Database notes for beginners pdf

This page contains my latest class notes for my database, data storage and analytics, system analysis and design, device relationship modeling, and notes for ER models and charts. The course notes are detailed and full of information. Multiple class notes are updated regularly. Please enjoy and feel free to send me comments and suggestions and share with friends. Introduction to the database course and Introduction to databases Provides a general introduction to database systems and their components, some history of database models and their use. Systems Analysis and Design Systems Analysis uses a structured, formal method to model and improve business processes. This set of notes introduces system analysis concepts (SA), how to plan and model systems, and how to use SA on your database. Device relationship modeling This section of the notes covers the basics of entity relationship modeling. E-R models are an important component for designing an effective database solution to a business problem. Good E-R models lead to easy application development, maintenance and use. Drawing unit relationship diagram training The following guides describe how to draw ERDs using a variety of drawing tools. Convert e-R models to relationship models This part of your notes covers the conversion of entity relationship models to relationship models. Sometimes this mapping is called an ER chart or model to the relationship model. Normalization This section of notes covers the process of normalization where relationships created from the conversion from the E-R model are parsed for potential errors, and these errors are corrected. Structured Query Language (SQL) This section of notes covers SQL – the language used to define and manipulate data in relational databases. A sample database and many examples are provided. File organization and indexing This section of notes covers the organization of data on the disk and the various data structures used to speed up data retrieval. Database transaction processing This set of notes describes transactions in database management systems. DatabaseCompency Control This set of notes introduces the topic Concurrent Control in databases. Two phase locking are discussed and demonstrated. Database Recovery Database hardware and software may crash or fail. Database recovery is the process of restoring the database to its most recent state. In this section, we cover two common approaches used to restore databases. Database system architectures A database system consists of data, the database management system, business logic and the user interface. Database Systems Architecture refers to how these four components are arranged, distributed and communicate with each other. In this section of notes, we discuss 5 different architectures that often occur. Web Database Integration Modern database systems typically use or mobile apps as a user interface. This means that the business rules, the business rules, and data will be on other systems. This section of notes introduces some basic ways database systems can incorporate web and mobile apps. Database Security and Management This set of notes provides a brief introduction to the security subsystem available on most commercial database management systems. It also introduces the roles and responsibilities of the database administrator (DBA) and the data administrator (or chief data officer). The advanced database and data storage course notes notes below are generally useful for an advanced database course. In particular, they are used in data storage and analysis courses. Database Reverse Engineering Often we do not have the luxury of designing a database from scratch, and we have to work on an existing form. Reverse development for databases is the process of recreating physical or conceptual computer models based on an existing form. A dedicated tutorial about Reverse Engineering a database schema using Oracle SQL Developer is available here. A dedicated training on Reverse Engineering using MySQL Workbench is also available. Introduction to data storage In this set of notes, we cover a very basic introduction to Data Warehousing, including how DW systems differ from OLTP systems, how DW systems are designed, how the DW database schema looks and finally what are some general business and technology issues around DW projects. Data warehouse project planning In this collection of notes, we cover what makes data warehouse development projects different from traditional system development projects. Dimensions, Facts and Cubes – Some examples This is a short set of examples that show the concepts of dimensions, facts, and data cubes, including the concepts of cutting and dice. Dimension modeling In this section of the notes, we introduce dimensional modeling, including defining dimensions, factual cubes, and discuss the development of dimensions. Extract, Transform and Load (ETL) In this part of the notes, we will cover the key subsystems and issues involved in the extract, transform and load process, as well as compare ETL with ELT approaches. Technical architecture for data warehouses and product selection This part of the notes covers the technical architecture and product selection of the BI system. In particular, we compare OLAP implementations using traditional relationship star forms and multidimensional databases. CAP Theorem In this set of notes, we discuss CAP Theorem of consistency, availability and partition tolerance as it applies to large data stores. A brief review of ACID properties and traditional data replication architectures using two phase execution is given as an introduction. NoSQL Databases The purpose of this section of notes is to introduce some flavors of NoSQL databases such as key value stores, document stores, column store databases and graph databases. Introduction to Hadoop and Apache Spark is a highly scalable storage storage increasing processing architecture is used to support data storage and Big Data application. This section of notes introduces Hadoop Eco System and Spark distributed processing architecture. Introduction to web services and web APIs Development of distributed systems requires coordination between service providers and service consumers. Web services are the general term for this architecture. In this set of notes, the components of service-oriented architecture are presented in addition to the lightweight services exemplified by REST/JSON used in various Web APIs. BI Application Design and Development Business Intelligence Applications (BI Applications) are the