Don`t always say yes to SQL

Erik Renaud, Modelon Solutions

What is SQL anyways?

language

tables, columns & rows

relationships

normalisation

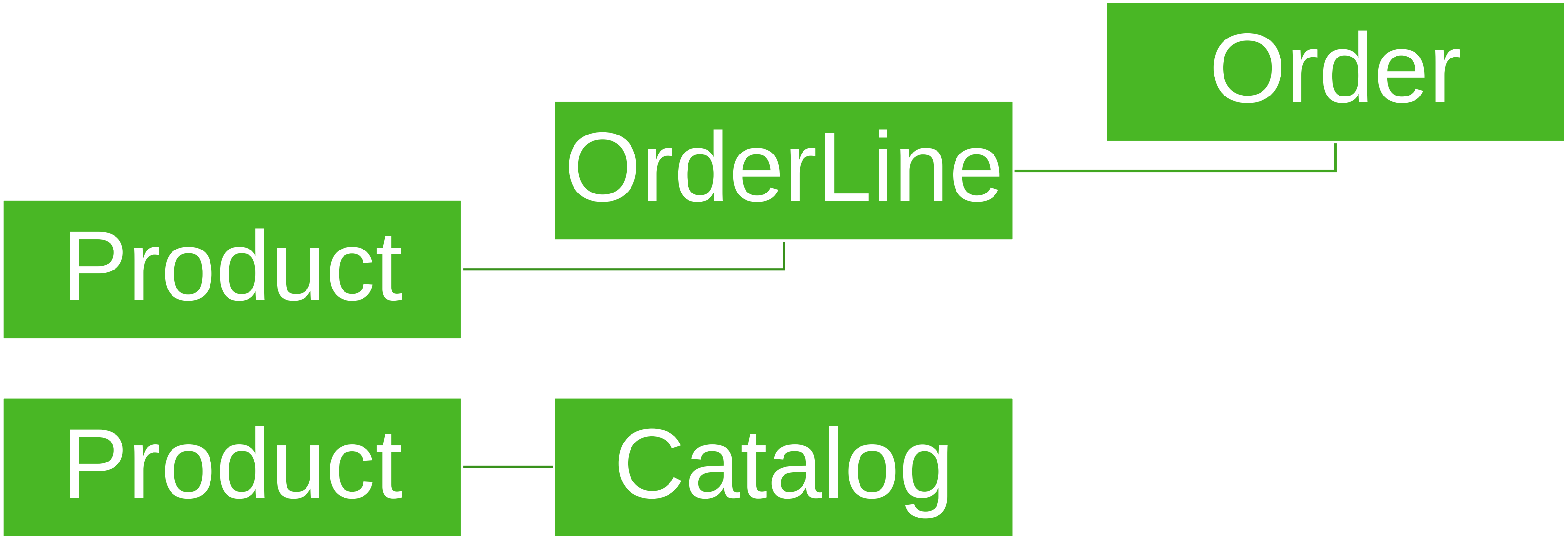
transactions

indexes

Why we say yes?

- we don’t like to change things
- we already have all the tools
- business intelligence, we’ll need that
- we have database administrators
- we like to design single row screens
- object relational mappers now exist

