Instructor: Mohammad Sadoghi
TAs: Dakai Kang
Shaokang Xie

Due Date: January 17, 2026
Submission Method: Canvas
Score: 5%

Pre-Milestone Handout

ECS 165A - Winter 2026

For the pre-milestone assignment, you are going to research and explore modern database technologies to learn more about their overall architecture, key components, and their strengths and limitations. This is the only individual assignment in which you need to write a short summary, at least five pages long, summarizing your findings.

This handout offers a few suggested technologies to get you started. You are not limited to any of the technologies listed here. You have complete freedom.

Think Long-term, Plan Carefully.
Be curious, Be creative!

One way to start your exploration is to learn about databases in general through Google. Essentially, you need to research what it takes to build a database. What are the key components that are necessary? Along the way, document all your findings and keep track of the sources of all materials. The materials below are merely suggestions to get you to start thinking about your journey.

There are two prominent modes of a database that one may employ. You can use an *embedded database*, essentially shipping your database as part of the application package, or an *externally hosted database*, in which a database is hosted on an external server, and your application connects to it. Broadly speaking, there are several flavors of databases, namely, **SQL** (a relational database with transactional capability), **NoSQL** database (key-value store with limited transactional capability), **NewSQL** databases (e.g., Google Spanner), **HTAP** databases (similar to **L-Store**, which is the focus of the course project), or **Non-Relational** databases (supporting data models such as a graph, JSON, XML, document, may also be categorized under NoSQL brand).

Ranking: You may find a ranking of the most widely used database systems here. An interesting read on the most popular database engines for mobile apps.

Benchmarks: TPC is the gold industry standard for benchmarking and evaluating major database technologies. For example, the two most well-known benchmarks are TPC-C and TPC-H.

Here are just a few starting examples.

Apache ResilientDB (Incubating) is an open-source permissioned blockchain fabric at UC Davis, which is designed with the aim of fostering academic and industrial research. It offers a high-throughput yielding distributed ledger built upon scale-centric design principles to democratize and decentralize computation. The key aim behind the development of ResilientDB is to illustrate that the design and architecture of the underlying system are as important as optimizing BFT consensus protocols. ResilientDB raises a simple yet intriguing question: *can a well-crafted system-centric architecture based on a classical BFT protocol outperform a protocol-centric architecture?* **Therefore, we strongly encourage everyone to explore and fork the ResilienDB codebase,** and more importantly, we invite everyone to contribute to the ResilienDB open-source project.

SQLite is a software library (an embedded database) that provides a relational database management system. The lite in SQLite means lightweight in terms of setup, database administration, and required resources. Note: SQL is the standard language to communicate with relational databases (which you will learn in class). [Tutorial, Architecture]

MySQL is the world's most popular open-source database. Whether you are a fast-growing web property, technology ISV, or large enterprise, MySQL can cost-effectively help you deliver high-performance, scalable database applications. [Documentation, Architecture]

Firebase is a platform (an externally hosted cloud service) that allows the building of web and mobile apps. You can store users' data on its real-time database, which syncs data among users' data in no time. It is owned by Google and is easy to integrate into your project. [Firebase-console, Intro Video]

PostgreSQL is a powerful, open-source object-relational database system with over 30 years of active development that has earned it a strong reputation for reliability, robustness, and performance [Documentation, Architecture].

Minibase is a database management system intended for educational use. It has a parser, optimizer, buffer pool manager, storage mechanisms (heap files, secondary indexes based on B+ Trees), and a disk space management system. The goal is not just

to have a functional DBMS but to have a DBMS where students can study and implement the individual components.[Documentation, Architecture]

Forming Team and Finalizing Roles
As part of this milestone, you need to form your group, including the names of all members, and fill out the form here. If you are looking for a team, please use this form.

Additionally, in your report, you need to specify the role of each member. For each group, it is important that each member lead one aspect of the project while contributing and learning about other parts. There are several leadership roles as we go through various stages of the project, such as **(1)** team coordinator, **(2)** system architect, **(3)** developers, and **(4)** tester. The design and development itself may further be categorized as **(1)** query evaluation (APIs and access methods); **(2)** bufferpool management; **(3)** crash, recovery, and logging; and **(4)** synchronization and concurrency. Please consider and specify the role of each member, which may change as we make progress throughout the course.

For each milestone, each group must submit an attribution section in which they specifically state each person's role and the percentage of their contribution. In a group of 5, it is expected that a person will complete 15-20% of the overall project. Of course, determining the percentage is not approximate, but the point is that every member contributes fairly. **No contribution means a grade of 0.**

Course Policy
In this class, we adopt the UC Davis Code of Academic Conduct, available here.

Disclaimer
The external links and resources that are being provided on this handout serve merely as a convenience and for informational purposes only; they do not constitute an endorsement or approval of their products, services, or opinions of the corporation or organization, or individual. As a student, developer, or researcher, it is your sole responsibility to learn how to assess the accuracy and validity of any external site. This is a crucial skill in the age of the Internet, **where anyone can publish anything!**

Changelog
Milestone Handout Version v1: January 1, 2026 (initial posted version)