

HOW DID NETFLIX
BEAT BLOCKBUSTER?
DATABASE SYSTEMS
PART 2

ANNOUNCEMENT

- Bonus 5 to be posted soon.

HOW NETFLIX BEAT
BLOCKBUSTER?

SOME ARGUMENTS

- 2005, Blockbuster worthed \$8 billion
- Netflix at the beginning: stock was a “worthless piece of crap” by WSJ
- 2010, Blockbuster filed for bankruptcy; NetFlix: 116 million in earning
- Use of emerging tech: good piece of SW, streaming, user friendly service, etc.
- Further readings:
 - <http://billhalal.com/?p=295>
 - <http://www.businessinsider.com/how-netflix-bankrupted-and-destroyed-blockbuster-infographic-2011-3>



IN 2011, **NETFLIX** SURPASSED **20 MILLION SUBSCRIBERS** IN THE **UNITED STATES** AND **CANADA**, MAKING IT THE WORLD'S LEADING INTERNET SUBSCRIPTION SERVICE FOR ENJOYING MOVIES AND TV SHOWS.



<http://www.businessinsider.com/how-netflix-bankrupted-and-destroyed-blockbuster-infographic-2011-3>

NETFLIX TAKES OVER PRIME TIME

NETFLIX STREAMING REPRESENTS MORE THAN

20% OF DOWNSTREAM INTERNET TRAFFIC DURING PEAK TIMES 8 -10 PM.

(AND IT'S BEING USED BY JUST UNDER 2% OF NETFLIX SUBSCRIBERS)



PEAK TIME DOWNSTREAM INTERNET TRAFFIC

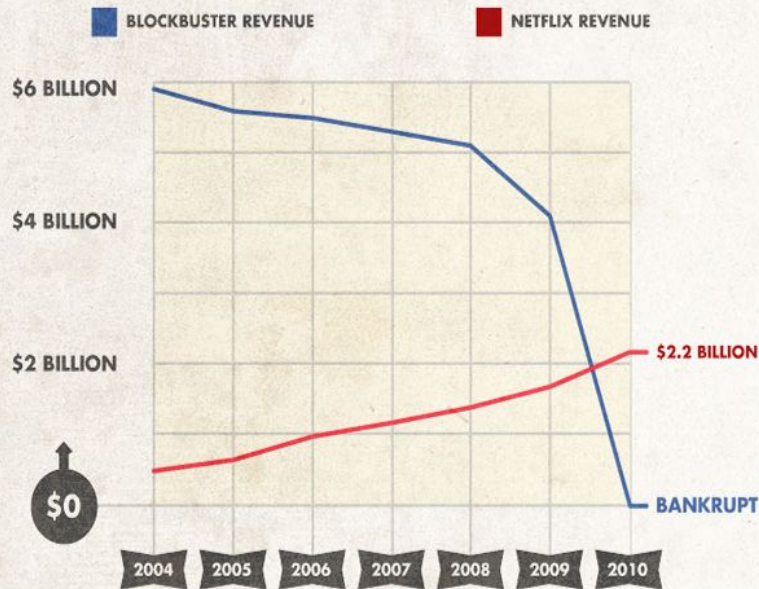


<http://www.businessinsider.com/how-netflix-bankrupted-and-destroyed-blockbuster-infographic-2011-3>



BACK IN 2000, **BLOCKBUSTER** DECLINED SEVERAL OFFERS TO PURCHASE **NETFLIX** FOR A MERE **\$50 MILLION**.
IN 2010, **BLOCKBUSTER** DECLARED BANKRUPTCY.