Why I do not like SQL



Order

OrderLine

Product

Product

Catalog

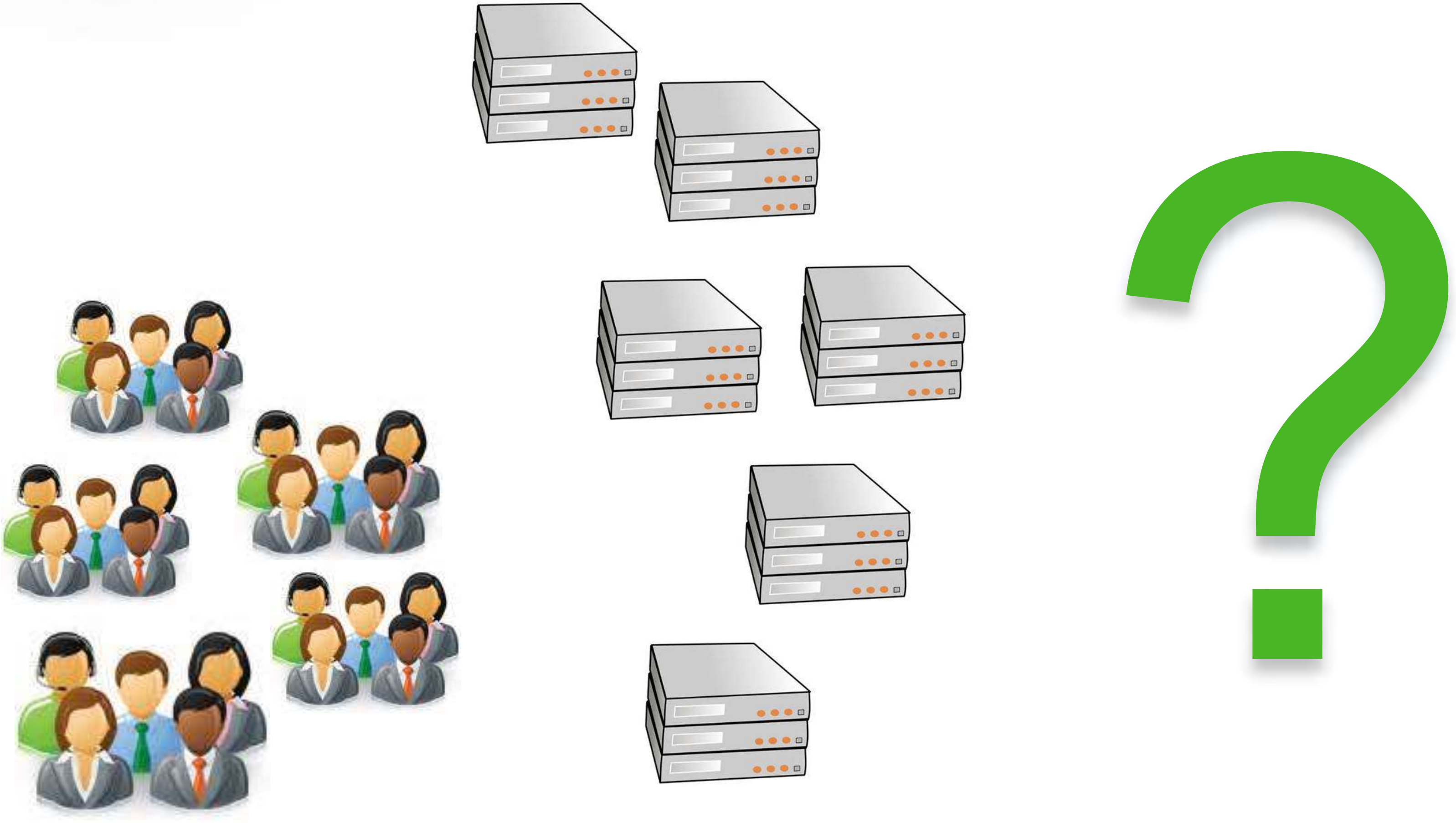
A small architecture?

