Relational Databases

Databases can be organized in many different ways, and thus take many forms. The most popular form of database today is the relational database. Popular examples of relational databases are Microsoft Access, MySQL, and Oracle. A relational database is one in which data is organized into one or more tables. Each table has a set of fields, which define the nature of the data stored in the table. A record is one instance of a set of fields in a table. To visualize this, think of the records as the rows of the table and the fields as the columns of the table. In the example below, we have a table of student information, with each row representing a student and each column representing one piece of information about the student.

Fields (Columns)

Records (Rows)

First Name	Last Name	Major	Birthdate
Ann Marie	Cortez	Political Science/Pre-Law	02/01/1993
Evan	Nguyen	Business	3/12/1994
Michelle	Smith	Computer Science	5/10/1992

In a relational database, all the tables are related by one or more fields, so that it is possible to connect all the tables in the database through the field(s) they have in common. For each table, one of the fields is identified as a primary key. This key is the unique identifier for each record in the table. To help you understand these terms further, let's walk through the process of designing a database.

Designing a Database

Suppose a university wants to create an information system to track participation in student clubs. After interviewing several people, the design team learns that the goal of implementing the system is to give better insight into how the university funds clubs. This will be accomplished by tracking how many members each club has and how active the clubs are. From this, the team decides that the system must keep track of the clubs, their members, and their events. Using this information, the design team determines that the following tables need to be created:

- Clubs: this will track the club name, the club president, and a short description of the club.
- Students: student name, e-mail, and year of birth.
- Memberships: this table will correlate students with clubs, allowing us to have any given student join multiple clubs.
- Events: this table will track when the clubs meet and how many students showed up.

Now that the design team has determined which tables to create, they need to define the specific information that each table will hold. This requires identifying the fields that will be in each table. For example, Club Name would be one of the fields in the Clubs table. First Name and Last Name would be fields in the Students table. Finally, since this will be a relational database, every table should have a field in common with at least one other table (in other words: they should have a relationship with each other).

In order to properly create this relationship, a primary key must be selected for each table. This key is a unique identifier for each record in the table. For example, in the Students table, it might be possible to use students' last name as a way to uniquely identify them. However, it is more than likely that some students will share a last name (like Rodriguez, Smith, or Lee), so a different field should be selected. A student's e- mail address might be a good choice for a primary key, since e-mail addresses are unique. However, a primary key cannot change, so this would mean that if students changed their e-mail address we would have to remove them from the database and then re-insert them – not an attractive proposition. Our solution is to create a value for each student — a user ID — that will act as a primary key. We will also do this for each of the student clubs. This solution is quite common and is the reason you have so many user IDs!

You can see the final database design in the figure below:

With this design, not only do we have a way to organize all of the information we need to meet the requirements, but we have also successfully related all the tables together. Here's what the database tables might look like with some sample data. Note that the Memberships table has the sole purpose of allowing us to relate multiple students to multiple clubs.

Club ID	Club name	President	Short desc
1	Cheese Club	14	To talk about our love of cheese.
2	Chess Club	1	To learn how to become better chess players.
3	Archery Club	6	To compete in archery tournaments.

Table: Clubs

Club ID	Event name	Date	Attendance
1	Cheese promo	1/10/2013	6
2	MLK Tournament	1/21/2013	17
3	January meeting	1/22/2013	12
2	January meeting	1/28/2013	10

Table: Events

ID	First Name	Last Name	Email	Year of Birth
1	Peter	Lee	plee@university.edu	1992
2	Jonathan	Edwards	jedwards@university.edu	1994
3	Marilyn	Johnson	mjohnson@university.edu	1993
6	Joe	Kim	jkim@university.edu	1992
12	Haley	Martinez	hmartinez@university.edu	1993
14	John	Mfume	jmfume@university.edu	1991
15	David	Letty	dletty@university.edu	1995

Table: Students

Club ID	Student ID
1	1
1	2
1	14
2	1
2	3
2	5
2	6
3	1
3	6
3	12
3	14
3	15

Table: Memberships

Normalization

When designing a database, one important concept to understand is *normalization*. In simple terms, to normalize a database means to design it in a way that: 1) reduces duplication of data between tables and 2) gives the table as much flexibility as possible.

In the Student Clubs database design, the design team worked to achieve these objectives. For example, to track memberships, a simple solution might have been to create a Members field in the Clubs table and then just list the names of all of the members there. However, this design would mean that if a student joined two clubs, then his or her information would have to be entered a second time. Instead, the designers solved this problem by using two tables: Students and Memberships.

