Data Modelling

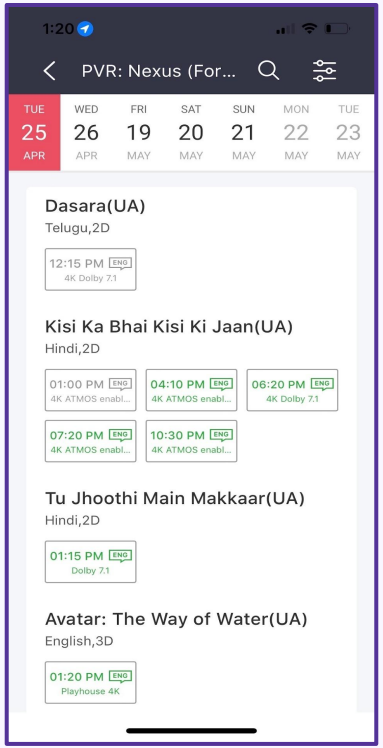
Sancheeta Kaushal | Ex-EM Blinkit

Problem Statement

Problem Solving Case: Bookmyshow

Bookmyshow is a ticketing platform where you can book tickets for a movie show. As part of this assignment, we need to build API's for the following feature. As a user, I can select any theatre in the city. On selecting the theatre, I should be able to see the dates of next 7 days. I can click on any date and the page should load to give me all the movies in that theatre on that given date. Movies should contain details of all the showtimes.

Problem Statement



The image represents the feature described on the next page. You have to code the APIs along with appropriate table structures for powering this UI.

Airtribe

What all skills you need to learn to build this ?

- Storing data in a table.
- Accessing data using SQL Queries.
- Normalizing data for efficient storage.
- Joining data across multiple tables.
- Understanding the role of transactions.
- Solving for the database challenges
- Accessing data from the tables in the application layer using ORMs.

What all skills you need to learn to build this ?

- Storing data in a table.
- Accessing data using SQL Queries.
- Normalizing data for efficient storage.
- Joining data across multiple tables.
- Understanding the role of transactions.
- Solving for the database challenges
- Accessing data from the tables in the application layer using ORMs.

Airtribe

Agenda

- Challenges with databases
- Monitoring Database
- Monitoring Queries
- ORMs

Challenges with databases

All these challenges can break our systems and bring downtime.

How do we know which challenges to tackle first ?

Database Monitoring

But what is it that we should be monitoring ?

Metrics - Database and Queries

Percona Monitoring and Management & Datadog

Alerting based on 99 percentile values

Metrics for Database

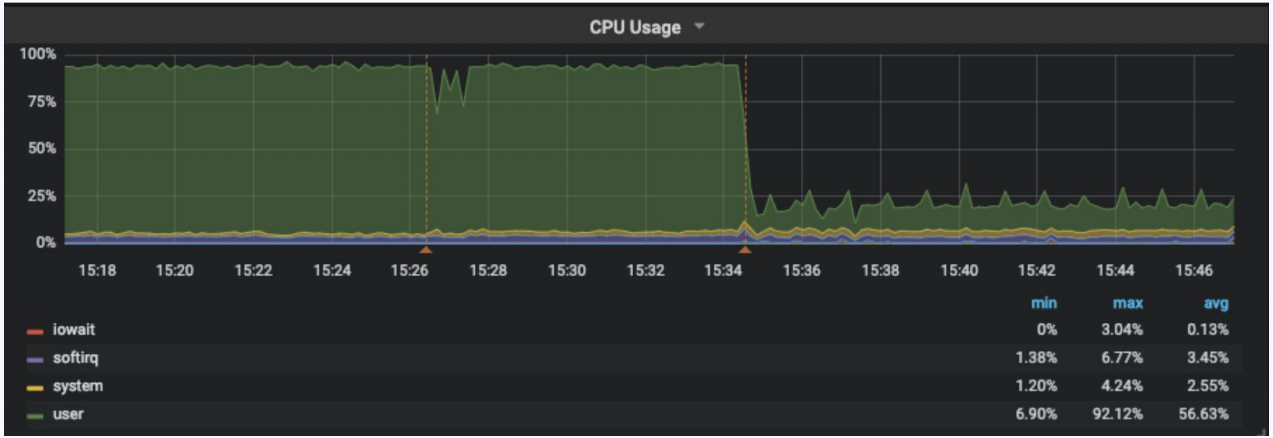
- CPU utilisation (Load on the database)
- Database/Table growth (Disk utilisation)
- Queries per second (Performance)
- Database connection count (Open connections)
- Buffer Pool size for RAM (Temporary Storage Memory from disk)
- Replication Delay (Primary and Replica Sync)
- Network Latency (Servers in India vs Oregon)
- Error and Exception

