

E-R model, Relational model, SQL

Hogeschool Rotterdam Rotterdam, Netherlands



General Stuff

E-R model, Relational model, SQL

Course

- Three assignment (Relational Databases, Document Database, Graph Database)
- Two Exams: Written and practicum (practicum check)



General Stuff

E-R model, Relational model, SQL

Course

- Relational databases: advantages and limitations (PostgreSQL)
- NoSQL databases: advantages, limitation, two cases studies (MongoDB and Neo4J)
- Advance topics for big data such as Hadoop, Spark will not be discussed



Introduction

E-R model, Relational model, SQL

Lecture topics

- E-R model.
- Relational model.
- SQL, and examples.



E-R model, Relational model, SQL

database design process

• Requirements analysis



E-R model, Relational model, SQL

- Requirements analysis
- Conceptual database design using E-R models



E-R model, Relational model, SQL

- Requirements analysis
- Conceptual database design using E-R models
- Logical database design (sometimes conceptual and logical are merged into one step)



E-R model, Relational model, SQL

- Requirements analysis
- Conceptual database design using E-R models
- Logical database design (sometimes conceptual and logical are merged into one step)
- Schema refinement through normalization



E-R model, Relational model, SQL

- Requirements analysis
- Conceptual database design using E-R models
- Logical database design (sometimes conceptual and logical are merged into one step)
- Schema refinement through normalization
- Physical database design



E-R model, Relational model, SQL

- Requirements analysis
- Conceptual database design using E-R models
- Logical database design (sometimes conceptual and logical are merged into one step)
- Schema refinement through normalization
- Physical database design
- Application and Security Design



E-R model, Relational model, SQL

Overview

- Highest level of database modelling.
- Model the conceptual aspect of the database.
- Far from the physical representation in the DBMS.



E-R model, Relational model, SQL

Entity

- Anything which can exist on its own on the database
- Consider a database for a space shooter game
- Starships, asteroids are entities, they have a meaning on their own



E-R model, Relational model, SQL

Attributes

- They model characteristics of the entity.
- Starship: velocity, shield, armour, weapon, [...]
- Asteroid: velocity, mass, integrity, [...]



E-R model, Relational model, SQL

Relations

- They describe the associations among entities (two or more).
- They have a cardinality: number of participants for each side.



E-R model, Relational model, SQL

Relations - 1:1

- Entity modelling a pilot and one modelling a starship.
- Related by "drives".
- The cardinality is 1:1: one pilot drives at most one starship, and one starship can contain only one pilot.



E-R model, Relational model, SQL

Relations - 1 : N

- Entity modelling a starship and one modelling a weapon.
- Related by "mounted"
- The cardinality is 1:N: a weapon can be mounted only on one starship, but a starship can mount more than one weapon.



E-R model, Relational model, SQL

Relations - N : M

- Entity modelling a starship and one modelling an asteroid.
- Related by "collides with"
- The cardinality is N : M : several starships can collide with several asteroids.



E-R model, Relational model, SQL

Keys

- A way to uniquely identify an entity.
- A key is a set of attributes that have unique values among entities.
- Starship: id number.



E-R model, Relational model, SQL

Weak entities

- Entities which do not have a key attribute.
- **Asteroids:** There can be two asteroids with the same position, same mass, velocity, etc.



E-R model, Relational model, SQL

Overview

- Halfway between a conceptual model and the physical model.
- Contain an abstraction of physical elements.
- Can be easily mapped to a physical implementation in a DBMS.
- There are mapping rules from E-R model to the relational model.



E-R model, Relational model, SQL

Relation

- A relation is a collection of tuples.
- Each element of a tuple is a value taken from an attribute set.
- Each attribute set is identified by a name

Ships					
<u>id</u>	name	shields	pilot	armour	integrity

(38258269, "Battlestar Galactica", 3000, "Captain Jack" ,5000, 1.0)



E-R model, Relational model, SQL

Keys

- A Primary key is a set of attributes with unique values in each tuple.
- A Candidate key is the smallest set of attributes which form a superkey.