In this design, when a student joins their first club, we first must add the student to the Students table, where their first name, last name, e-mail address, and birth year are entered. This addition to the Students table will generate a student ID. Now we will add a new entry to denote that the student is a member of a specific club. This is accomplished by adding a record with the student ID and the club ID in the Memberships table. If this student joins a second club, we do not have to duplicate the entry of the student's name, e-mail, and birth year; instead, we only need to make another entry in the Memberships table of the
Second club's ID and the student's ID.

The design of the Student Clubs database also makes it simple to change the design without major modifications to the existing structure. For example, if the design team were asked to add functionality to the system to track faculty advisors to the clubs, we could easily accomplish this by adding a Faculty Advisors table (similar to the Students table) and then adding a new field to the Clubs table to hold the Faculty Advisor ID.

Data Types

When defining the fields in a database table, we must give each field a data type. For example, the field Birth Year is a year, so it will be a number, while First Name will be text. Most modern databases allow for several different data types to be stored. Some of the more common data types are listed here:

- Text: for storing non-numeric data that is brief, generally under 256 characters. The database designer can identify the maximum length of the text.
- Number: for storing numbers. There are usually a few different number types that can be selected, depending on how large the largest number will be.
- Yes/No: a special form of the number data type that is (usually) one byte long, with a 0 for "No" or "False" and a 1 for "Yes" or "True".
- Date/Time: a special form of the number data type that can be interpreted as a number or a time.
- Currency: a special form of the number data type that formats all values with a currency indicator and two decimal places.
- Paragraph Text: this data type allows for text longer than 256 characters.
- Object: this data type allows for the storage of data that cannot be entered via keyboard, such as an image or a music file.

There are two important reasons that we must properly define the data type of a field. First, a data type tells the database what functions can be performed with the data. For example, if we wish to perform mathematical functions with one of the fields, we must be sure to tell the database that the field is a number data type. So if we have, say, a field storing birth year, we can subtract the number stored in that field from the current year to get age.

The second important reason to define data type is so that the proper amount of storage space is allocated for our data. For example, if the First Name field is defined as a text (50) data type, this means fifty characters are allocated for each first name we want to store. However, even if the first name is only five characters long, fifty characters (bytes) will be allocated. While this may not seem like a big deal, if our table ends up holding 50,000 names, we are allocating 50 * 50,000 = 2,500,000 bytes for storage of these values. It may be prudent to reduce the size of the field so we do not waste storage space.

Enterprise Databases

A database that can only be used by a single user at a time is not going to meet the needs of most organizations. As computers have become networked and are now joined worldwide via the Internet, a class of database has emerged that can be accessed by two, ten, or even a million people. These databases are sometimes installed on a single computer to be accessed by a group of people at a single location. Other times, they are installed over several servers worldwide, meant to be accessed by millions. These relational enterprise database packages are built and supported by companies such as Oracle, Microsoft, and IBM. The open-source MySQL is also an enterprise database.

As stated earlier, the relational database model does not scale well. The term *scale* here refers to a database getting larger and larger, being distributed on a larger number of computers connected via a network. Some companies are looking to provide large-scale database solutions by moving away from the relational model to other, more flexible models. For example, Google now offers the App Engine Datastore, which is based on

NoSQL. Developers can use the App Engine Datastore to develop applications that access data from anywhere in the world. Amazon.com offers several database services for enterprise use, including Amazon RDS, which is a relational database service, and Amazon DynamoDB, a NoSQL enterprise solution.

Big Data

A new buzzword that has been capturing the attention of businesses lately is *big data*. The term refers to such massively large data sets that conventional database tools do not have the processing power to analyze them. For example, Walmart must process over one million customer transactions every hour. Storing and analyzing that much data is beyond the power of traditional database-management tools. Understanding the best tools and techniques to manage and analyze these large data sets is a problem that governments and businesses alike are trying to solve.

What Is Metadata?

The term *metadata* can be understood as "data about data." For example, when looking at one of the values of Year of Birth in the Students table, the data itself may be "1992″. The metadata about that value would be the field name Year of Birth, the time it was last updated, and the data type (integer). Another example of metadata could be for an MP3 music file, like the one shown in the image below; information such as the length of the song, the artist, the album, the file size, and even the album cover art, are classified as metadata. When a database is being designed, a "data dictionary" is created to hold the metadata, defining the fields and structure of the database.

Data Warehouse

As organizations have begun to utilize databases as the centerpiece of their operations, the need to fully understand and leverage the data they are collecting has become more and more apparent. However, directly analyzing the data that is needed for day-to-day operations is not a good idea; we do not want to tax the operations of the company more than we need to. Further, organizations also want to analyze data in a historical sense: How does the data we have today compare with the same set of data this time last month, or last year? From these needs arose the concept of the data warehouse.