Metrics for Database

Culprit Query

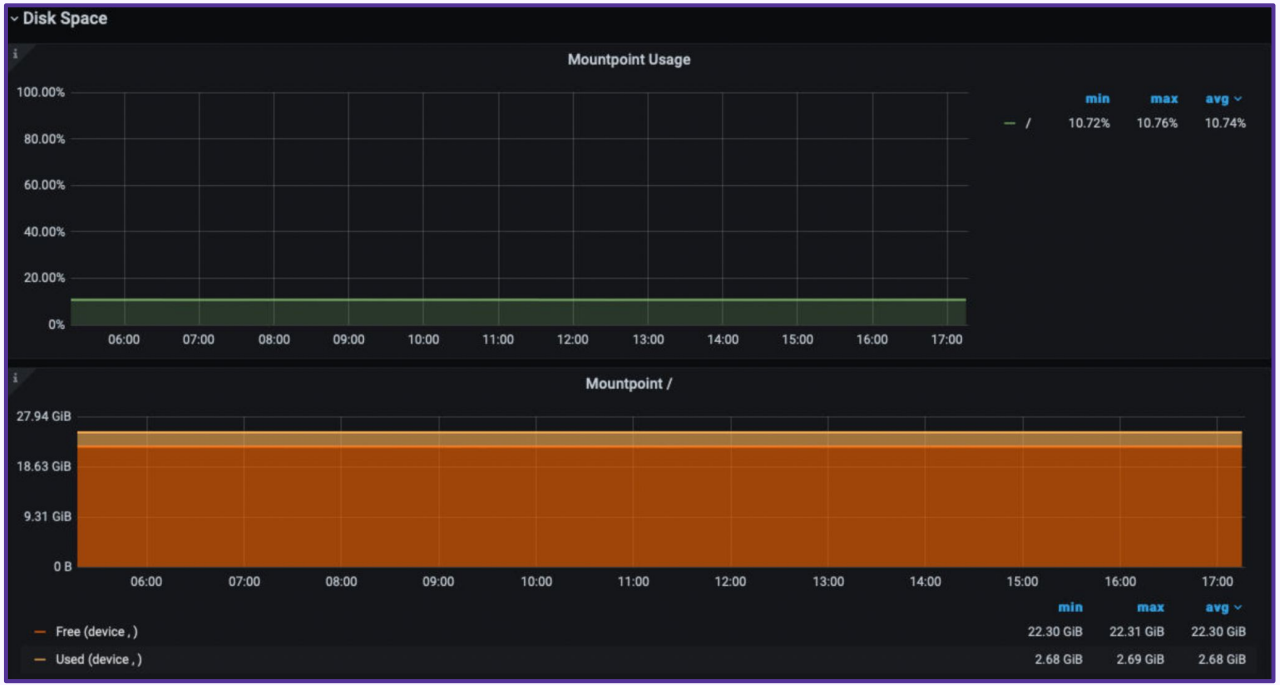
Select * from purchase_orders where status in (‘CREATED’, ‘DELIVERED’, ‘IN PROGRESS’) and other relevant conditions;

