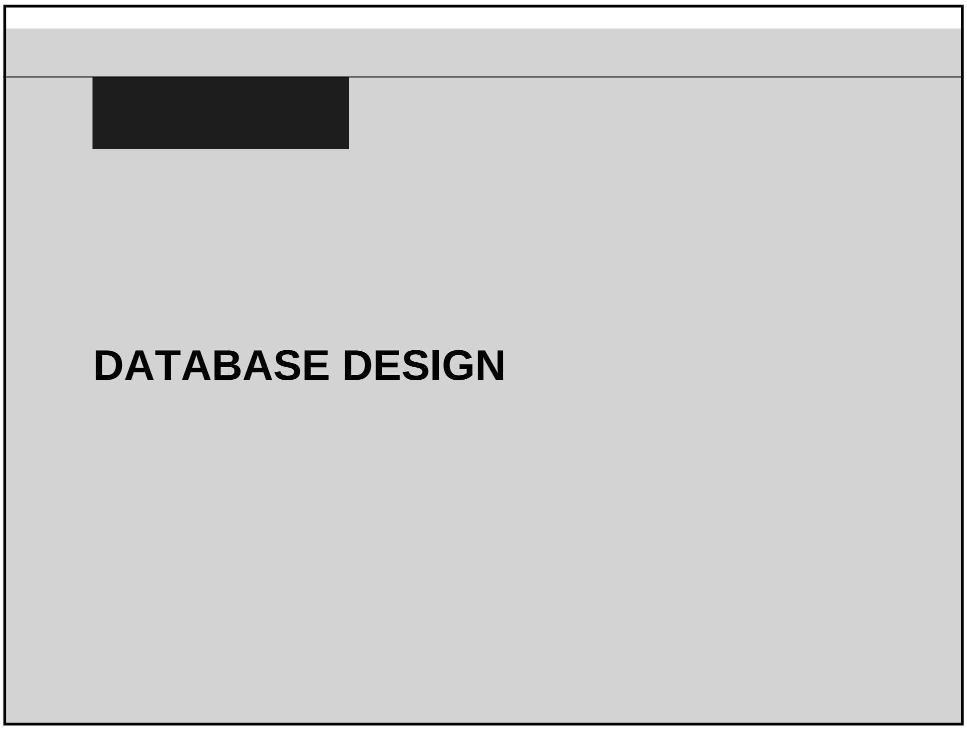
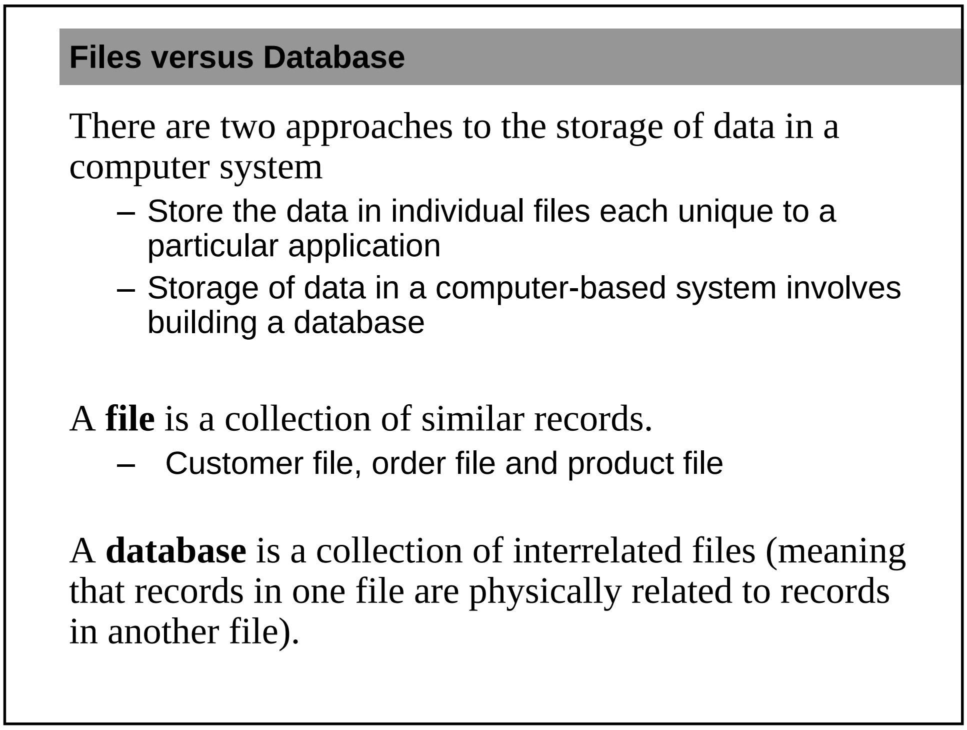


