Design and Modeling with MySQL and PostgreSQL

ALKIN TEZUYSAL

September 2024

Istanbul, Türkiye 2024

PERCONA UNIVERSITY

Open Source Databases Meetup

In partnership with FerretDB

@ask_dba

Let’s get connected!

Alkin Tezuysal - Director of Services @AltinityDB

- Linkedin : https://www.linkedin.com/in/askdba/

Open Source Database Evangelist

- Previously ChistaDATA, PlanetScale, Percona and Pythian as Senior Technical Manager, SRE, DBA
- Previously Enterprise DBA , Informix, Oracle, DB2 , SQL Server

@ask_dba

About Altinity

Founded in 2017 by Alexander Zaitsev, Mindaugas Zukas, Peter Zaitsev, Robert Hodges, Vadim Tkachenko

Focusing on ClickHouse infrastructure engineering and performance operations

What's ClickHouse anyway?

Services and Products around dedicated Altinity.Cloud, Support and Training

www.altinity.com

Catching winds
@svrubato

How to contribute to open source community?

Born to Sail, Forced to Work!

@ask_dba

Recognitions

- Most Influential in Database Community 2022 - The Redgate 100
- MySQL Cookbook, 4th Edition 2022 - O'Reilly Media, Inc.
- MySQL Rockstar 2023 - Oracle (MySQL Community)
- Database Design and Modeling with PostgreSQL and MySQL 2024 - <Packt>

Born to Sail, Forced to Work!

@ask_dba

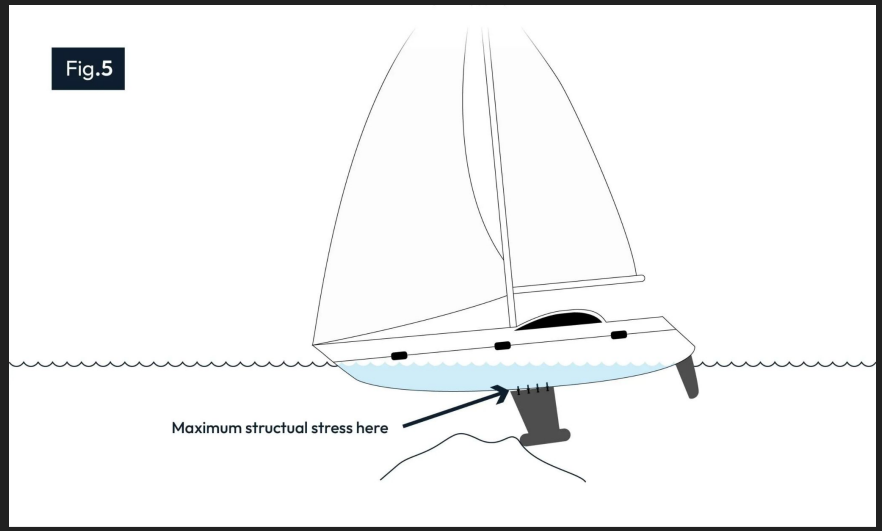
Maritime Trivia

What's a keel?

@ask_dba

Maritime Trivia

What's a keel?



@ask_dba

Keel

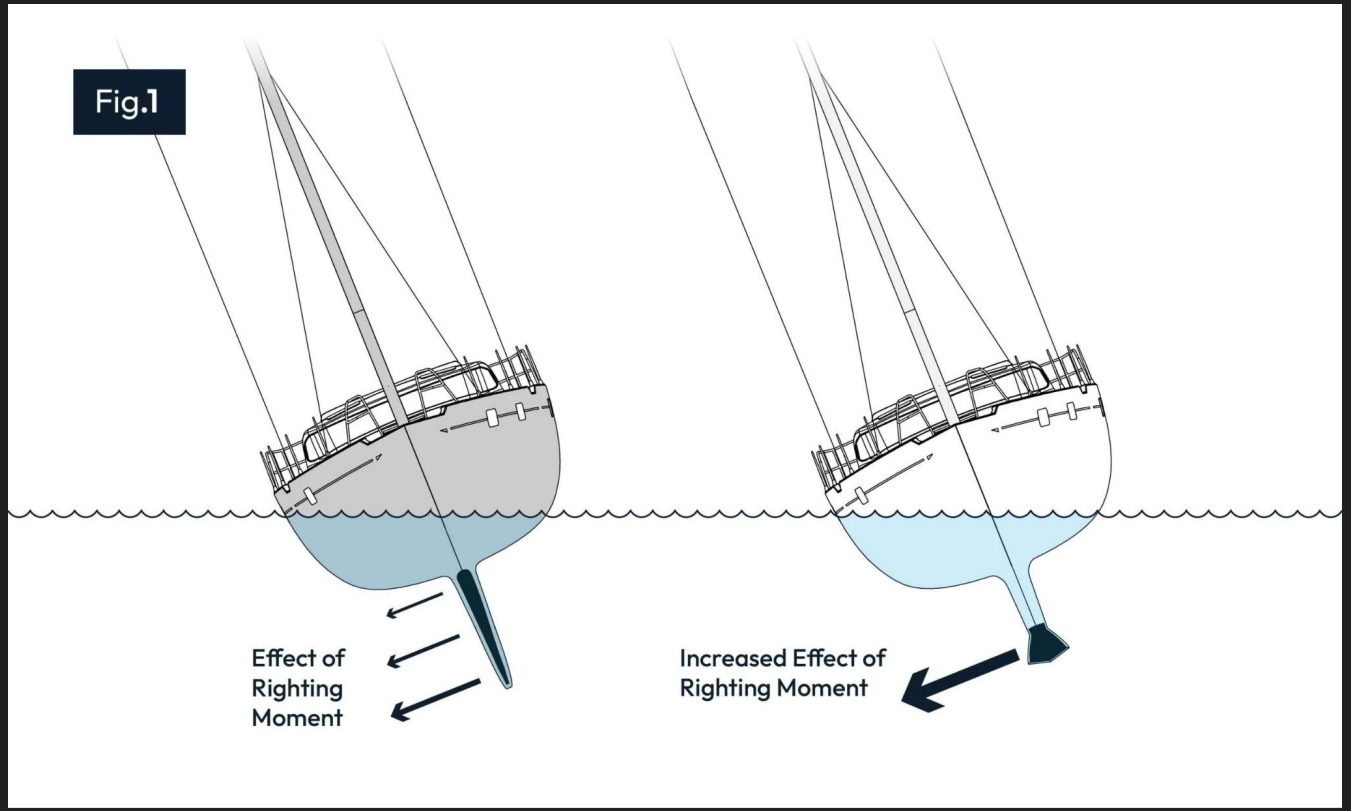


Fig.1
Effect of
Righting
Moment
Increased Effect of
Righting Moment

@ask_dba

Agenda

Database design

Design a schema, create ERDs, and apply normalization techniques

Scaling databases

Scale databases with sharding, replication, load balancing, and implement backup and recovery strategies

