E-R model, Relational model, SQL

Hogeschool Rotterdam
Rotterdam, Netherlands

Lecture topics

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

database design process

- Requirements analysis

database design process

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

database design process

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

database design process

- 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

database design process

- 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

database design process

- 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

Overview

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

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

Attributes

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

Relations

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

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.

Relations - 1 : N

- Entity modelling a starship and one modelling a weapon.
- Realted 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.

Relations - N : M

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

Keys

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

Weak entities

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

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.

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	armour	integrity

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

E-R model,
Relational
model, SQL

Introduction

Data models

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:
Primary key: (id, name, shield)
Candidate key: (id)

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				
id	name	shields	armour	integrity

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

Overview

- Used to create relations (tables).
- Used to insert/modify/extract data from relations (tables).
- Declarative language ("What" not "How").

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

Select all ships from the game

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

Select all ships from the game

```
SELECT *
FROM Ships
```

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

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

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

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

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

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

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

Ships				
id	name	shields	armour	integrity

Mounts	
shipid	weaponName

Weapons		
name	damage	type

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

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

Mounts	
<u>shipid</u>	<u>weaponName</u>

Weapons		
<u>name</u>	damage	type

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

```
SELECT s.id
FROM Ship s, Mounts m
WHERE s.id = m.shipid AND
      m.weaponName = 'StealthBlade MKII'
```

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

Mounts	
<u>shipid</u>	<u>weaponName</u>

Weapons		
<u>name</u>	damage	type

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

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

Mounts	
<u>shipid</u>	weaponName

Weapons		
<u>name</u>	damage	type

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

```sql
SELECT w.name
FROM Ship s, Mounts m, Weapon w
WHERE s.id = m.Shipid AND
      m.weaponName = w.name AND
      s.pilot = 'Apollo'
```

Ship					
Serial	Name	Pilot	Shields	Armour	Integrity

Mounts		
ShipSerial	WeaponName	Count

Weapons		
name	damage	type

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

Ship					
<u>Serial</u>	Name	Pilot	Shields	Armour	Integrity

Mounts		
<u>ShipSerial</u>	<u>WeaponName</u>	Count

Weapons		
<u>name</u>	damage	type

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

```sql
SELECT SUM(w.damage) AS damage
FROM Ship s, Mounts m, Weapon w
WHERE s.id = m.shipid AND
      m.weaponName = w.name AND
      s.pilot = 'Athena'
```

Ships

id	name	shields	armour	integrity

Mounts

shipid	weaponName

Weapon

name	damage	type

Count all the ships having more than 3 weapons

E-R model,
Relational
model, SQL

Introduction

Data models

E-R model

Relational
model

SQL

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

Mounts	
<u>shipid</u>	<u>weaponName</u>

Weapon		
<u>name</u>	damage	type

Count all the ships having more than 3 weapons

```
SELECT COUNT(*)
FROM (
 SELECT COUNT(*) 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)
```