DATABASE DESIGN



Files versus Database

There are two approaches to the storage of data in a computer system

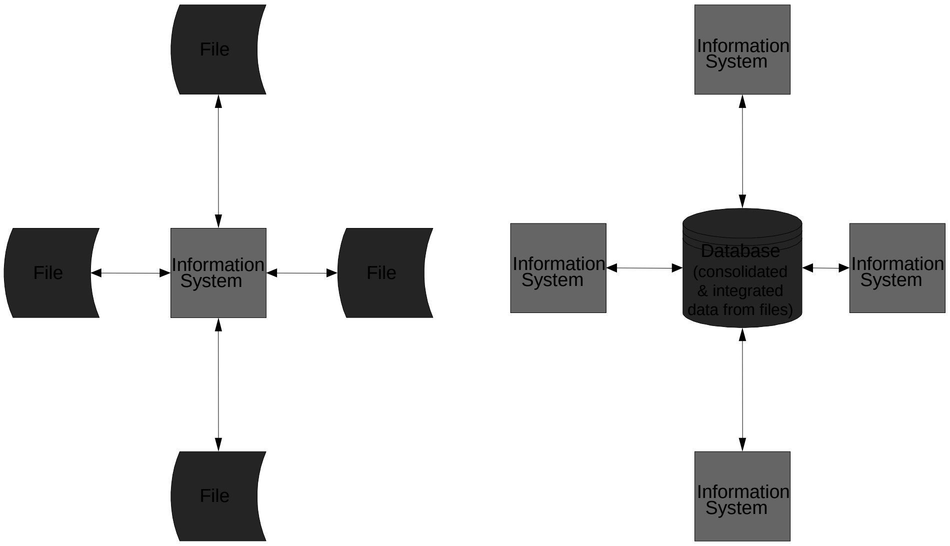
- Store the data in individual files each unique to a particular application
- Storage of data in a computer-based system involves building a database

A **file** is a collection of similar records.

- Customer file, order file and product file

A **database** is a collection of interrelated files (meaning that records in one file are physically related to records in another file).

Files versus Database



Problems with Conventional Files

- Harder to adapt to sharing across applications
- Harder to adapt to new requirements
- Need to duplicate attributes in several files.
- Somewhat slower performance
- Higher development costs
- Greater data vulnerability

Data base concepts

A **table** is the relational database equivalent of a file.

- A table is a named, two-dimentional arrays of data.
- Each table consists of a set of named columns.
- Each row of a table corresponds to a record that contains data

Example

Customer

Customer_id	Name	Salesperson	Region
8038	Anderson	Smith	South
9167	Hobbs	Hicks	West
7927	Tucker	Smith	South

Data base concepts

Properties of Table

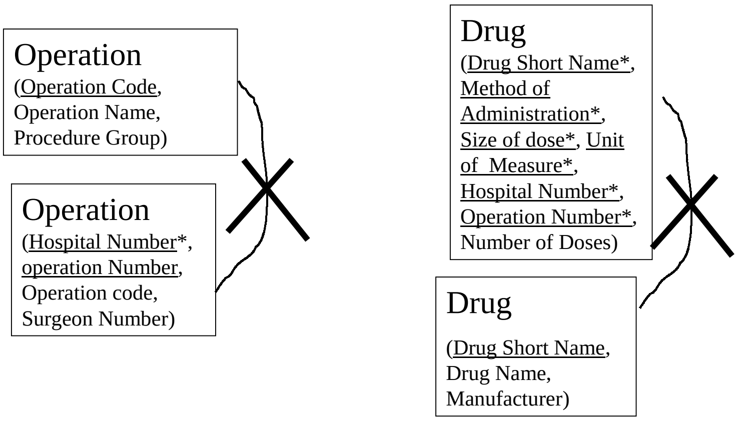
- Each table in a database has a unique name.
- Each row is unique; no two rows in a table are identical
- Each attribute (column) within a table has a unique name.
- The sequence of columns (left to right) is insignificant.
- The sequence of rows (top to bottom) is insignificant.

