Galvia. DIGITAL

CASE STUDY

Optimizing Arches ORM: 99% Faster PostgreSQL Queries with Galvia Digital

AT A GLANCE

Flax & Teal developers on the Coral project were blocked by sluggish Arches ORM queries, with response times over 30 seconds. Galvia Digital engineered a performance-first solution, codenamed "Emerald", that optimized the PostgreSQL query layer, slashing times to 300 milliseconds and boosting speed by 99% without interrupting ongoing development.

Project Duration: 3 Months

" Galvia Digital's work on Arches ORM has been a game-changer for our teams and customers, and will benefit any future project we run on the platform. As always, they are creative, adaptable, collaborative, and technically solid. "

Phil Weir, Director, Flax & Teal

→ CHALLENGES

- **Arches ORM was returning bloated data in 25–30 seconds**
- **Limited filters forced developers to manually sort full datasets**
- **Inefficient models caused query and performance bottlenecks**
- **Sparse documentation made onboarding and upgrades tough**
- **Risk of disrupting global teams relying on active workflows**

→ SOLUTIONS

Galvia Digital developed "Emerald," a performance-first overhaul of the Arches ORM query process. By replacing heavy default models with lean, upgradeable ones and introducing a dynamic query builder, the team enabled efficient, filter-rich PostgreSQL queries. Native filtering was added at the ORM layer, and all changes integrated seamlessly with existing systems, delivering dramatic speed gains without disrupting active workflows.

Galvia. DIGITAL

520 HOURS
Saved thanks to Galvia Digital's Developers

99% SPEED INCREASE

galviadigital.com

Galvia.
DIGITAL

CASE STUDY

→ RESULTS & BENEFITS

01 Massive Performance Gains

Query times dropped from 30 seconds to as little as 300ms, delivering a 99% speed improvement with no disruption to ongoing development.

02 Real-Time Data Access

Optimized query models and filtering enabled real-time responses, empowering developers to work faster and more efficiently.

03 Elevated Developer Experience

Reduced friction in querying and data retrieval boosted team productivity across the Coral project and simplified future extensions.

04 Seamless System Integration

All improvements were fully compatible with existing tools, ensuring legacy workflows continued uninterrupted while unlocking long-term maintainability.

TECH INVOLVED

POSTGRESQL

PYTHON

ARCHES

SQLITE

ORM

UNIT TESTING

How can we help?

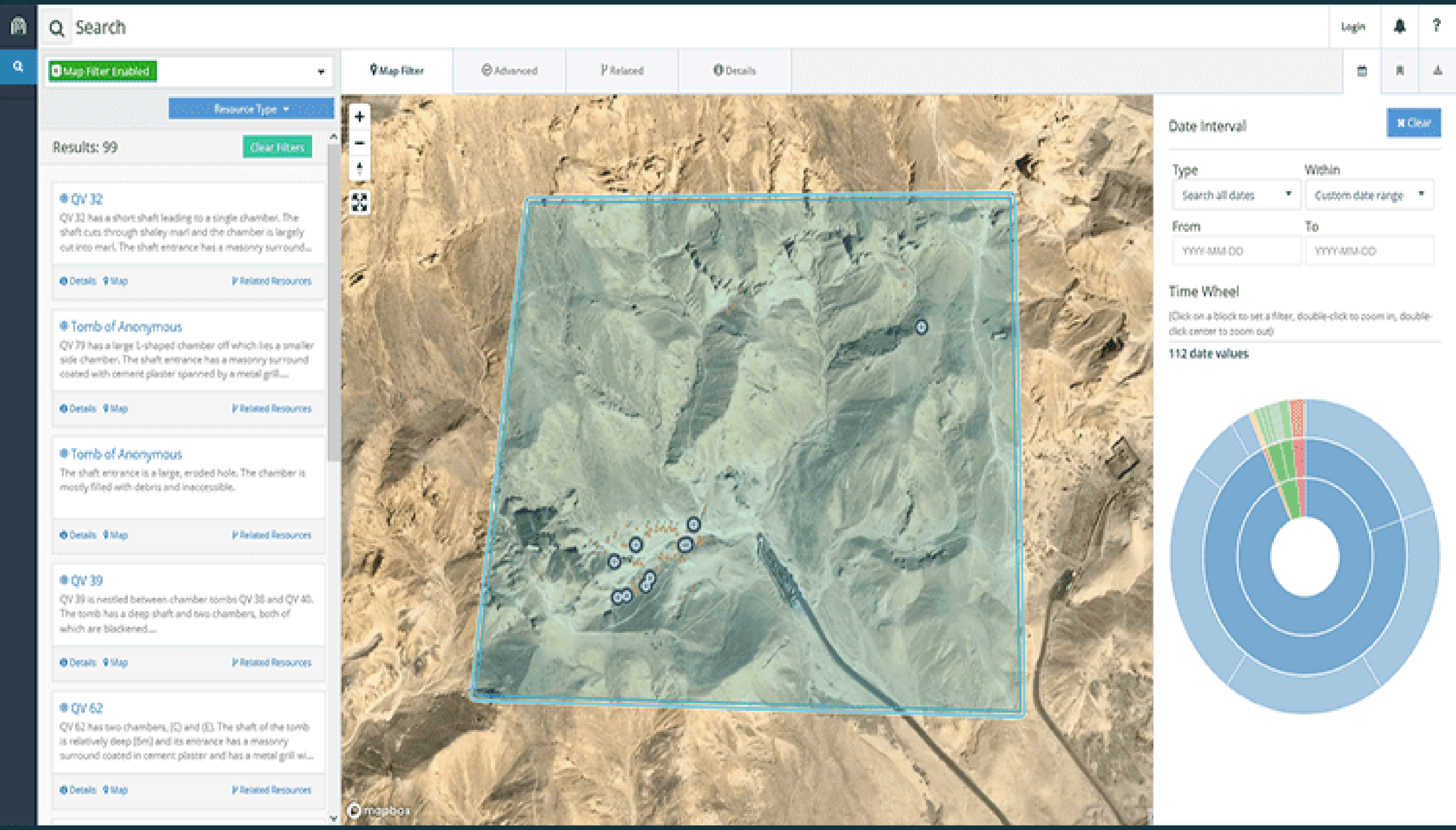
Bespoke Software Solutions For Your Business. Contact Us For a Free Diagnostic Call!

