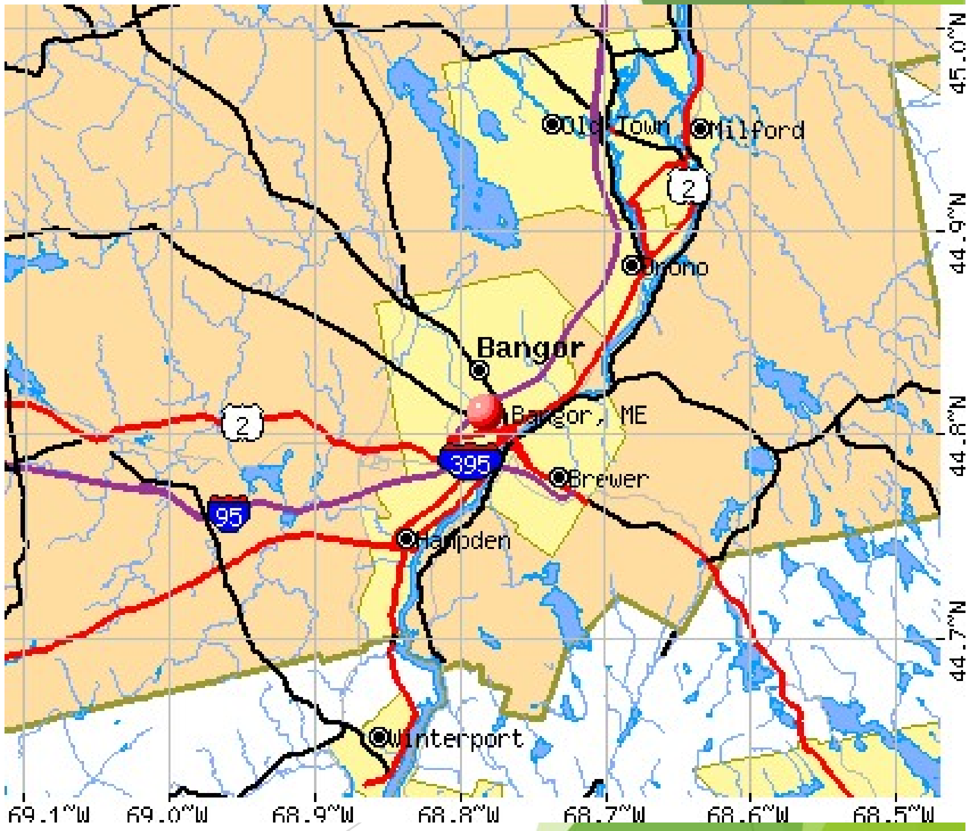
Lecture 4:
The GIS Database

Entity

Bangor

- Penobscot County, Maine, United States
- Centroid - 44.801N , -6778W
- Area 34.4 square miles
- Elevation – 158 feet
- Population 31,473



What is a database?

A database is any organized collection of data. Some examples common examples:

- a telephone book
- T.V. Guide
- airline reservation system
- motor vehicle registration records
- papers in your filing cabinet
- files on your computer hard drive.

Database Definitions

What is a database?

It's an organized collection of data, it need not be a computer based system.

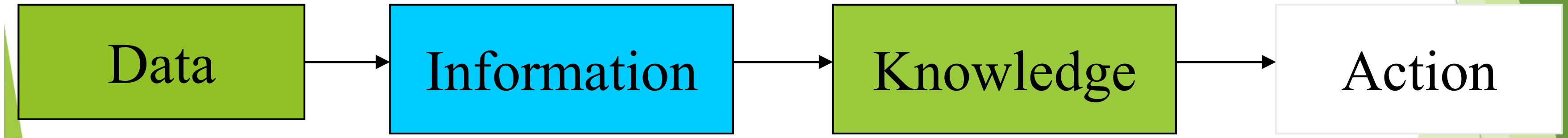
What is a database management system (DBMS)?

A software system designed to:

- Organize that data in a flexible manner,
- Provide tools to add, modify or delete data from the database,
- Query the data,
- Produce reports summarizing selected contents.

What is the ultimate purpose of a database management system?

Is to transform



Data → Information → Knowledge → Action

Features of a DBMS

Database Management Systems provide features to maintain database:

- Data independence - It refers to the immunity of user applications to make changes in the definition and organization of data.
- Integrity and security - refers to maintaining and assuring the accuracy and consistency of data over its entire life-cycle

Features of a DBMS

Database Management Systems provide features to maintain database:

- Transaction management - A **transaction** comprises a unit of work performed within a DBMS against a database, and treated in a coherent and reliable way independent of other transactions.

Features of a DBMS

Database Management Systems provide features to maintain database:

- Concurrency control - ensures that correct results for concurrent operations are generated, while getting those results as quickly as possible.
- Backup and recovery
- Provides a language for the creation and querying of the database.
- A language for writing application programs

Selecting a Database Management System

Database management systems (or DBMSs) can be divided into two categories:

- Desktop databases are oriented toward single-user applications and reside on standard personal computers (hence the term desktop).
- Server databases contain mechanisms to ensure the reliability and consistency of data and are geared toward multi-user applications.

Relational Databases

- The relational database model is the most dominant model in both the corporate and GIS world, due to its flexibility, organization, and functioning..
- It can accommodate a wide range of data types.

