DOWNLOAD PDF BASIC RELATIONAL DATABASE CONCEPTS

Chapter 1 : Basic Relational Database Concepts/Design, The SQL Language, & The mySQL Database Se

1 Relational Database Concepts for Beginners A database contains one or more tables of information. The rows in a table are called records and the columns in a table are called fields or attributes.

Our life is frittered away by detail Users can insert new information into the database, and delete, change, or retrieve existing information in the database, by issuing requests or commands to the software that manages the databaseâ€"which is to say, the database management system DBMS for short. Throughout this book, I take the term user to mean either an application programmer or an interactive user[2] or both, as the context demands. Now, in practice, those user requests to the DBMS can be formulated in a variety of different ways e. Given a human resources database, for example, we might write: The suppliers-and-parts databaseâ€"sample values As you can see, this database contains three files, or tables. Table S represents suppliers under contract. Table P represents kinds of parts. Table SP represents shipmentsâ€"it shows which parts are shipped, or supplied, by which suppliers. Examples throughout the rest of this book are based for the most part on the foregoing database. That logical database is contrasted with the corresponding physical database, which is the database as perceived by the DBMS i. As that figure is meant to suggest, the DBMSâ€"the software that manages the databaseâ€"effectively serves as a kind of intermediary between the logical and physical levels of the system: Database system architecture One general function provided by the DBMS is thus the shielding of users from details of the physical level of the system very much as programming language systems also shield users from details of the physical level of the system. The point is this: Most requests indeed, very likely all requests are capable of being implemented in a varietyâ€"typically a very large varietyâ€"of different ways. Moreover, those different ways will typically have widely differing performance characteristics; in particular, they could have execution times that vary, quite literally, from fractions of a second to many days. One immediate and significant implication of the foregoing is this: As a matter of factâ€"to jump ahead of myself for a momentâ€"let me state here for the record that it is and always was a major objective of the relational model that it should be the system, not the user, that has to worry about performance issues. Indeed, to the extent this objective is not met, the system can be said to have failed or certainly to be less than fully successful, at any rate. Data Independence The fact that the logical and physical databases are distinguished and ideally, at least kept rigidly apart is what allows us to achieve the important goal of data independence. Data independenceâ€"not a very good term, by the way, but we seem to be stuck with itâ€"means we have the freedom to change the way the database is physically stored and accessed without having to make corresponding changes to the way the database is perceived by the user. Now, the reason we might want to change the way the database is physically stored and accessed is almost always performance; and the fact that we can make such changes without having to change the way the database looks to the user means that existing application programs, queries, and the like can all still work after the change. Very importantly, therefore, data independence means protecting existing investmentâ€"investment in user training, in existing applications, and in existing database designs among other things. Thus, the DBMS is responsible for a accepting user requests, be they queries or updates, that are expressed in terms of the logical database and b responding to those requests by interpreting and implementing them, or in other words executing them, in terms of the physical database. The term update, in lower case,[5] is used to refer generically to requests that insert new data or delete or change existing data. We might also say, somewhat glibly, that it protects the data from users! Security controls are needed to ensure that user requests are legitimate, in the sense that the user in question is requesting an operation he or she is allowed to carry out on data he or she is allowed to access. In the case of the suppliers-and-parts database, for example, some users might not be allowed to see supplier status values; others might not be allowed to see suppliers at all; others might be allowed to see suppliers in London but not in other cities; others might be allowed to retrieve supplier information but not to update it; and so on. Concurrency controls have to do with the possibility that several users might be using the database at the same time. Likewise, an attempt to update the status for supplier S1 to must also be rejected, if status values are supposed never to exceed The problem is, if you call the DBMS a database, what do you call the database? So now we know what a DBMS is. In other words, the relational model can be thought of as a kind of recipe for what the user interface is supposed to look like in such a DBMS. Easier for the system too, in certain respects; but the emphasis is on the user. In fact, it seems to me that the concepts in question are much simpler than their counterparts were in older, prerelational and nonrelational systems such as IMS and IDMS. The data looks relational. Now, a standard concrete language does exist: As noted in the preface, however, SQL is very deeply flawed: So what I plan to do in this book is this: I believe Tutorial D is pretty much self-explanatory; however, a comprehensive description can be found if needed in the book Databases, Types, and the Relational Model: You might or might not know but I hope you do that the relational model was originally the invention of E. Codd, when he was employed as a researcher at IBM E for Edgar and F for Frankâ€"but he always signed with his initials; to his friends, among whom I was proud to count myself, he was Ted. It was late in that Codd, a mathematician by training, first realized that the discipline of mathematics could be used to inject some solid principles and rigor into a field, database management, that prior to that time was all too deficient in any such qualities. His original description of the ideas of the relational model appeared in an IBM Research Report in see Appendix E for further discussion. The purpose of that code fragmentâ€"which is expressed in a hypothetical but self-explanatory languageâ€"is to compute and display the sum of the integers in a certain one-dimensional array called A. A code fragment Note the following points: The code overall consists of nine statements. A statement in a programming language is a construct that causes some action to occur, such as defining or updating a variable or changing the flow of control. A type is a named set of values: Every value and every variable see further discussion below is of some type. A variable is a container for a value different values at different times, in general. In fact, to be a variable is to be assignable to, and to be assignable to is to be a variable. Note in particular that literals and variable references are both expressions, since they certainly both denote values. Note too that every value effectively carries its type around with it. For example, many programming languages support a read-only operator called RANDOM or some such for generating pseudorandom numbers. More on Types I need to say a little more about the concept of types in particular. They can also be arbitrarily complex. In the case of variables, parameters, and read-only operators, the type in question is specified when the construct in question is defined. For example, see the variable definitionsâ€"i. In the case of expressions, the type in question is simply the type of the result returned when the expression in question is evaluated. For example, in the case of type INTEGER, which for simplicity I take to be system defined, the agency responsible for defining the typeâ€"in other words, the system, by my assumptionâ€"must define: This requirement is sometimes referred to as The Assignment Principle. Note the following important corollary:

Chapter 2 : Relational Database Design/Basic Concepts - Wikibooks, open books for an open world

Basic Concepts on how relational databases work. Explains the concepts of tables, key IDs, and relations at an introductory level. For more info on Crow's Fe.

Well, a database is quite simply an electronic, organized, collection of data, in a repository that can be accessed, manipulated and retrieved when necessary. One can create different types of databases. The most common type of database in use currently is the relational database, however, one can also create non-relational databases which are also referred to as NoSQL databases. What is a relational database? The key structural fundamental unit of a database is a table. Tables in relational databases are all related to one another through the use of keys, which facilitate the creation of relationships. So a relational database is quite simply put, a group or collection of related tables. How are tables structured in relational databases? Tables in relational databases are made up of records rows and columns fields. Each table must contain a primary key field. Each row can then consequently be identified by a unique identifier through the use of the primary key field. Tables store information about single entities, a single theme or idea. For example, a database that stores information about a bakery would have a Products table, an Orders table, and a Customers table, among others. These would all be fields or column headings in the Customers table. Relationships would consequently be created between these tables. Many people are already quite familiar with creating tables in Excel. However, while rows and columns have a relational database analog in terms of records and fields. The rules governing table creation in relational databases are more strict than those in Excel. Excel tables can also be and are frequently standalone â€" as in they do not need to be related to other tables in the workbook. Relationships are created through the use of keys. A primary key in a table is a unique identifier field that allows for unique identification of each row in the table. The Owners table would have a column called OwnerID, which would uniquely identify every owner in the table. In order to create a relationship, this primary key is placed on another table. It is now referred to as a foreign key when placed on another table. In order to create a relationship between the two tables, one investigates how they are linked, and then creates the relationship. The most common type of relationship and the relationship that you are most likely to encounter when viewing databases created by others, is a relationship type called the one-to-many relationship. Occasionally you may also encounter a relationship type called one-to-one, in this type of relationship, one record in one table has one record related to it in another table. SQL in 10 Minutes, Sams Teach Yourself Affiliate Link One may also encounter a many-to-many relationship type occasionally, in this situation, one or more records in one table are linked to one or more records in another table. The usual way many-to-many situations are handled in a database is to create a third table that links the other two tables through a standard one-to-many relationship. What is referential integrity? Referential integrity is a term used to describe a situation where a relationship has been created between two tables and every record in the one table is correctly matched to its associated record in the other table. In other words, the primary key table cannot have an unrelated foreign key in the related table. What is an entity-relationship diagram? While going over database theory and models, you will come across the term entity-relationship diagram. When planning a database, database administrators look at the different tables needed or entities. Each entity describes a theme or idea in and of itself. A relationship between entities or tables describes the relationship between the tables in the database. Entities also have attributes â€" these are the fields in a table which describe the table, for example, an Employees table would have EmployeeID, EmployeeFullName, employee title, job description etc. Entity-relationship diagrams can be set up on paper or be using the software. When visually representing entity-relationship diagrams in a comprehensive manner, the entities are denoted by rectangles and the relationships are denoted by lines. The primary keys, foreign keys, and other columns should also be identified in a comprehensive entity relationship diagram. However when first planning your database, you may just make a simplified entity-relationship diagram and then later once you have identified your main entities or tables and the relationships between them, you can design a more comprehensive entity-relationship diagram that further specifies the attributes and primary and foreign keys. What is a database management system