Database concepts

Properties of Table

- Each table in a database has a unique name.

- Example



Operation
(Operation Code, Operation Name, Procedure Group)

Operation
(Hospital Number*, operation Number, Operation code, Surgeon Number)

Drug
(Drug Short Name*, Method of Administration*, Size of dose*, Unit of Measure*, Hospital Number*, Operation Number*, Number of Doses)

Drug
(Drug Short Name, Drug Name, Manufacturer)

Database concepts

Properties of Table

- Each row is unique; no two rows in a table are identical.

- Example

Emp-Id	*Name*	*Salary*	*Course_title*	
100	Simpson	48,000	C++	✗
100	Simpson	48,000	C++	
100	Simpson	48,000	Unix	
140	Susan	42,000	Java	
140	Susan	42,000	DBMS	✗
140	Susan	42,000	DBMS	
140	Susan	42,000	DBMS	

Database concepts

Properties of Table

- Each attribute (column) within a table has a unique name.

- Example

Operation
(Hospital Number*, **operation Number**, Operation code, Surgeon Number, **operation Number**)

✗

Drug
(Drug Short Name*, Method of Administration*, Size of dose*, Unit of Measure*, **Hospital Number***, Operation Number*, Number of Doses, **Hospital Number)**

✗

Hospital
(Hospital Number, **Hospital Name**, Hospital Category, Contact Person, **Hospital Name**)

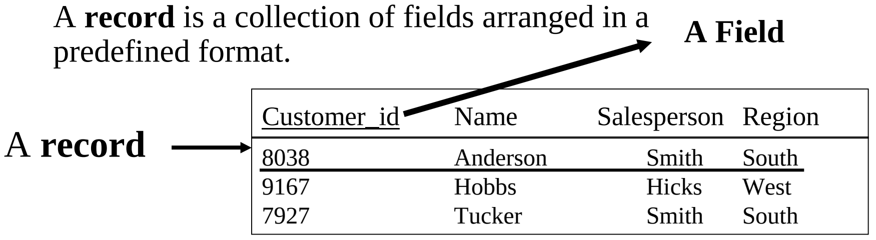
✗

Database concepts

A **field** is the physical implementation of a data attribute. They are the smallest unit of meaningful data.

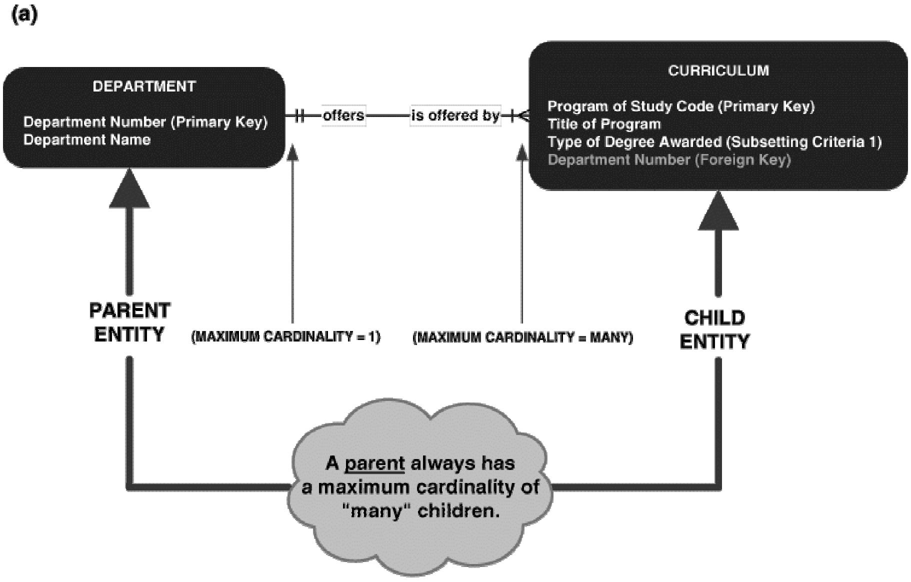
- A **primary key** is a field whose values identify one and only one record in a file.
- A **foreign key** is a pointer to a record in a different file (A foreign key is an attribute or combination of attributes that is the primary key of another entity)

A **record** is a collection of fields arranged in a predefined format.