Relational Database Terminology

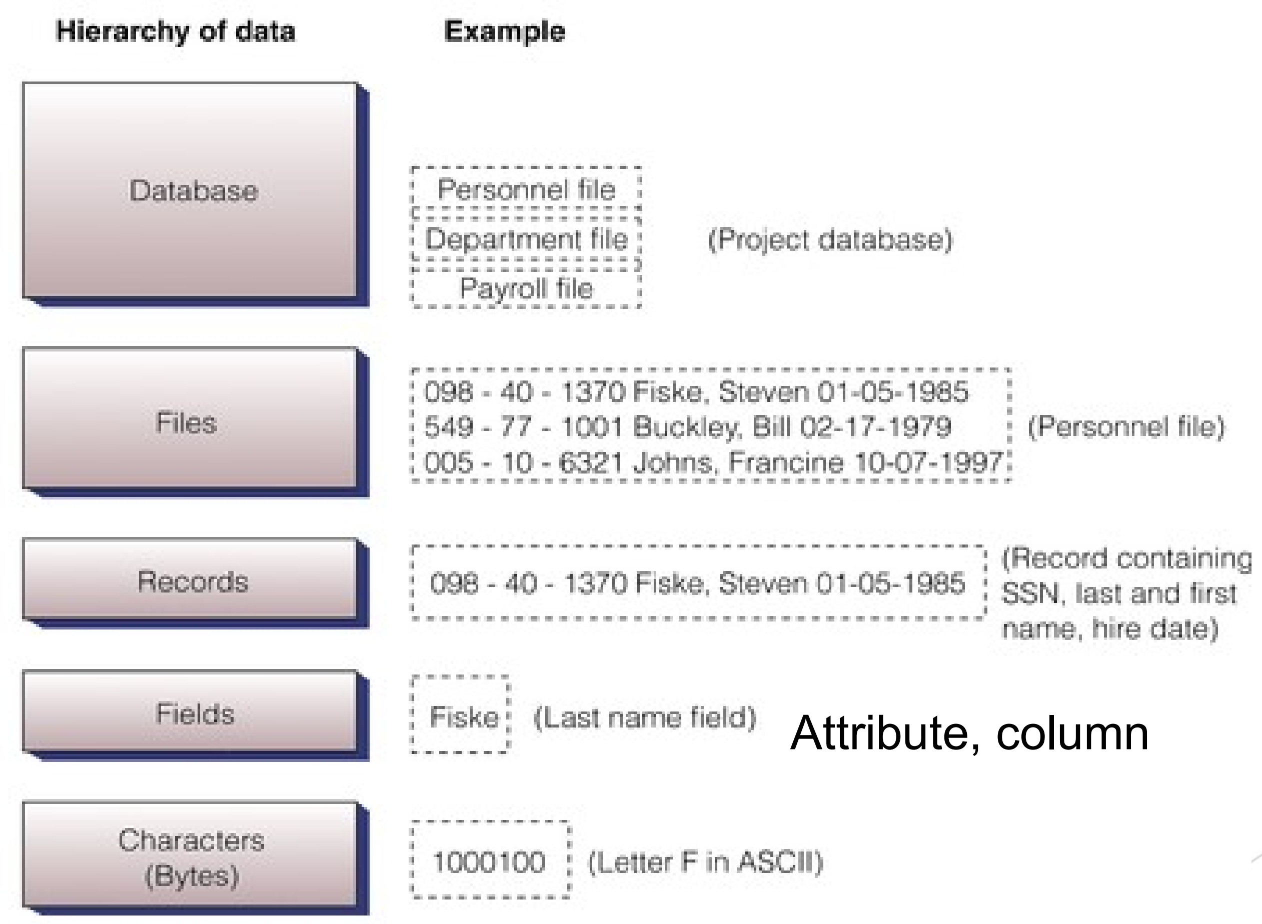
- Each table contains the data for a single entity.
- Each instance of an entity is a row/record/tuple in the table. This is a specific instance of the entity.
- Columns contain attributes/fields that describe the entity.
  - Attributes in a column must be from the same domain (text, integer, date).
  - An attribute may have a range (e.g.; 0 ≤ integers ≤ 100)
  - Column order has no significance.
- Tables are related through keys.

Attributes

- An entity is represented by a set of attributes, that is descriptive properties possessed by all members of an entity set.

  Domain – the set of permitted values for each attribute
- Attribute types:
  - *Simple* and *composite* attributes.
  - *Single-valued* and *multi-valued* attributes
    - E.g. multivalued attribute: *phone-numbers*
  - *Derived* attributes
    - Can be computed from other attributes
    - E.g. *age*, given date of birth

Relational Database Terminology



Hierarchy of data	Example	
Database	Personnel file Department file Payroll file	(Project database)
Files	098 - 40 - 1370 Fiske, Steven 01-05-1985 549 - 77 - 1001 Buckley, Bill 02-17-1979 005 - 10 - 6321 Johns, Francine 10-07-1997	(Personnel file)
Records	098 - 40 - 1370 Fiske, Steven 01-05-1985	(Record containing SSN, last and first name, hire date)
Fields	Fiske	(Last name field)
Characters (Bytes)	1000100	(Letter F in ASCII)