Selected rows - 150000



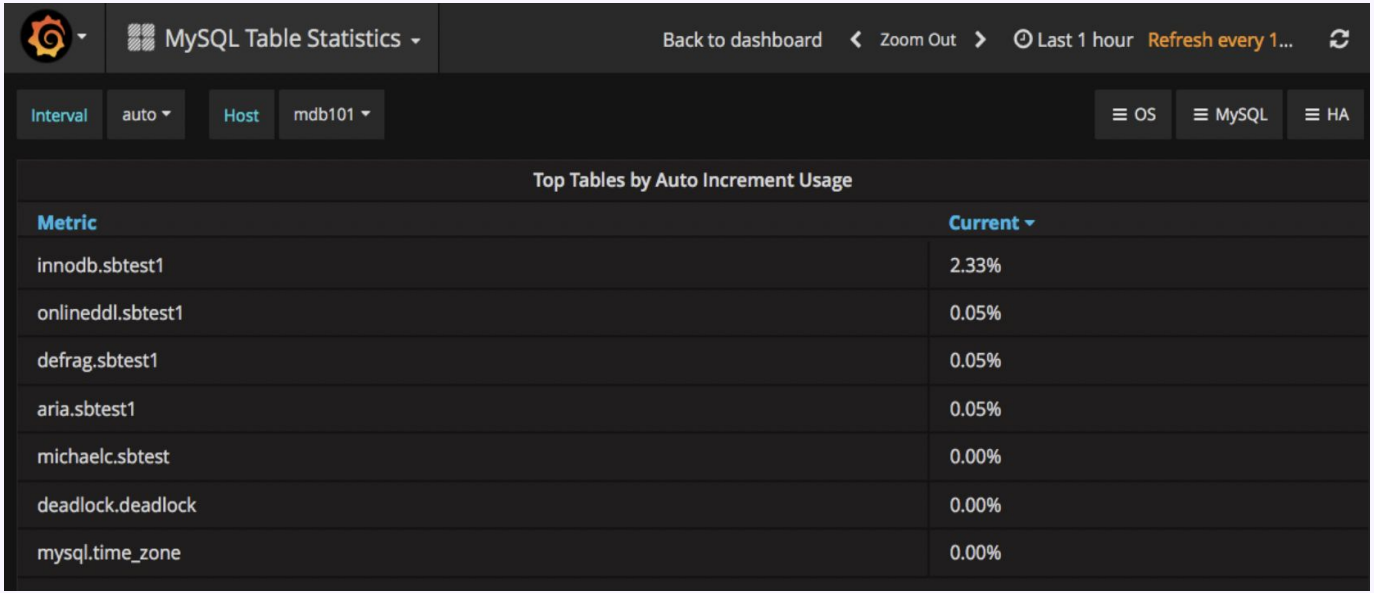
Airtribe

Metrics for Database



Airtribe

Metrics for Database

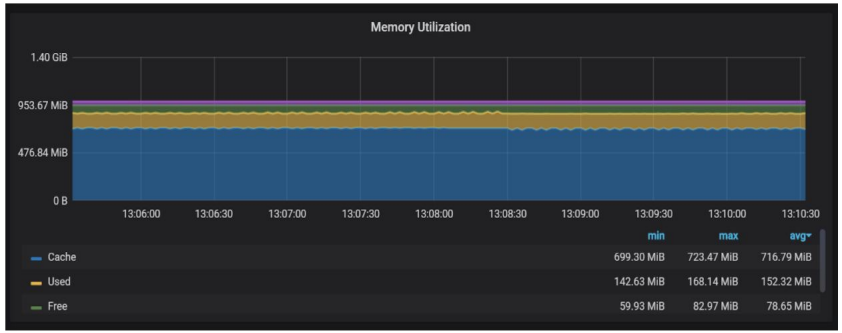


Airtribe

Metrics for Database

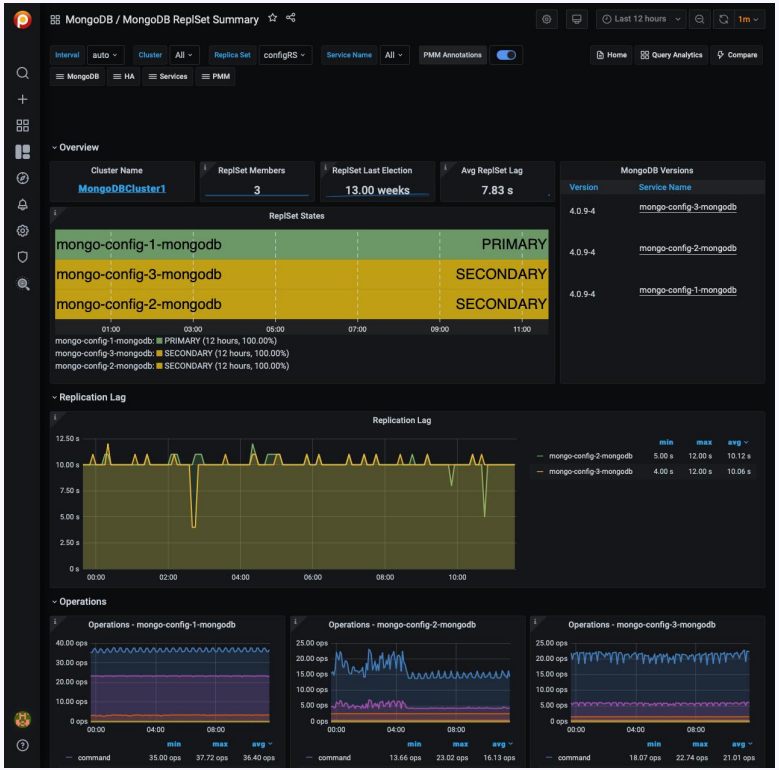
Airtribe

Metrics for Database



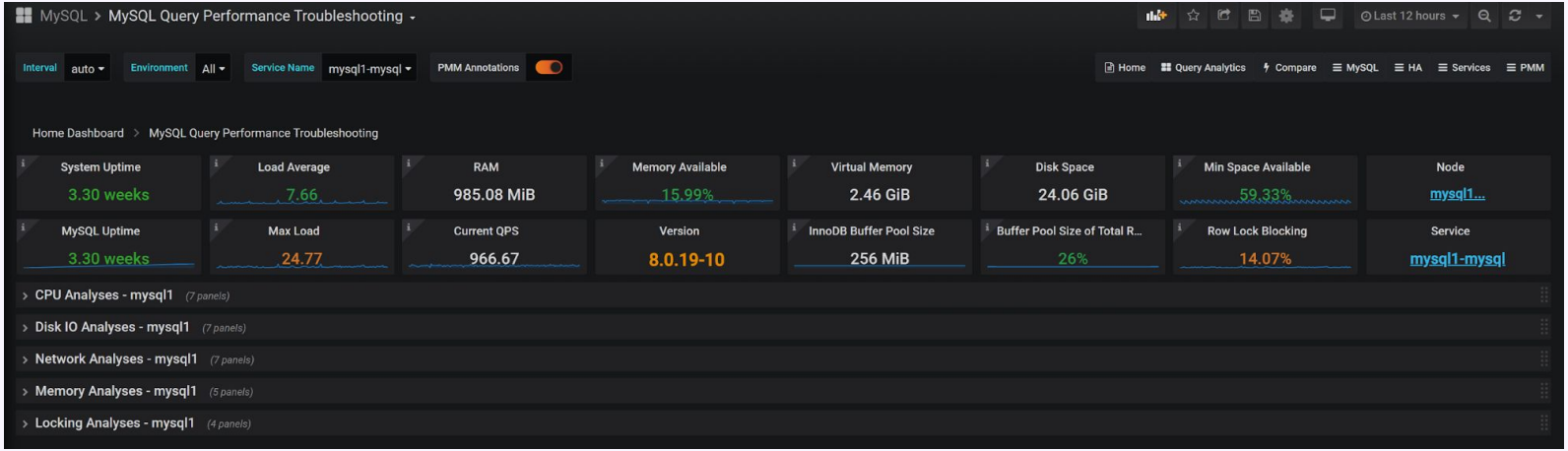
Airtribe

Metrics for Database