DBMS? A database management system DBMS is a program that allows database administrators to create, update and manipulate databases and the data stored in databases. There are many database management systems on the market â€” some are commercially available while others are open source. Many are industrial level and one only needs to really consider these types of packages for enterprise level transactions or if one is studying to be a database architect. SQL or Structured Query Language is the language used by relational databases that allows one to create, update and manipulate data in databases. It is a relatively simple language to learn and through the use of statements incorporating the words: Each vendor has their own version of SQL, which differs slightly in some aspects, however, once you have learned one version, you can easily pick up any other version of SQL. What is a query? A typical simple query could be â€” select all the customers from a particular region. The database, should if designed correctly be able to retrieve the answer or information that the query is asking for or execute the command given by the query. Forms are database objects that can be created in order to separate front-end users and their data entry requirements from the back-end tables. Forms allow database administrators to create a graphical user interface that simplifies data entry for the front-end user, and also protects the back-end tables from alteration. One may not want to expose the back-ends tables or relationships to front-end users since they may edit the relationships for example and then queries will not work. So forms resolve this issue. You can choose to add only fields or calculations from one table in your report, or you can choose to add fields of interest from multiple tables to your report. You can add calculation summaries such as the average, maximum and minimum on your reports, as well as a group your data. So how does Access fit into the picture? Microsoft Access is a relational database management system provided by Microsoft. It is the most popular desktop database software package currently in use. It is a good middleman between Microsoft Excel â€” the popular and widely utilized spreadsheet application and Microsoft SQL Server, the industrial enterprise level relational database management system. If tables are correctly structured in Excel, Access can import them, and these tables can further be manipulated in Access. In addition you can create tables, queries, forms and reports â€” basically, build a very comprehensive and useful database in Access without ever having to use SQL statements. Access allows one to create tables, queries, forms and reports via the familiar ribbon interface. However, despite the fact that you can create perfectly functional databases in Access without ever having to learn SQL if you start looking at creating more advanced data-centric applications in Access for specialized needs, then having knowledge of how to create SQL queries directly is very useful. You can also enhance your databases through the use of Visual Basic for Applications or VBA and Access programming is a useful skill for advanced or specialized needs or organizations. You will, however, need to understand and master the concepts associated with basic relational database theory. Ultimately your main goal when creating databases is to create a central repository that efficiently stores data and upon querying returns the correct, needed information. And there you have it. Go from SQL Beginner to Expert Affiliate Link Conclusion When embarking on the Microsoft Access or any other relational database learning curve and journey, there are a few concepts and terms that one has to master and be familiar with in order to accurately design relational databases. You will frequently come to certain terms in books, tutorials and be learning manuals and as you get started creating your database objects you will apply certain methods such as creating relationships frequently. Therefore, in order to not be confused and fully exploit the power of the relational database management system, you need to have a passing familiarity with some relational database theory. Once you have mastered the basics, you can proceed full speed ahead, with the actual practical application of some of the concepts. Please feel free to comment and tell us if you use Access currently or are thinking of getting started with Access or other relational database management systems.

Chapter 3 : A Tutorial on Database Concepts, SQL using MySQL.

Stay ahead with the world's most comprehensive technology and business learning platform. With Safari, you learn the way you learn best. Get unlimited access to videos, live online training, learning paths, books, tutorials, and more.