Entity

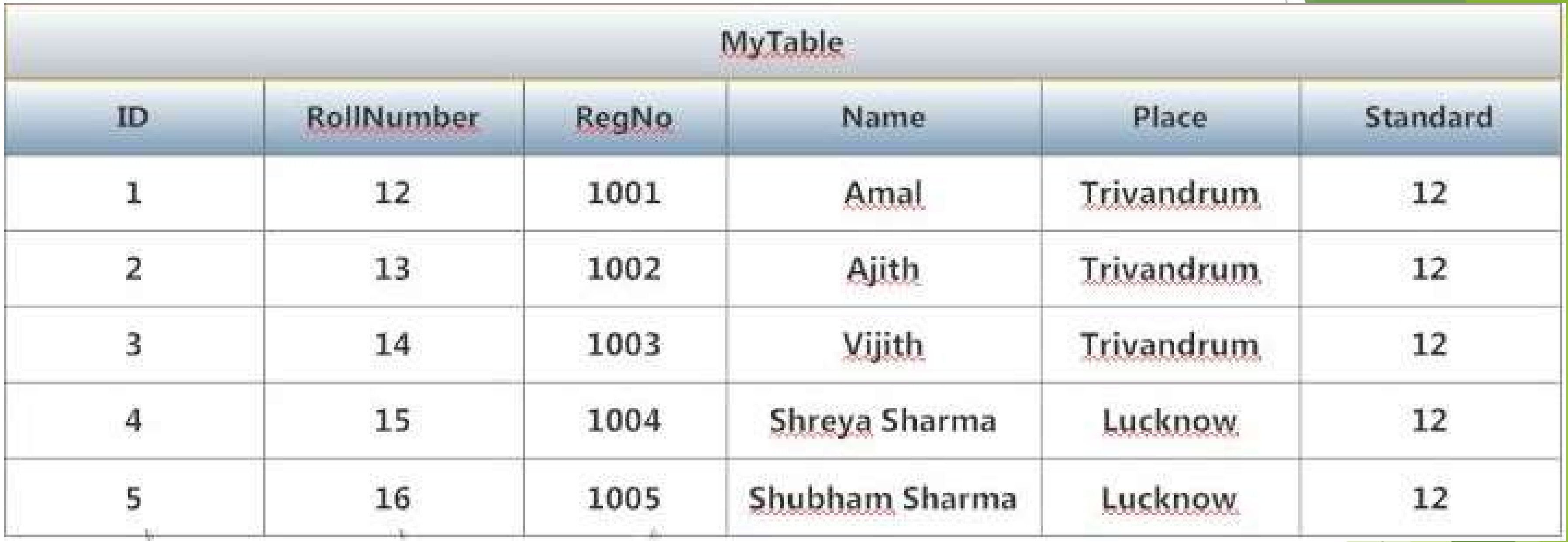
Record, row, tuple
A specific instance of the entity

Attribute, column

Keys

- A *super key* of an entity set is a set of one or more attributes whose values uniquely determine each entity.
- A *candidate key* of an entity set is a minimal super key
  - *Customer-id* is candidate key of *customer*
  - *account-number* is candidate key of *account*
- Although several candidate keys may exist, one of the candidate keys is selected to be the *primary key*.

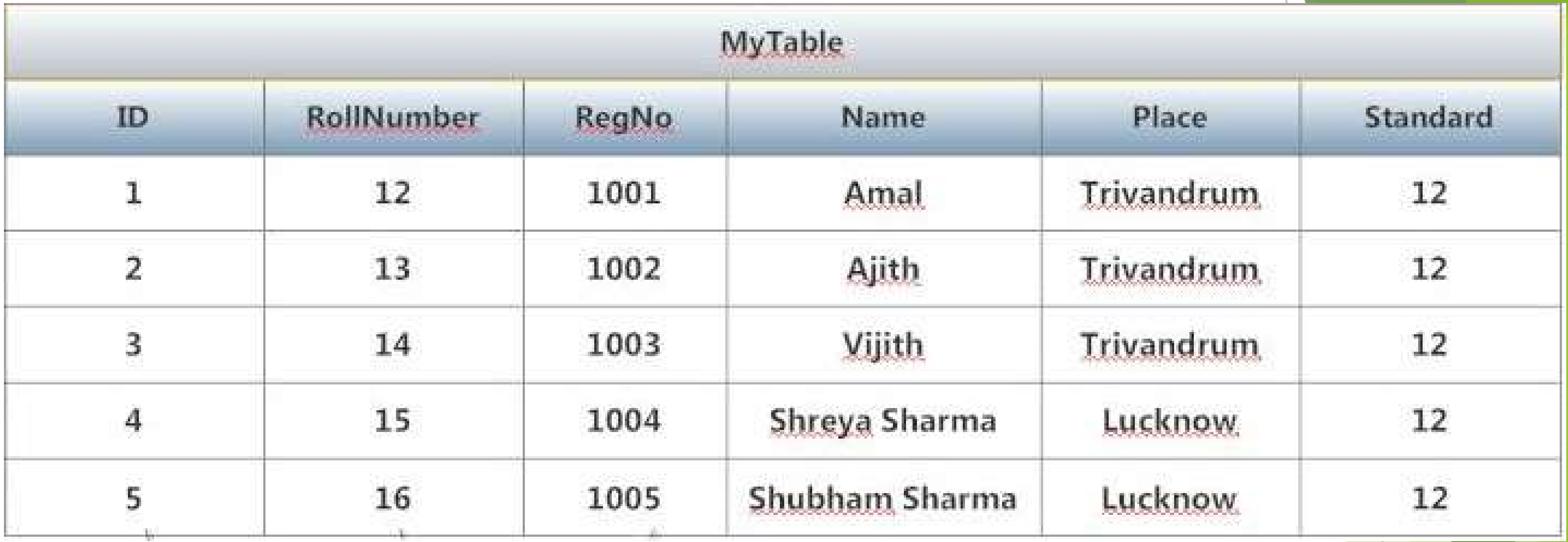
Super Key



MyTable					
ID	RollNumber	RegNo	Name	Place	Standard
1	12	1001	Amal	Trivandrum	12
2	13	1002	Ajith	Trivandrum	12
3	14	1003	Vijith	Trivandrum	12
4	15	1004	Shreya Sharma	Lucknow	12
5	16	1005	Shubham Sharma	Lucknow	12

ID, RollNumber, RegNo, Name, Place, Standard
ID, RegNo, Name
RegNo, Name, Place, Standard
RollNunmber, Name Place
ID, Place
RegNo

