Republic of Iraq
The Ministry of Higher Education
& Scientific Research
Al-Ma'arif University College
Department of Computer Science

Lecturer Name: Shihab Hamad Khaleefah
Academic Status: BhD. In Computer Science
Qualification: - Lecturer
Course Material: Database Management System
Grade Level: Second Stage, Lec. 1-3

OVERVIEW TO THE MAIN CONCEPT IN DATABASE DESIGN & MANAGEMENT

1. Introduction to Database Design and Management
   1.1 History of Database
2. Data Types and Information
   2.1 Information Vs. Databases Types
   2.2 Basic Assumptions of Database
   2.3 Structured Vs. Unstructured Data
   2.4 The Problems with Unstructured Data
   2.5 The Main Terms in Database:
      2.5.1 Columns - Fields & Rows – Records & Cells.
3. Data Models of Database
   3.1 The Relationships in Database:
      3.1.1 One to One
      3.1.2 One to Many
      3.1.3 Many to One
      3.1.4 Many to Many
4. The Relation Table Criteria
5. The Three Levels of the Database
   5.1 Notional Approach
   5.2 Logical Approach
   5.3 Physical Approach
6. The Three Factors of the Data Model in DB
7. Summary

Introduction to Database Design and Management

When we are talking about computers, we instantly think about the speed that allows us to execute complex operations. This speed is mainly used during searches. This means the extraction of a datum from an archive or database. This requires a database, which stores the data. The operation system 'Windows XP, seven, 8, 8.1 and 10' lacks the programme required for such issues. However, there are many database handlers on the market such as MySQL, SQL Server, MS Access, Oracle, Sybase, Informix, Postgres, and other database systems. I will introduce the usage and way of acquisition with SQL which is one of the most important commercial languages it is a standard language for storing, manipulating and retrieving data in databases. Despite the existence of such standards, most SQL code is completely portable among different database systems without adjustments.

The aim of lectures notes to understand what we are saying, and not just learn it, we need to clarify the rudiments. The most precision and attention is required by the construction and planning of a database. Choosing the most suitable model, shaping the structure of our database is the basis of our work. It is easy to build on good foundations, that is why the first part - in which we will clarify the rudiments of database administration - has a key importance.

In the second part we will discuss the first steps of database design and management. The structure and usage of the SQL itself is not so complicated. Adding and searching for information is not a challenge either, all this add up to the fact, that it is the most widely used database handler today. This also played a part in my choice of database handlers.

The third and biggest part of the lectures will be dedicated to the possibilities that are offered by the SQL. This field is concerned with the all the possible objects that can be created with the help of SQL, such as reports, forms, queries, etc.

The last section of the essay contains a collection of exercises - designed and management solely for the practice of the acquired knowledge – in connection with database administration. I will illustrate the usage of the SQL mainly through practical examples and images to model the process. Since the chapters build on each other, it is important that no part should be omitted, otherwise information gaps may emerge. The course material is also fit for a possible revision.

Data Types and Information

Information is not the same as data, but rather a kind of meaning of the data. Data is, in contrast to information, objective. Database administration is used for storing facts in databases, and to present information in such form that carry information for the user. Therefore, data is understood here as a series of signs that become information during the processing of the data.

The Data is a coherent mass of facts, which includes all the date that are required for the realization of a given goal.

"The datum is a piece of knowledge that can be interpreted (it is perceptible, sensible, comprehensible, and understandable)." (Halassy 1994, p. 8)

Is the main content in a database and its which we want to process it to get the information, So the data is anything in a form suitable for use with a computer. Data is often distinguished from programs. A program is a set of instructions that detail a task for the computer to perform. In this sense, data is thus everything that is not program code.
In one view, databases can be classified according to types of content: bibliographic, full-text, numeric, and images.

The Information is a datum that becomes a newly interpreted knowledge." (Halassy 1994, p. 9) . Is collection of data (after process it) and we will use it in a program. So the information is same data stored inside Database (But After Process it).

The database is a collection of data, which stores the data required for a given task in an organised way, grants access to them, and at the same time safeguards the integrity of the units, and protects them from any harm.

*** Is a set of data which, order, regular, related with another some in logical or relational view by one of the mathematics relationship (why?), to use these data in another time and process it to be clear information.so the database is a collection of information that is organized so that it can easily be accessed, managed, and updated.

Base Concept

"We call the thing-that-something that we want to describe with the help of our knowledge an entity." (Halassy 1994, 24)

The specific entities are known as **entity occurrences**.