Customer_id	Name	Salesperson	Region
8038	Anderson	Smith	South
9167	Hobbs	Hicks	West
7927	Tucker	Smith	South

Database concepts (Primary and foreign keys)


(a)
DEPARTMENT
Department Number (Primary Key)
Department Name
offers
is offered by
CURRICULUM
Program of Study Code (Primary Key)
Title of Program
Type of Degree Awarded (Subsetting Criteria 1)
Department Number (Foreign Key)
PARENT ENTITY
(MAXIMUM CARDINALITY = 1)
(MAXIMUM CARDINALITY = MANY)
CHILD ENTITY
A parent always has a maximum cardinality of "many" children.

Database concepts (Primary and foreign keys)

Department

Department number	Department name
101	Computer
102	Math
103	Physics
104	Chemistry
~~101~~	Engineering

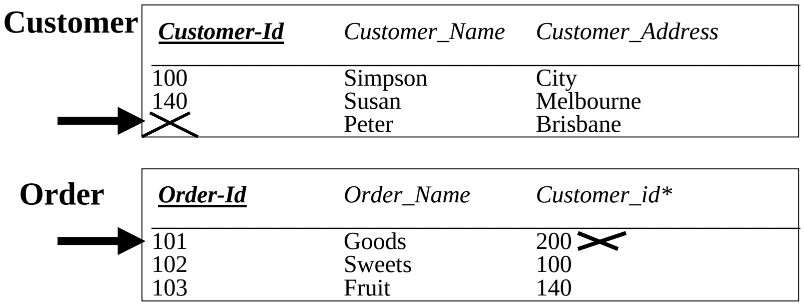
Curriculum

Program of study code	Title of program	Type of degree	Department number
901	Science	Master	101
902	Math	Master	102
903	IT	Master	101
904	IS	Master	101
	Physics	Master	~~115~~

Database Integrity

- Key integrity
- Domain integrity
- Referential integrity

A **referential integrity** error exists when a foreign key value in one table has no matching primary key value in the related table.



Customer

Customer-Id	*Customer_Name*	*Customer_Address*
100	Simpson	City
140	Susan	Melbourne
	Peter	Brisbane

Order

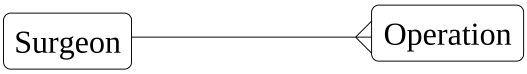
Order-Id	*Order_Name*	*Customer_id**
101	Goods	200
102	Sweets	100
103	Fruit	140

Logical Data Model (Entity relationship model)

• An *entity* is something about which we want to store data.
Examples: Student, University, Book, house etc

• An *attribute* is a descriptive property or characteristic of an entity.
Examples: Student name, student address, student GPA.

• A relationship is a natural business association that exists between one or more entities. Relationships are important to identify and describe because they are the logical link between business functions. They can be interpreted in both directions between multiple entities.
Example:

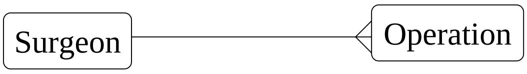


Why ERD

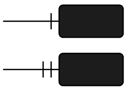
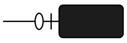
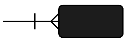
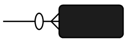
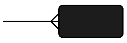
- To facilitate database design.
- To facilitate communication between the database designer and the end user.
- To enable the database designer to display the overall database view of the enterprise.

- Operation (Operation Number, Operation code, Surgeon Number*)
- Surgeon (Surgeon Number, Surgeon Specialty)

The format of the above tables does not encourage a quick appreciation of the main concepts and rules. For example, each operation can be performed by only one surgeon is not immediately apparent.