a few users, a few middletiers, a central database

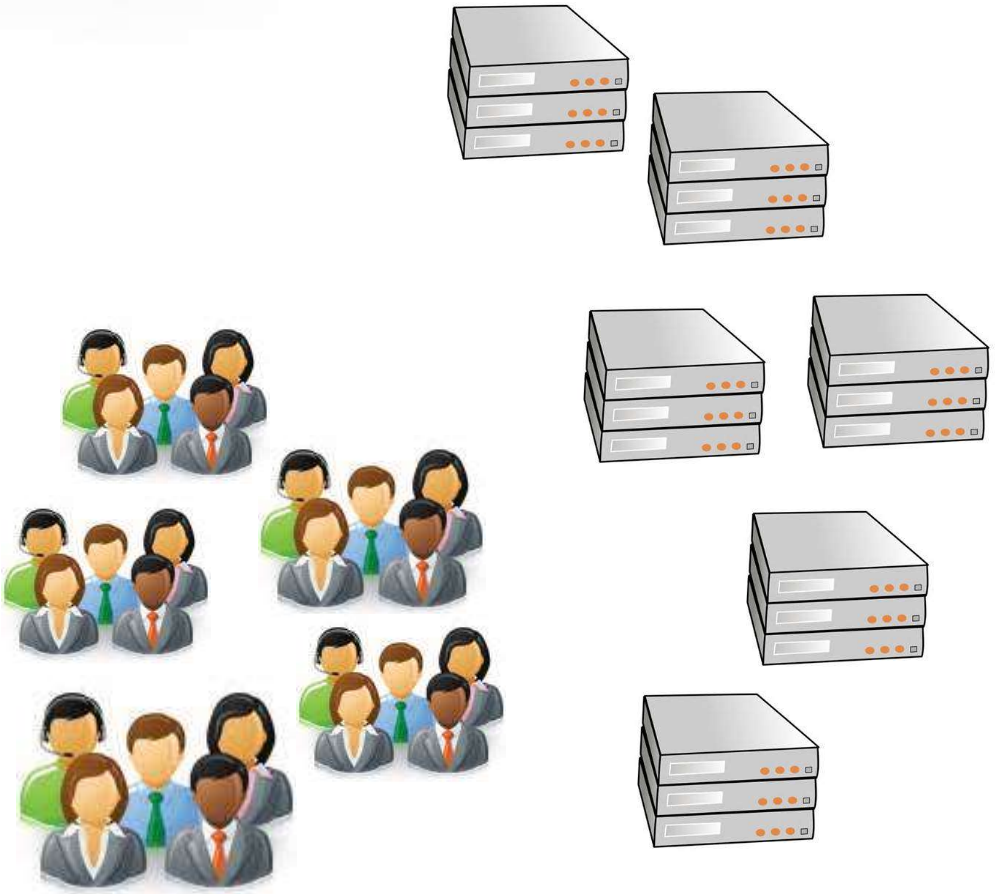
options :

backups,
replication,
clusters

A big architecture?



A big architecture idea?



Small database servers, that replicate data amongst themselves, that distribute processing on small nodes (kind of like how the web works)

Challenges

how do we group pertinent rows together to form useful entities/documents

how do we know where to read and save data

can we let go of backups because the topology is self protecting

without a sql server admin, who will take care of schemas, updates, migrations…

Enter nosql

documents (think json)

no schemas, no dbas, no downtime. burden on programmers

sharding & replication

distributed queries, map-reduce

low cost

But which one?

