	Sorgulama	Tablodan istenen sütun ve satırları getirir.	SELECT Ad, Soyad FROM Ogrenci;
INSERT	DML	Tabloya yeni kayıt ekler.	INSERT INTO Ogrenci (No, Ad) VALUES (1, 'Ece');
UPDATE	DML	Var olan kayıtların alan değerlerini günceller.	UPDATE Ogrenci SET Ad='Ece' WHERE No=1;
DELETE	DML	Koşula uyan kayıtları siler.	DELETE FROM Ogrenci WHERE No=1;
FROM	SELECT bileşeni	Sorgunun hangi tablo veya tablolardan veri alacağını belirtir.	FROM Ogrenci
WHERE	Koşul	Satırları koşula göre filtreler.	WHERE BolumNo=10
DISTINCT	Tekilleştirme	Sonuç kümesindeki tekrar eden satırları kaldırır.	SELECT DISTINCT Sehir FROM Ogrenci;
GROUP BY	Gruplama	Aynı grup değerlerine sahip kayıtları bir araya getirir.	GROUP BY BolumNo
HAVING	Grup koşulu	Gruplanmış sonuçlara koşul uygular.	HAVING COUNT(*) > 10
ORDER BY	Sıralama	Sonuçları bir veya daha fazla sütuna göre sıralar.	ORDER BY Soyad ASC
ASC	Ascending	Artan sıralama yönüdür.	A'dan Z'ye / küçükten büyüğe.
DESC	Descending	Azalan sıralama yönüdür.	Z'den A'ya / büyükten küçüğe.
INTO	Hedef tablo	Sorgu sonucunu yeni tabloya aktarma amacıyla kullanılabilir.	SELECT ... INTO YeniTablo ...

6.1 SQL koşul ve karşılaştırma operatörleri

Operatör / ifade	Anlamı	Örnek kullanım	Not
=, >, <, >=, <=, <>	Eşitlik, büyüktür, küçüktür, büyük/eşit, küçük/eşit, eşit değildir.	WHERE Not >= 60	<> standart eşit değildir gösterimidir; bazı sistemlerde != de desteklenir.
AND	Her iki koşul da doğru olmalıdır.	WHERE Bolum='Yazılım' AND Sinif=2	Koşulların kesişimi.
OR	Koşullardan en az biri doğru olmalıdır.	WHERE Sehir='Elazığ' OR Sehir='Malatya'	Koşulların birleşimi.
NOT	Koşulun mantıksal tersini alır.	WHERE NOT Sehir='Elazığ'	NULL mantığı ayrıca değerlendirilmelidir.
BETWEEN	Belirtilen aralıktaki değerleri seçer; uç değerler dahildir.	WHERE Not BETWEEN 50 AND 80	50 ve 80 de kapsanır.
IN	Bir listedeki değerlerden biriyle eşleşmeyi arar.	WHERE Sehir IN ('Elazığ','Ankara')	Birden çok OR koşuluna alternatif.
LIKE	Metin desenine göre eşleşme arar.	WHERE Ad LIKE 'A%'	% sıfır veya daha çok karakter; _ tek karakter jokeridir.
IS NULL	NULL değerleri seçer.	WHERE Telefon IS NULL	NULL için = NULL kullanılmaz.
IS NOT NULL	NULL olmayan değerleri seçer.	WHERE Telefon IS NOT NULL	Boş metinle aynı değildir.

6.2 Joker karakterler ve LIKE

Desen	Anlam	Örnek
'A%'	A harfiyle başlayan metinler	Ali, Ayşe, Aslı
'%A'	A harfiyle biten metinler	Zehra, Sema
'%veri%'	İçinde 'veri' geçen metinler	Veri tabanı, büyük veri
'A_e'	A ile başlayan, üçüncü karakteri e olan ve arada tek karakter bulunan desen (SQL Server gibi standart LIKE kullanımında _ tek karakterdir).	Ali desen eşleşmesi için örnek değildir; 'Abe' gibi.

6.3 SQL toplama ve özet fonksiyonları

Fonksiyon	İşlev	Örnek
COUNT()	Satır veya NULL olmayan değer sayısını hesaplar.	SELECT COUNT(*) FROM Ogrenci;
SUM()	Sayısal sütundaki değerleri toplar.	SELECT SUM(Bakiye) FROM Musteri;
AVG()	Sayısal sütunun aritmetik ortalamasını hesaplar.	SELECT AVG(Not) FROM Sinav;
MIN()	En küçük değeri döndürür.	SELECT MIN(Fiyat) FROM Urun;
MAX()	En büyük değeri döndürür.	SELECT MAX(Fiyat) FROM Urun;

Not: Kaynak dokümanda MAX/MIN ile başka sütunların birlikte seçildiği bazı örnekler, GROUP BY veya alt sorgu olmadan her VTYS'de geçerli değildir. Öğrenciler agregasyon fonksiyonlarının tek başına bir satırın diğer alanlarını otomatik olarak getirmediğini bilmelidir.