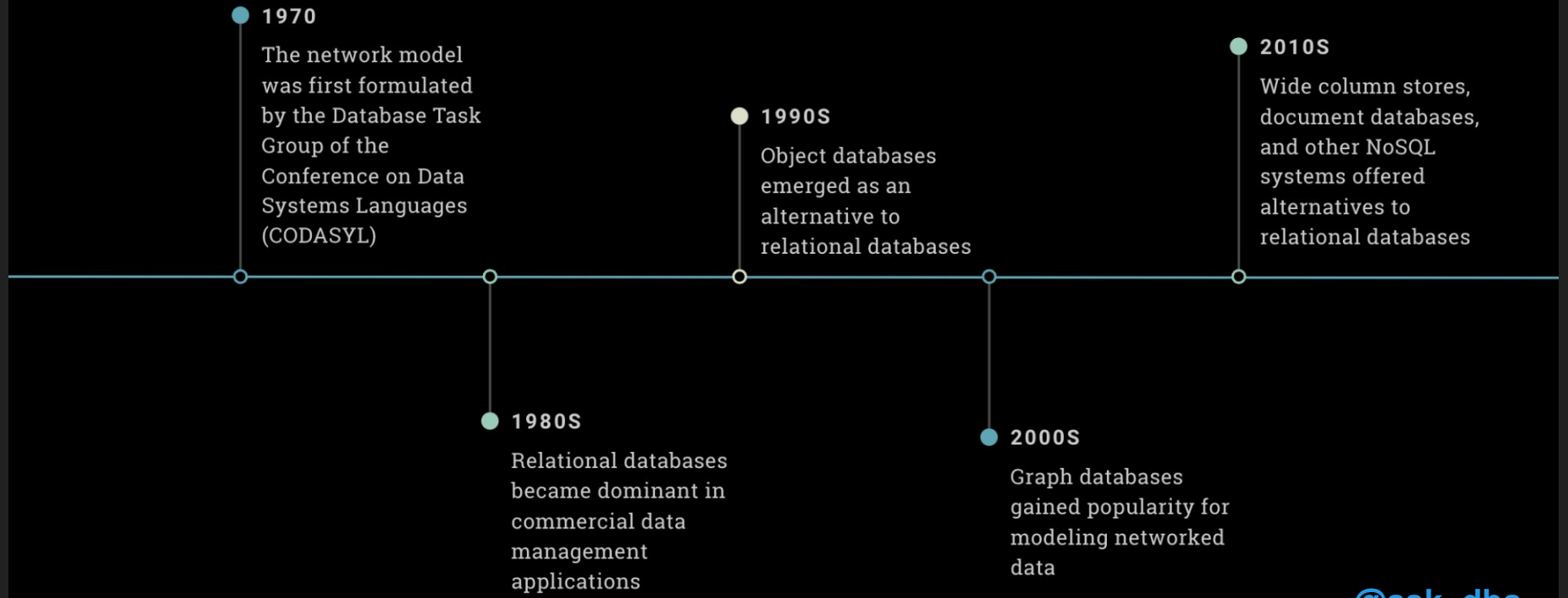
Integrating databases

Integrate databases with web apps, utilize SQL effectively and implement best practices

Emerging trends

Explore emerging trends, including NoSQL databases, cloud databases, and the impact of AI and machine learning

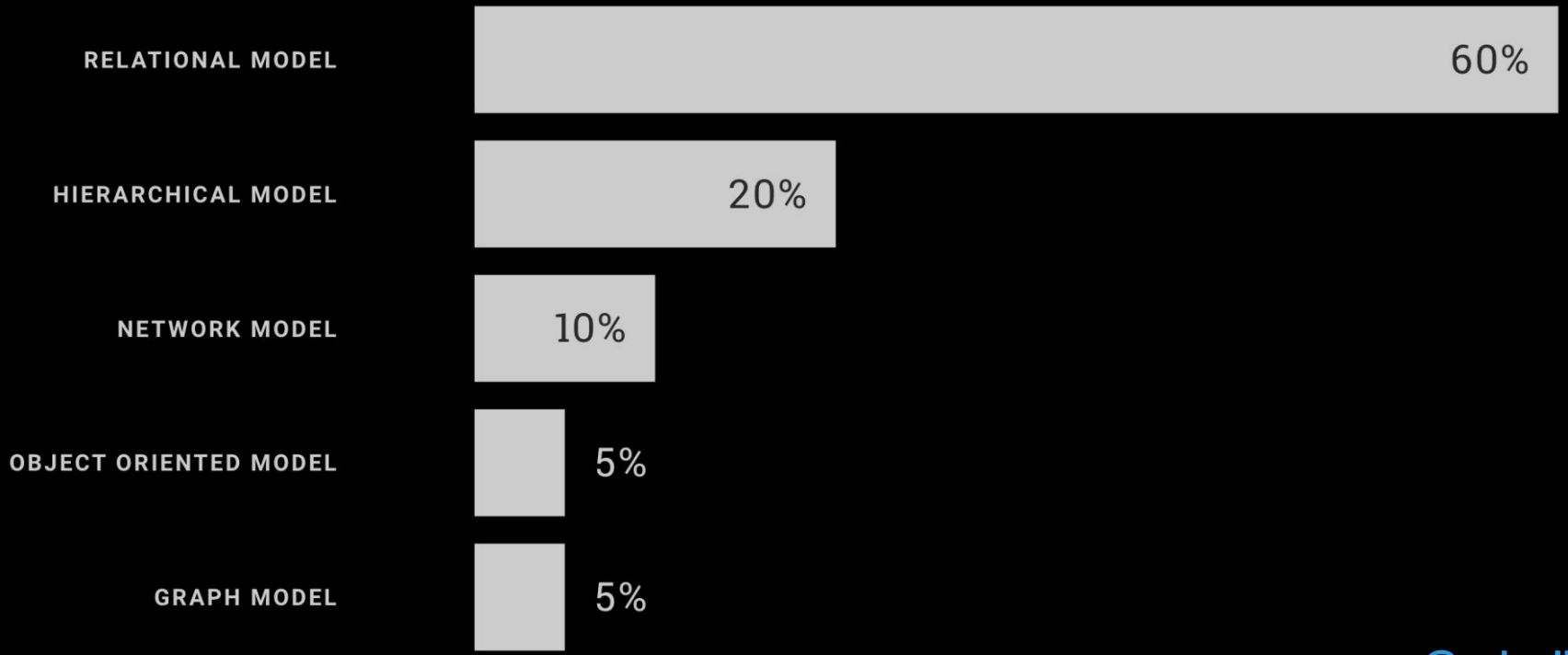
History of Data Models



@ask_dba

Database Design Models

Popularity percentages



@ask_dba

Relational Model

DATA ORGANIZED INTO TABLES

Tables consist of rows and columns representing records and attributes

ROWS REPRESENT RECORDS

Each row represents an individual data entry

COLUMNS REPRESENT ATTRIBUTES

Each column represents a property or attribute of the data

KEYS LINK TABLES

Keys establish relationships between tables and ensure data integrity

THE RELATIONAL MODEL STRUCTURES DATA INTO INTERCONNECTED TABLES VIA KEYS

Normalization

NORMALIZATION REDUCES DATA REDUNDANCY

Normalization organizes data into multiple tables so each piece of information is stored only once

NORMALIZATION IMPROVES DATA INTEGRITY

Normalization eliminates inconsistent data through techniques like removing duplicate data

THERE ARE DIFFERENT NORMAL FORMS

First Normal Form, Second Normal Form, and Third Normal Form have specific criteria for organizing data

NORMALIZATION IS KEY FOR REDUCING REDUNDANCY AND IMPROVING DATA INTEGRITY IN RELATIONAL DATABASES BEFORE APPLYING SQL.

ACID Properties

TRANSACTIONS ARE ATOMIC

Transactions are treated as a single unit that either completes fully or fails entirely.

TRANSACTIONS MAINTAIN CONSISTENCY

Transactions transform the database from one valid state to another, following predefined rules.