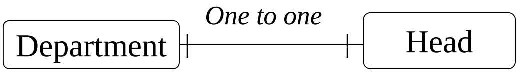
Data Modeling Concepts: Cardinality Notations

Cardinality Interpretation	Minimum Instances	Maximum Instances	Graphic Notation
Exactly one	1	1	
Zero or one	0	1	
One or more	1	Many	
Zero, one or more	0	Many	
More than one	Greater than 1	Greater than 1	

Data Modeling Concepts: Relationships

Draw the following relationships:

- Each department must have one head
- Each head must have one department

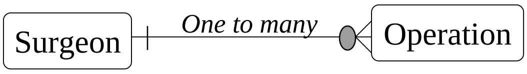


One to one

Department

Head

- A Surgeon could be associated with zero, one or more operations
- Each operation must be associated with only one Surgeon

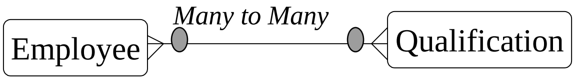


One to many

Surgeon

Operation

- Each employee can be awarded zero, one or more qualifications
- Each qualification can be awarded to zero, one or more employees



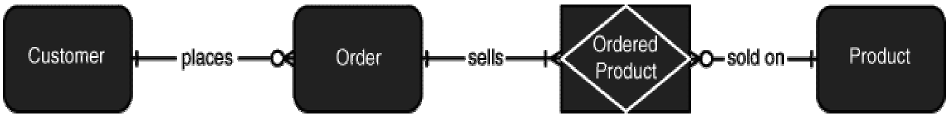
Many to Many

Employee

Qualification

Logical Data Model

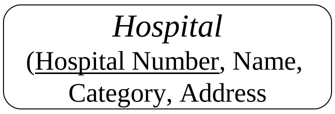
Entity relation diagram



Conversion of E-R diagram to Tables

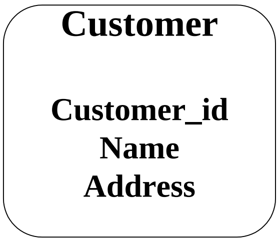
- Simplest solution involves
  - creating a table for each entity type
    - Each entity becomes a table
    - The entity name becomes the table name
    - Each attribute becomes a column
    - The identifier becomes the primary key

Example



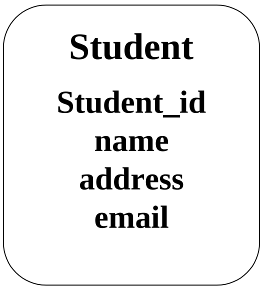
Hospital (Hospital_number, Name, Category, Address)

Conversion of E-R diagram to Tables



Customer (Customer_id, Name, Address)

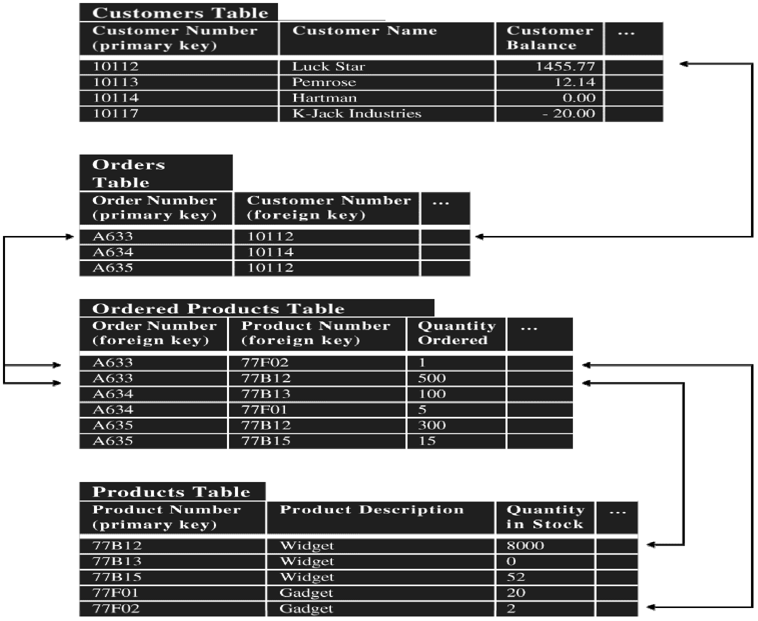
Customer_id	Name	Address



Student (Student_id, Name, address, email)