"We call the thing-that-something that we use to describe the phenomena, which we are interested in a property." (Halassy 1994, 28)

The concrete value of a property is the **property value**. The property value set is the term used for all the values present in a specific time.

Both entity and property are relative terms. A thing can be both an entity and a property at the same time. It is up to us to decide; which properties should be treated as individual entities.

Structured vs. Unstructured Data

Flat file = unstructured data

Database = structured data

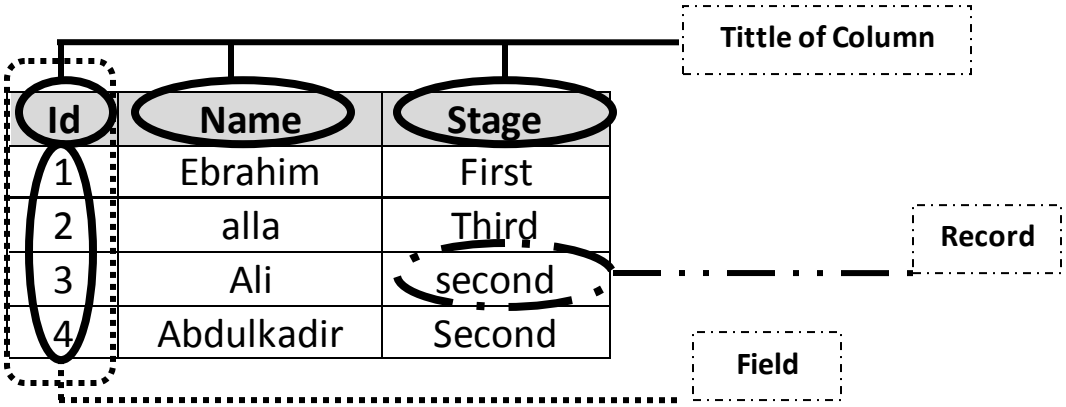
The Problems with Unstructured Data:

- High maintenance costs
- Data Redundancy: the same data will be represented multiple times in the file.
- Data dependence: if you change things about the file format then there will be expensive changes to programs that use it.
- Ensuring data consistency and controlling access to the data is difficult (i.e.) you cannot finely control multi-user access to the file.

```
>Q4U9M9|104K_THEAN 104 kDa microneme/rhoptry antigen precursor - Theileria annulata
MKFLVLLFNILCLFPILGADELVMSPIPTTDVQPKVTFDINSEVSSGPLYLNPVEMAGVK
YLQLQRQPGVQVHKVVEGDIVIWENEEMPLYTCAIVTQNEVPYMAYVELLEDPDLIFFLK
EGDQWAPIPEDQYLARLQQLRQQIHTESFFSLNLSFQHENYKYEMVSSFQHSIKMVVFTP
KNGHICKMVYDKNIRIFKALYNEYVTSVIGFFRGLKLLLLNIFVIDDRGMIGNKYFQLLD
DKYAPISVQGYVATIPKLKDFAEPYHPIILDISDIDYVNFYLGDATYHDPGFKIVPKIPQ
CITKVVDGNEVIYESSNPSVECVYKVTYYDKKNESMLRLDLNHSPPSYTSYYAKREGVWV
TSTYIDLEEKIEELQDHRSTELDVMFMSDKDLNVVPLTNGNLEYFMVTPKPHRDIIIVFD
GSEVLWYYEGLENHLVCTWIYVTEGAPRLVHLRVKDRIPQNTDIYMVKFGEYWVRISKTQ
YTQEIKKLIKKSKKKLPSIEEEDSDKHGGPPKGPEPPTGPGHSSSESKEHEDSKESKEPK
EHGSPKETKEGEVTKKPGPAKEHKPSKIPVYTKRPEFPKKSKSPKRPESPKSPKRPVSPQ
RPVSPKSPKRPESLDIPKSPKRPESPKSPKRPVSPQRPVSPRRPESPKSPKSPKSPKSPK
VPFDPKFKEKLYDSYLDKAAKTKETVTLPPVLPTDESFTHTPIGEPTAEQPDDIEPIEES
VFIKETGILTEEVKTEDIHSETGEPEEPKRPDSPTKHSPKPTGTHPSMPKKRRRSDGLAL
STTDLESEAGRILRDPTGKIVTMKRSKSFDDLTTVREKEHMGAEIRKIVVDDDGTEADDE
DTHPSKEKHLSTVRRRRPRPKKSSKSSKPRKPDSAFVPSIIFIFLVSLIVGIL
>P15711|104K_THEPA 104 kDa microneme/rhoptry antigen precursor - Theileria parva
MKFLILLFNILCLFPVLAADNHGVGPQGASGVDPITFDINSNQTGPAFLTAVEMAGVKYL
QVQHGSNVNIHRLVEGNVVIWENASTPLYTGAIVTNNDGPYMAYVEVLGDPNLQFFIKSG
DAWVTLSEHEYLAKLQEIRQAVHIESVFSLNMAFQLENNKYEVETHAKNGANMVTFIPRN
GHICKMVYHKNVRIYKATGNDTVTSVVGFFRGLRLLLINVFSIDDNGMMSNRYFQHVDDK
YVPISQKNYETGIVKLKDYKHAYHPVDLDIKDIDYTMFHLADATYHEPCFKIIPNTGFCI
TKLFDGDQVLYESFNPLIHCINEVHIYDRNNGSIICLHLNYSPPSYKAYLVLKDTGWEAT
THPLLEEKIEELQDQRACELDVNFISDKDLYVAALTNADLNYTMVTPRPHRDVIRVSDGS
EVLWYYEGLDNFLVCAWIYVSDGVASLVHLRIKDRIPANNDIYVLKGDLYWTRITKIQFT
QEIKRLVKKSKKKLAPITEEDSDKHDEPPEGPGASGLPPKAPGDKEGSEGHKGPSKGSDS
SKEGKKPGSGKKPGPAREHKPSKIPTLSKKPSGPKDPKHPRDPKEPRKSKSPRTASPTRR
PSPKLPQLSKLPKSTSPRSPPPPTRPSSPERPEGTKIIKTSKPPSPKPPFDPSFKEKFYD
DYSKAASRSKETKTTVVLDESFESILKETLPETPGTPFTTPRPVPPKRPRTPESPFEPPK
DPDSPSTSPSEFFTPPESKRTRFHETPADTPLPDVTAELFKEPDVTAETKSPDEAMKRPR
SPSEYEDTSPGDYPSLPMKRHRLERLRLTTTEMETDPGRMAKDASGKPVKLKRSKSFDDL
TTVELAPEPKASRIVVDDEGTEADDEETHPPEERQKTEVRRRRPPKKPSKSPRPSKPKKP
KKPDSAYIPSILAILVVSLIVGIL
>Q43495|108_SOLLC Protein 108 precursor - Solanum lycopersicum (Tomato) (Lycopersicon esculentum)
MASVKSSSSSSSSSFISLLLLILLVIVLQSQVIECQPQQSCTASLTGLNVCAPFLVPGSP
TASTECCNAVQSINHDCMCNTMRIAAQIPAQCNLPPLSCSAN
>P18646|10KD_VIGUN 10 kDa protein precursor - Vigna unguiculata (Cowpea)
MEKKSIAGLCFLFLVLFVAQEVVVQSEAKTCENLVDTYRGPCFTTGSCDDHCKNKEHLLS
GRCRDDVRCWCTRNC
>P13813|110KD_PLAKN 110 kDa antigen - Plasmodium knowlesi
FNSNMLRGSVCEEDVSLMTSIDNMIEEIDFYEKEIYKGSHSGGVIKGMDYDLEDDENDED
EMTEQMVEEVADHITQDMIDEVAHHVLDNITHDMAHMEEIVHGLSGDVTQIKEIVQKVNV
AVEKVKHIVETEETQKTVEPEQIEETQNTVEPEQTEETQKTVEPEQTEETQNTVEPEQIE
ETQKTVEPEQTEEAQKTVEPEQTEETQKTVEPEQTEETQKTVEPEQTEETQKTVEPEQTE
ETQKTVEPEQTEETQKTVEPEQTEETQKTVEPEQTEETQNTVEPEPTQETQNTVEP
>Q9XHP0|11S2_SESIN 11S globulin seed storage protein 2 precursor - Sesamum indicum (Oriental sesame) (Gingelly)
MVAFKFLLALSLSLLVSAAIAQTREPRLTQGQQCRFQRISGAQPSLRIQSEGGTTELWDE
RQEQFQCAGIVAMRSTIRPNGLSLPNYHPSPRLVYIERGQGLISIMVPGCAETYQVHRSQ
RTMERTEASEQQDRGSVRDLHQKVHRLRQGDIVAIPSGAAHWCYNDGSEDLVAVSINDVN
HLSNQLDQKFRAFYLAGGVPRSGEQEQQARQTFHNIFRAFDAELLSEAFNVPQETIRRMQ
SEEEERGLIVMARERMTFVRPDEEEGEQEHRGRQLDNGLEETFCTMKFRTNVESRREADI
FSRQAGRVHVVDRNKLPILKYMDLSAEKGNLYSNALVSPDWSMTGHTIVYVTRGDAQVQV
VDHNGQALMNDRVNQGEMFVVPQYYTSTARAGNNGFEWVAFKTTGSPMRSPLAGYTSVIR
AMPLQVITNSYQISPNQAQALKMNRGSQSFLLSPGGRRS
>P19084|11S3_HELAN 11S globulin seed storage protein G3 precursor - Helianthus annuus (Common sunflower)
MASKATLLLAFTLLFATCIARHQQRQQQQNQCQLQNIEALEPIEVIQAEAGVTEIWDAYD
QQFQCAWSILFDTGFNLVAFSCLPTSTPLFWPSSREGVILPGCRRTYEYSQEQQFSGEGG
RRGGGEGTFRTVIRKLENLKEGDVVAIPTGTAHWLHNDGNTELVVVFLDTQNHENQLDEN
```