7. VTYS yazılımları ve araç ekosistemi

Araç / ürün	Türü	Sözlük açıklaması	Ders dokümanındaki bağlam
Microsoft Access	Masaüstü ilişkisel veri tabanı uygulaması / VTYS	Küçük ölçekli veri tabanı oluşturma ve yönetme ortamı.	Access veri tipleri: Text, Memo, Number, AutoNumber vb.
MySQL	İlişkisel VTYS	Açık kaynak ekosisteminde yaygın kullanılan ilişkisel veri tabanı sunucusu.	MySQL veri tipleri ve SQL sorguları.
PostgreSQL	İlişkisel VTYS	Açık kaynaklı, gelişmiş özelliklere sahip ilişkisel veri tabanı sistemi.	Kaynakta PostgreSQL adı geçer.
Oracle Database	İlişkisel VTYS	Kurumsal uygulamalarda kullanılan ilişkisel veri tabanı sistemi.	Oracle veri tipleri; NUMBER, VARCHAR2, ROWID.
Microsoft SQL Server	İlişkisel VTYS	Microsoft'un ilişkisel veri tabanı yönetim sistemi.	T-SQL ve SQL Server veri tipleri.
Sybase	İlişkisel VTYS ailesi	Kurumsal veri tabanı ürünleriyle bilinen VTYS ailesi.	Kaynakta T-SQL desteğiyle anılır.
Informix	İlişkisel VTYS	Kurumsal veri tabanı yönetim sistemi.	Kaynakta orta/büyük ölçekli sistem olarak listelenir.
SQL istemcisi / yönetim aracı	Client / Administration Tool	Veri tabanına bağlanma, sorgu çalıştırma, şema inceleme ve yönetim için kullanılan arayüzdür.	Belirli bir ürün adı kaynakta ayrıntılandırılmamıştır.
SQL editörü	SQL Editor	SQL komutlarının yazılıp çalıştırıldığı araç bölümü.	SELECT, CREATE, UPDATE sorgularını yürütme.
Veri tabanı sunucusu	Database Server	Veri tabanı hizmetini sağlayan ve veri dosyalarını yöneten sunucu ortamıdır.	İstemcilerden gelen sorguları işler.
Veri tabanı istemcisi	Database Client	Sunucuya bağlanarak sorgu gönderen kullanıcı uygulaması veya yazılım bileşenidir.	Yönetim aracı veya uygulama.

