

# Rethink-ing your data store

All about RethinkDB

by: Brian Maula



# Outline

- A developer's tale
- Why did I choose RethinkDB?
- Why is it awesome?
- Caveats
- Demos



# Purpose

- This is a high-level introduction
- More of a story, really
- Whet your appetite for RethinkDB



# Who is this guy?

- C# Developer, ASP.NET, MVC, Entity Framework, SQL Server
- Loves playing around with new technology
- Pretends he's Carter Beauford on drums
- Enjoys cooking
- ILM Consultant



# Info

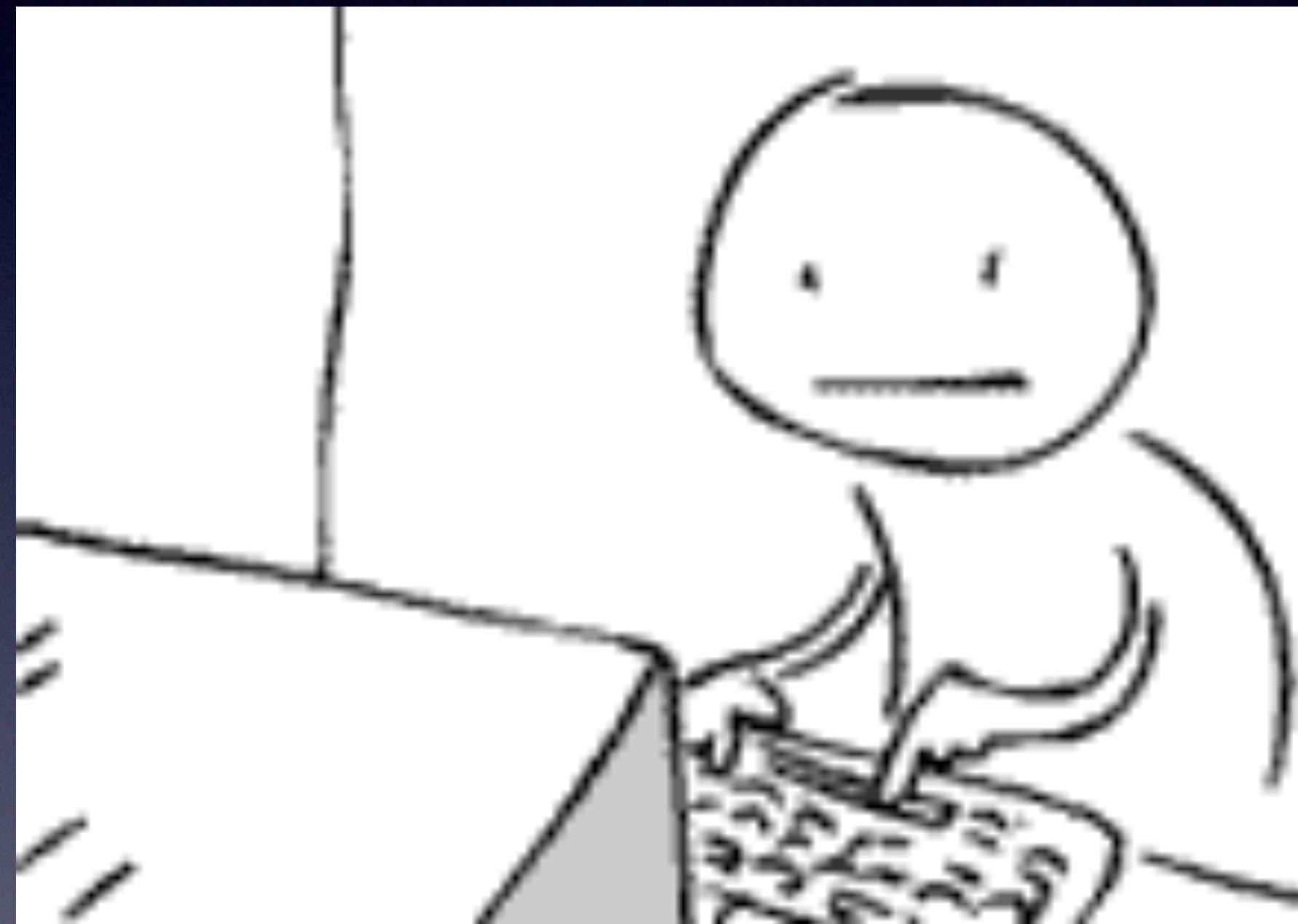
- [brian.maula@ilmservice.com](mailto:brian.maula@ilmservice.com)
- @maulowski (not necessarily tech stuff)
- [linkedin.com/in/bmaula](https://www.linkedin.com/in/bmaula) (I don't check this often...like never)



I AM NOT AN EXPERT



# A Developers Tale





# I want to build an app

- I wanted to build an app using the Microsoft “Trinity” stack: MVC, SQL Server, Entity Framework.
- Open Visual Studio 2013, new solution
- Application done!





# Application done!

No, not really. You just hit some roadblocks.