Airtribe

Metrics for Database

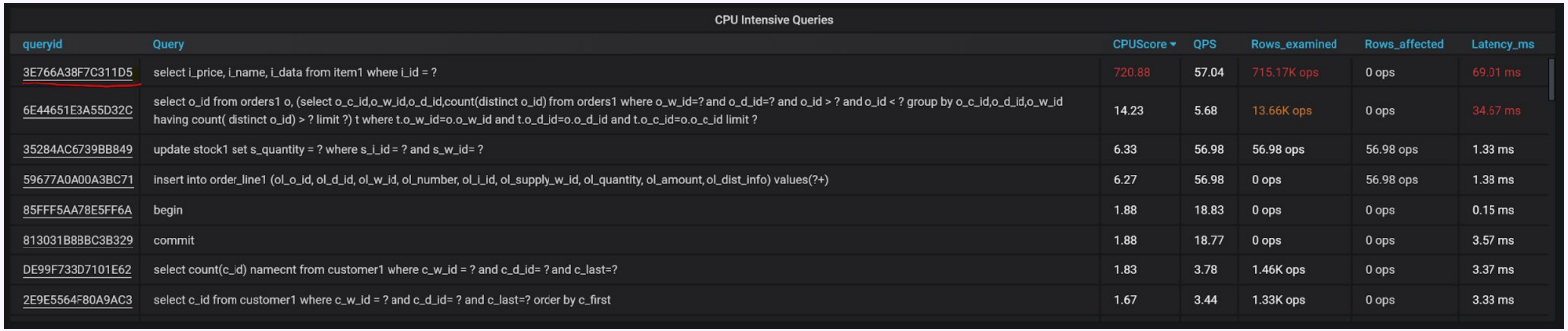


Airtribe

Metrics for Queries

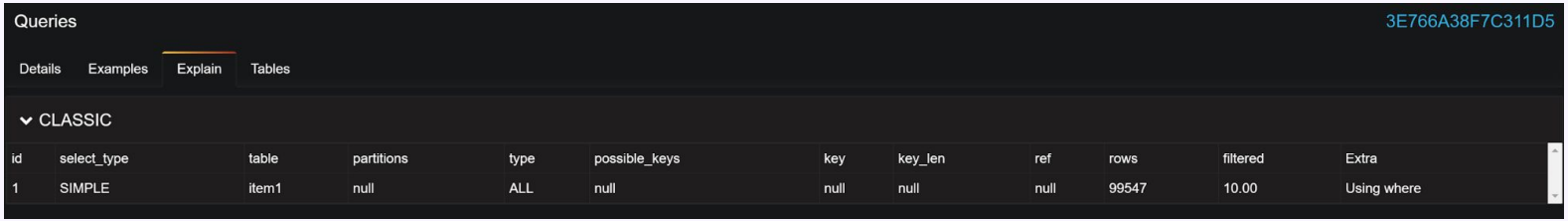
- Average query latency
- Rows returned per query
    - I/O Heavy Queries
    - Long Running Queries
- Critical Queries & Most Frequently Run Queries Runtime
- Queries waiting for disk I/O