Candidate Keys

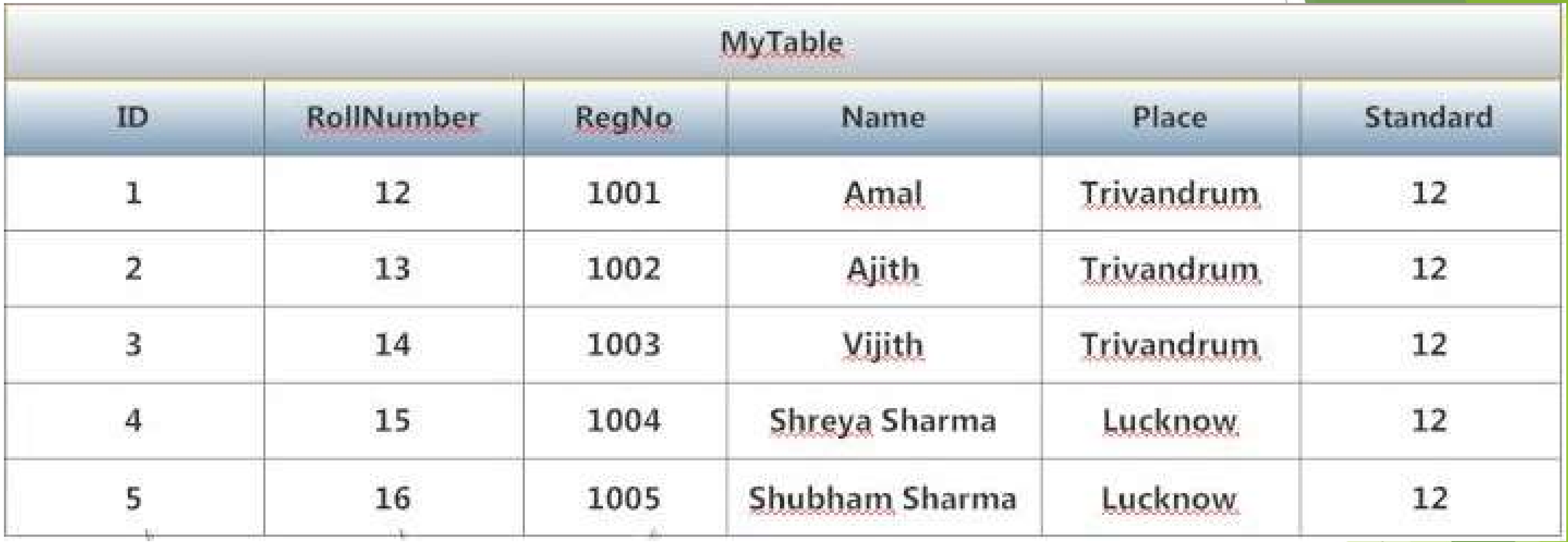


MyTable

ID	RollNumber	RegNo	Name	Place	Standard
1	12	1001	Amal	Trivandrum	12
2	13	1002	Ajith	Trivandrum	12
3	14	1003	Vijith	Trivandrum	12
4	15	1004	Shreya Sharma	Lucknow	12
5	16	1005	Shubham Sharma	Lucknow	12

ID
RollNumber
RegNo

Primary Key



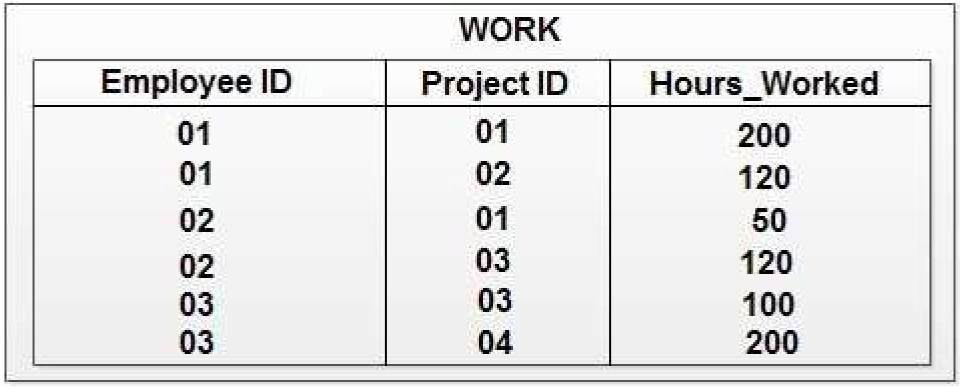
MyTable

ID	RollNumber	RegNo	Name	Place	Standard
1	12	1001	Amal	Trivandrum	12
2	13	1002	Ajith	Trivandrum	12
3	14	1003	Vijith	Trivandrum	12
4	15	1004	Shreya Sharma	Lucknow	12
5	16	1005	Shubham Sharma	Lucknow	12

ID

Keys

- A *composite key/ concatenated* is a key with more than one attribute.



WORK

Employee ID	Project ID	Hours_Worked
01	01	200
01	02	120
02	01	50
02	03	120
03	03	100
03	04	200

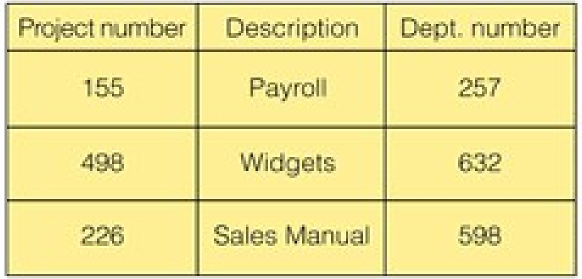
http://ecomputernotes.com/images/Composite%20Key.jpg

Keys

- A *foreign key* is an attribute that is a key of one or more relations other than the one in which it appears.