Roadblock: Real-time



# Solutions?

- Use a message queue system
- Long polling that calls an API
- Using SSIS and triggers



Roadblock: data modeling



# Why data modeling sucks

- Relational databases have schema enforcement
- You think of your data in terms of tables
- Difficult to deviate from the relational model



# Roadblock: Entity Framework/ Migrations



# Why does EF migration suck?

- Simple tables yields less headaches when writing out your Fluent queries
- Not good when your data store deviates from relational model
- Entity Framework Performance still sucks (last I checked)
- Changes to data model is costly
- The tool adds more steps to your workflow



# Conclusions

- Data modeling sucks for SQL
- Entity Framework can get in the way
- Iterating data models with migrations is time consuming
- Real-time using SQL Server is a pipe dream
- SQL Server is difficult to scale







**The open-source database**  
for the realtime web

source: <http://www.rethinkdb.com>



# Why RethinkDB?

- Is real-time, it **pushes** your data
- SQL Server, Entity Framework, and Migrations
- I didn't want a schema enforced data store
- I wanted to rapidly prototype
- Easy to manage and scale
- Have the best of both worlds



# Why RethinkDB?

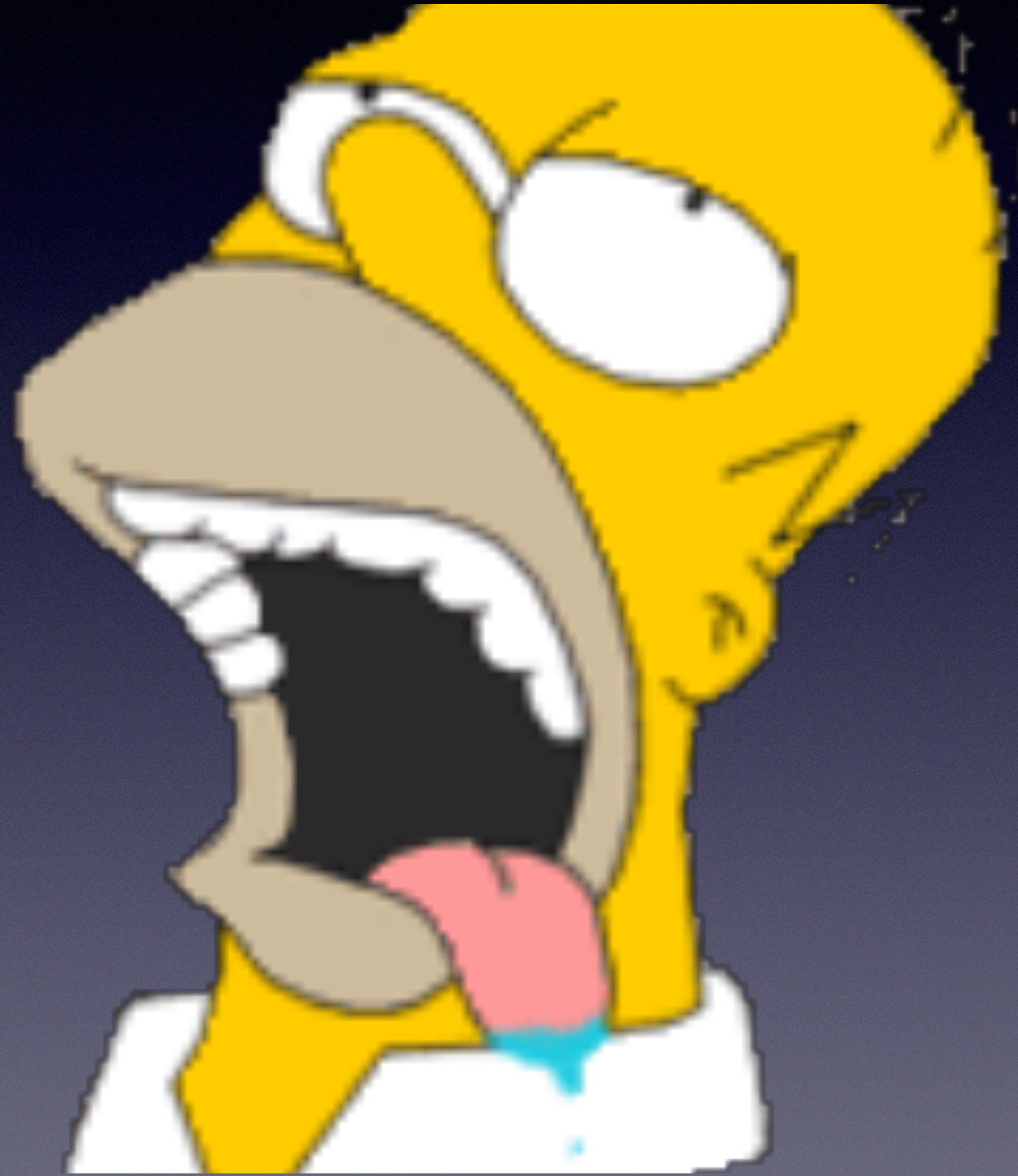
- geospatial querying
- Support for JOINS
- Chainable query language
- Real-time API is one method call
- Scaling the cluster can be done pretty intuitively



# Why RethinkDB?

- Robust architecture
- Support for secondary, compound, and arbitrary indexes
- Automatic failover
- Integration with RabbitMQ
- Integration with Elasticsearch









# Caveats

Because reality sucks



# Caveats

- Not ACID compliant
- You need strong schema enforcement
- Running deep, computationally-intensive analytics
- Running a high write system (e.g. twitter)
- Performance considerations
- You need Windows (currently only supports OS X and Linux)



# Caveats

- Be aware of how you model your data
- Atomic operations are limited to single operations only
- Queries aren't considered atomic operations
- Limited C# support
- Official drivers only support JavaScript, Ruby, and Python



# Demos

With somewhat contrived, trivial examples



# Conclusions

- If you need real-time
- If you're wanting to break up with SQL
- It has caveats, it's not perfect