TRANSACTIONS ARE ISOLATED

Transactions are executed independently to prevent visibility of intermediate states.

TRANSACTIONS ARE DURABLE

Once committed, a transaction's effects are permanently preserved despite failures.

TOGETHER, THESE ACID PROPERTIES MAKE SQL DATABASES ROBUST AND RELIABLE FOR DEMANDING DATA INTEGRITY NEEDS.

@ask_dba

CAP Theorem

CAP THEOREM IS ABOUT THE TRADE-OFF BETWEEN CONSISTENCY, AVAILABILITY AND PARTITION TOLERANCE

No distributed system can simultaneously provide consistency, availability, and partition tolerance

AVAILABILITY MEANS EVERY REQUEST GETS A RESPONSE

High availability systems aim to always be operational but can't guarantee consistency

PARTITION TOLERANCE MEANS THE SYSTEM WORKS DESPITE NETWORK FAILURES

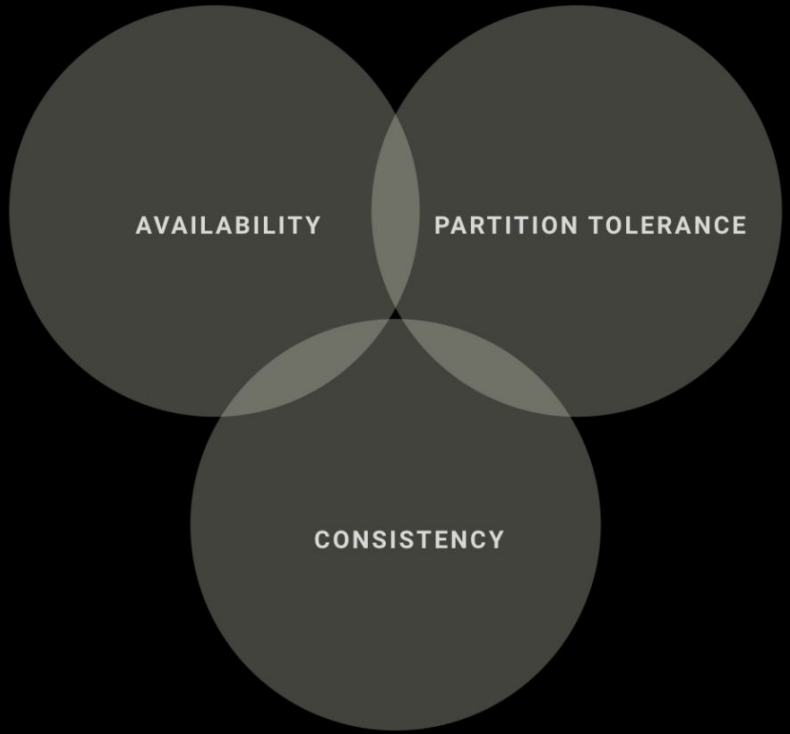
A partitioned system can remain available but consistency is difficult to maintain

CONSISTENCY MEANS ALL NODES SEE THE SAME DATA AT THE SAME TIME

Strong consistency provides linearizability but reduces availability

CAP THEOREM STATES IT'S IMPOSSIBLE FOR A DISTRIBUTED SYSTEM TO PROVIDE STRONG CONSISTENCY AND HIGH AVAILABILITY IN PRESENCE OF NETWORK PARTITIONS. SYSTEMS NEED TO TRADE-OFF BETWEEN THESE GUARANTEES. @ask_dba

CAP Theorem Visualized



@ask_dba

Data Models in NoSQL

KEY-VALUE MODEL

Stores data as key-value pairs, ideal for caching and simple data retrieval

COLUMN-FAMILY MODEL

Organizes data in column families, suitable for wide-column stores

GRAPH MODEL

Represents data as nodes and edges, making it suitable for complex relationships and graph-based operations

NOSQL DATABASES ADOPT DIFFERENT DATA MODELS LIKE KEY-VALUE, COLUMN-FAMILY, AND GRAPH MODELS TO STORE AND ORGANIZE DATA BASED ON THE USE CASE REQUIREMENTS.

CAP Theorem for NOSQL

CAP THEOREM PROPOSED BY ERIC BREWER

States consistency, availability and partition tolerance cannot be achieved simultaneously in distributed systems

MUST MAKE TRADEOFFS BETWEEN CAP PROPERTIES

Designers must choose which properties to optimize based on requirements and constraints

CAP THEOREM IS FUNDAMENTAL FOR NOSQL DATABASE DESIGN CHOICES INVOLVING DISTRIBUTED SYSTEMS

Key Terms and Data Models

ENTITY

An entity is a distinct object or concept represented in a database, like a person, place, thing or event.

ATTRIBUTE

Attributes are characteristics or properties of an entity that define specific pieces of information stored about it.

RELATIONSHIP

A relationship is a connection between entities defining how they are associated with each other.

KEY TERMS LIKE ENTITY, ATTRIBUTE AND RELATIONSHIP ARE THE BUILDING BLOCKS OF DATA MODELS IN DATABASE DESIGN.

Hierarchical Data Model

HIERARCHICAL MODEL ORGANIZES DATA IN PARENT-CHILD RELATIONSHIPS

Data is structured in a tree with each parent node having multiple children nodes and each child having only one parent

RELATIONSHIPS ARE UNIDIRECTIONAL FROM PARENT TO CHILD

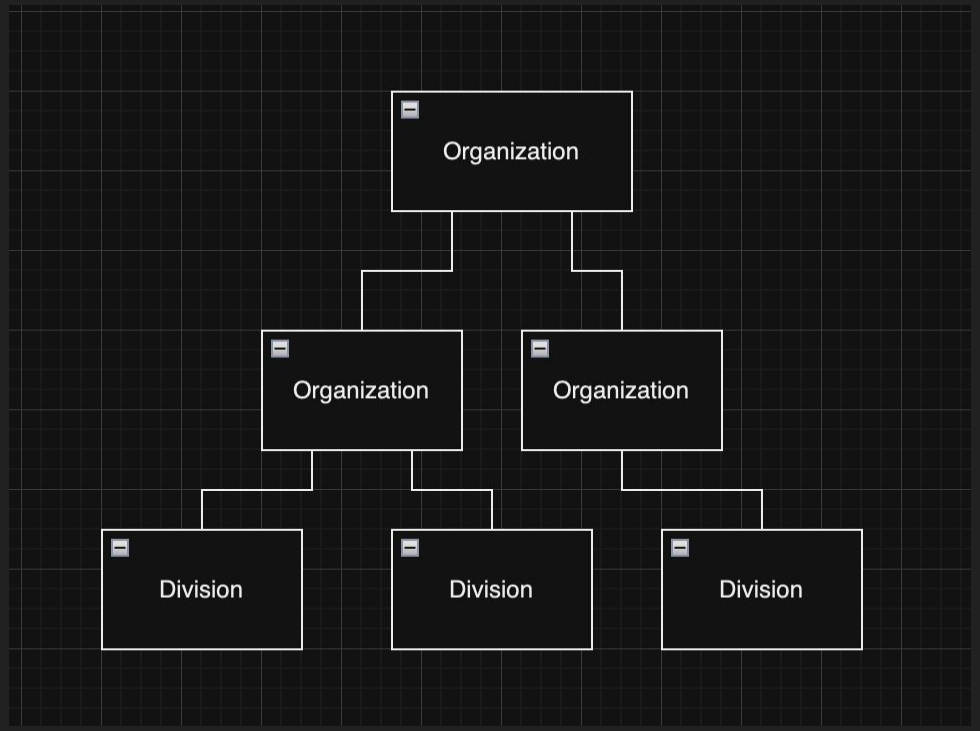
The relationship flows only from parent to child, a child cannot be parent of its own parent