Foreign Key

Data table 1: Project table

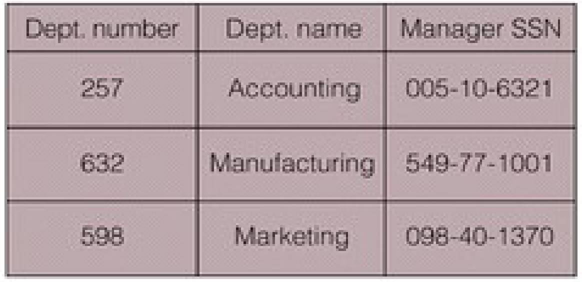


Project number	Description	Dept. number
155	Payroll	257
498	Widgets	632
226	Sales Manual	598

Primary Key

Foreign Key

Data table 2: Department table

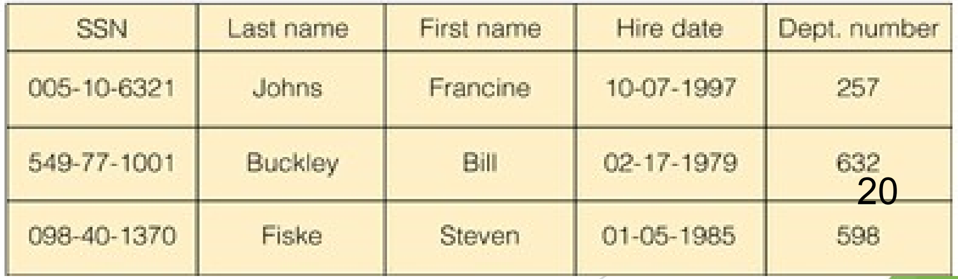


Dept. number	Dept. name	Manager SSN
257	Accounting	005-10-6321
632	Manufacturing	549-77-1001
598	Marketing	098-40-1370

Primary Key

Foreign Key

Data table 3: Manager table



SSN	Last name	First name	Hire date	Dept. number
005-10-6321	Johns	Francine	10-07-1997	257
549-77-1001	Buckley	Bill	02-17-1979	632
098-40-1370	Fiske	Steven	01-05-1985	598

Primary Key

Foreign Key

Keys

- Given the importance of keys, there are usually some restrictions on them: e.g., null values are not allowed.

Physical Database Structure

The physical design of the database specifies the physical configuration of the database on the storage media.

- This includes detailed specification of data elements, data types, indexing options and other parameters residing in the DBMS data dictionary.
- It is the detailed design of a system that includes modules & the database's hardware & software specifications of the system.

https://en.wikipedia.org/wiki/Database_design

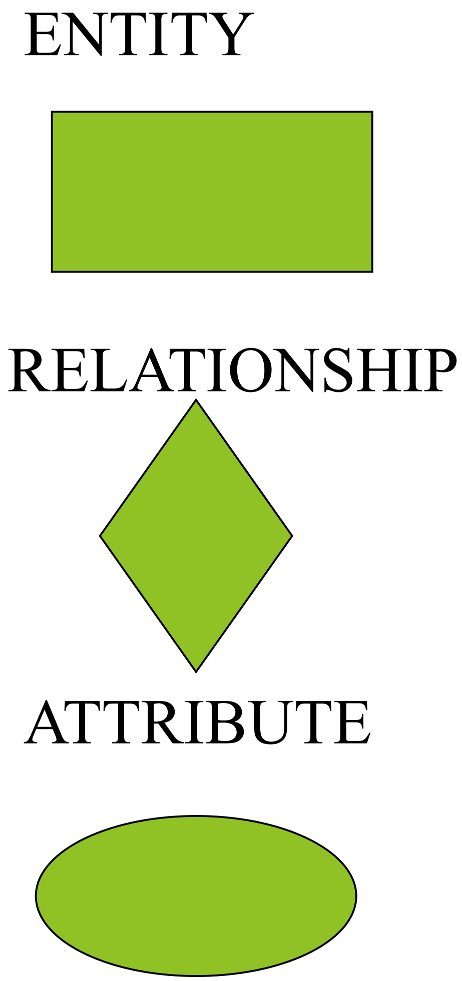
Logical Database Structure

- Several logical data structures are used to express the relationships between individual data elements or records in a database.
- Common logical data structures are hierarchical, network, and relational.

Conceptual Structure

- The conceptual structure is often represented as a schema.
- One example is the entity-relationship (ER) diagram.

Entity Relationship Diagram



ENTITY

RELATIONSHIP

ATTRIBUTE

□**Rectangles** represent entity sets.

□**Diamonds** represent relationship sets.

□**Lines** link attributes to entity sets and entity sets to relationship sets.