NETFLIX VS. BLOCKBUSTER (2004-2010)



<http://www.businessinsider.com/how-netflix-bankrupted-and-destroyed-blockbuster-infographic-2011-3>

SQL BASICS

DATABASE REVIEW

- Database
 - Organized collection of data
- Database management system
 - Control the creation and use of a database
- Record
 - Related data stored in a row
- Field
 - A named piece of data (e.g. Last name: Phillips)

DATABASE REVIEW

We ignore commands to create and delete database information

SQL command:

- ❑ `DROP TABLE table_name`
- ❑ Delete the table



SQL PRACTICE

Database queries are done with a **select**

```
select attribute-list from table-list where conditions
```

An experienced person can extract just a few relevant records from a huge database

We will practice on a database: <http://sqlzoo.net/>

SQL EXAMPLE

CustomerId	Name	Address	CreditCardNumber
101	Dennis Cook	123 Main Street	2736 2371 2344 0382
102	Doug Nickle	456 Second Ave	7362 7486 5957 3638
103	Randy Wolf	789 Elm Street	4253 4773 6252 4436
104	Amy Stevens	321 Yellow Brick Road	9876 5432 1234 5678
105	Robert Person	654 Lois Lane	1122 3344 5566 7788
106	David Coggin	987 Broadway	8473 9687 4847 3784
107	Susan Klaton	345 Easy Street	2435 4332 1567 3232

`SELECT * FROM Customer where Name = 'Amy Stevens'`

- Show just the record for Amy Stevens.
- This type of query is called a **selection** since it selects particular rows from the table

SQL EXAMPLE

Customer			
CustomerId	Name	Address	CreditCardNumber
101	Dennis Cook	123 Main Street	2736 2371 2344 0382
102	Doug Nickle	456 Second Ave	7362 7486 5957 3638
103	Randy Wolf	789 Elm Street	4253 4773 6252 4436
104	Amy Stevens	321 Yellow Brick Road	9876 5432 1234 5678
105	Robert Person	654 Lois Lane	1122 3344 5566 7788
106	David Coggin	987 Broadway	8473 9687 4847 3784
107	Susan Klaton	345 Easy Street	2435 4332 1567 3232

SELECT Name, Address FROM Customer

- ❑ Show the name and address of all customers
- ❑ This type of query is called a **projection** since it reduces the amount of info shown
- ❑ A shadow is a projection of a person

SQL EXAMPLE

Movie

MovieId	Title	Genre	Rating
101	Sixth Sense, The	thriller horror	PG-13
102	Back to the Future	comedy adventure	PG
103	Monsters, Inc.	animation comedy	G
104	Field of Dreams	fantasy drama	PG
105	Alien	sci-fi horror	R
106	Unbreakable	thriller	PG-13
107	X-Men	action sci-fi	PG-13
5022	Elizabeth	drama period	R
5793	Independence Day	action sci-fi	PG-13
7442	Platoon	action drama war	R

`SELECT Title FROM Movie WHERE Rating = 'PG'`

- ❑ Combined **projection and selection**
- ❑ Result: a list of all titles that have a PG rating

SQL EXAMPLE

Movie

MovieId	Title	Genre	Rating
101	Sixth Sense, The	thriller horror	PG-13
102	Back to the Future	comedy adventure	PG
103	Monsters, Inc.	animation comedy	G
104	Field of Dreams	fantasy drama	PG
105	Alien	sci-fi horror	R
106	Unbreakable	thriller	PG-13
107	X-Men	action sci-fi	PG-13
5022	Elizabeth	drama period	R
5793	Independence Day	action sci-fi	PG-13
7442	Platoon	action drama war	R

```
SELECT * FROM Movie WHERE Genre like '%action%'
```

SQL EXAMPLE

Movie

MovieId	Title	Genre	Rating
101	Sixth Sense, The	thriller horror	PG-13
102	Back to the Future	comedy adventure	PG
103	Monsters, Inc.	animation comedy	G
104	Field of Dreams	fantasy drama	PG
105	Alien	sci-fi horror	R
106	Unbreakable	thriller	PG-13
107	X-Men	action sci-fi	PG-13
5022	Elizabeth	drama period	R
5793	Independence Day	action sci-fi	PG-13
7442	Platoon	action drama war	R

`SELECT * FROM Movie WHERE Rating = 'R' order by Title`

SQL EXAMPLE

Movie			
MovieId	Title	Genre	Rating
101	Sixth Sense, The	thriller horror	PG-13
102	Back to the Future	comedy adventure	PG
103	Monsters, Inc.	animation comedy	G
104	Field of Dreams	fantasy drama	PG
105	Alien	sci-fi horror	R
106	Unbreakable	thriller	PG-13
107	X-Men	action sci-fi	PG-13
5022	Elizabeth	drama period	R
5793	Independence Day	action sci-fi	PG-13
7442	Platoon	action drama war	R

`SELECT * FROM Movie WHERE Rating = 'R' or Rating = 'PG-13'`

Conditions for selection can use Boolean operations.