Columns - Fields & Rows – Records & Cells

A column: is collection of cells aligned vertically in a table. Usually, a column in a table contains the values of a single field and all information associated with a specific field in a table. The first cell in any column refer to the title of next data. A field is an element in which one piece of information or data is stored. The field is every cell after the title of column (first Cell) aligned vertically.



A Row or record: A single entry in a table is called a Record or Row. A Record in a table represents set of related data Each row, which represents a complete record of specific item data, holds different data within the same structure. A row is occasionally referred to as a tuple.

A cell: every place in a table contains on a single data (vertical or horizontal).

DATA MODELS of DATABASE

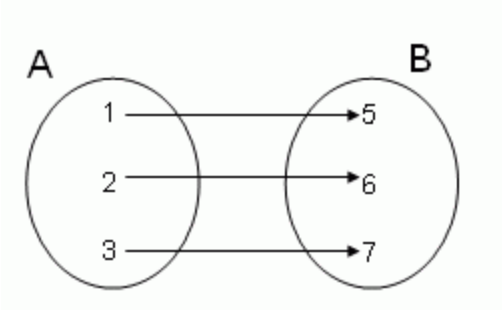
It happens in many cases that an individual data file, or a table does not have enough data to identify certain information. In such cases it may become necessary to handle the data files as a whole, according to a database structure, also known as data model. There are many data models existing such as Relational data model, Hierarchic data model, Network data model and Object-oriented data model. However, we will focus on the Relational data model which is widely known in database design and management.

The relationships in database

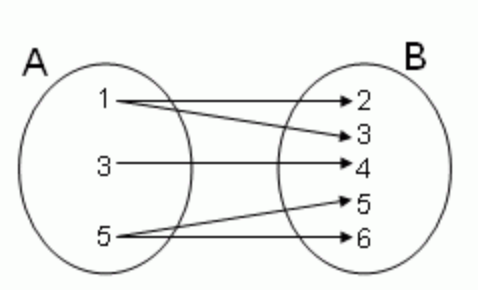
There are four relationships in database:

- One to One
- One to Many
- Many to One
- Many to Many

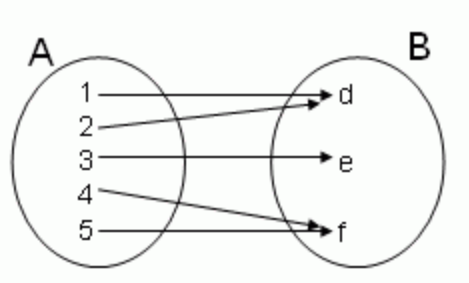
***One to One*:** One entity is associated with another entity. For Ex: Each employee is associated with one department. in this relation an item of table A has exactly one item from table B that it has a connection with. This relation is quite rare, simply because two entities can be easily merged. It is generally used to fix temporary problems. Also known as mutual relation.



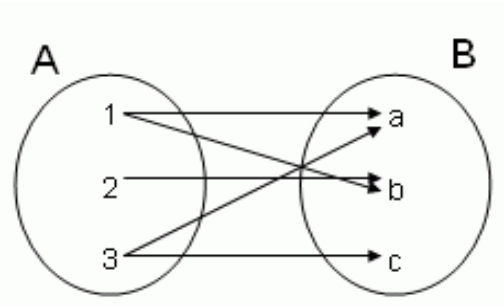
One to Many: One entity is associated with many other entities. For Ex: A company is associated with all working employees in one branch/office/country. if to an entity's 1 occurrence the other entity's more than one (N) occurrence can be connected then we call these type of relationships 1:N or 1 to more relation. We call the entity with a linking property that has an identifier role a superior, and the one with a switcher role an inferior. We also call this a hierarchic or inhomogeneous relation.