http://nosql-database.org

Cassandra

MongoDB

CouchDB

RavenDB

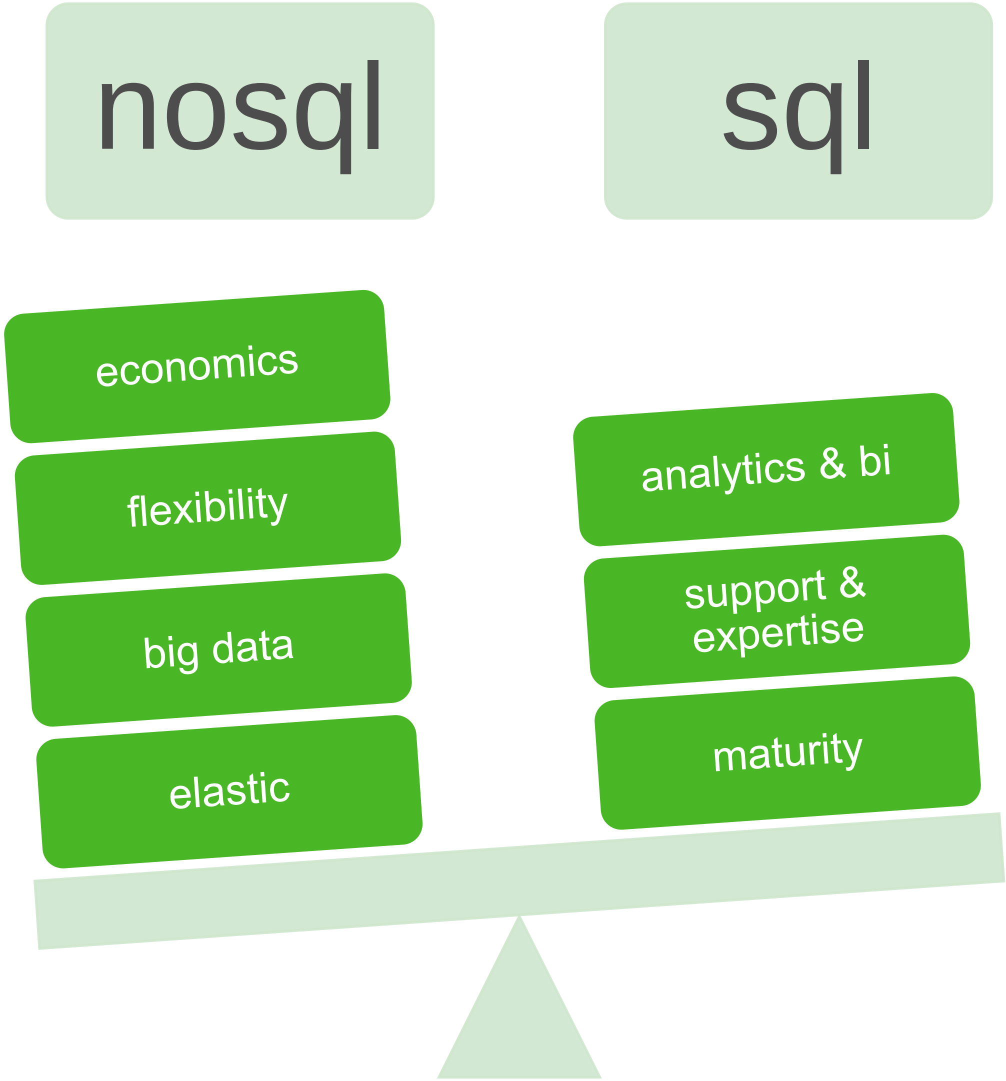
LevelDB

ElasticSearch

Demo - RavenDB

- Create a test project
- Create an entity (Order-OrderLine-Product)
- Add ravendb nugets (client-server)
- Launch ravendb
- Save the entity to ravendb
- Display what’s in the database
- Query ravendb for the entity
- Update the entity
- Show the rest concept
- Mention Esent & 30 seconds on RavenDB 3.0

Advantages



nosql

sql

economics

flexibility

big data

elastic

analytics & bi

support & expertise

maturity

BI

Acts on thousands of rows at once and aggregates them (sum, average, count…), also produces KPIs

Is very costly

A project by itself, requires architecture, analysis and design, mostly from dbas

