İLİŞKİSEL CEBİR / İLİŞKİSEL HESAB

- Verilere farklı yollarla erişmek için kullanılan tabloların kullanılan sorgu yapan prosedürel bir dilidir.

Prosedürel sorgu dili
Veri almak
veri ekleme, güncelleme
silme ...

Prosedürel olmayan sorgu dilidir.
Hangi verilerin alınacağını söyler ama nasıl alınacağını söylemez

---

Sorgunun girdileri ve çıktıları ilişkiler (Relations) dır.

Relational Operation

↳ SELECT ↳ PROJECT ↳ UNION
↳ SET INTERSECTION ↳ SET DIFFERENCE
↳ CARTASIEN PRODUCT ↳ RNAME

- σ Selection
- π Projection
- ⋈ natural join
- X cartesian product
- ∪ union
- − Difference
- ρ yeniden adlandırma

- **SELECT (σ)**
  Verilen koşulu sağlayan bir ilişkideki veya tablodaki tuple'ları veya satırları bulmak için kullanılır.
  Sadece bulur ekrana yazdırmaz
  Ekrana çıktı vermesi için π projection operatörü kullanılır.
- **PROJECTION (π)**
- **UNION (U)** iki ilişki arasından veri almak için kullanılır.
  Bu işlemin çalışması için tablolar aynı sayıda sütun adına sahip olmalıdır.
  Verilerin denetlenmesi sonucunda çıkartılır.

A1

Ayse	12
Mehmet	11
Ali	40

A2

Muhammed	80
Zübeyde	12
Ali	40

$\pi_{name}(A1) \cup \pi_{name}(A2)$

Ayse
Mehmet
Ali
Zübeyde
Muhammed

- **SET-DIFFERENCES (−)**
  İlk ilişkide bulunup ikinci ilişkide bulunmayanları ayırtlar (Farkını alır)

CARTASIAN (X) Kartezyen çarpım
R_1 ilişkisinin her bir demeti R_2 ilişkisinin her bir demeti ile birleştirilir.

A	123
B	456

888	---
999	+++

A	123	888	---
A	123	999	+++
B	456	888	---
B	456	999	+++

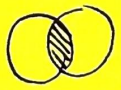
Koşulu sağlamalar koşuluyla birleşir

- **EQUI Join**
  Theta birleştirmesinin yalnızca (=) ile yapılmışsa o EQUI
- **Natural Join (⋈)**
  İki ilişki arasında en az ortak bir özellik varsa kullanabiliriz.
  Özellikle aynı adı ve etki alanına sahip olmalıdır

EMPLOYEE	Name
101	Stephen
102	John
103	Henry

EMPLOYEE	Salary
101	5000
102	4000
103	3000

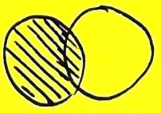
- **RENAME (ρ)**
- **INTERSECTION (∩)** Kesişim
  İki tablonun ortak satırlarını seçer



- **JOINS (⋈)**

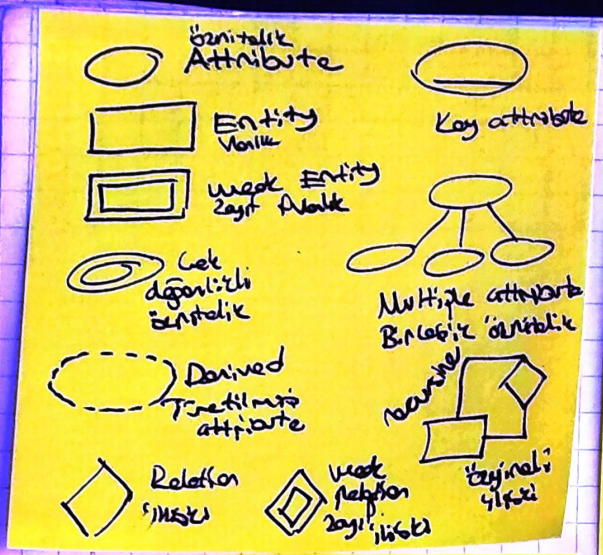
Inner joins	Outer Join
Theta	Left outher
EQUI	Right outher
Natural	Full outher

