**Database Management Overview**

|  |  |
| --- | --- |
| **What is a database?**   * An integrated collection of related data * Represents information in the real world * Traditional databases use the relational model which organizes data into relations (aka, tables) | A **University Database** may contain:  **Student Table** – contains information about a student  student name, address, phone, student ID, total grade points, total credit hours, major  **Course Table** – contains information shown in the catalog about a course  course number, title, description, department, credit hours  **Section Table** – contains an offering of a course as part of a semester schedule course number, semester, section, day, time, room, prof, max seats, current seats  **Enrollment Table** – contains students enrolled in particular sections  course number, semester, section, student ID, grade  **Prereq Table** – contains a course and its prerequisites course number, prereq course number |
| **What is a database management system (DBMS)?**  A software package for storing and managing databases. | **Some major DBMSes**  **Early DBMSes:**  IMS – (mid 1960s) IBM Information Management System, IBM mainframe-based hieararchical (tree structured database), no query optimizer, programmer had to navigate through the data  CODASYL Database Task Group (DBTG) – graph-structured database (entities can have multiple parents), no query optimizer, programmer had to navigate through the data  **Relational DBMSes**:  IBM DB2 – IBM mainframe-based, high performing, uses SQL  ORACLE – Larry Ellison's DBMS based on IBM's System R research, high performing on mostly non-mainframe, uses SQL  Microsoft SQL Server – Microsoft's relational DBMS to compete with ORACLE, uses SQL  MySQL – open source relational DBMS, uses SQL  IBM UDB – distributed version of IBM's relational product  **Big Data**  NOSQL – manage very large less-structured data (e.g., emails, social media posts, pictures, videos)   * Google's BigTable * Amazon's DynamoDB * Facebook's Cassandra * MongoDB – commercially available for documents   HaDoop – uses map reduce to divide and conquer |
| **Additional Terminology**  **Field** The smallest accessible piece of data. Usually a group of bytes (i.e., attribute, **column**). Examples:  **Record** A collection of fields.  **Physical File** A collection of records.  **Key** Specifies the identity of a record.  **Primary Key** A specially designated unique key used to access a particular database record.  **Secondary Key** Used to access possibly many database records by values in one or more fields.  **Data Model** A collection of concepts for describing data.  **Schema** A description of a particular collection of data including how it is organized and how the data may be manipulated.  **Relational Data Model** A data model where data is organized into ***relations***.    **Relation** A table with rows (database records) and columns (fields).  **Domain** The set of possible values for a column. | **Field (aka Column) Examples:**   * Student Id * Declared Major * Loan Balance * Total Grade Points * Automobile License Plate Number   **Primary Key vs Secondary Key**   * Student Table's primary key is student id * Course Table's primary key is ?? * Section Table's primary key is course number, section number, and semester * We may also want to access the Section table by   + ?? * Enrollment Table's primary key is the combination of ?? * We may also want to access Enrollment by   + ??   + ??   + ?? * Prereq Table's primary key is ?? * We may also want to access Prereq by   + ??   + ?? |
| **Why use a DBMS – Persistence of Information**   * Provide software with the ability to Create, Retrieve, Update, and Delete data. (CRUD) * Also see Backup and Recovery |  |
| **Why use a DBMS – Performance and Efficiency**   * Provide high-speed query processing against large quantities of data * DBMSes provide query optimizers which determine the best way to satisfy queries | **Example Queries:**   * Find each enrollment for student "ABC123" this semester.   + Why is it important for this query to be fast and efficient?     - Students do this fairly often * Determine students who may be a candidate for graduating this semester.   + Find students with ?? |
| **Why use a DBMS – Integrated Processing**   * Data can be viewed separately or as an integrated whole * Allows more information to be obtained from the data | **Example Queries:**   * What sections are both students "ABC123" and "DEF456" enrolled? * What prof teaches courses with the average grade > 3.0? |
| **Why use a DBMS – Data Independence**   * Programs should not have to change because of the following changes to the database:   + Addition of new fields   + Addition of new indexes   + Changes of structure from hashing to indexes   + Removal of fields which are not referenced by the program. * Changes to programs should not require changes to the database | In traditional file-based applications:   * adding a field to a record would require: * reorganizing the file to add the new field * since the record is longer, the code accessing the file would have to be changed:   + add the field to the struct   + recompile the programs * removing a field from a record would require: * reorganizing the file to remove that field (or adding filler) * since the record is shorter, the code accessing the file would have to be changed:   + delete the field from the struct   + recompile the programs |
| **Why use a DBMS – Reduction of Redundancy**   * With proper design, we can reduce redundancy * Save file space * Reduce processing requirements. (Do not have to update multiple files) * Improves integrity | Should Prof Office be a column in the Section Table?  ?? |
| **Why use a DBMS – Centralized Control of Data and Security**   * Centralized group of Database Administrators (DBAs) are responsible for defining the data and optimizing physical implementation * DBAs (or security personnel) can define who can access the data and what they can do with it (retrieve, insert, update, delete). * Enforce integrity constraints * DBAs can tune the database | DBAs typically make more money than programmers.  Some data is encrypted to help reduce identity theft.  DBAs can examine high demand queries and determine mechanisms to improve performance.  Some DBMSes provide:   * information about the cardinalities of values in specific columns * query optimization plans to explain to programmers and DBAs how specific queries are executed |
| **Why use a DBMS – Concurrent Access with Integrity**   * Allow multiple people to access the same databases simultaneously while preventing integrity problems * Provide transactions where all operations in a transaction are done or none of them are allowed * Since data is I/O intensive, having more users will better utilize the CPU | |  |  | | --- | --- | | **Application 1: dividend posting**  Retrieve Balance for Acct #111  Balance = Balance \* 1.01  Update Balance | **Application 2: deposits/withdrawals**  Retrieve Balance for Acct #111  Balance = Balance - WithdrawalAmt  Update Balance |   Suppose the Balance for Acct#111 is $200. WithdrawalAmt is $50.   |  |  |  | | --- | --- | --- | | Time | Application | Action | | t0 | 1 | Retrieves Balance for #111. Balance1 is 200. | | t1 | 2 | Retrieves Balance for #111. Balance2 is 200. | | t2 | 2 | Updates Balance for #111 to be 150. | | t3 | 1 | Updates Balance for #111 to be 202. |   What is the issue?  ?? |
| **Why use a DBMS – Backup and Recovery**   * Automated backups for restoring in the event of a major failure * Transaction logs to recover to a point in time | Since data is such an important asset to companies, what happens if the hard drive containing important data has hardware issues? Companies may go out of business if they lose their data. |
| **Why use a DBMS – Simplify Application Coding**   * Provide well-defined data * Provide easy-to-use data manipulation languages * Reduces application development time | **Structured Query Language (SQL)** is the most popular data manipulation language. At least one of our assignments will involve coding in either Java or C and using SQL. |
| **Why use a DBMS – Convienent Tools**   * Most major DBMSes also provide convienent tools for accessing the data without having to write code | **Many DBMSes provide (or these can be obtained from other vendors)**   * report writing * database table data display and modification tools |

**©2020 Larry W. Clark,** UTSA CS students may make copies for their personal use