The concept of the data warehouse is simple: extract data from one or more of the organization's databases and load it into the data warehouse (which is itself another database) for storage and analysis. However, the execution of this concept is not that simple. A data warehouse should be designed so that it meets the following criteria:

- It uses non-operational data. This means that the data warehouse is using a copy of data from the active databases that the company uses in its day-to-day operations, so the data warehouse must pull data from the existing databases on a regular, scheduled basis.
- The data is time-variant. This means that whenever data is loaded into the data warehouse, it receives a time stamp, which allows for comparisons between different time periods.
- The data is standardized. Because the data in a data warehouse usually comes from several different sources, it is possible that the data does not use the same definitions or units. For example, our Events table in our Student Clubs database lists the event dates using the mm/dd/ yyyy format (e.g., 01/10/2013). A table in another database might use the format yy/mm/dd (e.g., 13/01/10) for dates. In order for the data warehouse to match up dates, a standard date format would have to be agreed upon and all data loaded into

the data warehouse would have to be converted to use this standard format. This process is called extraction-transformation-load (ETL).

There are two primary schools of thought when designing a data warehouse: bottom-up and top-down. The bottom-up approach starts by creating small data warehouses, called data marts, to solve specific business problems. As these data marts are created, they can be combined into a larger data warehouse. The top-down approach suggests that we should start by creating an enterprise-wide data warehouse and then, as specific business needs are identified, create smaller data marts from the data warehouse.

Benefits of Data Warehouses

Organizations find data warehouses quite beneficial for a number of reasons:

- The process of developing a data warehouse forces an organization to better understand the data that it is currently collecting and, equally important, what data is not being collected.
- A data warehouse provides a centralized view of all data being collected across the enterprise and provides a means for determining data that is inconsistent.
- Once all data is identified as consistent, an organization can generate one version of the truth. This is important when the company wants to report consistent statistics about itself, such as revenue or number of employees.
- By having a data warehouse, snapshots of data can be taken over time. This creates a historical record of data, which allows for an analysis of trends.
- A data warehouse provides tools to combine data, which can provide new information and analysis.

Data Mining

Data mining is the process of analyzing data to find previously unknown trends, patterns, and associations in order to make decisions. Generally, data mining is accomplished through automated means against extremely large data sets, such as a data warehouse. Some examples of data mining include:

- An analysis of sales from a large grocery chain might determine that milk is purchased more frequently the day after it rains in cities with a population of less than 50,000.
- A bank may find that loan applicants whose bank accounts show particular deposit and Withdrawal patterns are not good credit risks.
- A baseball team may find that collegiate baseball players with specific statistics in hitting, pitching, and fielding make for more successful major league players.

In some cases, a data-mining project is begun with a hypothetical result in mind. For example, a grocery chain may already have some idea that buying patterns change after it rains and want to get a deeper understanding of exactly what is happening. In other cases, there are no presuppositions and a data-mining program is run against large data sets in order to find patterns and associations.

Business Intelligence and Business Analytics

With tools such as data warehousing and data mining at their disposal, businesses are learning how to use information to their advantage. The term *business intelligence* is used to describe the process that organizations use to take data they are collecting and analyze it in the hopes of obtaining a competitive advantage. Besides using data from their internal databases, firms often purchase information from data brokers to get a big-picture understanding of their industries. *Business analytics* is the term used to describe the use of internal company data to improve business processes and practices.

Business Intelligence

Sporadic usage of the term *business intelligence* dates back to at least the 1860s, but consultant Howard Dresner is credited with first proposing it in 1989 as an umbrella category for applying data analysis techniques to support business decision-making processes. What came to be known as BI technologies evolved from earlier, often mainframe-based analytical systems, such as decision support systems and executive information systems? Business intelligence is sometimes used interchangeably with business analytics; in other cases, business analytics is used either more narrowly to refer to advanced data analytics or more broadly to include both BI and advanced analytics.
Business intelligence (BI) is a technology-driven process for analyzing data and presenting actionable information to help corporate executives, business managers and other end users make more informed business decisions. BI encompasses a wide variety of tools, applications and methodologies that enable organizations to collect data from internal systems and external sources, prepare it for analysis, develop and run queries against the data, and create reports, dashboards and data visualizations to make the analytical results available to corporate decision makers as well as operational workers.