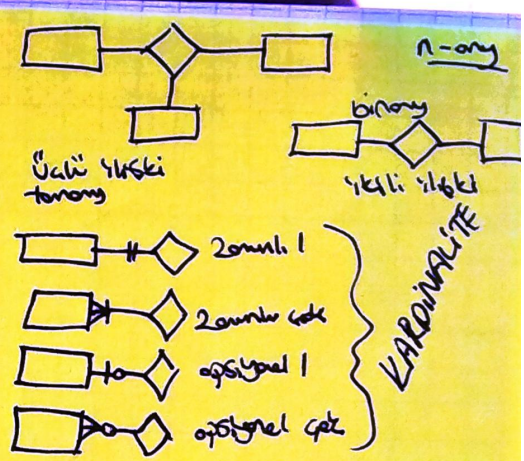
LEFT OUTHER JOIN (⟕)



RIGHT OUTHER JOIN (⟖)

FULL OUTHER JOIN (⟗)





ERD oluşturma adımları

1) Varlıklar tanımlanır (Entity Identification)
2) İlişkiler tanımlanır (Relationship Identification)
3) Kardinalite tanımlanır (Cardinality Identification)
4) Özellikler tanımlanır (Identify attributes)
5) Create ERD

VERİTABANI OLUŞTURMA YAŞAM GÖRÜŞÜ

1) Gereksinim analizi (Requirement Analysis)
2) Kavramsal Tasarım (Conceptual Design)
3) Mantıksal Tasarım (Logical Design)
4) Fiziksel Tasarım (Physical Design)
5) Create Database

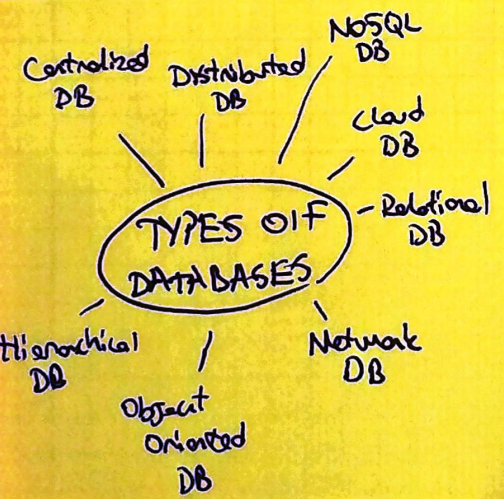
DBMS LANGUAGE

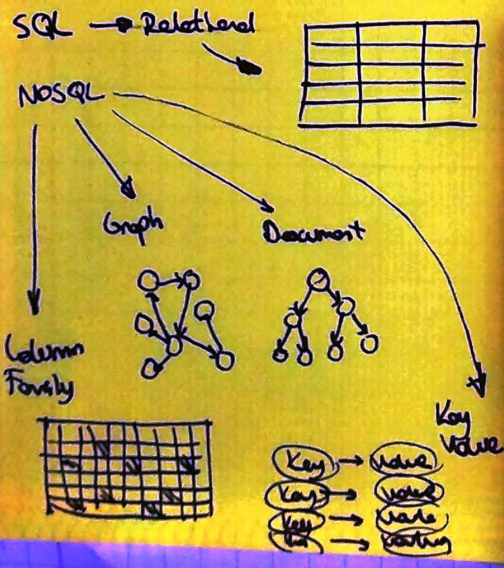
DDL — Data Definition Language
Veritabanı yapısını ve modelini tanımlarken kullanılır.
CREATE
ALTER
DROP
DELETE
RENAME
COMMENT

DCL — Data Control Lang.
Sahıcıyı ve kaydedilen verileri almak için kullanılır.
GRANT
REVOKE

DML — Data Manipulation Language
Verilere erişmek veya onları işlemek için kullanılır
INSERT
UPDATE
DELETE
JOIN
SELECT
LOCK
CALL