8. Öğrencilerin en çok karıştırabileceği kavramlar

Karıştırılan kavramlar	Doğru ayrım	Kısa hatırlatma / örnek
Veri – Bilgi	Veri ham girdidir; bilgi işlenmiş ve anlamlandırılmış çıktıdır.	'85' veri; 'öğrencinin notu 85' bağlamlı bilgidir.
Veri tabanı – VTYS	Veri tabanı saklanan düzenli veri kümesidir; VTYS onu yöneten yazılımdır.	Veri tabanı = içerik; VTYS = yönetim sistemi.
VTYS – RDBMS	RDBMS, VTYS'nin ilişkisel modeli kullanan türüdür.	Her VTYS ilişkisel olmak zorunda değildir.
ER modeli – İlişkisel model	ER modeli kavramsal çözümleme aracıdır; ilişkisel model tablolarla uygulama yapısını temsil eder.	ER diyagramı tasarım; tablolar ilişkisel temsil.
Varlık – Varlık kümesi	Varlık tek bir örnektir; varlık kümesi aynı türden örneklerin tümüdür.	Ayşe = varlık; Öğrenciler = varlık kümesi.
Nitelik – Etki alanı	Nitelik özelliğin adıdır; etki alanı alabileceği değerler kümesidir.	Not = nitelik; 0–100 = domain.
Türetilen – Birleşik nitelik	Türetilen başka veriden hesaplanır; birleşik alt parçalara ayrılır.	Yaş türetilir; adres bileşenlerine ayrılır.
İlişki – İlişki kümesi	İlişki tek bağlantı örneği; ilişki kümesi aynı tür bağlantıların kümesidir.	Ali'nin Matematik alması / tüm öğrenci-ders alma bağlantıları.
Bire çok – Çoka bir	Aynı ilişkinin karşı yönlerden anlatımıdır; hangi taraftan bakıldığı önemlidir.	Bölümden öğrenciye 1:N; öğrenciden bölüme N:1.
Çoka çok – Bire çok	Çoka çokta iki taraf da çoklu olabilir; bire çokta yalnız bir taraf çokludur.	Öğrenci-Ders M:N; Bölüm-Öğrenci 1:N.
PK – FK	PK tablonun kendi satırını tanımlar; FK başka tablodaki anahtara referans verir.	ÖğrenciNo PK; BölümNo FK.
Aday anahtar – Süper anahtar	Aday anahtar minimal benzersiz tanımlayıcıdır; süper anahtar gereksiz alanlar içerebilir.	ÖğrenciNo aday; ÖğrenciNo+Ad süper anahtar.
Bileşik anahtar – Bileşik nitelik	Bileşik anahtar birden çok alanla kimlik kurar; bileşik nitelik alt parçalara ayrılır.	(ÖğrenciNo,DersKodu) anahtar; Adres alt bileşenleri.
Satır – Sütun	Satır tek kayıttır; sütun kayıtların ortak özelliğidir.	Ayşe'nin kaydı satır; Ad sütunu.
Şema – Veri örneği	Şema yapıyı; instance belirli andaki içerikleri gösterir.	Tablo tanımı / tablonun mevcut satırları.
CHAR – VARCHAR	CHAR sabit uzunluk; VARCHAR değişken uzunluklu metin için kullanılır.	Sabit kod / değişken uzunluktaki şehir adı.
INT – DECIMAL	INT tam sayı; DECIMAL kesin basamaklı ondalık/sayısal değer saklar.	Adet = INT; fiyat = DECIMAL.
FLOAT – DECIMAL	FLOAT yaklaşık kayan nokta; DECIMAL belirli hassasiyetli ondalık gösterimdir.	Finansal tutarlarda DECIMAL çoğu zaman daha uygundur.
NULL – 0 – boş metin	NULL bilinmeyen/eksik/uygulanamaz; 0 sayısal sıfır; '' boş metindir.	Üçü birbirinin yerine kullanılamaz.
DELETE – DROP	DELETE tablodaki satırları siler; DROP tablo nesnesini kaldırır.	DELETE FROM Ogrenci; / DROP TABLE Ogrenci;
ALTER – UPDATE	ALTER tablo yapısını değiştirir; UPDATE mevcut satır değerlerini değiştirir.	Sütun ekleme / maaş güncelleme.
WHERE – HAVING	WHERE satırları gruplamadan önce filtreler; HAVING gruplar üzerinde koşul uygular.	WHERE BolumNo=1; HAVING COUNT(*)>5.

Karıştırılan kavramlar	Doğru ayrım	Kısa hatırlatma / örnek
ORDER BY – GROUP BY	ORDER BY sıralar; GROUP BY gruplar.	Listeyi sırala / aynı bölümleri grupla.
DISTINCT – GROUP BY	DISTINCT yinelenen sonuç satırlarını ayıklar; GROUP BY gruplama ve agregasyon için kullanılır.	Tekil şehir listesi / şehir başına öğrenci sayısı.
LIKE – IN	LIKE metin desenini arar; IN listeden değer eşleştirir.	Ad LIKE 'A%'; Şehir IN ('Elazığ','Ankara').
COUNT(*) – COUNT(kolon)	COUNT(*) satırları sayar; COUNT(kolon) NULL olmayan alan değerlerini sayar.	NULL bulunan sütunda sonuçlar farklı olabilir.
Yedekleme – Geri yükleme	Yedekleme kopya oluşturur; geri yükleme bu kopyadan sistemi döndürür.	Backup alma / restore etme.

9. Uçtan uca örnek: Üniversite veri tabanını tasarlama

Aşağıdaki mini senaryo, dokümandaki üniversite, öğrenci, ders ve not örneklerini tek bir tasarım akışında birleştirir. Amaç, terimlerin birbirine nasıl bağlandığını göstermektir.