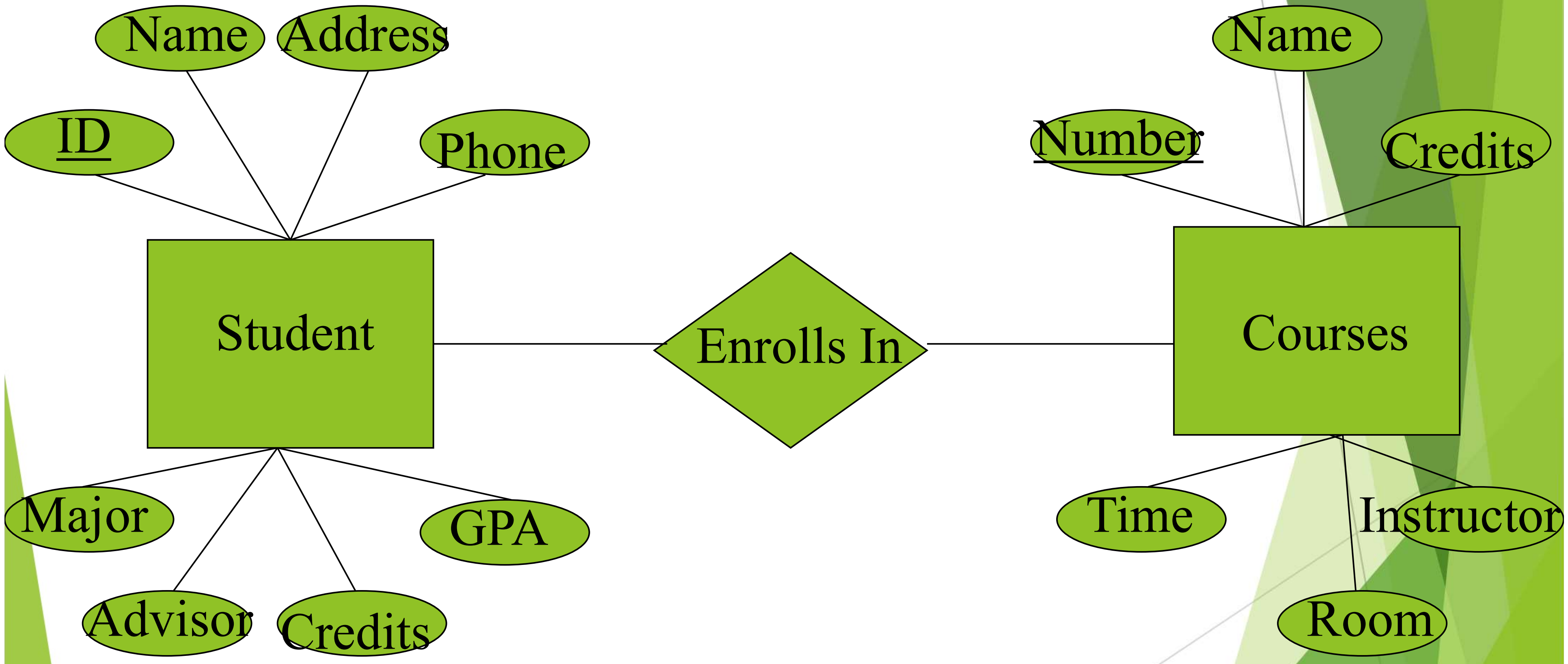
□**Ellipses** represent attributes

  □**Double ellipses** represent multivalued attributes.

  □**Dashed ellipses** denote derived attributes.

□**Underline** indicates primary key attributes.

Entity Relationship Diagram



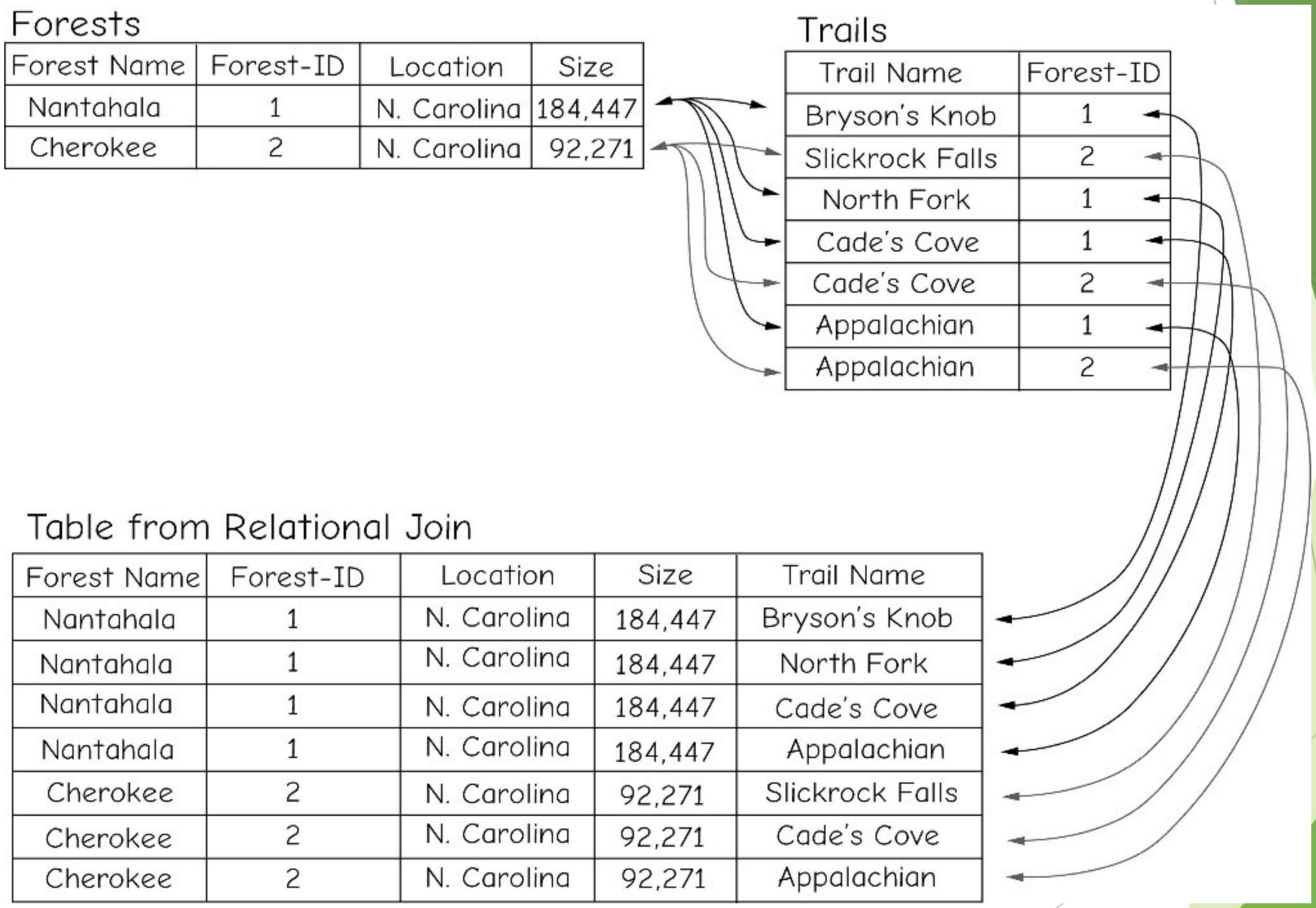
Entity Relationship Model

- The result is a diagram of all of the entities, their attributes, and the relationships between entities
  - Each entity becomes a table.
    - Student table
    - Course table
  - Each relationship (usually) becomes a table.
    - Enrolls, which allows you to join information from both tables.

Types of Relationships between Entities

- 1:1 – one faculty member is assigned to one office.
- 1:M (M:1) – one faculty member teaches many courses.
- M:N – many students take many courses.