Many to One: Many entities are associated with only one entity. For Ex: Many employees are associated with one project.



Many to Many: Many entities are associated with many other entities. For Ex: In a company many employees are associated with multiple projects(completed/existing), and at the same time, projects are associated with multiple employees. when two entities have a relation in which, to entity A's 1 occurrence there are more than one entity connected from entity B's 1 occurrence then we refer to this situation as N:M, or more to more relation. We also refer to these situations as network relations. If there is an existing N:M relation between two entities we can dissolve it by introducing a third entity, which will lead to two 1: N relations.



The relational data model should examine in more detail, So we can call those programs that store, organize, and search for data in tables (relations) on the computer relational database managers.

A relational database is a type of database which consists of more than one interconnected tables. The relational database manager system is capable of interconnecting data tables with each other on a logical basis, and search for the common information inside these tables.

In order to call a table a relation it needs to meet the following criteria:

- It cannot have two identical rows.
- Each and every column has its own name.
- The sequence of the rows and columns is optional.

The relational databases usually contain more than one logically interconnected table. There is a set relation between the tables. It is very important during the design stage of the process that we construct these relations carefully.

The basis of the structure of the relational database is the normalization (see later), which refers to the method used for determining the optimal place of the data.

THE THREE LEVELS OF THE DATABASE

'DESIGNER'S BLINDNESS'

Both computing designers and users are tool oriented. That means, they think in the data structure that is supported by their currently used database manager. The problem occurs when they change to a new system. Then they have to start everything over. However, the database has a **device-independent approach**. We differentiate between the **notional**, **logical** and **physical** level of the DB. The structure of the DB needs to be formed in three steps.

THE NOTIONAL AND LOGICAL STRUCTURE OF THE DB

Designing and modifying the DB should always occur on the required level. During the designing phase there are two possible attitudes:

- There are some, who use the file manager type of systems, like dBase, Paradox, or FoxPro. These can be accessed through a file-server structure in a network system. They have, primarily, historical importance.
- There are others, who design more complex systems, and the aim is to achieve a multi-user network environment. They use client-server architecture; in addition, they may further develop it to produce multilayer applications. Such systems are, for example, Oracle, MSSQL, IBM DB2, PostgreSQL, MySQL.

"We call the data structure conceptual, when they reflect the phenomena, its characteristics and its relations according to reality, and at the same time reflecting the natural concepts." (Halassy 1994, 45)

There are many factors that can influence what the structure of the database should look like:

- **Technical factor:** it is often the case that you have to get accustomed to the possibilities of the database manager. The designed data model may not be true to nature.
- **Accessibility:** it may happen that we need to modify a good structure, because of the privacy of the data.
- **Efficiency:** we may need to choose a data structure with a single report instead of one with multi-report, because our database manager supports this structure much better.

Therefore, we call those data structures that meet the technical, accessibility and efficiency criteria **logical structures**. The best solution of realization is when the principal planning follows the conceptual database designing.

THE PHYSICAL STRUCTURE OF THE DB

A database is only acceptable when the physical manifestation matches reality. However, there are some problems that may emerge:

- Assertion: the input data must be valid. For instance, if we add a date, its structure and value must be valid.
- Data presentation: when we give the type and size of a datum it is called data presentation. There are types like textual, imagery, logical, numerical, etc. These should be handled separately for each has its own executable operation.
- Organisation of data and way of storage: the more modern a database manager is, the less attention is required for the way the files are stored.

THE NAMES OF THE PHYSICAL DATA STRUCTURE

"We call the conscious order of storage, access and presentation of knowledge on a store **physical data structure**." (Halassy 1994, 49)

Data table, file: it matches the definition of the entity. The system handles the data in a table form, and thus we need to think in tables.

Field, column: this is the same as property. We refer to the name of the given property with **field name**, and to the property occurrence with **field value**. Field is also referred to as **column**.

Record, row: same as **entity occurrence**. Record is the value that can be found in a row of a report. These are only concerning one entity.

Elementary item: values appearing in the cells of a table

Entity: that of which we store the data about. We consider a person, for instance, an entity.

Attribute: a property, characteristic feature of the entity. A person's characteristic feat could be its height.

