



Noter

Adit (hjpg708)

AD 2023

11. februar 2025

Indhold

How to organize data?	3
Entity–relationship (ER) model	3
What Should We Learn Today?	3
Summary of ER	3
Relational model	3
How to query data?	4
Relational calculus	4
SQL	4
How to organize data consistently and efficiently?	4
Functional Dependencies	4
Normal Forms	4
ACID	4
Views triggers	4
How to query data efficiently?	4
Indexing	4
Join evaluation	4

How to organize data?

Entity–relationship (ER) model

What Should We Learn Today?

- Explain the concepts of entity (set), relationship (set) and express these concepts in entity-relationship (E/R) diagrams
- Explain and express constraints (key, uniqueness, and ref. integrity) in E/R diagrams
- Explain and capture in E/R diagrams weak entity sets and ISA hierarchies
- Analyze trade-offs and argue for the advantages and disadvantages of an E/R design
- Explain what a data model is, along with the distinction between schemas and instances
- Define and explain the relational data model
- Define and explain the main classes of integrity constraints in the relational model, in particular key and foreign key constraints
- Model relational schemas and express them in SQL

Summary of ER

- Several kinds of integrity constraints can be expressed in the ER model: key constraints, uniqueness constraints, and referential-integrity constraints. Some foreign key constraints are also implicit in the definition of a relationship set.
 - Some constraints (notably, functional dependencies) cannot be expressed in the ER model.
 - Constraints play an important role in determining the best database design for a use case.
- E/R design is subjective. There are often many ways to model a given scenario! Analyzing alternatives can be tricky. Common choices include:
 - Entity vs. attribute, entity vs. relationship, binary or n-ary relationship, whether or not to use ISA hierarchies, how to define constraints.
- Ensuring good database design: resulting relational schema should be analyzed and refined further.
- Functional dependencies and normalization techniques can be useful but cannot be captured in a E/R model. More on this later in the course

Relational model

How to query data?

Relational calculus

Schema No file chosen

```
Classes(class:string, type:string, country:string)
Ships(name:string, class:string, launched:int)
Battles(name:string, date:string)
Outcomes(ship:string, battle:string, result:string)
```

Instance No file chosen

```
Classes("Bismarck","bb","Germany",8,15,42000)
Classes("Iowa","bb","USA",9,16,46000)
Classes("Kongo","bc","Japan",8,14,32000)
Classes("SouthDakota","bb","USA",9,16,37000)
Classes("Renown","bc","GreatBritain",6,15,32000)
Classes("Revenge","bb","GreatBritain",8,15,29000)
Classes("Tennessee","bb","USA",12,14,33000)
Classes("Yamato","bb","Japan",9,18,65000)
Ships("Alabama","SouthDakota",1942)
Ships("Haruna","Kongo",1915)
Ships("Hiei","Kongo",1914)
Ships("Idaho","Mississippi",1919)
Ships("Iowa","Iowa",1943)
Ships("Kirishima","Kongo",1915)
Ships("Kongo","Kongo",1913)
Ships("Missouri","Iowa",1944)
Ships("Musashi","Yamato",1942)
Ships("NewJersey","Iowa",1943)
Ships("NewMexico","Mississippi",1918)
Ships("Ramillies","Revenge",1917)
Ships("Renown","Renown",1916)
Ships("Repulse","Renown",1916)
Ships("Resolution","Revenge",1916)
Ships("Revenge","Revenge",1916)
Ships("RoyalOak","Revenge",1916)
Ships("RoyalSovereign","Revenge",1916)
Ships("SouthDakota","SouthDakota",1942)
Ships("Wisconsin","Iowa",1944)
Ships("Yamato","Yamato",1941)
Battles("DenmarkStrait","24-27.05.41")
Battles("Guadalcanal","15.11.42")
Battles("NorthCape","26.12.43")
Battles("SurigaoStrait","25.10.44")
Outcomes("Arizona","PearlHarbor","sunk")
Outcomes("Bismarck","DenmarkStrait","sunk")
```

RC query No file chosen

```
EXISTS t, n, b, d.
Classes(c1,t,co,n,b,d) AND
b = 16
```

Result

```
("Iowa","USA")
("SouthDakota","USA")
```

SQL

How to organize data consistently and efficiently?

Functional Dependencies

Normal Forms

ACID

Views triggers

How to query data efficiently?

Indexing

Join evaluation