Example:

Candidate key: (id, name)

Primary key: (id)



E-R model, Relational model, SQL

Keys

- A Primary key is the chosen key for a relation among all the candidate keys.
- A Foreign key is a set of attributes in one relation which is a primary key in another relation.

Example (Foreign key):

Mounts		
shipid	weaponName	

Ships					
<u>id</u>	name	shields	pilot	armour	integrity

In the relation Mounts the attribute shipid is a foreign key to Ships.

SQL

E-R model, Relational model, SQL

Overview

 Declarative language ("What" not "How"). Consists of 4 categories

SQL

E-R model, Relational model, SQL

Overview

- Declarative language ("What" not "How"). Consists of 4 categories
- Data Definition Language (DDL): used to create relations (tables).
- Data Manipulation Language (DML): used to insert/modify/extract data from relations (tables).
- Data Control Language (DCL): grant control to tables, views and database
- Transaction Control Language TCL: used to create transactions and to control them.



	Ships				
<u>id</u>	name	pilot	shields	armour	integrity

Select all ships from the game



	Ships				
<u>id</u>	name	pilot	shields	armour	integrity

Select all ships from the game

SELECT *
FROM Ships



	Ships				
<u>id</u>	name	pilot	shields	armour	integrity

Select all ships in the game whose pilot is "William Adama"



Ships					
<u>id</u>	name	pilot	shields	armour	integrity

Select all ships in the game whose pilot is "William Adama"

```
SELECT *
FROM Ships s
WHERE s.pilot = 'William_Adama'
```



Ships					
<u>id</u>	name	pilot	shields	armour	integrity

Find the name of the ships whose pilot is "Starbucks"



	Ships				
<u>id</u>	name	pilot	shields	armour	integrity

Find the name of the ships whose pilot is "Starbucks"

```
SELECT s.name
FROM Ship s
where s.pilot = 'Starbucks'
```



Ships					
<u>id</u>	name	pilot	shields	armour	integrity

Mounts			
shipid	weaponName		

Weapons				
<u>name</u>	damage	type		

Find the id of the ships mounting the weapon "Stealthblade MKII"





Mounts			
shipid	weaponName		

```
Weapons

name damage type
```

Find the id of the ships mounting the weapon "Stealthblade MKII"



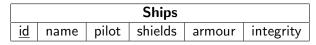
Ships					
<u>id</u>	name	pilot	shields	armour	integrity

Mounts		
shipid	weaponName	

Weapons			
<u>name</u>	damage	type	

Find the name of all the weapons mounted in the ships flown by "Apollo"





Mounts		
shipid	weaponName	

Weapons			
<u>name</u>	damage	type	

Find the name of all the weapons mounted in the ships flown by "Apollo"



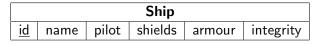
Ship					
<u>id</u>	name	pilot	shields	armour	integrity

Mounts			
shipid	weaponName	Count	

Weapons			
<u>name</u>	damage	type	

Find the total damage output of the ships flown by "Athena"





Mounts			
shipid	weaponName	Count	

Weapons		
<u>name</u>	damage	type

Find the total damage output of the ships flown by "Athena"



	Ships				
<u>id</u>	name	pilot	shields	armour	integrity

Mounts		
shipid	weaponName	

Weapon			
<u>name</u>	damage	type	

Count all the ships having more than 3 weapons



Ships					
<u>id</u>	name	pilot	shields	armour	integrity

Mounts		
shipid	weaponName	

Weapon			
<u>name</u>	damage	type	

Count all the ships having more than 3 weapons

```
SELECT COUNT(*)

FROM (
SELECT * AS ShipCount
FROM Ship s, Mounts m, Weapon w
WHERE s.id = m.shipid AND
m.weaponName = w.Name
GROUP BY s.id
HAVING COUNT(*) > 3)
```