+44 77 330 100 28

info@galviadigital.com

galviadigital.com

WHAT IS ARCHES & CORAL?



Results: 99

QV 32
QV 32 has a short shaft leading to a single chamber. The shaft cuts through shaley marl and the chamber is largely cut into marl. The shaft entrance has a masonry surround...
Details Map Related Resources

Tomb of Anonymous
QV 79 has a large L-shaped chamber off which lies a smaller side chamber. The shaft entrance has a masonry surround coated with cement plaster spanned by a metal grill...
Details Map Related Resources

Tomb of Anonymous
The shaft entrance is a large, eroded hole. The chamber is mostly filled with debris and inaccessible.
Details Map Related Resources

QV 39
QV 39 is nestled between chamber tombs QV 38 and QV 40. The tomb has a deep shaft and two chambers, both of which are blackened...
Details Map Related Resources

QV 62
QV 62 has two chambers, (C) and (E). The shaft of the tomb is relatively deep (5m) and its entrance has a masonry surround coated in cement plaster and has a metal grill wi...
Details Map Related Resources

Date Interval

Type: Search all dates
Within: Custom date range
From: YYYY-MM-DD
To: YYYY-MM-DD

Time Wheel
(Click on a block to set a filter, double-click to zoom in, double-click center to zoom out)
112 date values

Arches is an open-source platform built to manage cultural heritage data at scale. It's designed around structured, ontology-rich records. Things like historic buildings, protected sites, and conservation information.

Under the hood, Arches uses PostgreSQL as its database and ORM patterns to connect its models to SQL. That setup allows developers to work with high-level data structures while the system handles all the translation to database queries in the background.

Coral is a custom implementation of Arches developed for a government body. It builds on the Arches core but adds a layer of accessibility for non-technical users.

Through workflows and custom interfaces, Coral lets archaeologists, planning officers, and other civil servants interact with structured records in a way that feels natural, without needing to understand the underlying schema or database design.

While Arches provides the foundation, Coral shapes it into a day-to-day working tool.

PROJECT OVERVIEW

The Coral project plays a vital role in helping organisations protect and manage cultural heritage data. From historic buildings and archaeological sites to contributor notes and geospatial references, Coral deals with complex, structured information. It runs on Arches, a platform tailored for ontology-rich datasets. Arches does a lot well, but under the hood, the system was dragging.

Developers were running into serious bottlenecks when they queried data.

Basic filters were taking upwards of 25 to 30 seconds to return results. Even small pages took ages to load. For a team trying to move quickly, iterate, and ship updates, that's a killer.

These weren't just minor hiccups. They were bringing day-to-day work to a standstill.

Flax & Teal brought Galvia Digital in to tackle this head-on. Not to replace Arches or overhaul Coral, but to go straight at the root of the issue: query speed.

We focused entirely on performance. After three months of hands-on work, we delivered Emerald. a lightweight, developer-friendly query layer that made Arches fast again without breaking a thing.

Flax & Teal is a leading software consultancy dedicated to the digital preservation of cultural heritage. As registered suppliers and contributors to Arches, an open-source platform designed for managing immovable heritage assets like historic buildings and archaeological sites, Flax & Teal works with institutions across government, academia, and conservation to securely manage and enrich complex datasets.

Duration: The project was delivered in under 3 months

WHAT WAS HAPPENING?

The root issue was Arches' ORM, the layer that pulled data from the database. It loaded full records by default, even when only a small piece was needed. Models were packed with extra logic, and filtering only supported basic "equals," forcing developers to load full datasets and filter in memory.

The ORM lived in a single, undocumented file containing loops and repeated code. It was hard to read, debug, or update, making onboarding slow and changes risky.

UNDERSTANDING THE PROBLEM

We started with a deep investigation into how Arches ORM was structured.
We studied the query functions, examined what was actually happening in filtering methods like where, find, all, and first, and bench-marked the time and complexity of those operations.