Metrics for Queries



CPU Intensive Queries

queryid	Query	CPUScore	QPS	Rows_examined	Rows_affected	Latency_ms
3E766A38F7C311D5	select i_price, i_name, i_data from item1 where i_id = ?	720.88	57.04	715.17K ops	0 ops	69.01 ms
6E44651E3A55D32C	select o_id from orders1 o, (select o_c_id,o_w_id,o_d_id,count(distinct o_id) from orders1 where o_w_id=? and o_d_id=? and o_id > ? and o_id < ? group by o_c_id,o_d_id,o_w_id having count(distinct o_id) > ? limit ?) t where t.o_w_id=o.o_w_id and t.o_d_id=o.o_d_id and t.o_c_id=o.o_c_id limit ?	14.23	5.68	13.66K ops	0 ops	34.67 ms
35284AC6739BB849	update stock1 set s_quantity = ? where s_i_id = ? and s_w_id= ?	6.33	56.98	56.98 ops	56.98 ops	1.33 ms
59677A0A00A3BC71	insert into order_line1 (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_i_id, ol_supply_w_id, ol_quantity, ol_amount, ol_dist_info) values(?+)	6.27	56.98	0 ops	56.98 ops	1.38 ms
85FFF5AA78E5FF6A	begin	1.88	18.83	0 ops	0 ops	0.15 ms
813031B8BBC3B329	commit	1.88	18.77	0 ops	0 ops	3.57 ms
DE99F733D7101E62	select count(c_id) namecnt from customer1 where c_w_id = ? and c_d_id= ? and c_last=?	1.83	3.78	1.46K ops	0 ops	3.37 ms
2E9E5564F80A9AC3	select c_id from customer1 where c_w_id = ? and c_d_id= ? and c_last=? order by c_first	1.67	3.44	1.33K ops	0 ops	3.33 ms



Queries 3E766A38F7C311D5

Details Examples Explain Tables

CLASSIC

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	item1	null	ALL	null	null	null	null	99547	10.00	Using where