Relation database and Table database A relation is defined as a set of tuples that have the same attributes. A tuple usually represents an object and information about that object. Objects are typically physical objects or concepts. A relation is usually described as a table , which is organized into rows and columns. All the data referenced by an attribute are in the same domain and conform to the same constraints. The relational model specifies that the tuples of a relation have no specific order and that the tuples, in turn, impose no order on the attributes. Applications access data by specifying queries, which use operations such as select to identify tuples, project to identify attributes, and join to combine relations. Relations can be modified using the insert, delete, and update operators. New tuples can supply explicit values or be derived from a query. Similarly, queries identify tuples for updating or deleting. Tuples by definition are unique. If the tuple contains a candidate or primary key then obviously it is unique; however, a primary key need not be defined for a row or record to be a tuple. The definition of a tuple requires that it be unique, but does not require a primary key to be defined. Because a tuple is unique, its attributes by definition constitute a superkey. Base and derived relations[edit] Main articles: Relvar and View database In a relational database, all data are stored and accessed via relations. Relations that store data are called "base relations", and in implementations are called "tables". Other relations do not store data, but are computed by applying relational operations to other relations. These relations are sometimes called "derived relations". In implementations these are called " views " or "queries". Derived relations are convenient in that they act as a single relation, even though they may grab information from several relations. Also, derived relations can be used as an abstraction layer. Mathematically, attaching a domain to an attribute means that any value for the attribute must be an element of the specified set. The character string "ABC", for instance, is not in the integer domain, but the integer value is. Another example of domain describes the possible values for the field "CoinFace" as "Heads","Tails". So, the field "CoinFace" will not accept input values like 0,1 or H,T. Constraints[edit] Constraints make it possible to further restrict the domain of an attribute. For instance, a constraint can restrict a given integer attribute to values between 1 and Constraints provide one method of implementing business rules in the database and support subsequent data use within the application layer. SQL implements constraint functionality in the form of check constraints. Constraints restrict the data that can be stored in relations. These are usually defined using expressions that result in a boolean value, indicating whether or not the data satisfies the constraint. Constraints can apply to single attributes, to a tuple restricting combinations of attributes or to an entire relation. Since every attribute has an associated domain, there are constraints domain constraints. The two principal rules for the relational model are known as entity integrity and referential integrity. Unique key A primary key uniquely specifies a tuple within a table. In order for an attribute to be a good primary key it must not repeat. While natural attributes attributes used to describe the data being entered are sometimes good primary keys, surrogate keys are often used instead. A surrogate key is an artificial attribute assigned to an object which uniquely identifies it for instance, in a table of information about students at a school they might all be assigned a student ID in order to differentiate them. The surrogate key has no intrinsic inherent meaning, but rather is useful through its ability to uniquely identify a tuple. Another common occurrence, especially in regard to N: M cardinality is the composite key. A composite key is a key made up of two or more attributes within a table that together uniquely identify a record. For example, in a database relating students, teachers, and classes. Classes could be uniquely identified by a composite key of their room number and time slot, since no other class could have exactly the same combination of attributes. In fact, use of a composite key such as this can be a form of data verification , albeit a weak one. Foreign key A foreign key is a field in a relational table that matches the primary key column of another table. The foreign key can be used to cross-reference tables. Foreign keys do not need to have unique values in the referencing relation. Foreign keys effectively use the values of attributes in the referenced relation to restrict the domain of one or more attributes in the referencing relation. A foreign key could be described formally as: Stored procedure A stored procedure is executable code that is associated with, and generally stored in, the database. Stored procedures usually collect and customize common operations, like inserting a tuple into a relation , gathering statistical information about usage patterns, or encapsulating complex business logic and calculations. Frequently they are used as an application programming interface API for security or simplicity. Stored procedures are not part of the relational database model, but all commercial implementations include them. Index database An index is one way of providing quicker access to data. Indexes can be created on any combination of attributes on a relation. Queries that filter using those attributes can find matching tuples randomly using the index, without having to check each tuple in turn. This is analogous to using the index of a book to go directly to the page on which the information you are looking for is found, so that you do not have to read the entire book to find what you are looking for. Relational databases typically supply multiple indexing techniques, each of which is optimal for some combination of data distribution, relation size, and typical access pattern. Indices are usually not considered part of the database, as they are considered an implementation detail, though indices are usually maintained by the same group that maintains the other parts of the database. The use of efficient indexes on both primary and foreign keys can dramatically improve query performance. This is because B-tree indexes result in query times proportional to log n where n is the number of rows in a table and hash indexes result in constant time queries no size dependency as long as the relevant part of the index fits into memory. Relational algebra Queries made against the relational database, and the derived relvars in the database are expressed in a relational calculus or a relational algebra. In his original relational algebra, Codd introduced eight relational operators in two groups of four operators each. The first four operators were based on the traditional mathematical set operations: The union operator combines the tuples of two relations and removes all duplicate tuples from the result. The intersection operator produces the set of tuples that two relations share in common. The difference operator acts on two relations and produces the set of tuples from the first relation that do not exist in the second relation. The cartesian product of two relations is a join that is not restricted by any criteria, resulting in every tuple of the first relation being matched with every tuple of the second relation. The cartesian product is implemented in SQL as the Cross join operator. The remaining operators proposed by Codd involve special operations specific to relational databases: The selection, or restriction, operation retrieves tuples from a relation, limiting the results to only those that meet a specific criterion, i. The projection operation extracts only the specified attributes from a tuple or set of tuples. The join operation defined for relational databases is often referred to as a natural join. In this type of join, two relations are connected by their common attributes. The relational division operation is a slightly more complex operation and essentially involves using the tuples of one relation the dividend to partition a second relation the divisor. The relational division operator is effectively the opposite of the cartesian product operator hence the name. Database normalization Normalization was first proposed by Codd as an integral part of the relational model. It encompasses a set of procedures designed to eliminate non-simple domains non-atomic values and the redundancy duplication of data, which in turn prevents data manipulation anomalies and loss of data integrity. The most common forms of normalization applied to databases are called the normal forms.

Chapter 4 : Basic Database Concepts - Relational Theory for Computer Professionals [Book]

Basic Concepts A database is just what the name implies, a base collection of data. The data is organized in some manner so that the information contained within the database can be easily retrieved.