EACH PARENT CAN HAVE MULTIPLE CHILDREN

A parent node in the hierarchy can have multiple child nodes under it representing different branches

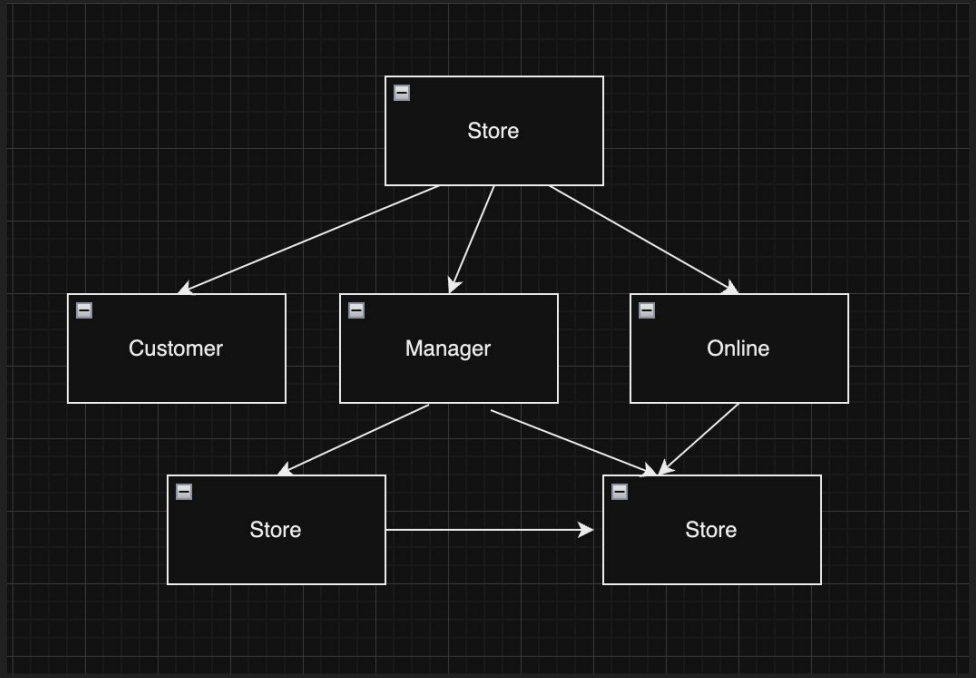
THE HIERARCHICAL MODEL ALLOWS ORGANIZING DATA IN A TOP-DOWN STRUCTURE WITH CLEAR RELATIONSHIPS BETWEEN PARENTS AND CHILDREN, SUPPORTING MULTIPLE CHILDREN UNDER EACH PARENT.