Airtribe

Metrics for Queries

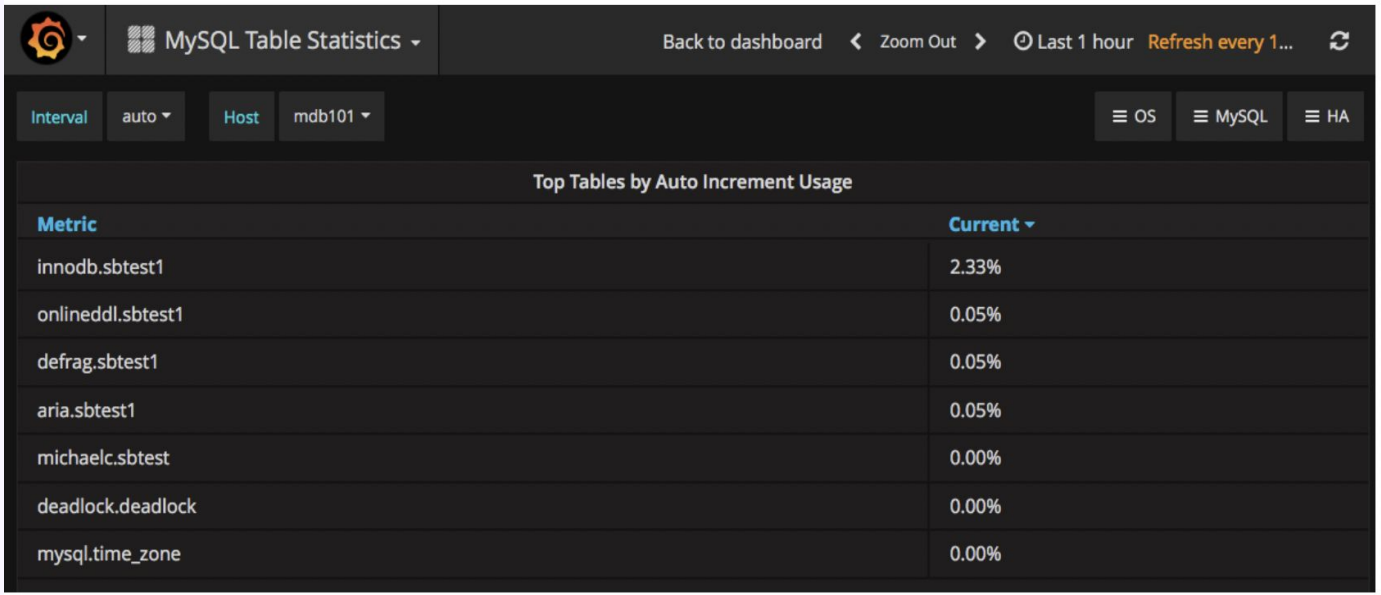
Airtribe

Github Case Study – Consistency

- ID column reaching max limit
- How could we have avoided this downtime ? What metric could have helped us here ?

Airtribe

Metrics for Database



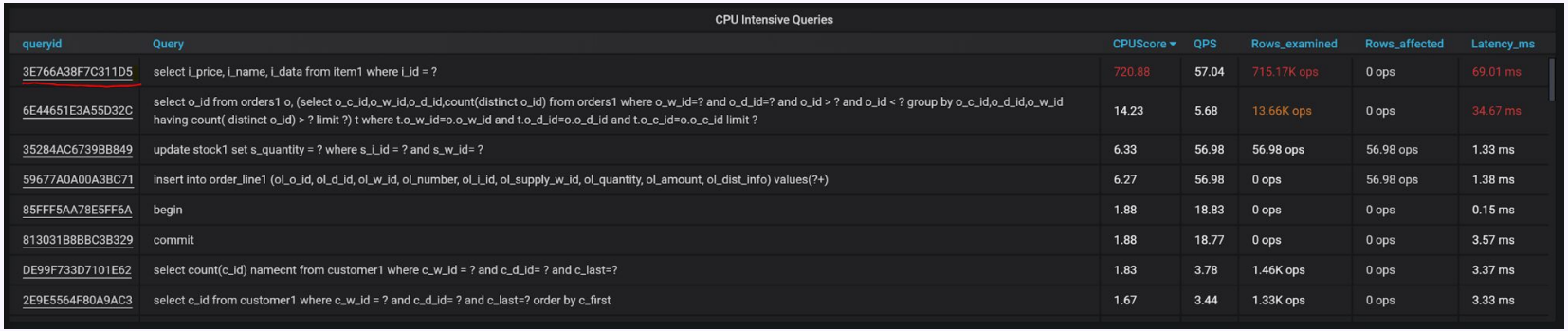
Airtribe

Github Case Study – Performance

- Inefficient Query leading to downtime
- SELECT count(id) FROM orders WHERE user_id = 123 AND start_time > 0;
- Full Table scan
- Executed multiple times in a day can put unnecessary load on the database and bring it down.