Oracle Database Documentation Roadmap About Relational Databases Every organization has information that it must store and manage to meet its requirements. For example, a corporation must collect and maintain human resources records for its employees. This information must be available to those who need it. An information system is a formal system for storing and processing information. An information system could be a set of cardboard boxes containing manila folders along with rules for how to store and retrieve the folders. However, most companies today use a database to automate their information systems. A database is an organized collection of information treated as a unit. The purpose of a database is to collect, store, and retrieve related information for use by database applications. Typically, a DBMS has the following elements: Repository of metadata This repository is usually called a data dictionary. Query language This language enables applications to access the data. A database application is a software program that interacts with a database to access and manipulate data. The first generation of database management systems included the following types: Hierarchical A hierarchical database organizes data in a tree structure. Each parent record has one or more child records, similar to the structure of a file system. Network A network database is similar to a hierarchical database, except records have a many-to-many rather than a one-to-many relationship. The preceding database management systems stored data in rigid, predetermined relationships. Because no data definition language existed, changing the structure of the data was difficult. Also, these systems lacked a simple query language, which hindered application development. Codd defined a relational model based on mathematical set theory. Today, the most widely accepted database model is the relational model. A relational database is a database that conforms to the relational model. The relational model has the following major aspects: Structures Well-defined objects store or access the data of a database. Operations Clearly defined actions enable applications to manipulate the data and structures of a database. Integrity rules Integrity rules govern operations on the data and structures of a database. A relational database stores data in a set of simple relations. A relation is a set of tuples. A tuple is an unordered set of attribute values. A table is a two-dimensional representation of a relation in the form of rows tuples and columns attributes. Each row in a table has the same set of columns. A relational database is a database that stores data in relations tables. For example, a relational database could store information about company employees in an employee table, a department table, and a salary table. Essentially, an RDBMS moves data into a database, stores the data, and retrieves it so that it can be manipulated by applications. Logical operations In this case, an application specifies what content is required. For example, an application requests an employee name or adds an employee record to a table. For example, after an application queries a table, the database may use an index to find the requested rows, read the data into memory, and perform many other steps before returning a result to the user. The RDBMS stores and retrieves data so that physical operations are transparent to database applications. Oracle Database has extended the relational model to an object-relational model, making it possible to store complex business models in a relational database. Brief History of Oracle Database The current version of Oracle Database is the result of over 30 years of innovative development. Highlights in the evolution of Oracle Database include the following: Portable version of Oracle Database Oracle Version 3, released in , was the first relational database to run on mainframes, minicomputers, and PCs. The database was written in C, enabling the database to be ported to multiple platforms. Enhancements to concurrency control, data distribution, and scalability Version 4 introduced multiversion read consistency. Objects and partitioning Oracle8 was released in as the object-relational database, supporting many new data types. Additionally, Oracle8 supported partitioning of large tables. Internet computing Oracle8i Database, released in , provided native support for internet protocols and server-side support for Java. Oracle8i was designed for internet computing, enabling the database to be deployed in a multitier environment. Grid computing Oracle Database 10g introduced grid computing in This release enabled organizations to virtualize computing resources by building a grid infrastructure based on low-cost commodity servers. A key goal was to make the database self-managing and self-tuning. Manageability, diagnosability, and availability Oracle Database 11g, released in , introduced a host of new features that enable administrators and developers to adapt quickly to changing business requirements. The key to adaptability is simplifying the information infrastructure by consolidating information and using automation wherever possible. In Oracle Database, a database schema is a collection of logical data structures, or schema objects. A database schema is owned by a database user and has the same name as the user name. Schema objects are user-created structures that directly refer to the data in the database. The database supports many types of schema objects, the most important of which are tables and indexes. A schema object is one type of database object. Some database objects, such as profiles and roles, do not reside in schemas. You define a table with a table name, such as employees, and set of columns. In general, you give each column a name, a data type , and a width when you create the table. A table is a set of rows. A column identifies an attribute of the entity described by the table, whereas a row identifies an instance of the entity. For example, attributes of the employees entity correspond to columns for employee ID and last name. A row identifies a specific employee. You can optionally specify rules for each column of a table. These rules are called integrity constraints. This constraint forces the column to contain a value in every row. Chapter 5, "Data Integrity" Indexes An index is an optional data structure that you can create on one or more columns of a table. Indexes can increase the performance of data retrieval. When processing a request, the database can use available indexes to locate the requested rows efficiently. Indexes are useful when applications often query a specific row or range of rows. Indexes are logically and physically independent of the data. Thus, you can drop and create indexes with no effect on the tables or other indexes. All applications continue to function after you drop an index. In contrast to procedural languages such as C, which describe how things should be done, SQL is nonprocedural and describes what should be done. Users specify the result that they want for example, the names of current employees , not how to derive it. All operations on the data in an Oracle database are performed using SQL statements. For example, you use SQL to create tables and query and modify data in tables. A SQL statement can be thought of as a very simple, but powerful, computer program or instruction. Query data Insert, update, and delete rows in a table Create, replace, alter, and drop objects Control access to the database and its objects Guarantee database consistency and integrity SQL unifies the preceding tasks in one consistent language. The principal benefit of server-side programming is that built-in functionality can be deployed anywhere. Oracle Database can also store program units written in Java. A Java stored procedure is a Java method published to SQL and stored in the database for general use. Chapter 8, "Server-Side Programming: Transactions An RDBMS must be able to group SQL statements so that they are either all committed, which means they are applied to the database, or all rolled back, which means they are undone. A transaction is a logical, atomic unit of work that contains one or more SQL statements. An illustration of the need for transactions is a funds transfer from a savings account to a checking account. The transfer consists of the following separate operations: Decrease the savings account. Increase the checking account. Record the transaction in the transaction journal. Oracle Database guarantees that all three operations succeed or fail as a unit. For example, if a hardware failure prevents a statement in the transaction from executing, then the other statements must be rolled back. Transactions are one of the features that sets Oracle Database apart from a file system. If you perform an atomic operation that updates several files, and if the system fails halfway through, then the files will not be consistent. In contrast, a transaction moves an Oracle database from one consistent state to another. The basic principle of a transaction is "all or nothing": Chapter 10, "Transactions" Data Concurrency A requirement of a multiuser RDBMS is the control of concurrency , which is the simultaneous access of the same data by multiple users.

