I am Qiwen Chen, a student at the University of Liverpool majoring in Computer Science and a passionate future database systems engineer. When I was onboarding as an intern at PingCAP, a leading database system unicorn in China, my colleague sent me three academic papers on database systems: the paper on Google Spanner, the paper on the company's TiDB product, and the paper on self-driving databases from CMU Database Group. These papers intrigued me because they revealed the secrets behind some of the world-class databases, and each represented a leading trend in database systems. **Scaling with high availability, versatility across different workloads, and automation of database management**. Although my college life was greatly affected by the pandemic, I was able to continue my internship in some of the best infrastructure teams in China, mostly because the development of database systems has been persistently pushed and updated by the industry. Considering the valuable and challenging problems raised by the industry, I believe that I can focus on essential technical skills and computer science knowledge in my study, which will help me to become a pragmatic and visionary infrastructure engineer.

My previous education provided me with numerous opportunities to try my hand at different areas of computer science and engineering. Since my freshman year, I was always eager to learn the most current topics in computer science. Even before freshman orientation month was over, I took online lectures on machine learning, such as CS231n of Stanford University, and later completed Andrew Ng's Deep Learning specialization. I also joined an AI study group with international students, which later became the Camculator team, dedicated to developing a handwritten math solver for college students via camera. In the summer of 2019, I got the opportunity to work in the X- CHI lab of Xi'an Jiaotong-Liverpool University as a student researcher. The first year of study was quite exhausting, but I boosted my confidence in studying computer science: I was awarded the College Academic Achievement Award as one of the top 10% students in 2019; the Camculator team won the championship of the University Entrepreneurship Competition in 2019 and was later awarded as the Dandelion Project of Suzhou city in China; in the X- CHI Lab, I helped implement the experimental data pipelines and algorithms that examined EEG data for cybersickness signals, and our work was presented at ISMAR 2020. However, it was the challenges I encountered in my start-up and research endeavors that brought me to the field of database systems. It was through these challenges that I discovered the vulnerability of modern AI in manufacturing, the area I had previously pursued. As the team pushed forward with the Camculator app, we discovered that the backend server suffered slightly from high latency and scalability issues. In-game EEG research was severely limited by the scale and quality of the experimental data, which advanced AI algorithms struggled to compensate for. However, these challenges also pointed to the critical importance of data infrastructure systems in delivering data-driven applications. Therefore, in my sophomore year, I started looking for computer systems courses such as MIT6.024 and MIT6.814, which triggered my obsession with database systems. I found that database systems were particularly attractive to techies like me because they were a perfect combination of computer science and computer engineering, encompassing computer architecture, data science, and algorithm design. Moreover, these curricula showed me how exciting recent innovations were in changing database systems and how crucial they were in building data-intensive systems that enable machine learning and Big Data applications at scale.

Although the pandemic thwarted my plans to move overseas, the industry opened up a surprising path that ignited my passion. I worked on 3 different internship projects from the perspective of 3 different roles: an **infrastructure engineer**, an **upstream developer**, and an **AI-for-database practitioner**. I gained my perspective of an infrastructure engineer through the internship at PingCAP. After that, I worked as an upstream developer at ByteDance (the company behind TikTok) on a data infrastructure migration project. Finally, I worked with Alibaba Cloud (China's cloud market leader) to help develop an AI-based cloud database tuning system as a research and development intern - a wonderful opportunity to get a glimpse into AI-for-database applications. I realized that innovations in database

systems were constantly being driven and implemented by world-class technology companies like Google, Microsoft, and Amazon. These experiences also reinforced my belief that technical practice is more important than research in this field, which is perfect for me who is passionate about engineering. My experience as an intern can effectively support my studies in database systems and pave the way for me to join some of the world-class infrastructure development teams to achieve my goal of building world-class data infrastructure.

Being an **infrastructure engineer** means putting a high level academic education into practice. Reading academic papers on computer systems can never match the experience of building a system used by millions of people. At PingCAP, I helped improve the chaos-theoretic stability of TiDB, the company's well-known distributed SQL database. I worked closely with Qiang Zhou, the leader of the Engineering Efficiency Team and founder of Chaos Mesh, a renowned open-source project that exploits production system vulnerabilities by creating kernel-level turbulence. Through my work at PingCAP, I had the opportunity to analyze critical bugs in test cases and understand the architecture of TiDB. In doing so, I came to appreciate the high level of software engineering standards on which TiDB is built, from the carefully designed Golang code that calibrates memory offsets and detects read-write conflicts, to the overall architecture that includes the placement driver, row and column memory (TiKV and TiFlash), and coprocessors. A single failed case of optimistic lock testing can lead me to thousands of lines of code of various components and several thousand lines of logs originating from the internal test infrastructure. Striving for high technical standards is even more important when developing solutions that involve large services. At ByteDance, I worked on a two-step solution to adapt Abase, the company's internal implementation of flexible cached storage, to the privatized deployment of ByteDance's to-B products: I created a middleware query parser for upstream teams to ensure compatibility of over 500 microservices in the Larksuite service architecture, and then moved on to support data structures like Redis zip-list and bit arrays from the executor level of Abase. Even for my most research-oriented internal project, located in Alibaba Cloud, I had to carefully implement the system with fully automated RDS APIs and distributed frameworks that enable concurrent tuning sessions from multiple instances, which significantly reduced the time required for database tuning.