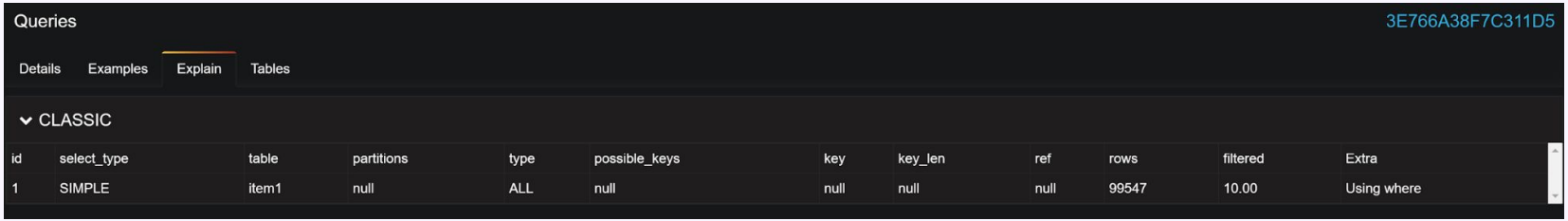
Airtribe

Metrics for Queries



CPU Intensive Queries

queryid	Query	CPUScore	QPS	Rows_examined	Rows_affected	Latency_ms
3E766A38F7C311D5	select i_price, i_name, i_data from item1 where i_id = ?	720.88	57.04	715.17K ops	0 ops	69.01 ms
6E44651E3A55D32C	select o_id from orders1 o, (select o_c_id,o_w_id,o_d_id,count(distinct o_id) from orders1 where o_w_id=? and o_d_id=? and o_id > ? and o_id < ? group by o_c_id,o_d_id,o_w_id having count(distinct o_id) > ? limit ?) t where t.o_w_id=o.o_w_id and t.o_d_id=o.o_d_id and t.o_c_id=o.o_c_id limit ?	14.23	5.68	13.66K ops	0 ops	34.67 ms
35284AC6739BB849	update stock1 set s_quantity = ? where s_i_id = ? and s_w_id= ?	6.33	56.98	56.98 ops	56.98 ops	1.33 ms
59677A0A00A3BC71	insert into order_line1 (ol_o_id, ol_d_id, ol_w_id, ol_number, ol_i_id, ol_supply_w_id, ol_quantity, ol_amount, ol_dist_info) values(?+)	6.27	56.98	0 ops	56.98 ops	1.38 ms
85FFF5AA78E5FF6A	begin	1.88	18.83	0 ops	0 ops	0.15 ms
813031B8BBC3B329	commit	1.88	18.77	0 ops	0 ops	3.57 ms
DE99F733D7101E62	select count(c_id) namecnt from customer1 where c_w_id = ? and c_d_id= ? and c_last=?	1.83	3.78	1.46K ops	0 ops	3.37 ms
2E9E5564F80A9AC3	select c_id from customer1 where c_w_id = ? and c_d_id= ? and c_last=? order by c_first	1.67	3.44	1.33K ops	0 ops	3.33 ms



Queries

3E766A38F7C311D5

Details | Examples | Explain | Tables

CLASSIC

id	select_type	table	partitions	type	possible_keys	key	key_len	ref	rows	filtered	Extra
1	SIMPLE	item1	null	ALL	null	null	null	null	99547	10.00	Using where

Airtribe

How do applications interact with database?

- ORM - Object Relational Mapper
- Object (Apps) → Relations (Tables)
- Done via models

ORM’s in action

Connection Setup, CRUD Operations,
Transactions in Sequelize

Airtribe

Setting up ORM in Codebase

- You have already installed the mysql server (brew install) and mysql client (DBeaver).
- Now you need to install mysql NPM Library and Sequalize ORM in package.json as dependencies.
- Next we create a config to write code to connect with MySQL.
- Once connection is tested, let's create the product model for our Instamart use case.
- Run the migration to create the table in the database using sequalize.sync(). You can learn how to run migrations on production at this link.

Models

Create model for the Product table from lecture 1.

id	name	price	stock
1	Onion	12	50
3	Tomato	25	40

```
const Product = sequalize.define('product', {
    id: {
        type: DataTypes.BIGINT,
        autoIncrement: true,
        primaryKey: true
    },
    name: {
        type: DataTypes.STRING(100)
    },
    price: {
        type: DataTypes.FLOAT,
        allowNull: false
    },
    stock: {
        type: DataTypes.INTEGER,
        allowNull: false
    }
})
```

Airtribe

Example – Models

Create model for the following table.

123 customer_id	ABC customer_name	ABC city	ABC country
1	John Smith	New York	USA
2	Jane Doe	San Francisco	USA
3	Bob Johnson	London	UK
4	Alice Brown	Paris	France
5	Tom Lee	Tokyo	Japan
6	Sancheeta	Bengaluru	India
8	Dhaval	Bengaluru	India

Airtribe

Example – Models

Create model for the following table.

123 id	order_date	123 customer_id	123 total_amount
1	2020-03-15	1,001	50
2	2020-08-22	1,002	75.5
3	2021-01-05	1,003	120.25
4	2021-07-10	1,004	95.8
5	2022-02-18	1,005	200
6	2022-06-30	1,006	150.75
7	2019-06-30	1,007	108.25

Airtribe

Creation & persistence to database

build, create, save, bulkCreate

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
  age: Sequelize.INTEGER,
});

// Build a new user instance
const newUser = User.build({ name: 'John', age: 25 });

// Save the new user to the database
await newUser.save();

console.log(newUser);

// Create a new user directly in the database
const createdUser = await User.create({ name: 'Jane', age: 30 });

console.log(createdUser);

// Bulk create multiple users in a single operation
const usersData = [
  { name: 'Alice', age: 28 },
  { name: 'Bob', age: 32 },
];

const createdUsers = await User.bulkCreate(usersData);
```

Airtribe

Update

```
User.update(
  { age: 30 }, // Updated values
  {
    where: { id: 1 }, // Condition for updating specific records
  }
)
  .then((result) => {
    console.log(result); // Number of rows affected
  })
  .catch((error) => {
    console.error(error);
  });
```

Delete & Reload

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
  age: Sequelize.INTEGER,
});

// Find a user by ID
const user = await User.findByPk(1);

// Reload the user from the database
await user.reload();

console.log(user);

// Update the user's attributes
user.name = 'John Doe';
user.age = 30;

// Save the updated user to the database
await user.save();

console.log(user);

// Destroy the user
await user.destroy();

console.log('User deleted');
```