Hierarchical Model

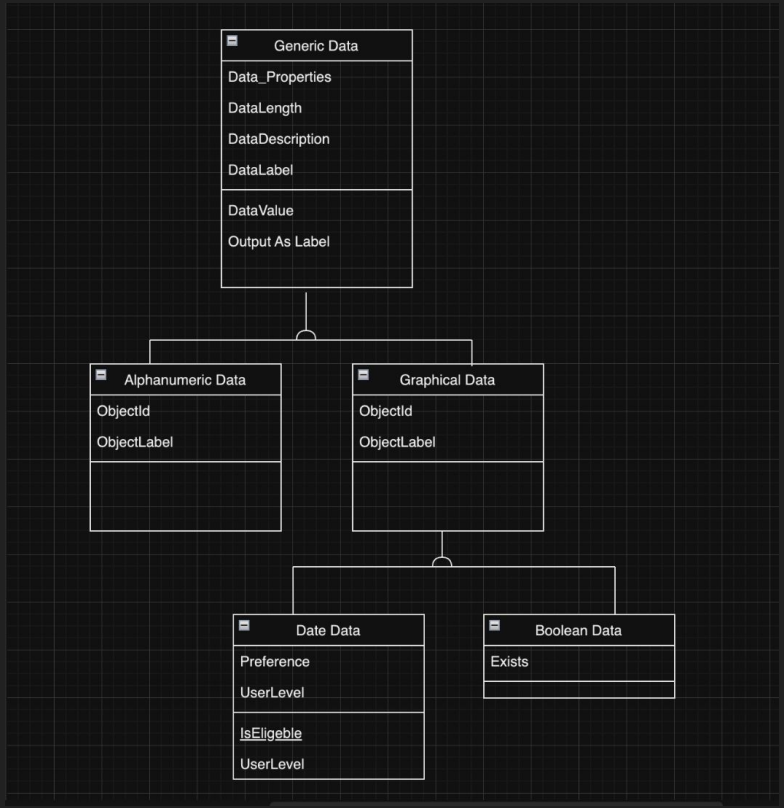


@ask_dba

Network Model

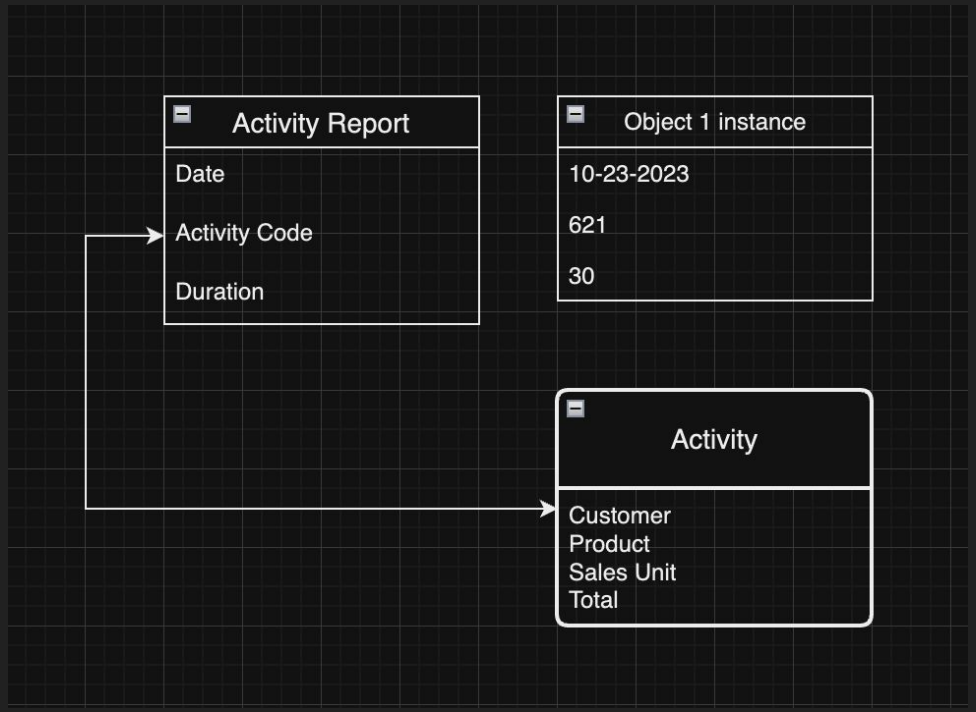


Object Oriented Model

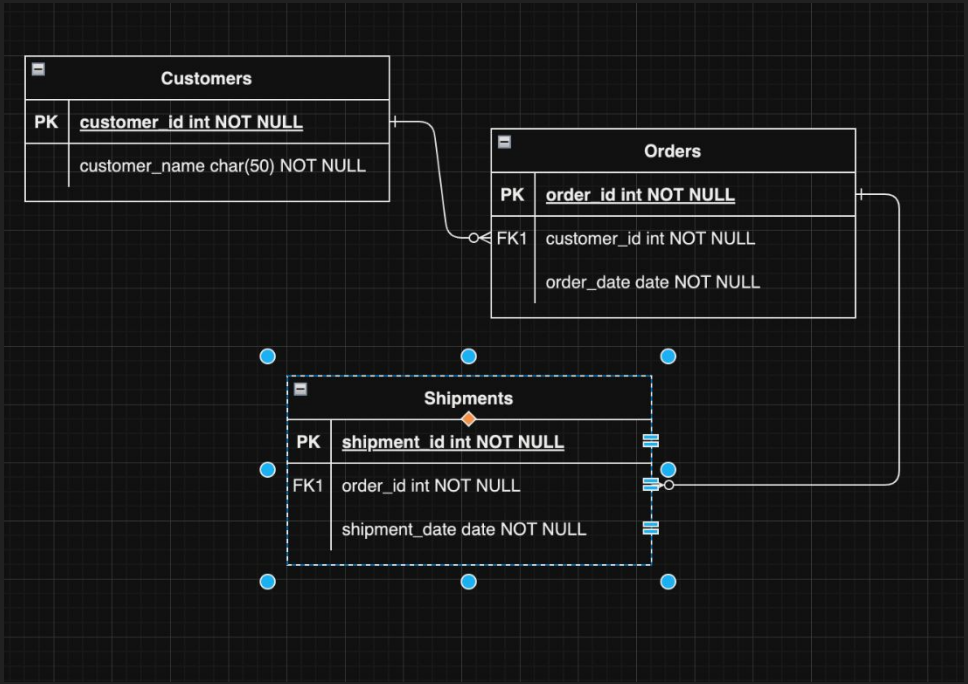


@ask_dba

Object Relational Model

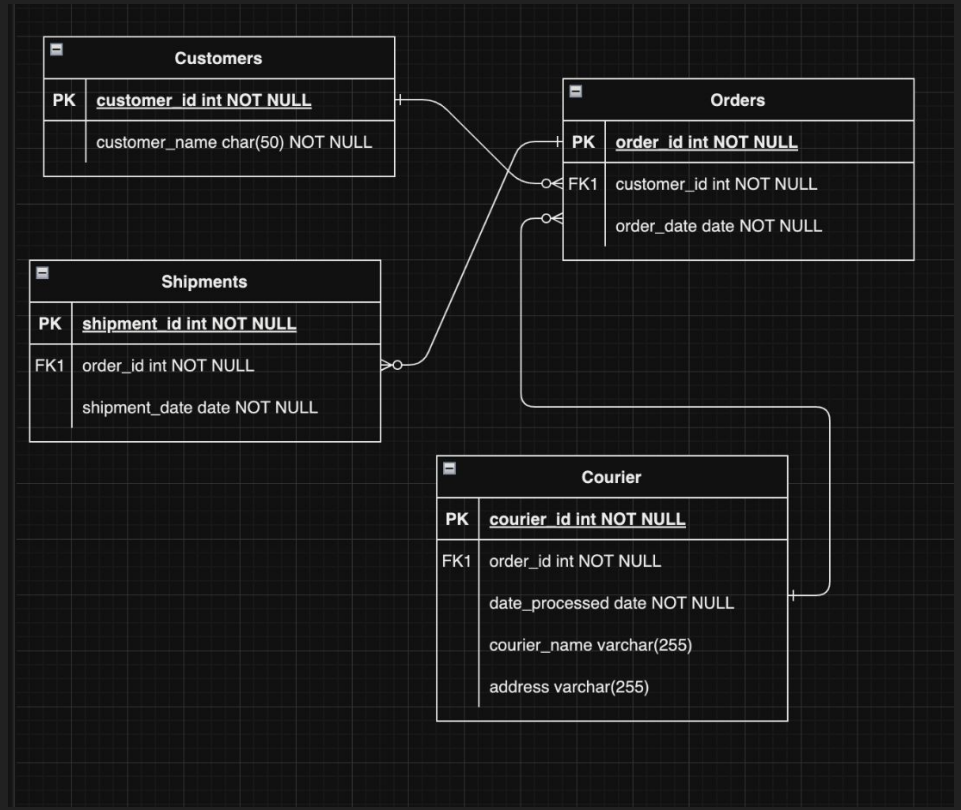


Relational Model



@ask_dba

Entity Relationship Model



@ask_dba

Let's focus on ER Model

ERD - ENTITY RELATIONSHIP DIAGRAM

ERD shows entities, attributes, relationships between entities

NORMALIZATION

Breaks data into multiple tables to avoid redundancy and anomalies

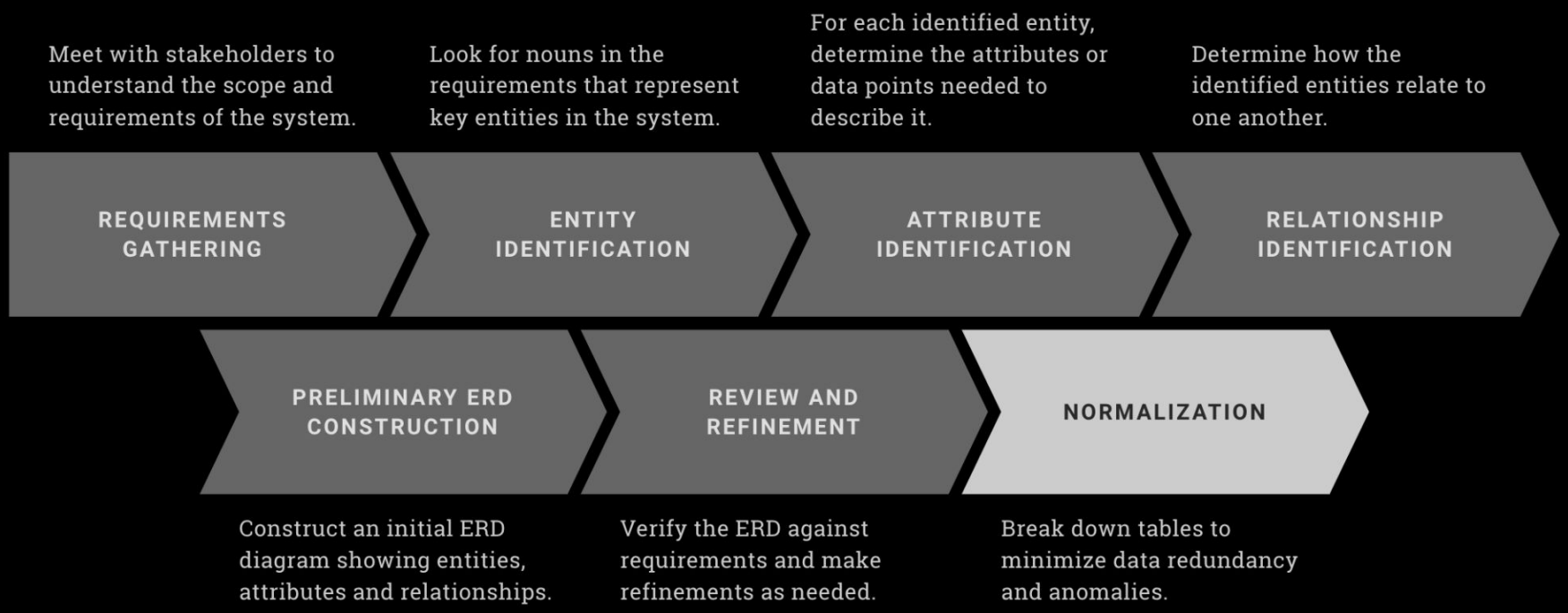
KEYS

Primary key uniquely identifies rows, foreign key links tables

DATABASE DESIGN INVOLVES USING MODELS LIKE ERD TO STRUCTURE DATA INTO NORMALIZED TABLES WITH PROPER KEYS.