software interfaces that end users interact with to analyze data. In this section of notes, we explore the issues and techniques surrounding the creation of BI Applications. Advanced database record and index storage In this collection of notes, I dive deeper into database internals to show how databases store data and indexes on the disk. How data is organized on the disk has implications for database performance and alignment. Database query management performance tuning and optimization database often require a deeper understanding of how queries are processed and optimized in DBMS. In this set of notes, we provide a general overview of how rule-based and cost-based query optimizations work, and then provide some specific examples of optimization in commercial DBMS. Data warehouse query optimization In this section of your notes, we discuss issues specific to query optimization and other performance issues related specifically to data storage, including star additions. . . . . Last updated on Nov 27,2019 4.4K Views As you all know, database management system (DBMS) is software used to manage databases. So, this article about DBMS Tutorial will help you understand both, basic and advanced concepts of DBMS. The topics discussed in this article are: Let's start! What is a database? The database is an organized collection of structured data to make it easily accessible, manageable and refreshable. In simple words, you can say a database in a place where the data is stored. The best analogy is the library. The library contains a large collection of books of different genres, here the library is database and books are the data. In the initial phase of the data time, data was collected and stored on tape, which were mostly read-only devices, which meant that when data was stored on it, it could never be read again. They were slow and bulky, and soon computer scientists realized that they needed a better solution to this problem. Together, the data and DBMS, along with the applications associated with them, are referred to as a database system, often shortened to just a database. The development of DatabaseDatabases has evolved since its inception in the early 1960s. In the 1980s, relational databases were followed by object-oriented databases in the 1990s. More recently, recently, databases came in response to the growth of the internet and the need for faster speed and processing of unstructured data. Today, cloud databases and self-driving databases are breaking new ground in terms of how data is collected, stored, managed, and exploited. Databases' is a very big topic. So, covering the themes under this topic is a very tedious task. DBMS Tutorial: Properties of DatabaseNow, the main characteristics of a database include: It uses a digital repository established on a server to store and manage the information.| The database should be able to store all kinds of data contained in this real world. It can provide a clear and logical view of the process that manipulates data. Most importantly, the database is used to provide security for data. DBMS contains all automatic backup and recovery procedures. It also contains ACID properties that maintain data in a healthy state in case of failure. The database can reduce the complex relationship between data. It is also used to support the manipulation and processing of data. You can view the database from different points of view according to the requirements specified by the user. Now, talking about the applications in a database, we will see how accurately you can make use of Database.DBMS Tutorial: Applications of DatabaseDatabase applications are applications designed to collect, manage and disseminate information very efficiently. So many small business owners create simple databases like customer contact and mailing lists with easy-to-use software, and there are companies that use the data manipulation pre-databases. Accounting ApplicationsWeb applicationsCRM applicationsAccounting programsStalking about the accounting system, it is a custom database application used to manage financial data. You can use the custom forms that are used to record fixed assets, liabilities, inventory, and transactions between customers and vendors. Get an overview of income statements, balances, purchase orders, and invoices that are generated, custom reports are based on information entered in the database. Accounting programs run on a single computer suitable for a small business or in a networked shared environment to meet the needs of multiple departments and locations in larger organizations. Web applicationsMane web applications also use databases to store data. This may be confidential information about an organization or other private information about the user. The database is used to store data in sequential order and helps you access data when necessary. Also, many web applications are created using database applications. There are websites that also combine an accounting database system to record sales transactions and a CRM database application to incorporate feedback and drive positive customer experience. We discuss the CRM database in the next topic. The most popular web-based application Facebook is a database built on the MySQL database system and is an indication of the increasing use of database applications as the basis for web-based applications. CRM applicationsA Customer Relationship Management System (CRM) is a perfect database application that is customized to manage marketing, sales, and support relationships between a business and its customers. The main goal is to maximize sales, minimize costs and promote strategic customer relationships. BenefitsReduced data redundancy. There are also reduced errors and increased consistency. Easier data integrity from application applications. Improved data access to users using host and query languages. Data security has also been improved. Reduced data entry, storage and retrieval costs. DisadvantagesComplexity: Databases are complex hardware and software systems. Cost: It requires significant upfront and ongoing financial resources. Security: Most leading companies need to know that their database systems can securely store data, including sensitive employee and customer information. Compatibility: There is a risk that a DBMS may not be compatible with the company's operational requirements. Now that you have got an idea of how the Database works, let's go ahead and understand the Database Management System.DBMSA Database Management System (DBMS) is a software used to manage the database. It receives instructions from