Presenter Note:

Slide 1:

Hello! Everyone. This is Weiting Li, welcome to this software engineering technical showcase. [space] In this video I'm going to talk about database technology in software engineering. More specificly, database technology in software engineering environment. [space] Of course, I am going to guide you through the very definition of these terminologies and then go forward to our topic. As we all know, the development of any software is a complex process involving enormous individual work, collaborations, and lifecycle management. [space] Due to the immaterial nature of software, its development can be regarded as the goal-oriented production of a set of complex structured, interrelated data that needs to be stored persistently and managed efficiently.[space] Database technology has provided this perfect solution to the problem.

Slide 2:

Some of you may be familiar with database or have worked with databases. But for the newbies, you may ask: what is database? Database is a collection of data! [space] Online or offline, centralized or distributed. It only consists data that meets certain criterion. So not any data goes into the database. [space]

Slide 3:

Database has certain properties listed below: Firstly, it is designed to reflect part of the real world called the mini-world. Secondly, it is a logically coherent and integrated collection of large amount of data. This data may have an explicitly controlled lifecycle etc.. [space]
Thirdly, access to the data should be user-friendly, flexible and efficient. [space] In addition, it is designed and populated with ideally non-redundant data so that users do not need to worry about too much data processing. [space] And finally, the data is able to be used and modified independently of the software that created it. This is called data independence.

Slide 4:

Next, since we know about database generally, let's take our attention to database technology. Database technology is a combination of concepts, systems, methods and tools to support the construction and operation of databases. [space]

Slide 5:

After we learn the definition, here are the characteristics of database technology: To begin with, a data model is needed to cornerstone the mini-world concept fully. Different model have different levels of expressiveness. The most popular models are the relational, the object-oriented, and object-relational.[space] Persistence : in short, the database is stored persistently. [space]Data integrity: Semantic integrity control mechanisms such as database triggers, are provided to ensure the consistency of the data during manipulation. [space] Querying: Stored data maybe retrieved according to need. [space] Concurrency control: a consistent database state must be presented to every user when accessed concurrently.[space] Security: Security mechanism are required to prevent malicious access and manipulation. [space] Secondary storage: database technology should provide various methodology to efficiently manage storage when the data is too big.[space] Distribution: database could be distributed on different physical locations. [space]

Slide 6:

Next, let's talk about DBMS which stands for database management systems. DBMS are designed to manage large quantities of data in a consistent manner. It is nothing more than computerized data-keeping system. The main purpose is to access and manipulate data user-friendly and efficiently. The four prominent DBMS are inverted list, hierarchic, network or relational DBMS.

Slide 7:

Is there any DBMS that you may heard of or used before?
[10s]

Slide 8:

One famous DBMS is MYSQL. MySQL is a relational database specified DBMS. It is open-source, fast, reliable and easy to use. The server of mySQL works in a client/server or embedded system.
[space] To give you a taste, here is a screen-shot of mysql interface in the ios system.

Slide 9:

In this slide, I want to just go over some basic operations in mySQL: [space]
Firstly, you type this line to select the database to use.
To insert a new input, you could do this and easily input a customized row into the database.
Finally, here are some codes to help us select the right information from the database. Because most of the time, we do not need all the columns in a dataset. We only need part of them.

Slide 10:

Later on, I want to finally talk about database technology in software engineering but more importantly in software engineering environment. [space]

Slide 11:

What is SEE, software engineering environment? It is a software system that provides support for the development, repair, and enhancement of software, and for the management and control of these activities. A typical see has a central database and a set of software tools. The database would act as a repository for all related data to the project through the entire cycle of the project. The software tools offer support for various purposes: technical or managerial. [space]
We want to discus about the integration of database functionality in software engineering environments (SEE) and its use by software development tools. SEE have been developed for the support and control of various aspects of the software development lifecycle and for the integration of development tools.

Slide 12:

SEE can advantageously use database functionality to create valuable software development services. Database can be used a s the data integration medium between the vairous applications of the SEE. In this slide, I want to talk about the functional requirements for efficient use of database technology in SEE. [space] Firstly, database technology requires persistent storage of software artifacts. Artifact means an item that is produced during the development process. The database model need to be powerful enough to allow the description of any potentially required artifact type, and the defined persistent artifact descriptions should be able to evolve through the life-cycle of the software repository. [space] Artifact integrity and consistency. It is very important that the maintenance of integrity and consistency of the artifacts are maintained properly. [space] Artifact querying and retrieval. Retrieval of artifacts from the library is a significant feature for software reuse. In any cases, it is important to achieve a high retrieval precision, high recall ratio, and a short response time. [space]

Next, version management. During the development process, it is possible that an artifact evolves multiple times. Therefore, it is also very important to keep track of the versions and the relationship between them. [space]Cooperative design: In a software development process, it is necessary to manage work in progress and consider the nature of the performed work which usually consists of cooperative work. This issue can be solved by having a interface that allow different users to work in a private, persistent workspaces.[space] Representation of the software process. Basically, all aspects of the software process need to be modelled including the types of artifacts, the step types and structure, the resources etc.. [space] Tool integration and portability. This basically means that it is important to have the integration of tools used for software development. We can achieve this by having a library structure to store the tools need. So that different artifacts have same access to the toolset. The portability means the portability of the tools between different SEE. This is realized by building a well-defined interface. [space] Finally, the collection and evaluation of software development: It is vital to instrument and measure the software development process. Some software metrics are developed to help monitor this.

Slide 13:

Enough of the requirements of database technology in SEE. I want to turn the topic back to DBMS and a system similar to it: OMS which stands for object management system. OMS are model-based distributed applications used for managing complex physical environments. The first a few OMS was developed during the 80s and early 90s of the last century. It has been widely used in the software engineering community. Here on the left shows the difference between DBMS and OMS. The real reason for the implementation and use of OMS has been the perceived or real limitations of available DBMS combined with the complexity of implementing a full-fledged DBMS.

Slide 14:

In this slide, I am going to show some examples of OMS that is used widely. Firstly the Adele-DB. It is organized around the concepts of software product, work environment and software process. The software processes is the set of actions executed in work Environments and which result in software products. There are four main components: Object Manager, Activity Manager, Configuration Manager and Process Manager contribute. On the leftside, this is an example diagram of the use of Adele DB. On the right side, this is another popular OMS system called Triton. A OMS that was developed as prototype to explore the requirements for software environments and to provide prototypical solutions.

Slide 15

Then, I want to talk about some proprietary software engineering databases. Proprietary database means software whose source code is unavailable to users. Firstly, the EPOSDB, basically is a private database under EPOS corporate. It is a general entity relationship based DBMS and it is structurally object-oriented, built upon the C-ISAM sequential file system and provides a novel change-oriented versioning model, and nested, cooperative transactions. Next, the GRAS DBMS, provides a innovative graph based data model in which artifacts are represented as attributed nodes and relationships as edges in between.

Slide 16:

Finally, I want to discuss with you guys about the commercial DBMS available nowadays. These databases have been used as component repositories for many SEEs. Sybase for example, a commercial relational DBMS, is used to store artifact descriptions in JBCL which is the JB component library system. Other databases like commercially object-oriented DBMS have also been used. For instance, O2 in Memphis which is used to store process and organizational data. And here is a table that shows some information about special-purpose DBMS that are used by corporates.