Aşama	Uygulama	İlgili terminoloji
1. Gereksinim	Üniversite öğrencileri, bölümleri, dersleri ve öğrencilerin ders notları tutulacaktır.	Veri tabanı, gereksinim, veri.
2. Kavramsal model	ÖĞRENCİ, BÖLÜM ve DERS varlık kümeleri belirlenir; öğrenci bölüme bağlıdır ve ders alır.	ER modeli, varlık, nitelik, ilişki.
3. Kardinalite	Bir bölümde çok öğrenci; öğrenciler çok ders alabilir ve her ders çok öğrenci tarafından alınabilir.	1:N ve M:N.
4. Anahtarlar	ÖğrenciNo, BölümNo ve DersKodu benzersiz tanımlayıcı olarak seçilir.	PK, aday anahtar.
5. İlişkisel dönüşüm	ÖĞRENCİ tablosuna BölümNo FK eklenir; M:N için ÖĞRENCİ_DERS ara tablosu oluşturulur.	Şema, FK, köprü tablo, bileşik anahtar.
6. Veri tipleri	ÖğrenciNo INT, Ad VARCHAR, DoğumTarihi DATE, Not DECIMAL gibi türler seçilir.	Data type, domain, NULL.
7. Veri tanımlama	CREATE TABLE ile tablolar ve kısıtlar oluşturulur.	SQL, DDL, CREATE, NOT NULL, PRIMARY KEY.
8. Veri girişi	INSERT ile öğrenci ve ders kayıtları eklenir.	DML, INSERT.
9. Sorgulama	SELECT, WHERE, ORDER BY ile öğrenciler filtrelenir ve sıralanır.	SELECT, FROM, WHERE, ORDER BY.
10. Analiz	AVG, MIN, MAX ve GROUP BY ile ders bazında not istatistikleri hesaplanır.	Agregasyon, GROUP BY, HAVING.

10. Hızlı tekrar: SQL komutlarının etkisi

Komut	Etkilediği şey	Örnek etki	Dikkat
CREATE	Veri tabanı nesnesi	Yeni tablo oluşturur.	Yapı oluşturur; kayıt ekleme komutu değildir.
ALTER	Nesnenin yapısı	Tabloya sütun ekler/değiştirir.	Satır içeriğini güncellemez.
DROP	Nesnenin kendisi	Tabloyu kaldırır.	Tablo yapısı ve içerik nesneyle birlikte ortadan kalkar.
SELECT	Sorgu sonucu	Kayıtları getirir.	Tek başına veri değiştirmez.
INSERT	Tablo verisi	Yeni satır ekler.	Sütun-değer eşleşmesi doğru olmalıdır.
UPDATE	Tablo verisi	Var olan satırları değiştirir.	WHERE unutulursa tüm satırlar etkilenebilir.
DELETE	Tablo verisi	Satırları siler.	WHERE unutulursa tüm satırlar silinebilir.

11. Kaynak dokümandaki konu kapsamı ve sayfa eşlemesi

Kaynak sayfaları	Konu başlığı	Bu sözlükteki karşılığı
1–6	Veri, bilgi, veri tabanı, VTYS, avantajlar, yazılımlar	Bölüm 1 ve 7
7–20	ER modeli, varlıklar, nitelikler, ilişkiler, kısıtlar, anahtarlar	Bölüm 2 ve 3
21	İlişkisel şema ve ilişkisel tablo	Bölüm 4
22–28	Access, MySQL, Oracle ve SQL Server veri tipleri	Bölüm 5
28–31	ER şemasının tablolara dönüştürülmesi	Bölüm 3.1 ve 9
32–40	SQL, SELECT, filtreleme, sıralama ve özet fonksiyonları	Bölüm 6
40–46	CREATE, ALTER, DROP, SELECT, INSERT, UPDATE, DELETE	Bölüm 6 ve 10

12. Öğrenme ve değerlendirme için önerilen kontrol listesi

- ☐ Veri tabanı ile VTYS arasındaki farkı kendi cümlelerimle açıklayabiliyorum.
- ☐ Bir ER diyagramında varlık, nitelik, anahtar nitelik ve ilişkiyi ayırt edebiliyorum.
- ☐ 1:1, 1:N ve M:N ilişkileri örnekleyebiliyorum.
- ☐ M:N ilişkiyi ara tabloya dönüştürebiliyorum.
- ☐ PK, FK, aday anahtar ve süper anahtar arasındaki farkı açıklayabiliyorum.
- ☐ Bir alan için uygun veri tipini ve etki alanını seçebiliyorum.
- ☐ SELECT sorgusunda FROM, WHERE, GROUP BY, HAVING ve ORDER BY görevlerini ayırt edebiliyorum.
- ☐ CREATE, ALTER, DROP, INSERT, UPDATE ve DELETE komutlarının etkisini biliyorum.
- ☐ NULL, 0 ve boş metnin farklı durumlar olduğunu biliyorum.
- ☐ UPDATE veya DELETE sorgularında WHERE koşulunun önemini açıklayabiliyorum.

Editoryal not ve kapsam sınırı

Bu sözlük, ders materyalindeki kavramları esas alır. Kaynakta ayrıntılı biçimde işlenmeyen normalizasyon, indeksleme, transaction/ACID, JOIN türleri, view, stored procedure, trigger, yetkilendirme komutları ve NoSQL gibi başlıklar kapsamı genişletmek amacıyla sözlüğe bağımsız konu olarak eklenmemiştir. Bunlar veri tabanı alanının önemli konuları olmakla birlikte, bu dokümanın görünen ders kapsamının dışındadır.