Airtribe

Find in Sequelize

findAll, findOne, findByPk, findOrCreate

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
  age: Sequelize.INTEGER,
});

// findAll: Retrieve all users
const allUsers = await User.findAll();
console.log(allUsers);

// findOne: Retrieve a single user
const user = await User.findOne({ where: { id: 1 } });
console.log(user);

// findByPk: Retrieve a user by primary key
const userById = await User.findByPk(1);
console.log(userById);

// findOrCreate: Find or create a user
const [foundUser, created] = await User.findOrCreate({
  where: { name: 'John' },
  defaults: { age: 25 },
});
```

Airtribe

Code – Demo

Airtribe

Associations

- One to one
  - Student - ContactAddress
- One to many
  - Student - Class in School
- Many to many
  - Student - Course in College

Associations

- Student - ContactAddress
  - Student.HasOne(ContactAddress);
  - ContactAddress.BelongsTo(Student);
- Student - Class in School
  - Class.HasMany(Students);
  - Student.BelongsTo(Class);
- Student - Course in College
  - Course.HasMany(Student);
  - Student.BelongsToMany(Course);

Associations

HasOne, BelongsTo, HasMany, BelongsToMany

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
});

const UserProfile = sequelize.define('UserProfile', {
  bio: Sequelize.TEXT,
});

User.hasOne(UserProfile);
UserProfile.belongsTo(User);
```

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
});

const Post = sequelize.define('Post', {
  title: Sequelize.STRING,
  content: Sequelize.TEXT,
});

User.hasMany(Post);
Post.belongsTo(User);
```

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
});

const Project = sequelize.define('Project', {
  title: Sequelize.STRING,
});

const UserProject = sequelize.define('UserProject', {
  role: Sequelize.STRING,
});

User.belongsToMany(Project, { through: UserProject });
Project.belongsToMany(User, { through: UserProject });
```

Airtribe

Example – Associations

Define associations in the following scenarios.

Scenario 1: Book can have only one author.

Scenario 2: Book has one author and many co-authors.

Scenario 3: Same author can write multiple books.

Code – Demo

Airtribe

Filtering & ordering parameters

Where, order, limit, offset, include (easy loading concept)

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
  age: Sequelize.INTEGER,
});

const Post = sequelize.define('Post', {
  title: Sequelize.STRING,
  content: Sequelize.TEXT,
});

User.hasMany(Post);
Post.belongsTo(User);

// Retrieve users with age greater than 25, order by name in descending orde
// limit the results to 10, and include associated posts
const users = await User.findAll({
  where: { age: { [Op.gt]: 25 } },
  order: [['name', 'DESC']],
  limit: 10,
  include: [{ model: Post }],
});
```

Airtribe

Grouping Parameters

```
const User = sequelize.define('User', {
  name: Sequelize.STRING,
  age: Sequelize.INTEGER,
  city: Sequelize.STRING,
});

// Group users by city and retrieve the count of users in each city
const groupedUsers = await User.findAll({
  attributes: ['city', [sequelize.fn('COUNT', sequelize.col('id')), 'userCount']],
  group: ['city'],
});
```

Syntax – Single Transaction

```
const result = await sequalize.transaction(async (t) => {
    const user = await User.update(data,{
        where: {
            id: requested_id,
        },
        transaction: t
    });

    const users = await User.findAll({
        where:{
            id: requested_id
        },
        transaction: t
    })
}
```

Airtribe

Syntax – Multiple Transactions

```
const result = await sequalize.transaction(async (t) => {
    const user = await User.update(data,{
        where: {
            id: requested_id,
        },
        transaction: t
    });

    const users = await User.findAll({
        where:{
            id: requested_id
        }
    })
}
```

Airtribe

Code – Demo

Airtribe

Problem Solving

Answer the following scenarios.

Scenario 1: What keyword would you use to get all seats in a bookmyshow app ?

FindAll

Scenario 2: What keywords help you define associations between related models like Theatre and Screen ?

HasMany, BelongsTo

Scenario 3: What are the different ways of creating a row in database ?

Create, Build + Save

Scenario 4: Which keyword can help you prefetch the data from a related table ?

Include

Scenario 5: Define the syntax for transactions in Sequalize ?

Model.update(data,{where:{column: value}, transaction:t})

Thank You!

See you again on next date here

Airtribe