TCL — Transaction Control Language
COMMIT
ROLLBACK
SAVEPOINT
SET TRANSACTION





ACID özellikleri

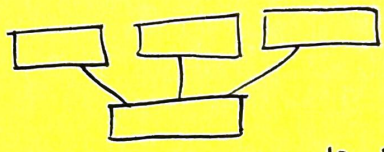
A - Atomicity (Atomiklik)
- Bir işlem gerçekleşirken ona başka bir işlemin girmemesi. Kurmalı İşlem

C - Consistency (Tutarlılık)
- İşlem öncesi ve sonrası işlem kurmalı sonuçları değişmemeli

I - Isolation (İzolasyon)
- Aynı anda veri artırılmamalı, veri durumunda veri atılamamalıdır

D - Durability (Dayanıklılık)
- İşlemi tamamlayıp verileri işlediğinde veri değişikliğinin kalıcı kalması sağlanır.

Generalization (Genelleme)



Genelleştirilmiş varlıkların tüm özellikleri içeren bir genel varlık bulunmasına genelleştirme denir.

Specialization (Özelleştirme) - Genellemenin tam tersi

Bir varlığın alt varlıklara bölündüğü süreçtir.

DBMS Bileşenleri

- Donanım
- Yazılım
- Veri
- Prosedürler
- Veritabanı Erişim Dili

3 LAYER DB

Presentation
↓ ↑
Application
↓ ↑
Database Tier

2 LAYER DB

Application
↓ ↑
Database Tier

DATA MODELS

- Relational Data Model → Satırlar Sütunlar Tabloları bulunur
- Semi-structural Data Model → Diğer 3'ünden farklıdır XML işaretleme dili yaygın olarak bulunur.
- Entity-Relation Data Model → Nesneler ve bunlar arasındaki ilişkiler
- Object Based Data Model → Kapsülleme nesne kimliği

Mantıksal Veri Bağımsızlığı

- Mantıksal veriler veritabanı ile ilgili verilerdir

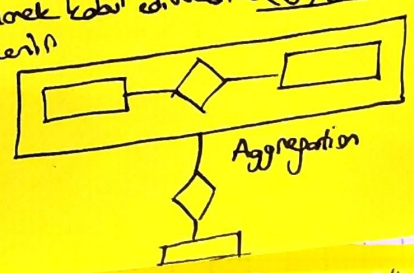
Mantıksal veri bağımsızlığı kendisine diskte depolanan gerçek verilerden uzaklaştıran bir tür mekanizmadır.

— Table formatında değişiklik yaparsak diskte bulunan veri değişmemeli dir

Fiziksel Veri Bağımsızlığı

Fiziksel veri bağımsızlığı, sonraki veya mantıksal veriyi değiştirmeden disk üzerindeki fiziksel veriyi değiştirmeyi sağlar.

- Varlıkların ayrıntılarını kullanıcıdan gizleme olayına soyutlama (abstraction) denir
- Kalıtım genelleme ve uzmanlaşmanın önemli bir özelliğidir. Alt seviye varlıkların, üst seviye varlıkların özelliklerini devralmasına kalıtım (inheritance) denir.
- İki varlık arasındaki ilişki tek bir varlık olarak kabul edilmesine (aggregation) toplama denir



ANAHTAR TÜRLERİ

- **Primary Key** → Benzersiz şekilde bir varlığı tanımlamamızı yardımcı olan → Birincil Anahtar
- **Candidate Key** → Bir demet varlığı benzersiz şekilde tanımlayabilen özelliktir. → Aday Anahtar
- **Super Key** → Aday anahtarların birleşimine süper anahtar denir → Süper Anahtar
- **Foreign Key** → Başka bir tablonun birincil anahtarını temsil etmek için kullanılan tablonun sütunudur. → Yabancı Anahtar

ER Diyagramını Tabloya Dönüştürme

- Varlıklar tablolara dönüşür
- Tek değerlikli özniteliklerin tabloların sütunları olur
- Birincil anahtarlar tabloların anahtar özelliği olur.
- Birden çok değerlikli öznitelik aynı bir tablo ile temsil edilir
- Tablolarda türetilmiş özellikler yer almaz