Learning about databases from the perspective of **upstream developers** makes me realize that there is no silver bullet in the world of databases. As a result, best-of-breed applications like TikTok often use different types of infrastructures for different functionalities, requiring a tremendous amount of maintenance and migration effort. At ByteDance, I worked in the Lark Service Architecture department under the guidance of Mr. Guoqiao Xiao, one of ByteDance's first backend architects. To understand the background of the team's data migration project, it took me more than 10 working days to investigate the details of almost all popular data storage products in the market: PostgreSQL, Amazon DynamoDB, Redis, and Microsoft Cosmos DB. In the process, I realized that each data infrastructure has its own focus, either high availability, strong consistency, rich data structures, or transactional and analytical performance. For example, at ByteDance, I worked with Abase, which was prevalent at ByteDance because it supported upstream enterprises for fast cache access while maintaining a persisted version on disk. PingCAP's TiDB, on the other hand, is less powerful due to its highly decoupled distributed architecture, but offers high availability and consistency, which is perfect for financial companies. Throughout the project, I was left with one question: if database systems can hardly be universal, could they be better adapted to different workloads? This led me to the very last internship project, which was dedicated to implementing an AI-for-database solution.

With the question of "universal database systems" in mind, I came across the field of **AI for databases** during my internship at Alibaba Cloud. During this project, I was introduced to existing academic and industry proposals on AI for databases, a topic that is still new to the industry but dedicated to solving one of the most valuable problems in the field. I was tasked with building RDS-

Tune, a large-scale database tuning system under the supervision of Xiang Peng, Director of Alibaba Cloud RDS and former Principal Engineer at Amazon Aurora. The implementation was based on ResTune, a paper by Alibaba Group's DAMO Research that proposed a meta-learning approach to cloud database tuning. Inspired by this proposal and existing work such as NoisePage and the CMU Database Group's OtterTune project, I quickly identified useful algorithms and technical techniques for the goal as well as subtasks such as database workload characterization and parameter set optimization. These approaches can be combined with the ResTune proposal, creating experiments that potentially produce better results. For example, I updated the machine learning pipelines to support workload characterization using internal DB metrics instead of normalized SQL statements, which was less portable to other relational databases but yielded more robust results. When the first version of RDSTune shipped, the throughput of a PostgreSQL instance running on a highly skewed workload increased by 10% after hundreds of iterations. As the result of an intern project, this was a slight improvement, but I realized that the idea of AI for databases still has a long way to go in the cloud computing industry and that real research needs to be built with extraordinary effort based on the vast majority of previous work.

My journey on the topic of AI for databases never ends. I am currently researching learned indexes with Professor Konstantinos Tsakalidis from the University of Liverpool as part of my final year project. My research focuses on evaluating and optimizing the performance of learned indexes when querying geospatial data. I believe this research experience is a perfect opportunity to dive into index design, an important skill for database kernel developers. Just as databases are changing the way data-intensive applications are built, trends in cloud computing and AI are also changing database systems. During my internships, I have witnessed startups and giant IT companies in China working diligently to move databases to a cloud-native architecture and automate database development. While essential cloud-native features are still missing in some of the most widely used cloud database services in China, technologies such as RDMA and disaggregation of storage and computing power are gradually being implemented in existing systems to better utilize hardware resources and reduce IO costs. Although the majority of the cloud computing industry in China is still skeptical of AI-for-systems practices, I can imagine the potential of projects like RDS-Tune in mass applications to have similar impact as the automatic tuning feature in Azure SQL databases. However, the database industry in China is still at an early stage. Database R&D is limited to short-term outcomes such as stability improvements and privatized deployments.

I am striving to find an ideal environment where I can pursue my passion of becoming an infrastructure engineer who can contribute to the positive change in the society. In China's IT industry, overtime is a common phenomenon as business growth is prioritized over manpower and short-term results are prioritized over software development quality and architectural changes. In China's universities, computer systems research is often unfairly evaluated by the number of accepted papers at top conferences, regardless of its value to production systems. In addition, the cloud computing industry in China lacks the scale and maturity to cultivate innovations from scratch, and most database curricula in universities in China and the UK are too outdated for future database talents. To participate in building the next-generation data infrastructure, I would like to pursue a master's degree to sharpen my skills and seek a position as an infrastructure engineer at one of the leading tech giants or a unicorn startup.

I am passionate to get into database engineering as I am eager to solve the most challenging technical problems in delivering data-driven applications to millions of users. It has been my true passion and working on the fundamentals behind the scenes, but at the same time creating real, tangible value brings me great pleasure and sense of achievement. I am constantly looking for opportunities to learn about database systems, live out my passion, and build next-generation

databases for the world. I hope I will be given the opportunity to pursue my passion and further academic education that will help me achieve my future goals. At the same time I am confident that I can fully respond to all the challenges and meet the expectations set in front of prospective students.