Table Join



Forests

Forest Name	Forest-ID	Location	Size
Nantahala	1	N. Carolina	184,447
Cherokee	2	N. Carolina	92,271

Trails

Trail Name	Forest-ID
Bryson's Knob	1
Slickrock Falls	2
North Fork	1
Cade's Cove	1
Cade's Cove	2
Appalachian	1
Appalachian	2

Table from Relational Join

Forest Name	Forest-ID	Location	Size	Trail Name
Nantahala	1	N. Carolina	184,447	Bryson's Knob
Nantahala	1	N. Carolina	184,447	North Fork
Nantahala	1	N. Carolina	184,447	Cade's Cove
Nantahala	1	N. Carolina	184,447	Appalachian
Cherokee	2	N. Carolina	92,271	Slickrock Falls
Cherokee	2	N. Carolina	92,271	Cade's Cove
Cherokee	2	N. Carolina	92,271	Appalachian

Table Joins

- Table joins depend on the **data** not the **attribute name**.
- There are many different types of table joins.
- Tables can be joined regardless of the relationship EXCEPT:
  - When joining to the feature attribute table in a GIS, the relationship must be 1:1 or M:1
  - Other relationships must use the relate.

One-to-One Join

Employee-id	Job
1	Digislave
2	Useless Supervisor

Employee-id	name
1	Tom
2	John

Join Employee-id to Employee-id

After join

Employee-id	Job	Name
1	Digislave	Tom
2	Useless Supervisor	John

A join does not permanently alter the table structure

Many-to-One Join

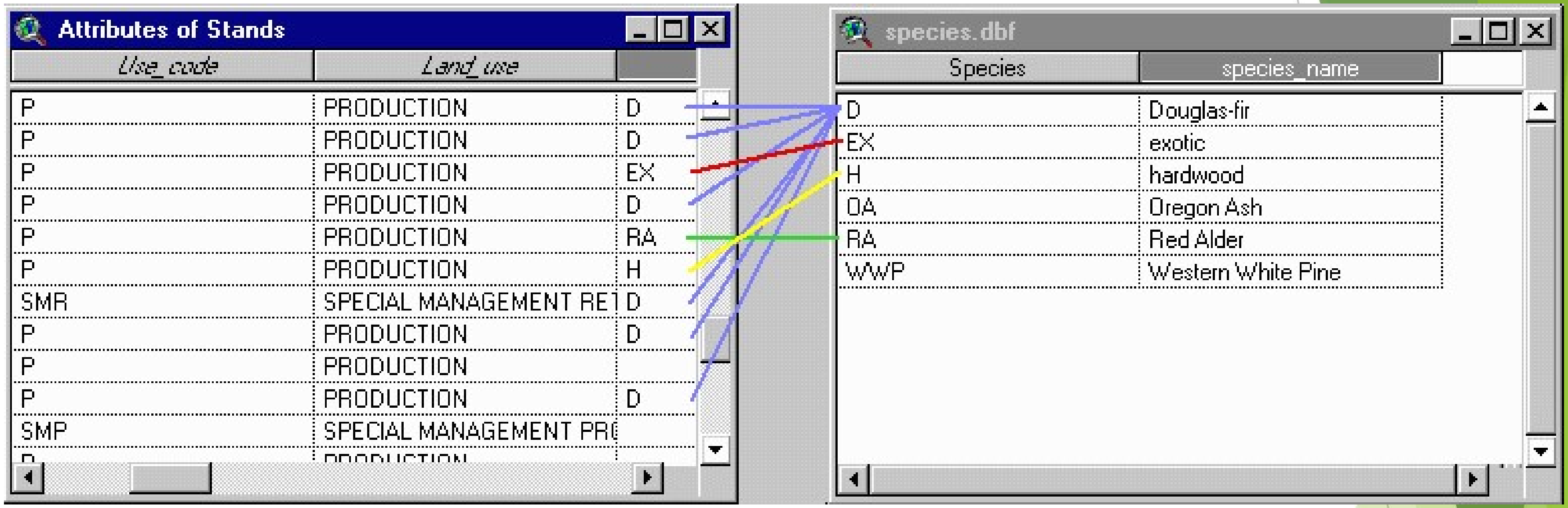
Polygon Id	Symbol
1	Qa
2	Qa
3	Pa
4	Qe

Symbol	Description
Qa	Quaternary Alluvium
Qe	Quaternary Eolian
Pa	Permian Abo

After Join on Symbol

Polygon ID	Symbol	Description
1	Qa	Quaternary Alluvium
2	Qa	Quaternary Alluvium
3	Pa	Permian Abo
4	Qe	Quaternary Eolian

Relate in a GIS



Use_code	Land_use	
P	PRODUCTION	D
P	PRODUCTION	D
P	PRODUCTION	EX
P	PRODUCTION	D
P	PRODUCTION	RA
P	PRODUCTION	H
SMR	SPECIAL MANAGEMENT RET	D
P	PRODUCTION	D
P	PRODUCTION	
P	PRODUCTION	D
SMP	SPECIAL MANAGEMENT PR	
P	PRODUCTION	

Species	species_name
D	Douglas-fir
EX	exotic
H	hardwood
OA	Oregon Ash
RA	Red Alder
WWP	Western White Pine