The potential benefits of business intelligence programs include accelerating and improving decision making; optimizing internal business processes; increasing operational efficiency; driving new revenues; and gaining competitive advantages over business rivals. BI systems can also help companies identify market trends and spot business problems that need to be addressed.

BI data can include historical information, as well as new data gathered from source systems as it is generated, enabling BI analysis to support both strategic and tactical decision-making processes. Initially, BI tools were primarily used by data analysts and other IT professionals who ran analyses and produced reports with query results for business users. Increasingly, however, business executives and workers are using BI software themselves, thanks partly to the development of self-service BI and data discovery tools.

Business intelligence data typically is stored in a data warehouse or smaller data marts that hold subsets of a company's information.

In addition to BI managers, business intelligence teams generally include a mix of BI architects, BI developers, business analysts and data management professionals; business users often are also included to represent the business side and make sure its needs are met in the BI development process.

Business Analytics

Business analytics (BA) is the practice of iterative, methodical exploration of an organization’s data with emphasis on statistical analysis. Business analytics is used by companies committed to data-driven decision making.
BA is used to gain insights that inform business decisions and can be used to automate and optimize business processes. Data-driven companies treat their data as a corporate

asset and leverage it for competitive advantage. Successful business analytics depends on data quality, skilled analysts who understand the technologies and the business and an organizational commitment to data-driven decision making.

Examples of BA uses include:

- Exploring data to find new patterns and relationships (data mining)
- Explaining why a certain result occurred (statistical analysis, quantitative analysis)
- Experimenting to test previous decisions (A/B testing, multivariate testing)
- Forecasting future results (predictive modeling, predictive analytics)

Once the business goal of the analysis is determined, an analysis methodology is selected and data is acquired to support the analysis. Data acquisition often involves extraction from one or more business systems, cleansing, and integration into a single repository such as a data warehouse or data mart. The analysis is typically performed against a smaller sample set of data. Analytic tools range from spreadsheets with statistical functions to complex data mining and predictive modeling applications. As patterns and relationships in the data are uncovered, new questions are asked and the analytic process iterates until the business goal is met. Deployment of predictive models involves scoring data records (typically in a database) and using the scores to optimize real-time decisions within applications and business processes. BA also supports tactical decision making in response to unforeseen events, and in many cases the decision making is automated to support real-time responses.

While the terms business intelligence and business analytics are often used interchangeably, there are some key differences:

BI vs. BA	**Business Intelligence**	**Business Analytics**
Answers the questions:	What happened? When? Who? How many?	Why did it happen? Will it happen again? What will happen if we change *x*? What else does the data tell us that never thought to ask?
Includes:	Reporting (KPIs, metrics) Automated Monitoring/Alerting (thresholds) Dashboards Scorecards OLAP (Cubes, Slice & Dice, Drilling) Ad hoc query	Statistical/Quantitative Analysis Data Mining Predictive Modeling Multivariate Testing

Why Learn About Information Systems in Organizations?

Organizations of all types use information systems to cut costs and increase profits. After graduating, a management major might be hired by a shipping company to help design a computerized system to improve employee productivity. Marketing major might work for a national retailer using a network to analyze customer needs in different areas of the country. An accounting major might work for an accounting or consulting firm using a computer to audit other companies' financial records. A real estate major might use the Internet and work in a loose organizational structure with clients, builders, and a legal team located around the world. A biochemist might conduct research for a drug company and use a computer to evaluate the potential of a new cancer treatment. An entrepreneur might use information systems to advertise and sell products and bill customers.
Although your career might be different from your classmates', you will almost certainly work with computers and information systems to help your company or organization become more efficient, effective, productive, and competitive in its industry. In this chapter, you will see how information systems can help organizations produce higher-quality products and services to increase their return on investment. We begin by investigating organizations and information systems.

Organization

A formal collection of people and other resources established to accomplish a set of goals.

The primary goal of a for-profit organization is to maximize shareholder value, often measured by the price of the company stock. Nonprofit organizations include social groups, religious groups, universities, and other organizations that do not have profit as their goal.
An organization is a system (see fig.1), which means that it has inputs, processing mechanisms, outputs, and feedback. An organization constantly uses money, people, materials, machines and other equipment, data, information, and decisions. As shown in figure below, resources such as materials, people, and money serve as inputs to the organizational system from the environment, go through a transformation mechanism, and then are produced as outputs to the environment. The outputs from the transformation mechanism are usually goods or services, which are of higher relative value than the inputs alone. Through adding value or worth, organizations attempt to achieve their goals.