The result is all contributes of PG-13 or R-rated movies.

SQL DATES

- ❑ SQL has sophisticated date and time handling
- ❑ Dates can be used in restricting selections
- ❑ Format is important

```
SELECT * FROM CheckedOut where DueDate >= '12/10/2007'
```

SQL JOINS

- ❑ The real power in databases comes from combining info in 2 tables with a shared field (usually a key)
- ❑ This is called a **join**
- ❑ What if we want to find the Phone number and Customer Names of a Patron
- ❑ That information is not stored in the Phones table or Patrons table alone
 - ❑ Select * from each table
 - ❑ How would you solve this problem?

SQL JOIN

- A shared field in Phones and Patrons: CardNumber

Select * from Patrons, Phones where

Patrons.CardNumber = Phones.CardNumber

- When a field is in 2 tables, you must specify the table before the field
Table.FieldName
- What would happen if you remove the where clause?
Select * from Patrons, Phones

SQL JOIN

Allows the full power of database queries

- ❑ Can store information in a natural form
- ❑ Reconnect records across tables
- ❑ Useful when there are one-to-many relationships
 - ❑ One person, many checked out books
- ❑ Better than building a single, huge table
- ❑ How would we get the title of checked out books? Do it by hand first

CONCLUSION

- SQL is a powerful tool for specifying queries
- It draws upon the logical operations used in programming to allow very precise results
- Databases are the engines of the information economy
 - Reliable, scalable, flexible data processing

SQL AND ADVANCED SQL

<http://sqlzoo.net/>

Select Basics

http://sqlzoo.net/wiki/SQLZOO:SELECT_basics

Select .. Where

http://sqlzoo.net/wiki/SQLZOO:SELECT_from_WORLD_Tutorial

#3

```
SELECT name, gdp/population FROM world  
WHERE population > 200000000
```

#4

```
SELECT name, population/1000000 FROM world  
WHERE continent like '%South America%'
```

#5

```
SELECT name, population FROM world  
WHERE name in ('France', 'Germany', 'Italy')
```

#6

```
SELECT name FROM world  
WHERE name like '%United%'
```

#7

```
SELECT name, population, area FROM world  
WHERE area > 3000000 or population > 250000000
```

#8

```
SELECT name, population, area FROM world  
WHERE area > 3000000 xor population > 250000000
```

#9

```
SELECT name, ROUND(population/1000000, 2), ROUND  
(gdp/10000000000, 2) FROM world  
WHERE continent='South America'
```


#10

```
SELECT name, ROUND(gdp/population, -3) FROM world  
WHERE gdp>10000000000000
```

Join

http://sqlzoo.net/wiki/More_JOIN_operations

2

```
SELECT yr  
FROM movie  
where title like '%Citizen Kane%'
```

3

```
SELECT id, title, yr
```

```
FROM movie
```

```
where title like '%Star Trek%' ORDER by yr
```

4

Select title
from movie
where id in ('11768', '11955', '21191')

5

```
Select id  
from actor  
where name like '%Glenn Close%'
```

6

```
select id
```

```
from movie
```

```
where title like '%Casablanca%'
```

7

```
select name  
from actor join casting on (id = actorid)  
where movieid=11768
```


8

```
select name  
from (movie join casting on (movie.id = casting.movieid)) join actor on  
(casting.actorid = actor.id)  
where title = 'Alien'
```

9

```
select title
from (movie join casting on (movie.id = casting.movieid)) join actor on
(casting.actorid = actor.id)
where name = 'Harrison Ford'
```



THANKS!

Any questions?

You can find me at
beiwang@sci.utah.edu

<http://www.sci.utah.edu/~beiwang/teaching/cs1060.html>

CREDITS

Special thanks to all the people who made and released these awesome resources for free:

- Presentation template by [SlidesCarnival](#)
- Photographs by [Unsplash](#)