Chapter 5 : Learn SQL and Database Concepts | 18+ Best Online Courses | Vertabelo

The theory of relational databases is built upon the mathematical theory of sets. In mathematics, a set is one of the most general ways of considering more than one object at once, and a substantial amount of theory has been built up to describe this. You won't need to know too much set theory to.

Basic Concepts A database is just what the name implies, a base collection of data. The data is organized in some manner so that the information contained within the database can be easily retrieved. Some of the simple databases that you might be familiar with are things like phone books or rolodexes. As data processing has become more sophisticated, so have methods for collecting, storing and retrieving information. Databases have become the cornerstone for an overwhelming amount of the computing environment in existence. Basic Database concepts for the opening section Field Column: Could be a name, or a number. In some cases, it may even be a null or empty value. A number of pieces of information that relate to the same object. If you keep records on an employee, you might have their name, address, social security number, phone number, etcâ€¦Each piece of the information relates back to one employee. If you put all the employee records together, you have a table of employees. If you were keeping the company records, you might have a table for employees, a table for customers, and another for sales records. All these tables would be combined as a database. The difference between a database and a relational database is in the way the tables are constructed. If you were keeping the company records on a series of Excel spreadsheets, you would have just a database. In a relational database, the tables are constructed so that there is a logical link between them. Relational Database versus Non-Relational Database In order to build and maintain a relational database, you use a relational database management system. This is one of a series of iterations of MS Access. Difference between Access and large commercial databases Access is a very powerful program, when used in the way that it is intended to be used. This means that it is designed for small-scale applications 50, records or less and no more than 2 or 3 concurrent users no more than 2 or 3 people trying to use the database at the same time. When these parameters are exceeded, the database may become slow and unresponsive. Access does not have very strong security or disaster recovery facilities. An Access database is allowed to be 1 Gigabyte in size and contain 32, Objects. Objects being tables, queries, reports, etcâ€¦everything is kept in a single container. This is the database file or mdb Microsoft database. All the files will be saved with the file extension mdb. Primary and Foreign Keys When we start thinking of constructing the various tables that our database requires, we need to consider how we will organize these tables and relate them to one another. For the most part, this will be done using keys. There are 2 types of keys in a table. Tables â€” Data collection objects Queries â€" Questions of your data Forms â€" Predefined format to display or enter data Reports â€" Printable version of database information The first section of this book, including the homework is meant to be an introduction to these elements. The computer you are using may be slightly different, so you may have to search through the different program groups to find MS Access After you start Access, turn to page 1.

Chapter 6 : Database Concepts and Standards

Relational Database Design 1 RELATIONAL DATABASE DESIGN Basic Concepts â€¢ a database is an collection of logically related records â€¢ a relational database stores its data in 2-dimensional.

It is the language used by relational database management systems RDBMS to access and manipulate data and to create, structure and destroy databases and database objects. Learn how relational databases are structured. Learn about some of the most popular relational databases. Learn about the major SQL statements. Codd published "A Relational Model of Data for Large Shared Data Banks," an article that outlined a model for storing and manipulating data using tables. Between and , Oracle then Relational Software, Inc. All major relational databases support this standard but each has its own proprietary extensions. Relational Databases A relational database at its simplest is a set of tables used for storing data. Each table has a unique name and may relate to one or more other tables in the database through common values. Tables A table in a database is a collection of rows and columns. Tables are also known as entities or relations. Rows A row contains data pertaining to a single item or record in a table. Rows are also known as records or tuples. Columns A column contains data representing a specific characteristic of the records in the table. Columns are also known as fields or attributes. Relationships A relationship is a link between two tables i. Relationships make it possible to find data in one table that pertains to a specific record in another table. For example, the FirstName column might be defined as varchar 20 , indicating that it can contain a string of up to 20 characters. Unfortunately, datatypes vary widely between databases. Primary Keys Most tables have a column or group of columns that can be used to identify records. For example, an Employees table might have a column called EmployeeID that is unique for every row. This makes it easy to keep track of a record over time and to associate a record with records in other tables. Foreign Keys Foreign key columns are columns that link to primary key columns in other tables, thereby creating a relationship. For example, the Customers table might have a foreign key column called SalesRep that links to EmployeeID, the primary key in the Employees table. It runs on both Unix and Windows. It has only a slightly higher market share than Oracle on Windows machines. Many people find it easier to use than Oracle. It is still very commonly used on mainframes and runs on both Windows and Unix. MySQL is available on both Windows and Unix, but it lacks some key features such as support for stored procedures. The most common DML statements are:

Chapter 7 : Table, Record, Fields etc in RDBMS Concept | Studytonight

This course introduces the world of database systems. It provides the foundation that will enable learners to master skills in data modeling and information, as well as extract information using existing database management systems. The following main topics are covered: database design/modeling.