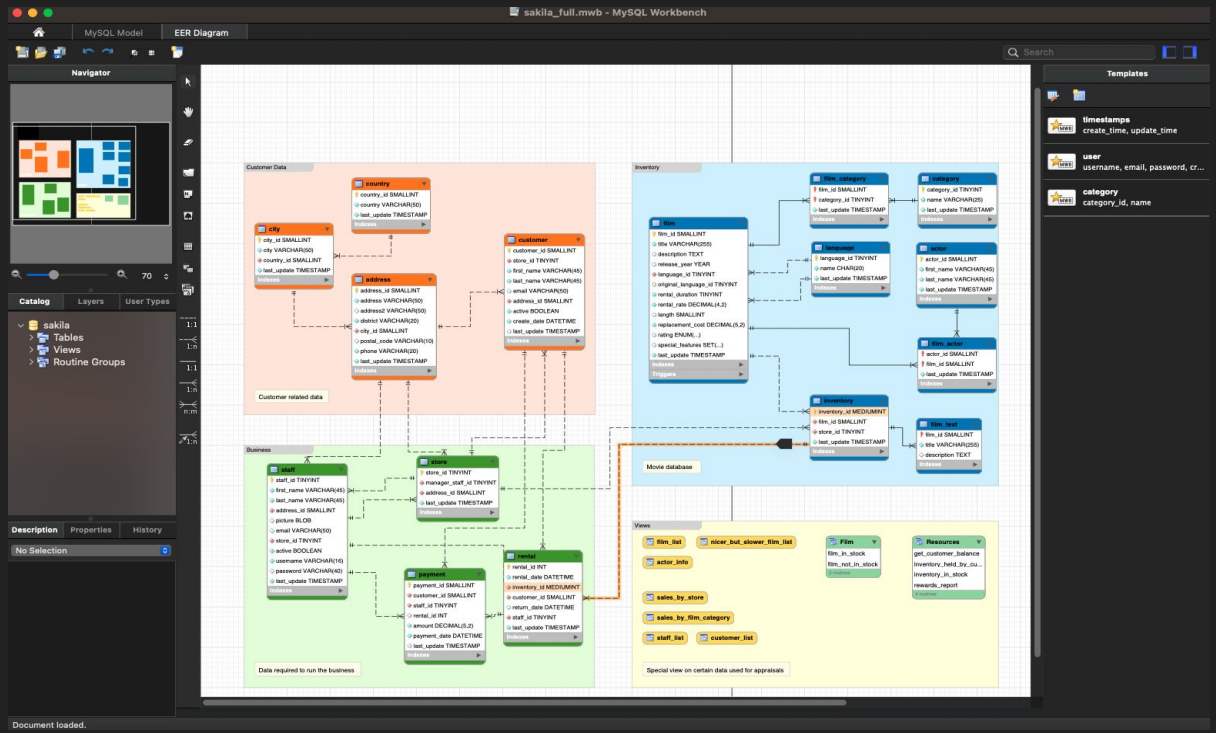
@ask_dba

Steps for ER Database Design



@ask_dba

MySQL Workbench



@ask_dba

MySQL Workbench Features

MODELING

Allows designing database models visually

QUERY BUILDING

Provides a visual query builder to construct complex queries

VALIDATION

Validates models for issues before creating databases

ADMINISTRATION

Allows administering MySQL servers and databases

MYSQL WORKBENCH PROVIDES POWERFUL TOOLS FOR DATABASE MODELING, VALIDATION, QUERYING, AND ADMINISTRATION.

First Normal Form

FIRST NORMAL FORM REQUIRES NO DUPLICATE ROWS

Each row in a table should be unique with a primary key to identify it

ONLY ATOMIC VALUES IN COLUMNS

Columns should not contain sets, arrays or lists

FOLLOWING 1NF ENSURES DATA INTEGRITY AND AVOIDS UPDATE ANOMALIES

Second Normal Form

A TABLE MUST BE IN 1NF TO QUALIFY FOR 2NF

2NF requires the table to already satisfy all conditions for 1NF before applying additional criteria

NO PARTIAL DEPENDENCIES ALLOWED

All non-key attributes must depend fully on the entire primary key, not just part of it

COMPOSITE KEYS COMPLICATE 2NF

With a composite primary key, ensuring no partial dependencies on just part of the key is trickier

2NF BUILDS ON 1NF BY FURTHER RESTRICTING ALLOWABLE DEPENDENCIES TO HELP NORMALIZE THE TABLE STRUCTURE

3rd Normal Form

ENTITY NORMALIZATION

Breaking down data into atomic units to remove redundancy

ATOMICITY

Each attribute contains only a single value

REMOVING PARTIAL DEPENDENCIES

No non-key attribute depends on just part of a key

3RD NORMAL FORM ELIMINATES DATA REDUNDANCY AND ANOMALIES BY ENSURING ATTRIBUTES ARE DEPENDENT ON THE PRIMARY KEY

@ask_dba

Isolation Levels - Ensuring Data Integrity

- **ISOLATION LEVELS**
  Isolation levels define the degree to which concurrent transactions can interact with each other, balancing data integrity and efficiency.
- **LEVELS OF ISOLATION**
  The choice of isolation level impacts data consistency, concurrency, and system performance.
- **ENSURING DATA INTEGRITY**
  Isolation levels play a key role in ensuring data consistency in databases beyond just ACID properties.

Isolation Levels in MySQL

MYSQL USES REPEATABLE READ ISOLATION LEVEL BY DEFAULT

REPEATABLE READ prevents dirty reads and non-repeatable reads but allows phantom reads

ISOLATION LEVELS BALANCE CONSISTENCY AND CONCURRENCY

Options like READ UNCOMMITTED offer less isolation but more concurrency

CHOOSE ISOLATION LEVEL CAREFULLY FOR APP NEEDS

Consider data consistency needs and performance goals

UNDERSTANDING ISOLATION LEVELS IN MYSQL HELPS OPTIMIZE FOR AN APP'S SPECIFIC DATABASE NEEDS

@ask_dba

Scaling Databases

SCALING DATABASES IS CRUCIAL

Database scaling is important for maintaining performance, reliability, and availability as data volumes grow.

EXPONENTIAL DATA GROWTH

Data from social media, IoT, and enterprise apps is growing exponentially, driving the need to scale databases.

STRATEGIES FOR SCALING

Common database scaling techniques include sharding, replication, caching, and moving to the cloud.