Student_id	Name	address	email

Physical Data Model (Relational Schema)



Customers Table

Customer Number (primary key)	Customer Name	Customer Balance	...
10112	Luck Star	1455.77	
10113	Pemrose	12.14	
10114	Hartman	0.00	
10117	K-Jack Industries	- 20.00	

Orders Table

Order Number (primary key)	Customer Number (foreign key)	...
A633	10112	
A634	10114	
A635	10112	

Ordered Products Table

Order Number (foreign key)	Product Number (foreign key)	Quantity Ordered	...
A633	77F02	1	
A633	77B12	500	
A634	77B13	100	
A634	77F01	5	
A635	77B12	300	
A635	77B15	15	

Products Table

Product Number (primary key)	Product Description	Quantity in Stock	...
77B12	Widget	8000	
77B13	Widget	0	
77B15	Widget	52	
77F01	Gadget	20	
77F02	Gadget	2	

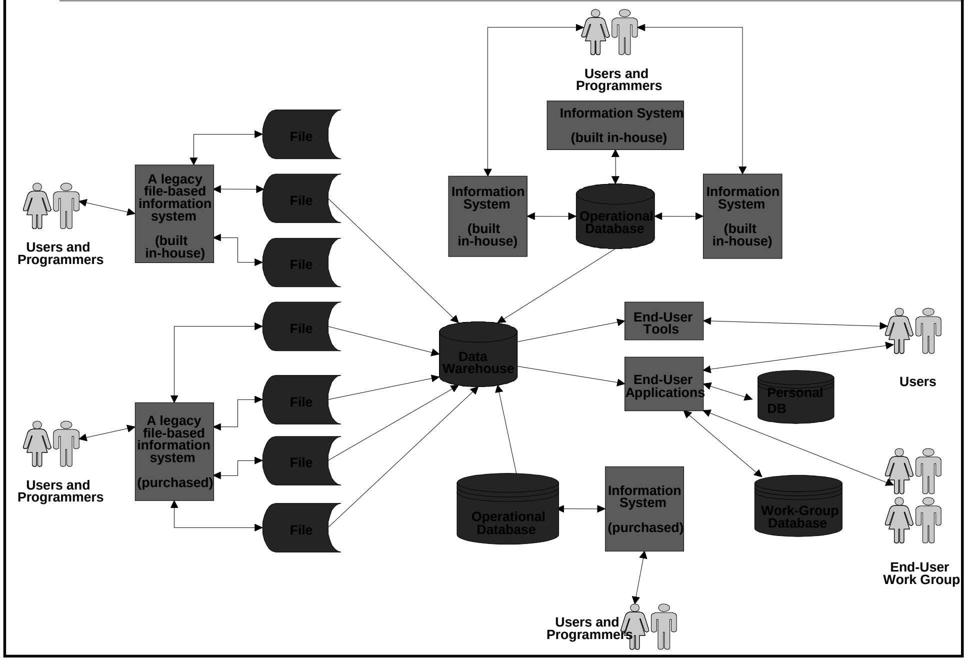
Database concepts (Data Architecture)

A business's **data architecture** defines how that business will develop and use files and databases to store all of the organization's data; the file and database technology to be used; and the administrative structure set up to manage the data resource.

Data is stored in some combination of:

- **Conventional files**
- **Operational databases** (also called transactional databases)
- **Data warehouses**
    - To support data mining
- **Personal databases**
- **Work group databases**

A Modern Data Architecture


Users and Programmers

Information System (built in-house)

Information System (built in-house)

Operational Database

Information System (built in-house)

File

File

File

File

File

File

File

A legacy file-based information system (built in-house)

Users and Programmers

A legacy file-based information system (purchased)

Users and Programmers

Data Warehouse

End-User Tools

End-User Applications

Personal DB

Users

Operational Database

Information System (purchased)

Work-Group Database

End-User Work Group

Users and Programmers

Database concepts (Data Administrators)

A **data administrator** is responsible for the data planning, definition, architecture, and management.