Entity type: all the properties of the entity as a whole.

Entity occurrence: the concrete properties of the entity. For example, Opel Astra, 10 years old, 1400 cm3, blue colored.

Primary key: the property of an entity, which takes up a different value for every entity occurrence, is known as the atom or identifier of the entity. It is also known as primary key.

We call the relation between the entities a relationship. When it comes to practice, it is not self-evident what kind of relationship we crate between the entities, and it is a difficult task at the same time. This is the core of both database designing and its hardships.

"Forgin Key, Candidate Key and Super Key have discussed in next lecture."

THE THREE FACTORS OF THE DATA MODEL

The data model has three factors: entity, property, relation. These are all equal members of the data model; therefore, none of them is superior to the other.

- The properties of the entity are known as its internal structure.
- The relations of the entity are the entities external structure.

THE ROLE OF THE PROPERTIES

The properties have four roles:

- **Atom/identifier or primary key:** the given property clearly identifies the entity occurrence.
- **Descriptive:** those properties that, considering the entity occurrence, are not unique. Most of the properties of an entity are like this.
- **Switcher or foreign key:** a property that is being identifier in one entity and descriptive in another. It could ensure the relation between the two entities.
- **Super key:** if the relation has one column, which clearly identifies every single record.

The roles of the properties are not of the same importance. "We call the function of the property within the entity its relative role, and the most important relative role its absolute role." (Halassy, 1994, 75)

Relative means that the task of the property depends on, which entity contains it. Another thing is, a property can have the same relative and absolute role.

"Two entities only have a relationship with each other, if one of them as a switcher property contains the others identifier property." (Halassy 1994, 76)

Requirements a primary key must meet:

- All entities must have an identifier.

- The identifier's value cannot be empty or unknown in any entity occurrence.
- Every entity can have only one identifier property.
- "The same property can only be used by one entity." (Halassy 1994, 74).

LECTURE QUESTIONS:

1- Why we create database and store the data inside it?

2- What is the differences between data and information?

3- Does the program code are the same data in the database? why?

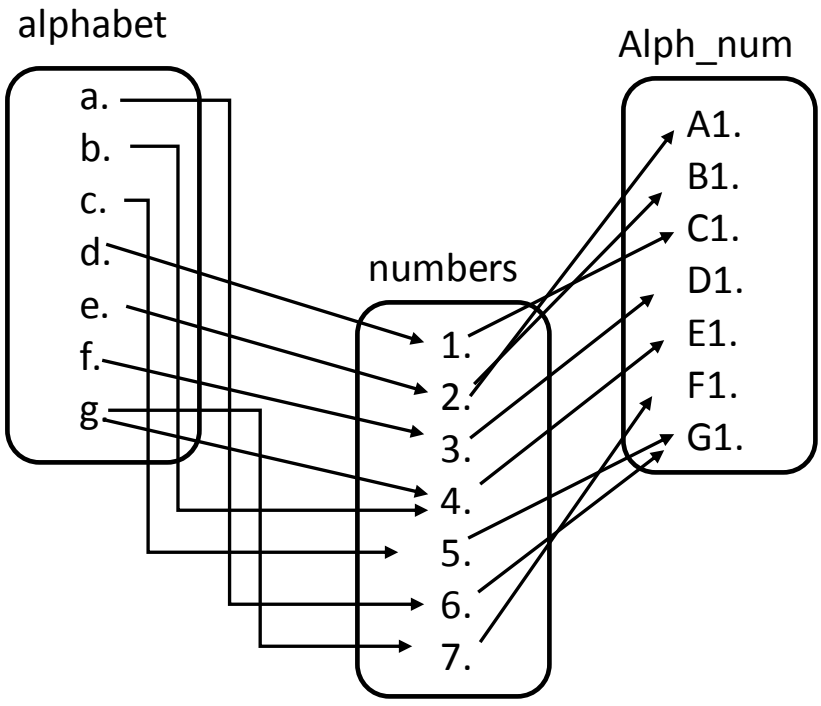
4- What is main part for database, and what Is the main part for table?

5- What is the type of relationship between the following tables?

A- Alphabet table & alph_num table

B- Numbers table & Alph_num table

C- Alphabet table & numbers table



Summary

In this lecture,

The student's will be able to learn how to make the definition of database terminology from tables, queries, reports and forms. also will be able to create and manage databases, use data stored in the databases. The most important point is to understanding the data models of database, three levels of the database, the physical data structure, three factors of the data model with entire example illustrates the work of this function.