What we found: The default models were bloated with unused methods, loops were slowing everything down, and every query loaded the full record, even when only a subset of fields was needed. There were also no built-in ways to do greater than/less than filtering. Developers had to load every record and filter manually in Python.

We traced load times back to specific parts of the ORM, especially around how the TileProxyModel and from_resource methods were used. The structure wasn't designed for performance or flexibility, so we broke it down, tested the time impact of every piece, and worked from the bottom up.

THE CHALLENGES

- Querying even small data sets took 25–30 seconds
- Models were overloaded with logic and methods not always needed
- Filtering was basic: only "equals" was supported by default
- Developers had to load entire datasets and filter in memory
- Code was hard to read, undocumented, and tangled into a single large file
- Debugging and onboarding new devs was difficult
- The existing system couldn't be changed too much without breaking live projects

OUR SOLUTIONS

1. Streamlining the Models

We started by overhauling the models that power the ORM. Instead of always pulling in the full object structure, we introduced lean models designed to return only the necessary data. If a developer asked for something more complex that couldn't be handled by the new model, the model could automatically switch to the full version behind the scenes. This change alone cut down the size and weight of most queries dramatically, improving response time right out of the gate.

2. Smarter Filtering and Query Logic

Next, we addressed one of the biggest developer pain points: filtering. The original system only supported exact matches. We added built-in support for dynamic filters and range queries so developers could finally run commands like person.where(age > 5 & age < 10) without fetching every record and sorting manually in code. We also removed duplicate loops and unnecessary operations in the existing ORM logic, cutting out dead weight and improving clarity.

We bypassed the TileProxyModel, which was one of the main sources of overhead. Direct model access replaced proxy calls wherever possible, slashing the time it took to run each query. These backend changes made the system not only faster but also easier to extend.

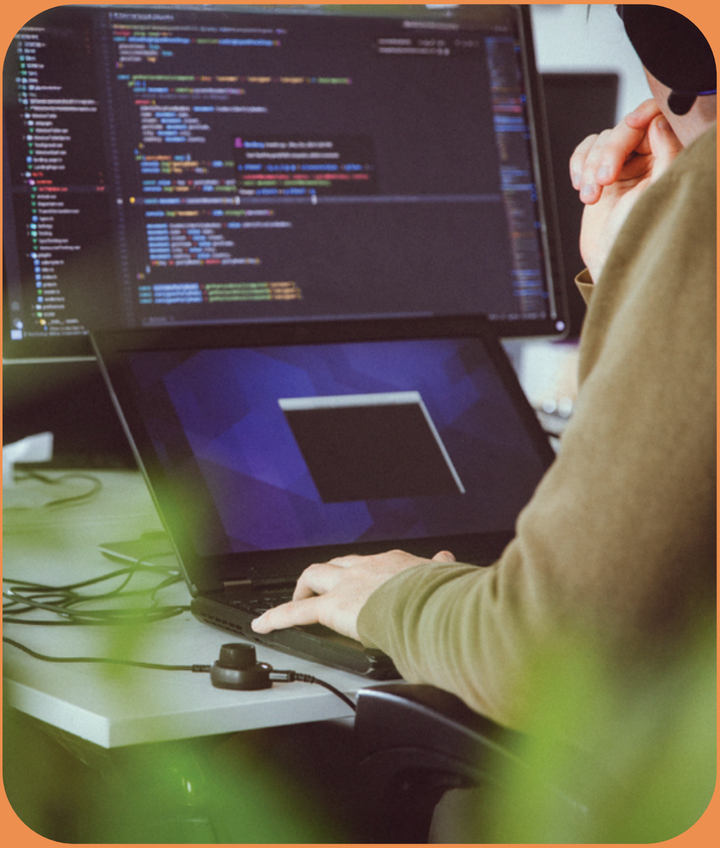
3. Compatibility and Integration

We wrapped Emerald in a compatibility layer that made it work seamlessly with Coral's existing setup. No changes were needed to admin panels, legacy systems, or front-end tools. Developers kept working the same way, just faster. The full solution was built in Python with PostgreSQL, combining Django ORM principles with custom SQL builders to modernize performance without breaking what already worked.

BUSINESS OUTCOMES

After three months, Coral's developers weren't just getting faster queries, they were working smoother across the board. Queries that once took 30 seconds dropped to milliseconds, pages loaded instantly, and filters delivered results without extra work.

Teams didn't need to change anything. Emerald gave them better tools, better speed, and a platform that's now easier to build on and ready for whatever comes next.



arches®
An open source data management platform for the heritage field

"Galvia Digital work on Arches ORM has been a game-changer for our teams and customers, and will benefit any future project we run on the platform. As always, they are creative, adaptable, collaborative, and technically solid."

Phil Weir,
Director,
Flax & Teal

How can we help?

Galvia Digital helps organisations tackle complex performance and scalability challenges in data-heavy systems. We specialise in optimising query layers, modernising legacy infrastructure, and building tools that make life easier for developers and decision-makers alike.

If your platform is slowing your team down, we can help speed things up—without tearing everything down to start over.

Galvia. DIGITAL

+44 77 330 100 28

info@galviadigital.com