Patrycja Dybka We did it again! Each year, we review all SQL courses and tutorials on the market to help you find the best online resources to learn SQL. Find out which SQL courses are worth a try! You can learn about SQL queries , data management , and table creation. Check out how to learn SQL. You will get to know how to use and define constraints. Finally, you will find out how to create views. Each course features dozens of interactive exercises with a detailed explanation of the theory behind the lesson, and an interactive console that verifies your solution. After completing all lessons, you can take a final quiz to test your knowledge. You can start the course without registration but you will be asked to sign up after completing the first few exercises. Their online SQL Tutorial guides you through syntax and most important statements, keywords, and functions. Short but concrete descriptions and many examples make this tutorial easy to read and understand. A distinctive feature is the ability to test queries online. At the start of the tutorial, there is a sample database which you can modify and restore back to its original content at any time. The entire tutorial is available without registration. It covers the basics of database fundamentals: The course Learn SQL online consists of 4 lessons that focus on: The left one contains a description of the exercise with a brief theoretical introduction. The panel on the right features a visual representation of the database schema with the query result. Unfortunately, both quizzes and projects are available only for users who have subscribed to a paid Pro version of Codecademy. In order to participate in the course, you need to register using an email address or a Google or Facebook account. Combined, these tools empower learners to study at their own pace inside and outside of the classroom. Querying and Managing Data. The window on the left is a video tutorial, while the window on the right shows real-time changes in the database structure and the results of queries performed under the SQLite database. Exercises are supported by documentation and useful hints. After finishing the first part of the course, learners complete a more advanced project. The whole course contains 5 parts, starting with SQL fundamentals. Khan Academy allows users to watch videos without registering; however, after registration via Facebook, Google, or email they can ask questions regarding videos and participate in discussions. This site has three major sections: Tutorials, Assessments, and Reference. A tutorial is a set of exercises that aims to acquaint you with new topics. At any time, you can consult the Reference section for background on the appropriate theory, explained with interactive examples. After completing each tutorial, you can take a quiz to show how well you understand the topic. The Assessments section offers several mini projects; each includes 15 questions of varying levels of difficulty. The course is available without registration, however, you can create an account with your email. This is recommended if you want to publish comments or contribute to the site. There are plenty of easy-to-understand examples, but no interactive exercises to practice what you have learned. All courses available at TutorialsPoint. Although Udacity is profit-oriented, some courses are available for free. For this reason, Python fundamentals are required. The course consists of video lessons, and each lesson includes a final quiz. The lessons are divided into four parts: The tutorial explains the basic concepts and constructs of SQL and provides examples at various levels of expertise. Unlike video courses, all tutorials and articles are completely free. Based around specific projects, they include step-by-step written instructions and screenshots to help you practice and master your skills. In fact, this is more of an article than tutorial. It consists of two parts: In the second article, users learn about indexes, data types, and some rather more complex query structures. Both articles are available without registration. The course is based on Microsoft SQL Server, which is why the author provides newbie users with a guide on how to get started using this database engine. The tutorial presents free tools to get you started, a guide leading you through the setup process, step-by-step explanations of how to download and activate a sample database, and much more. An especially great thing with Essential SQL is that Kris is personally involved in the teaching process. When there are problems or doubts, he strongly encourages readers to contact him. A significant drawback of this course is its lack of a structured learning program. Most of the articles are available without registration; however, some content requires users to sign up with an email account. You can learn the basics of creating and manipulating tables, get familiar with data migration, and start understanding data manipulation and transactions. To keep the book simple, the site uses SQLite3 to teach you how to design data and work with it. The entire contents of the website is available without registration. Udemy â€" SQL Tutorials Udemy is a leading global platform where experts create courses and offer them to the public, either at no charge or for a tuition fee. There are several free SQL courses for both beginners and advanced users. Here are some free courses worth checking into:

Chapter 8 : SQL Introduction

RDBMS stands for Relational Database Management System. RDBMS is the basis for SQL, and for all modern database systems like MS SQL Server, IBM DB2, Oracle, MySQL, and Microsoft Access. A Relational database management system (RDBMS) is a database management system (DBMS) that is based on the.

This tutorial may be reproduced without any modifications to this document for non-commercial, non-profit purposes. All other forms of usage requires the permission of the author - Ranjan Chari, email: A lucid and quick understanding of databases, RDBMS and related jargon such as triggers, replication, procedures etc. A clear understanding of how SQL works. A database is a collection of data organized in a particular way. This was based on Dr. Though SQL is the current universally adopted standard. SQL is the language used to query all databases. Understanding SQL and using it efficiently is highly imperative in designing an efficient database application. This concept was first described around by Dr. A relational database uses the concept of linked two-dimensional tables which comprise of rows and columns. A user can draw relationships between multiple tables and present the output as a table again. A user of a relational database need not understand the representation of data in order to retrieve it. Relational programming is non-procedural. Programming languages are procedural if they use programming elements such as conditional statements if-then-else, do-while etc. SQL has none of these types of statements. These software packages are used to manipulate a database. It may be a subset or a superset of the instructions provided by SQL Normalization is the process where a database is designed in a way that removes redundancies, and increases the clarity in organizing data in a database. In easy English, it means take similar stuff out of a collection of data and place them into tables. The important thing here is to know when to Normalize and when to be practical. That will come with experience. For now, read on Normalization of a database helps in modifying the design at later times and helps in being prepared if a change is required in the database design. Normalization raises the efficiency of the datatatabase in terms of management, data storage and scalability. Each Column Type is Unique. The entity should already be in the 2NF and no column entry should be dependent on any other entry value other than the key for the table. If such an entity exists, move it outside into a new table. Now if these 3NF are achieved, the database is considered normalized. The database should be in 3NF and all tables can have only one primary key. Tables cannot have multi-valued dependencies on a Primary Key. There should be no cyclic dependencies in a composite key. Well this is a highly simplified explanation for Database Normalization. One can study this process extensively though. After you go through the tutorial examples and also the example contacts database, look at the example provided in creating a normalized database near the very end of this tutorial. And then try to think how you would like to create your own database. Much of database design depends on how YOU want to keep the data. In real life situations often you may find it more convenient to store data in tables designed in a way that does fall a bit short of keeping all the NFs happy. Making your life simpler.