Organizational Structures

Organizational structure refers to organizational subunits and the way they relate to the overall organization. An organization's structure depends on its goals and approach to management, and can affect how it views and uses information systems. The types of organizational structures typically include traditional, project, team, and virt**ua**l. Organizational structure can have a direct impact on the organization's information system.
Organization structure is a setup or a framework which determines the hierarchy of people, its function, workflow, and the reporting system in an organization.

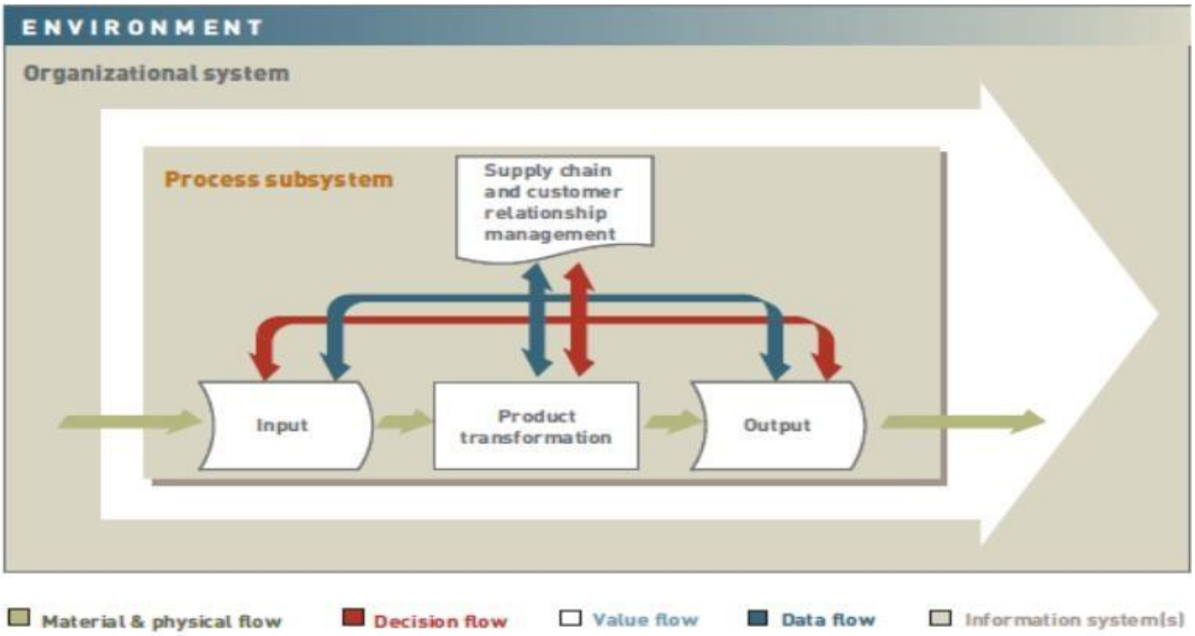

Fig (1) Organization system

Why an Organization Structure is needed?

Every organization must have a well-defined organization structure if it wants to work efficiently and achieve its goal with fewer hurdles.

Nowadays, organizations have to be very competitive, productive, efficient and dynamic if they want to survive and grow. Clients and consumers are becoming more quality conscious and demanding and they need a quick response from organizations.

Therefore to fulfill customer requirements, market demand and for self-survival, it is very necessary for any organization to define a structure for itself which is best suited to its requirements and supports its objective.

Let’s say your organization is producing a product whose specifications rarely change. Your customers are satisfied and not more demanding. In this case, what kind of organization structure will you select?

Of course you will select the functional organization structure because the functional structure is good for an organization involved in production operation.

Okay, let’s discuss another case.

Suppose your organization consists of 10 people, which is involved with small projects. Once you complete a project, you start hunting for another project. In this case, what kind of structure will you select?

In this case you will select the projectized organization structure, because here you are dealing with only projects.

Now, let’s see the final case.

Your organization is very big, spread out over a large geographical area and it works in a dynamic environment where customer requirements and market demand is constantly changing. In this case, what kind of structure will you select?

In this case you cannot select the functional organization because the functional organization structure does not allow quick changes to its operations.

You can also not select the projectized structure because, although this structure supports a dynamic nature, it cannot account for other things because your organization is too big and you need some permanent functional units with supportive staff to keep your organization functioning smoothly, and this is not possible with a pure projectized organization.

Therefore, you will select a matrix organization structure. This structure has qualities of the functional organization structure as well as the projectized organization. So, you will select this structure because only this structure can support your operations and help you achieve your organization's objective.

Therefore, you can say that an organization needs an organization structure if it wants to survive, and if an organization does not have a proper structure to support its objective, it will not succeed.