a database administrator (DBA) and therefore instructs the system to make the corresponding changes. These are basically commands used to load, retrieve, or modify existing data from the system. Database Management Systems also aims to provide an overview of the databases, by providing a variety of administrative operations such as tuning, performance monitoring and backup recovery. Database management systems allow users to do the following:Define data – Allows users to create, modify, and delete the definitions that define the organization of the database. Refresh data – Allows users to insert, modify, and delete data from the database. Retrieve data – Allows users to retrieve data from a database based on the requirement. User management – Records users and monitors their action, enforces data security, maintains data integrity, monitors performance, and handles concurrentity control. PropertiesTo restrict access permissions for usersRegiating multiple views of the individual database schemaFacilitates security and removes data redundancyAllow's multi-user transaction processing and data sharingGet the ACID propertyOffers both physical and logical data independenceNow, let's see how to create a database. We use the CREATE DATABASE statement to create a new database. Syntax:CREATE DATABASE DATABASE NAME; Example:CREATE DATABASE College; So the database named College will be created. This is how easily you can create a Database.Now let's understand the applications of DBMS. Applications of and productionUniversitiesThese are some of the remarkable applications of DBMS. Now, let's go further and understand the features of DBMS. DBMS Tutorial: FeaturesMinimum duplication: There are many users who use the database so the chances of data duplicity are very high. In the database management system, data files are shared that in turn minimize data duplication. Saving storage: DBMS has a lot to save, but the integration of data into a DBMS saves much more space. Cost-effective: Many companies pay so much money to store their data. If they have managed data to store, it will save the cost of data entry. Security: DBMS stores all data files permanently, and there is no chance that you may lose data. For example, you lose some data, then there's also a backup and recovery method that can store your organization's data files. So DBMS is very secure. Let's understand the architecture of DBMS. ArchitectureDesigning DBMS, depends mainly on the architecture. The architecture can either be centralized or decentralized or hierarchically. It can be seen as a single-tier or multi-tier. You can also have an n-layer architecture that divides the entire system into related but independent n-modules, which can be independently modified, modified, modified, or replaced. You can have:Single-tierTwo-tierThree-tierSingle-tierHere a database is directly available to the user. This means that the user can lie directly on a DBMS and use it. Any changes made here will be made directly to the database itself. And it does not provide a convenient tool for end users.1 level is used where the client, server and database all res located on the same machine. When you install a database in the system and access SQL queries, the 1-layer architecture is used. However, this architecture is rarely used in the production section.2-TierArchitecture is the same as the basic client server. In this architecture, client-side applications can communicate directly with the server-side database. To communicate with DBMS, the client-side application establishes a server-side connection. When the client computer makes a request to access the database that exists on the server by using SQL, the server executes the request on the database and returns the result back to the client. The three-layer architecture of 3 layers contains a layer between the client and the server. Here, the client cannot communicate directly with the server. The end user has no idea about the application server. The database also has no idea of any other user beyond the program. The application that exists on the client side interacts with an application server that in turn communicates with the database system. It has three layers or levels namely Presentation Layers, Program Layers, and Database Layers. Database level: At this level, a database exists together with the processing languages (Query). You also have the relationships that define the data and limitations at this level. Application level: It is also called the middle level. This level consists of the application server and the applications that have access to the database. For a user, this application level displays an abstract view of the database. At the other end, the database level is not aware of other users outside the application level. Therefore, the application layer sits in the middle and acts as an intermediary between the end user and the Database.User level: This is also called as a presentation level. The end users operate at this level and know nothing about the existence of the database beyond this layer. In this layer, multiple views of the database can be provided by the application. All views are generated by applications that exist at the application level. Now that you have understood the architecture, let's go ahead and understand the components of DBMS. DBMS Tutorial: ComponentsStalking about the components of DBMS, we have: This consists of a set of physical electronic devices such as I/O devices, storage devices and many more. It also provides an interface between computers and real systems. This is the set of applications used to control and manage the general database. It also includes the DBMS software itself. The operating system, the network software used to share the data between users, the applications used to access data in DBMS. Database Management System collects, stores, processes, and accesses data. The database contains both actual or operational data and the metadata. These are the rules and instructions for using the database to design and run DBMS to guide the users who operate and manage it. It is used to access the data to and from the database. In order to enter new data, requires updating or retrieving data from databases. You can type a set of appropriate commands in the database access language, send them to DBMS, which then processes the data and generates it, displays a set of results in a user-readable form. Now that you have understood the components of a database, let's move on and understand the types. DBMS training: The following types are the different types of