This poster presents our implementation strategy and pipeline architecture for visualising very large-scale graphs in an immersive environment, using high-performance graphics approaches. The innovation lies in utilizing GPUs for real-time cluster-based interactive rendering in conjunction with an intermediate graph representation that utilizes Khronos Group's GLTF file format, as well as a novel interaction design.

The following outlines the processing pipeline (see figure 1), highlighting several novel contributions, such as extracting graphs from relational databases, processing them offline or in the cloud, and finally representing them in GLTF format for quick loading and transfer to GPU memory.

**Graph Extraction**
Any non-trivial database instance of a given schema will generally contain multiple graphs. In particular, given an entity relationship model, the available graphs are identified as follows: Every entity corresponds to a set of nodes, and every relationship between two entities corresponds to a set of edges. Consequently, we treat every database instance as a collection of heterogeneous graphs. The graph extraction stage concludes with the original data converted into an intermediate graph representation suitable for subsequent algorithmic processing.

**Graph Preparation**
To detect the community structure of the extracted graphs, we adopt the state-of-the-art Louvain method, which has been proven to be computationally suitable even when large-scale graphs are involved. Subsequently, the following stage is concerned with spatially laying out the graph. The previous algorithms work purely on the abstract graph structure, without assigning concrete positions to nodes or edges. For the purpose of generating layouts, we utilise the ForceAtlas2 and OpenOrd algorithms.

**Graph Visualisation**
The final stage is concerned with the graphical representation and interactive manipulation of the graph. At this stage, the entire graph is loaded into GPU memory. To display the massive networks, we adopt GPU-based point-cloud rendering techniques, producing large numbers of node and edge geometries on-the-fly with the aid of geometry shaders. The use of shaders also allows users to interactively adjust the visual appearance of the graph to aid the analysis (animated transitions or view distortions in real-time).

Figure 1: Overview of graph processing pipeline.

Figure 2: Successive graph summarisation steps.

Figure 3: Collaborative exploration of massive network in immersive environment.