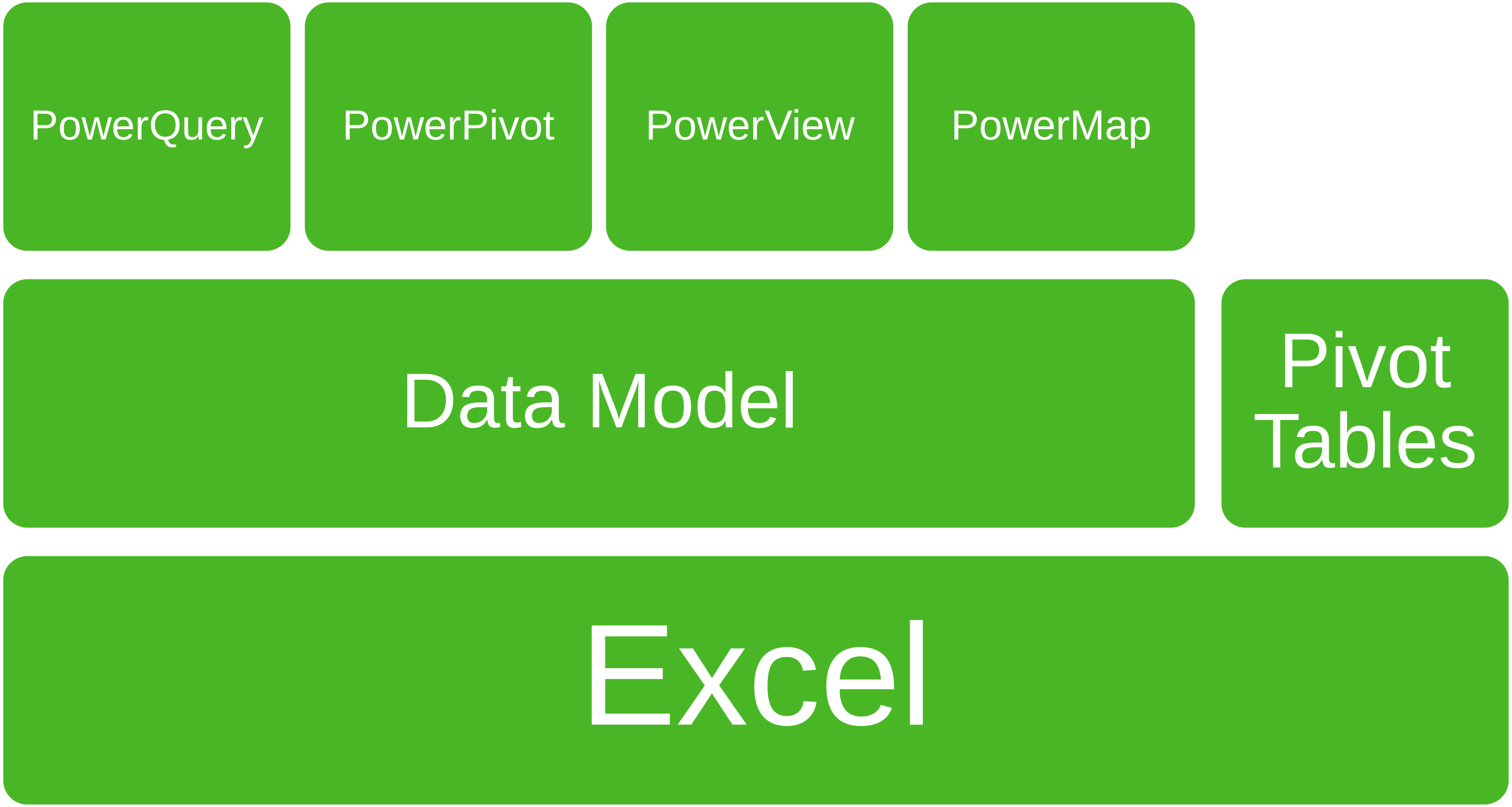
Specialised tools, users who want to “drill through” and perform analysis are left hanging

data-first philosophy, not consistent with software architecture and big-data

Self Service BI

Think data mashups for end-users using commodity software

Excel and friends



PowerQuery

PowerPivot

PowerView

PowerMap

Data Model

Pivot Tables

Excel

Demo – Self Service BI

- Create extra data
- Open RavenDB interface
- Show indexes
- Create the transformer, run the query
- Open the query+transformer url
- Open Excel
- Put url in “From web” in PowerQuery
- Select “Result”, “To Table”, click on expand columns
- Create a PowerView with “Product” and “Quantity”
- Switch to Map, add ZipCode

Big Data

“ Big data is the term for a collection of data sets so large and complex that it becomes difficult to process using on-hand database management tools or traditional data processing applications. The challenges include capture, curation, storage, search, sharing, transfer, analysis and visualization.

thank you !

erik.renaud@modelon.net

www.modelon.net