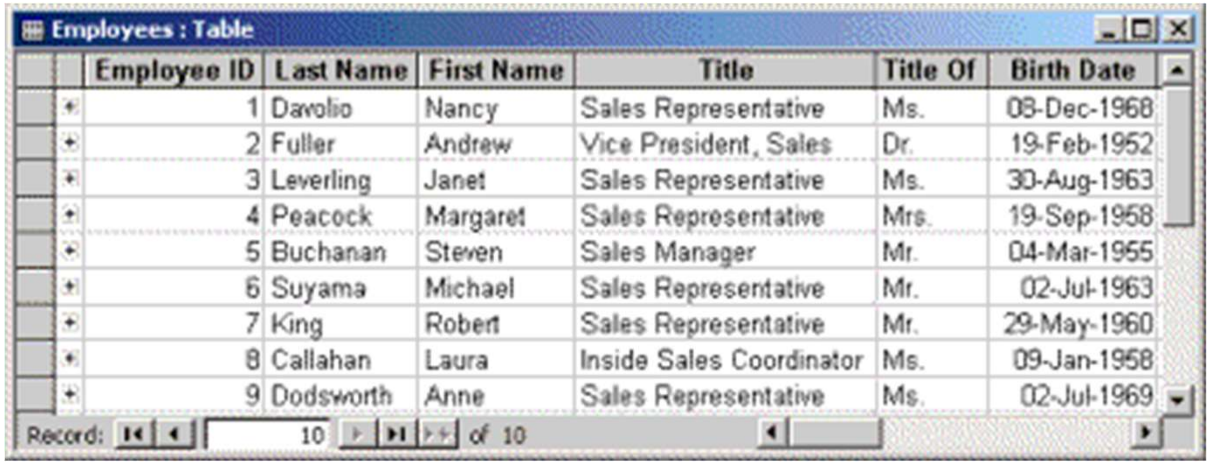
https://courses.washington.edu/gis250/lessons/tables/images_av3/relate_table1.gif

Relational Algebra

- Relational database relied on relational algebra.
- Relational algebra takes tables/relations as inputs and returns tables as outputs.
- The algebra combines or splits tables by rows or by columns to generate either a subset of tables or an expanded tables.

Fundamental Building Blocks

Tables comprise the fundamental building blocks of any database.



Employees : Table

Employee ID	Last Name	First Name	Title	Title Of	Birth Date
1	Davolio	Nancy	Sales Representative	Ms.	08-Dec-1968
2	Fuller	Andrew	Vice President, Sales	Dr.	19-Feb-1952
3	Leverling	Janet	Sales Representative	Ms.	30-Aug-1963
4	Peacock	Margaret	Sales Representative	Mrs.	19-Sep-1958
5	Buchanan	Steven	Sales Manager	Mr.	04-Mar-1955
6	Suyama	Michael	Sales Representative	Mr.	02-Jul-1963
7	King	Robert	Sales Representative	Mr.	29-May-1960
8	Callahan	Laura	Inside Sales Coordinator	Ms.	09-Jan-1958
9	Dodsworth	Anne	Sales Representative	Ms.	02-Jul-1969

Record: 10 of 10

The table above contains the employee information for an organization -- characteristics like name, date of birth and title.

Relational Algebra

- Five basic operators
  - select: σ
  - project: Π
  - union: $\cup$
  - difference: –
  - Cartesian product: x
- The operators take one or two relations as inputs and produce a new relation as a result.

Relational Algebra

- Derived Relational operators
  - Intersection $\cap$
  - Divide (not used very often)
  - Join $\bowtie$
- These can be expressed using different combinations of the fundamental operators.

Select Operation – Example

- Relation r

A	B	C	D
α	α	1	7
α	β	5	7
β	β	12	3
β	β	23	10

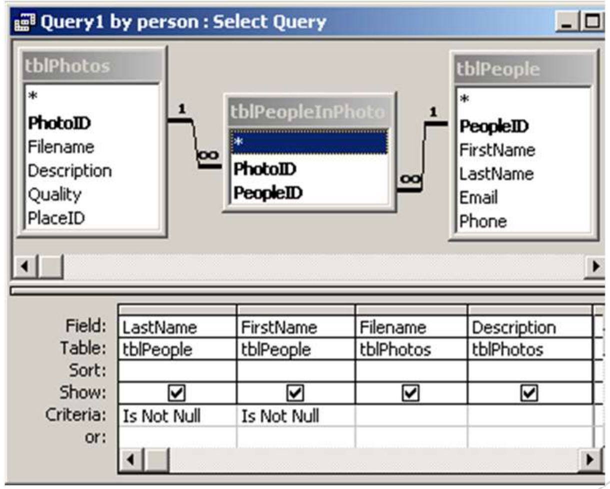
- $\sigma_{A=B \wedge D > 5}(r)$

Select from relation r
where A=B AND D>5

A	B	C	D
α	α	1	7
β	β	23	10

Database Queries

- Queries may be made of one table or several tables at the same time.
- In many systems querying is facilitated by icons, or menus, or queries by example (QBE – a graphical query language).



Field:	LastName	FirstName	Filename	Description
Table:	tblPeople	tblPeople	tblPhotos	tblPhotos
Sort:				
Show:	☑	☑	☑	☑
Criteria:	Is Not Null	Is Not Null		
or:				

Structured Query Language (SQL)

- DDL – Data Definition Language; used to create and manage the database.
- DDM – Data Manipulation Language; used to query the database.

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

```
select   customer.customer-name
from     customer
where    customer.customer-id = ‘192-83-7465’
```

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

SQL

1. Records with Area>20.0
2. Records with LandUse=Urban and Municip=City

ID	Area	LandUse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

Attribute Queries

Simple selection:

records with Area > 20.0

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

AND selection:

records with (Landuse = Urban) **and** (Municip = City)

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

SQL

3. Records with Area>20.0 OR Municip=City
4. Records with LandUse NOT Urban

ID	Area	LandUse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

OR selection:

records with (Area > 20.0)
OR (Municip = City)

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

NOT selection:

records with
Landuse NOT Urban

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

SQL

5. NOT [LandUse=Urban AND Municip=County
6. [NOT (LandUse=Urban)] AND [NOT (Municip=County)]

ID	Area	LandUse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

NOT [(Landuse = Urban) AND
(Municip = County)]

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County

[NOT (Landuse = Urban)] AND
[NOT (Municip = County)]

ID	Area	Landuse	Municip
1	10.5	Urban	City
2	330.3	Farm	County
3	2.4	Suburban	Township
4	96.0	Suburban	County
5	22.1	Urban	City
6	30.2	Farm	Township
7	4.4	Urban	County