CODD Kuralları

1) Bilgi Kuralı
2) Garantili Erişim Kuralı
3) Boş değerlerin sistematik olarak değerlendirilmesi
4) Etkin Çevrimiçi Katalog
5) Kapsamlı veri alt dil kuralı
6) Güncelleme Kuralı
7) Yüksek düzeyli ekleme, güncelleme, silme
8) Fiziksel veri bağımsızlığı
9) Mantıksal veri bağımsızlığı
10) Bütünlük bağımsızlığı
11) Dağıtım bağımsızlığı
12) Yıkmama kuralı

SQL – Yapılandırılmış Sorgu Dili

NORMALLEŞTİRME – NORMALIZATION

Veritabanı tasarımının amacı, gereksiz fazlalık olmadan bilgileri depolamamıza hadde.. Bu durumu gerçekleştirmek için normal formlar kullanılır

Bilgi Teknoloji

İşlevsel bağımlılık

dept_name	budget
Physics	70000
Biology	90000
Math	10,000

Bir değer başka bağlı bağımlılık

İşlevsel Bağımlılık (FD)

İki öznitelik arasındaki bir dizi kısıtlama bağlılıktır.

Functional Dependency (İşlevsel Bağımlılık)

- Trivial Functional Dependency — Önemsiz İşlevsel bağımlılık
- Not-trivial Functional Dependency — Önemsiz olmayan İşlevsel bağımlılık

Eklenme Anormalliği

Silme Anormalliği

Normalleştirme verileri düzenleme işlemidir.

Bir ilişkiden veya bir dizi ilişkiden fazlalığı en aza indirmek için kullanılır. Büyük tabloları küçük tablolara böler.

NORMAL FORMS

1NF 2NF 3NF 4NF 5NF

Cover	Sayed	Şoför	Araç	Semt
Orhan	Yılmaz	Ahmet	Toyota	Levent, Etiler
Metin	Sayar	Mehmet	Honda	Bakırköy, Ataköy,
Metin	Sayar	Tolga	Ford	Kadıköy, Beylerbeyi

↓ 1NF

Orhan	Yılmaz	Ahmet	Toyota	Levent
Orhan	Yılmaz	Ahmet	Toyota	Etiler
~~Orhan~~	~~Yılmaz~~	~~Ahmet~~	Honda	Bakırköy
Metin	Sayar	Mehmet	Honda	Ataköy
Metin	Sayar	Mehmet	Ford	Kadıköy
Metin	Sayar	Tolga		
Metin	Sayar	Tolga	Ford	Beylerbeyi

Consistency — Veri bütünlüğü

Availability — Erişilebilirlik

CAP Teorisi

Partition Tolerance — Bölüm Toleransı

Normalleştirme

Veri fazlalığını, ekleme anormalliğini, güncelleme anormalliğini ve silme anormalliğini önlemek için verileri veritabanında düzenleme işlemidir.

Güncelleme Anormalliği:

emp_id	emp_name	emp_address
101	Rick	Delhi
101	Rick	Delhi
123	Maggie	Agra
166	Glen	Chennai
166	Glen	Chennai

Rick çalışanın iki satırda varsa ve adresin değiştirmek istersek iki satırda da değişmeli

- 1NF Bir atomik değer içeriyse bu ilişki 1NF'dir
- Bir ilişki 1NF'e uyuyorsa ve anahtar olmayan tüm özellikler birincil bir anahtara bağlı olarak tamamen işlevselse 2NF olacaktır.
- 2NF'de ise ve geçişli bağımlılığı yoksa 3NF'dir
- Boyce Codd normal formunda çok değerli bağımlılık yoksa 4NF olacaktır.

XML Platformdan bağımsız datanın tanımıyla ilgili işaretleme dili

Farklı programlama dilleri ve platformlar arasında veri aktarımı sağlayacak bir dil.

XML belgelerinin kök öğesi olmak zorundadır.

+ bir kapanış etiketine sahip olmalıdır