Chapter 9 : Introduction to Oracle Database

Basic Relational Database Concepts/Design, The SQL Language, & The mySQL Database Server A database is a software program which allows for storage and retrieval of information on a computer hard drive's file system or other device. A relational database is a database that allows for queries which.

Row Record When you are designing your database, make sure that each table in the database holds information about a specific thing, such as employees, products, or customers. By designing a database this way, you can set up a structure that eliminates redundancy and inconsistencies. For example, both the sales and accounts payable departments may look up information about customers. In a relational database, the information about customers is entered only once, in a table that both departments can access. A relational database is a set of related tables. You use primary and foreign keys to describe relationships between the information in different tables. Primary and foreign keys Primary and foreign keys define the relational structure of a database. These keys enable each row in the database tables to be identified, and define the relationships between the tables. Tables have a primary key All tables in a relational database should have a primary key. The primary key is a column, or set of columns, that allows each row in the table to be uniquely identified. No two rows in a table with a primary key can have the same primary key value. If no primary key is assigned, all the columns together become the primary key. It is good practice to keep your primary key for each table as compact as possible. Examples In a table holding information about employees, the primary key may be an ID number assigned to each employee. In the sample database, the table of sales order items has the following columns: An order number, identifying the order the item is part of A line number, identifying each item on any order A product ID, identifying the product being ordered A quantity, showing how many items were ordered A ship date, showing when the order was shipped To identify a particular item, both the order number and the line number are required. The primary key is made up of both these columns. Tables are related by foreign keys The information in one table is related to that in other tables by foreign keys. Example The sample database has one table holding employee information and one table holding department information. The department table has the following columns: This is the primary key for the table. This is called a foreign key to the department table. A foreign key references a particular row in the table containing the corresponding primary key. In this example, the employee table which contains the foreign key in the relationship is called the foreign table or referencing table. The department table which contains the referenced primary key is called the primary table or the referenced table. Other database objects A relational database holds more than a set of related tables. Among the other objects that make up a relational database are: Indexes Indexes allow quick lookup of information. Conceptually, an index in a database is like an index in a book. In a book, the index relates each indexed term to the page or pages on which that word appears. In a database, the index relates each indexed column value to the physical location at which the row of data containing the indexed value is stored. Indexes are an important design element for high performance, however their use is transparent to the user. Views Views are computed tables, or virtual tables. They look like tables to client applications, but they do not hold data. Instead, whenever they are accessed, the information in them is computed from the underlying tables. The tables that actually hold the information are sometimes called base tables to distinguish them from views. Stored procedures and triggers These are routines held in the database itself that act on the information in the database. You can create and name your own stored procedures to execute specific database queries and to perform other database tasks. Stored procedures can take parameters. For example, you might create a stored procedure that returns the names of all customers who have spent more than the amount that you specify as a parameter in the call to the procedure. A trigger is a special stored procedure that automatically fires whenever a user updates, deletes, or inserts data, depending on how you define the trigger. You associate a trigger with a table or columns within a table. Triggers are useful for automatically maintaining business rules in a database. Users and groups Each user of a database has a user ID and password. You can set permissions for each user, so that confidential information is kept private and users are prevented from making unauthorized changes. Users can be assigned to groups, in order to make the administration of permissions easier. Java objects You can install Java classes into the database. Java classes provide a powerful way of building logic into your database, and a special class of user-defined datatypes for holding information. The basic query operations in a relational system are projection, restriction, and join. A projection is a subset of the columns in a table. A restriction also called selection is a subset of the rows in a table, based on some conditions. For example, you may want to select the item identification numbers and product names for all items for which more than a dozen has been shipped: SQL includes statements that create tables, views, and other database objects. It also includes statements that modify tables and commands that perform many other database tasks discussed in this manual. The system tables Every database contains a set of system tables, which are special tables that the system uses to manage data and the system. These tables are sometimes called the data dictionary or the system catalog. System tables contain information about the database. You never alter the system tables directly in the way that you can alter other tables. The system tables hold information about the tables in a database, the users of a database, the columns in each table, and so on. This information is data about data, or metadata.