DATABASE SCALING THROUGH SHARDING, REPLICATION, CACHING, AND THE CLOUD IS KEY FOR HANDLING EXPONENTIAL DATA GROWTH FROM MODERN APPLICATIONS

Primary Methods of Database Scaling

HORIZONTAL SCALING

Adding more machines or nodes to existing database system

VERTICAL SCALING

Upgrading existing hardware of database server

HORIZONTAL SCALING EXPANDS CAPACITY BY ADDING RESOURCES, VERTICAL SCALING UPGRADES EXISTING RESOURCES FOR BETTER PERFORMANCE. @ask_dba

Scaling MySQL Vertically vs Horizontally

VERTICAL SCALING COMES NATURALLY

Vertical scaling is easier and faster to implement initially

VERTICAL SCALING ADDS RESOURCES

Additional CPU, RAM can be added to handle more load

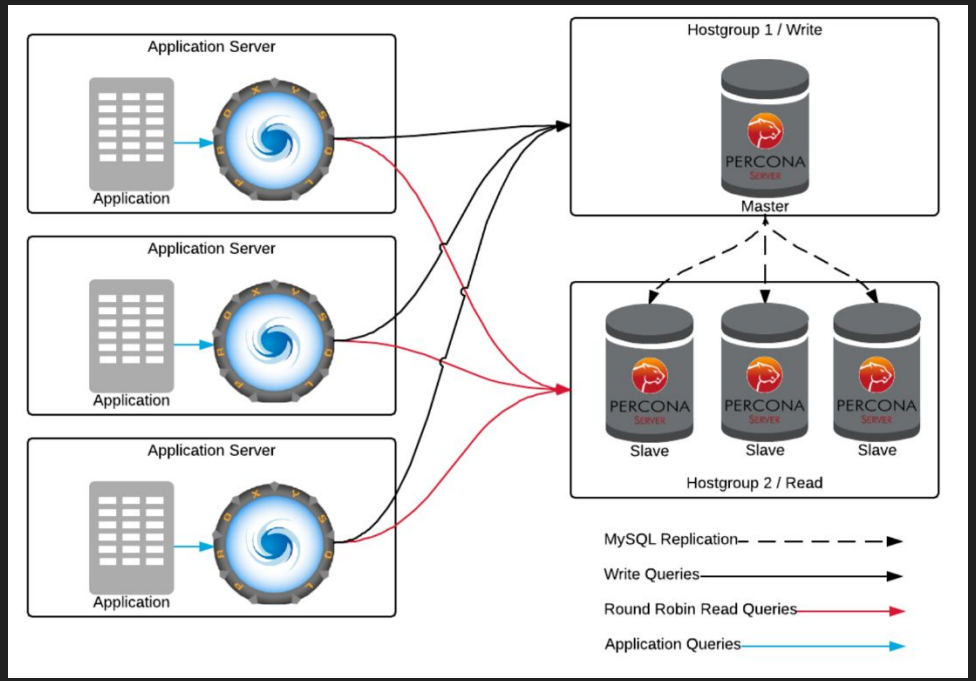
VERTICAL SCALING HAS LIMITS

There is a physical limit to how much you can vertically scale

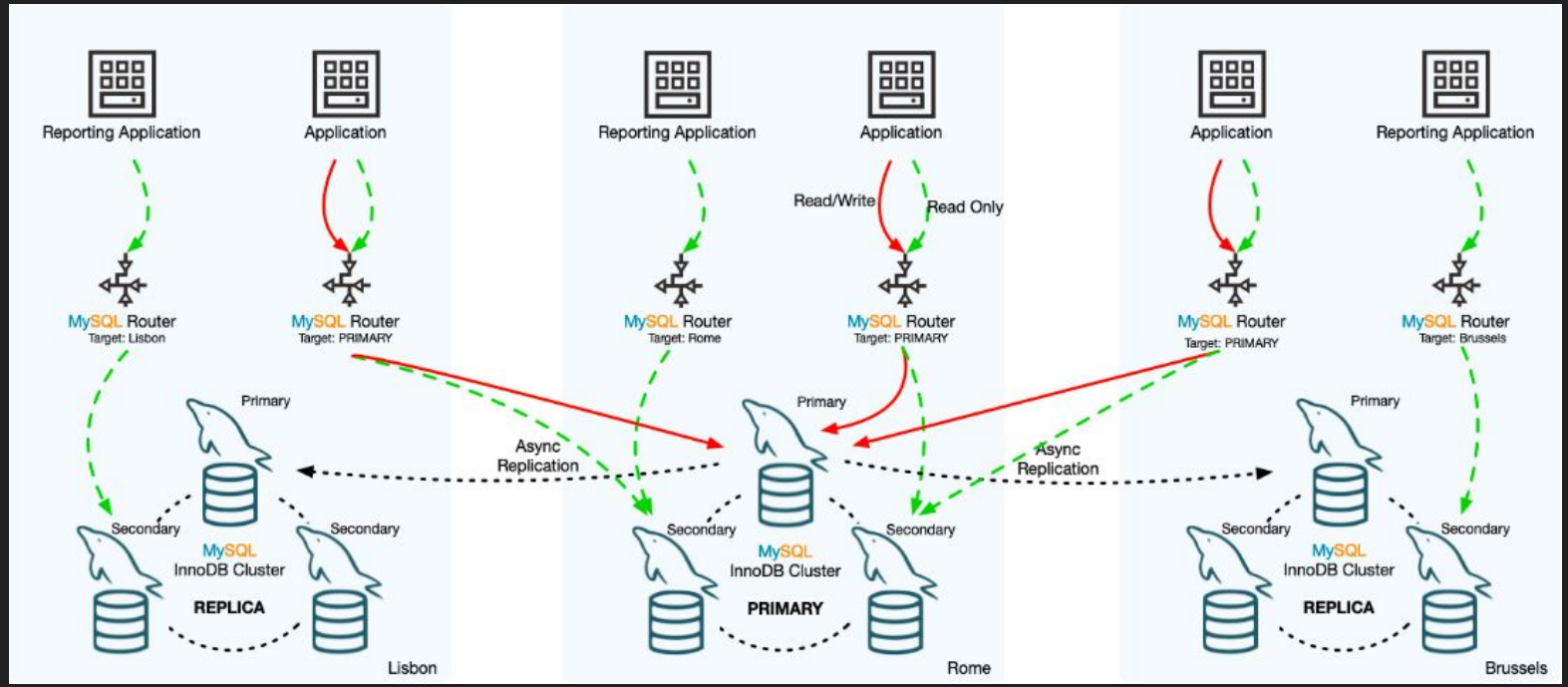
VERTICAL SCALING IS A GOOD SHORT-TERM TACTIC, BUT HORIZONTAL SCALING IS BETTER FOR THE LONG-TERM