DBMS:Hierarchical: This type of DBMS displays a predecessor-trailing type of relationship style. You can consider it to be similar to a tree, where the nodes in the tree represent records and the branches of the tree represent fields. Relational Database(RDBMS): This type has a structure that allows users to identify and access data relative to another part of the data in the database. Here the data is stored in the form of tables.Network: This type of database management system supports many to many relationships where multiple user records can be linked. Object-oriented: It uses small individual software called objects. Here, each object contains some data and the instructions for the actions to be performed with the data. DBMS Tutorial: Computer modelsData models in DBMS help define how structure of a database is modeled. Computer models are basically the basic entities that introduce abstraction in DBMS. These data models also define how data is connected to each other and how it is processed and stored inside the system. Why do you need this computer model? It ensures that all data objects required by the database are accurately represented. Omission of data at times will lead to the creation of erroneous reports and give incorrect results. A data model helps design the database at conceptual, physical, and logical levels. The structure helps define relationship tables, primary and foreign keys, and stored procedures. It is also useful to identify missing and redundant data. This data model can further be divided into these types:Types of data modelConceptualPhysicalLogicalNow, let's see the work of these data models. ConceptualThis type of data model defines what the system contains. The concept model is made by computer architects in general. The purpose is to organize, scope and define business concepts and rules. There are 3 basic styles under Conceptual Data Models:EntityAttributeRelationshipThis can be referred to as the Entity-Relationship Model.Entity-Relationship (ER) model is based on the idea of real entities and relationships between them. This ER model is best used for the conceptual design of a database. Entity: An entity in an ER model is a real entity that has properties that are named attributes. Each attribute is defined by the set of values called Domains. For example, consider a student's details. The details such as name, age, class, part and all of these come under the device. Relationship: The logical association between entities is called a relationship. These relationships are assigned with entities in different ways. The mapping (one-to-one, one-to-many, many-to-many) defines the number of association between two entities. Now let's understand physical data model. PhysicalA Physical Data Model helps describe the database-specific implementation of the data model. The Physical Data model offers an abstraction of the database and helps generate schema. This Physical Data model also helps visualize the database structure. It also helps model database column keys, constraints, indexes, triggers, and other RDBMS functions. Now let's understand the logical data model. LogicalLogical data models help add more information to the concept model elements. This model defines the structure of the data elements and also specifies the corresponding relationships between them. At this level, no primary or foreign key is defined, and you must check and adjust the link details that were previously specified for relationships. The main advantage of this logical computer model is to provide a basis for forming the basis of the physical model. I hope this is clear to you. Go ahead with dbms tutorial, let's take a look at the keys in DBMS. DBMS Tutorial: KeysKeys the most important concept of databases. Keys play an important role in the relational database. This is used to identify unique rows from the table. It also establishes the relationship between tables. Why do you need these keys in the database? The answer to this would be,In a real application, a table can contain thousands or even more number of records. Moreover, the records can also be duplicated. Keys ensure that you can uniquely identify a table record despite many challenges. The keys also allow you to create a relationship and also identify the relationships between tablesKeys also help you enforce identity and integrity in the relationship. Types of KeysDBMS have different keys that have different functions. Super KeyPrimary KeyCandidate KeyForeign KeyCompound KeyLet's discuss the most commonly used keys in DBMS. Candidate key: The minimal set of attributes that can uniquely identify a tip is called a candidate key. A relationship can contain more than a single candidate key, where the key is either a simple or complex key. Super Key: The set of attributes that can uniquely identify a tip is called Super Key. So, a candidate key is a superkey, but vice-versa is not true. Primary key: A set of attributes that can be used to identify each tip is also a primary key. So, if there are 3-4 candidate keys present in a relationship, then out of these, one can be chosen as a primary key. Alternate key: Candidate key other than primary key is called as an alternate key. Foreign key: An attribute that can only take the values that exist as the values for another attribute is the foreign key of the attribute that it refers to. Move on to the last topic of this article on DBMS Tutorial, let's learn about normalization in DBMS. NormalizationNormalization is the process of reducing the redundancy of data in the table and also improving data integrity. So why is this necessary? without normalization in SQL, we can face many problems such asInsertion anomaly: It occurs when we cannot insert data to the table without the presence of another attributeUpdate anomaly: There is a data inconsistency caused by data redundancy and a partial refresh of data. Deletion anomaly: It occurs when certain attributes are lost due to deleting other attributes. This image below shows how normalization in SQL works. So, with this, we come to the end of this DBMS Tutorial. I hope you are clear about the topics discussed in this tutorial. If you want to learn more about MySQL and get to know this open source relational database, then check out our MySQL DBA Certification Training which comes with instructor-led live training and real project experience. This tutorial will help you understand MySQL in depth and help you achieve mastery over the topic. In case of questions you can put them in the comments box in DBMS Tutorial and we will go back at the earliest. Earliest.