@ask_dba

Proxy Solution

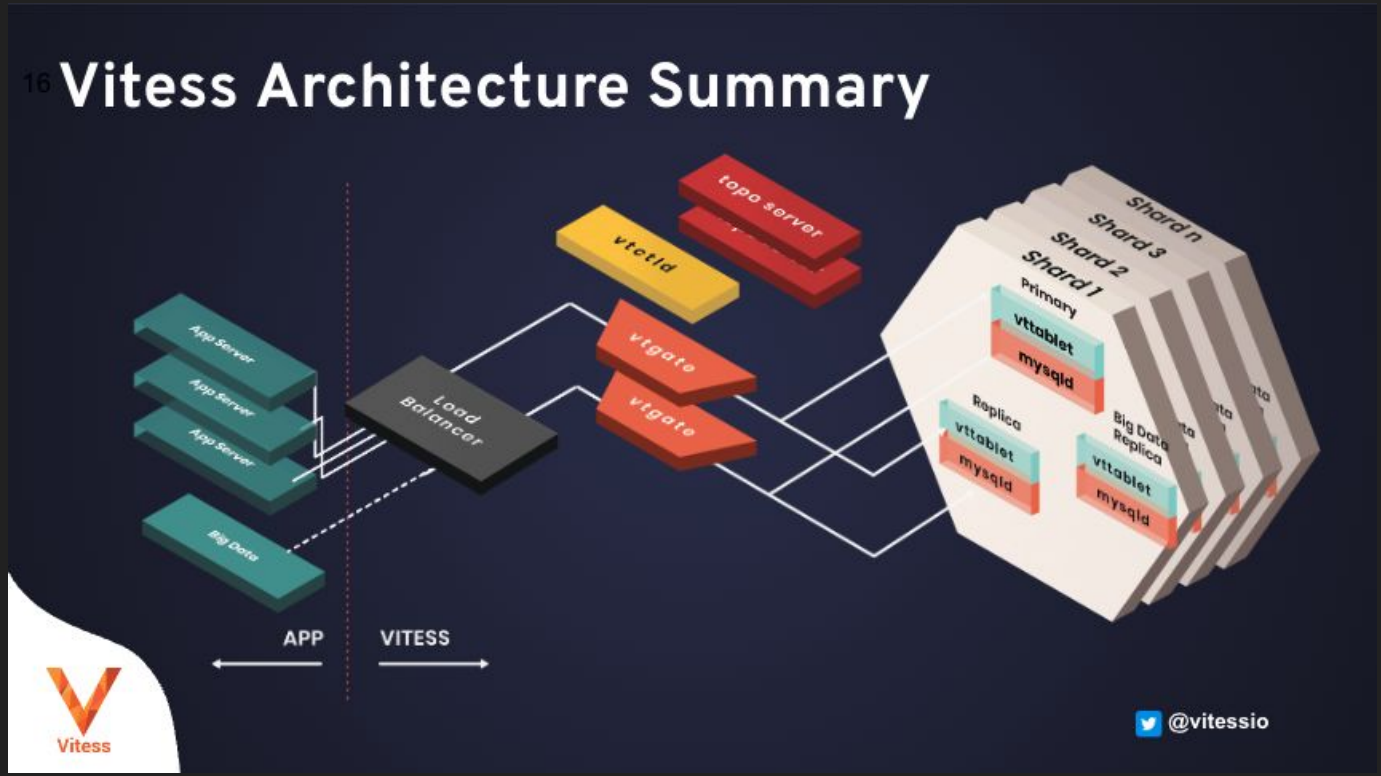


Clustering Solutions



@ask_dba

Sharding Solutions



Serverless MySQL Overview

SERVERLESS MYSQL SOLUTIONS

Several cloud providers offer serverless MySQL options like AWS Aurora Serverless, Google Cloud SQL, Azure Database for MySQL, and PlanetScale.

INSTANT SCALING

These solutions provide instant and automatic scaling based on application needs, aligning with serverless principles.

HIGH AVAILABILITY

They offer high availability and resilience by dynamically adjusting capacity and resources.

REDUCED OPERATIONAL BURDEN

Serverless MySQL offloads database management, freeing developers to focus on apps.

SERVERLESS MYSQL SOLUTIONS FROM MAJOR CLOUD PROVIDERS OFFER AUTOMATIC SCALING, HIGH AVAILABILITY, AND REDUCED OPS BURDEN.

@ask_dba

The Future of Database Management

INTEGRATION OF ADVANCED TECHNOLOGIES

The future of database management will see the integration of advanced technologies like AI, ML, automation etc. to enable more intelligent and efficient data handling.

SHIFT TOWARDS FLEXIBLE AND SECURE SOLUTIONS

There will be a move towards more flexible, scalable and secure database solutions that can meet changing business needs and safeguard data integrity.

LEVERAGING DATA IN NEW WAYS

New opportunities will emerge to analyze, visualize and extract value from data across industries in ways that drive innovation and digital transformation.

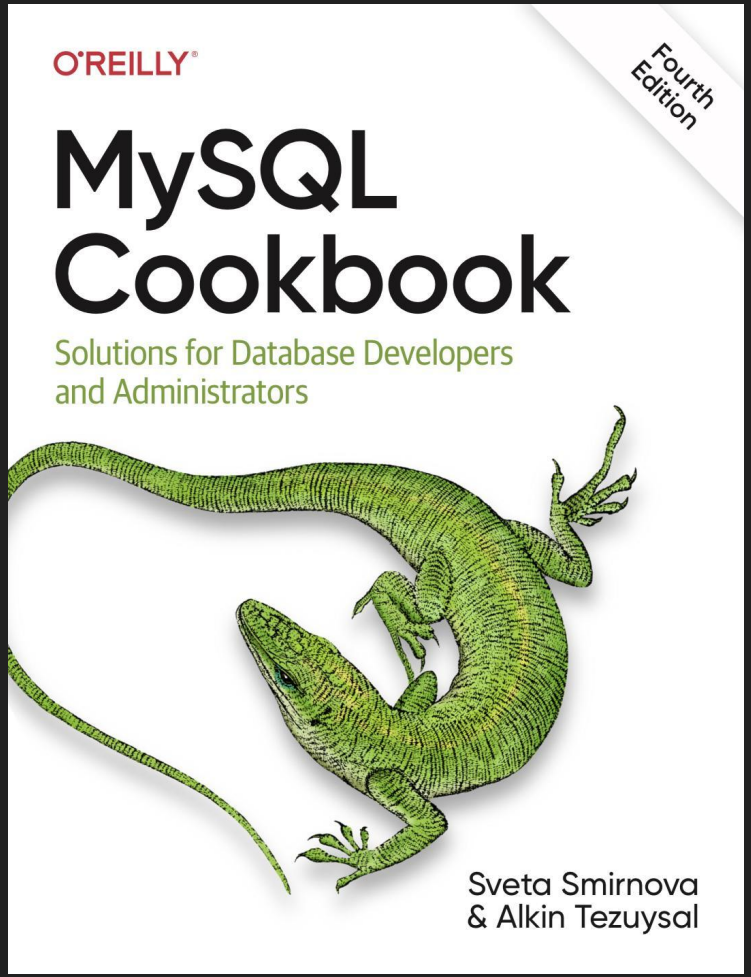
THE FUTURE OF DATABASE MANAGEMENT LOOKS PROMISING, WITH TECHNOLOGY INTEGRATION, FLEXIBLE ARCHITECTURES AND INNOVATIVE DATA USE CASES SET TO ADDRESS EXISTING CHALLENGES AND UNLOCK NEW POTENTIAL.

1ST EDITION

Database Design and Modeling with PostgreSQL and MySQL

Build efficient and scalable databases for modern applications using open-source databases

ALKIN TEZUYSAL | IBRAR AHMED


O'REILLY
Fourth Edition
MySQL Cookbook
Solutions for Database Developers and Administrators
Sveta Smirnova & Alkin Tezuysal

@ask_dba

In partnership with FerretDB

References & Credits

- Vitess
- MySQL 8.0 Reference Manual :: 23 InnoDB Cluster
- ProxySQL
- MySQL Workbench
- Percona XtraDB Cluster - MySQL Software
- PlanetScale
- TiDB: The Advanced Distributed SQL Database
- YugabyteDB: High-Performing Distributed SQL Database