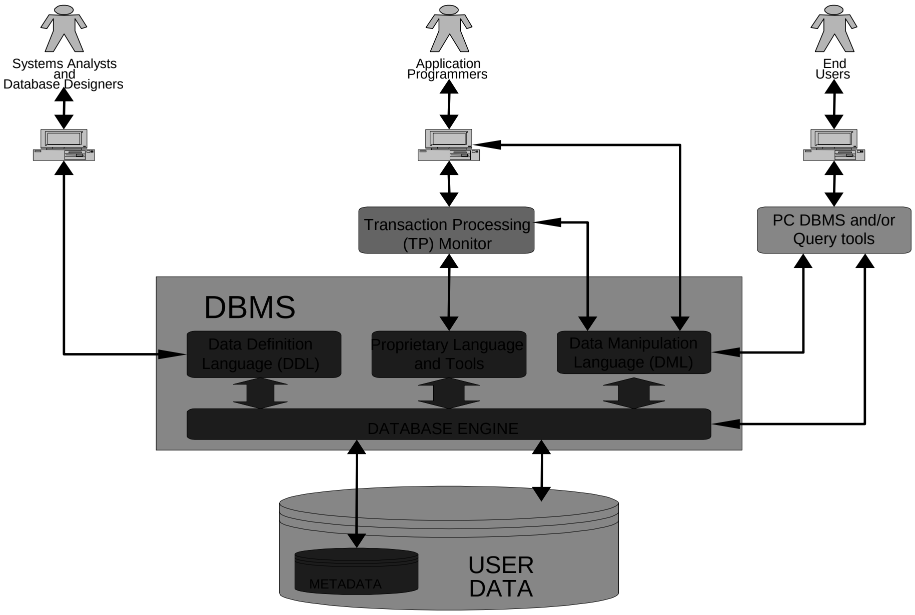
One or more **database administrators** are responsible for the database technology, database design and construction, security, backup and recovery, and performance tuning.

Database concepts (DBMS)

A **database management system (DBMS)** is specialized software that is used to create, access, control, and manage the database. The core of the DBMS is a **database engine**.

- A data definition language (DDL) is that part of the engine used to physically define tables, fields, and structural relationships.
- A data manipulation language (DML) is that part of the engine used to create, read, update, and delete records in the database, and navigate between different files (tables) in the database.

Typical DBMS Architecture



Systems Analysts and Database Designers

Application Programmers

End Users

Transaction Processing (TP) Monitor

PC DBMS and/or Query tools

DBMS

Data Definition Language (DDL)

Proprietary Language and Tools

Data Manipulation Language (DML)

DATABASE ENGINE

METADATA

USER DATA

Database concepts (Relational Databases)

Relational databases implement stored data in a series of two-dimensional tables that are "related" to one another via foreign keys.

- The physical data model is called a **schema**.
- The DDL and DML for a relational database is called **SQL** (Structured Query Language).
- **Triggers** are programs embedded within a table that are automatically invoked by updates to another table.
- **Stored procedures** are programs embedded within a table that can be called from an application program.

Data Normalization (also see Chapter 8)

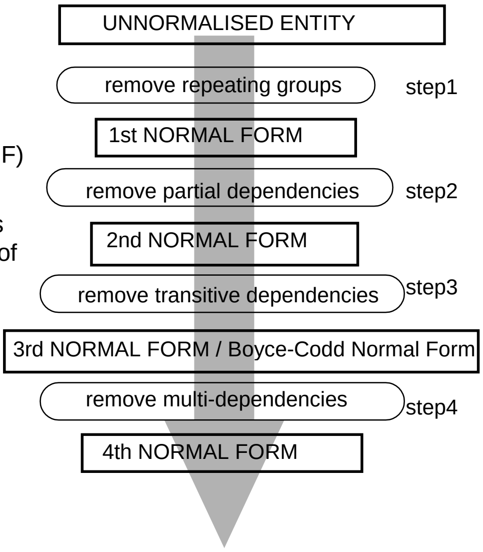
- Normalization is the process of simplifying the relationship between data elements in a record.
- The purpose of normalization is to produce a stable set of relations.
- To achieve a design that is highly flexible.
- To increase the quality of database design.

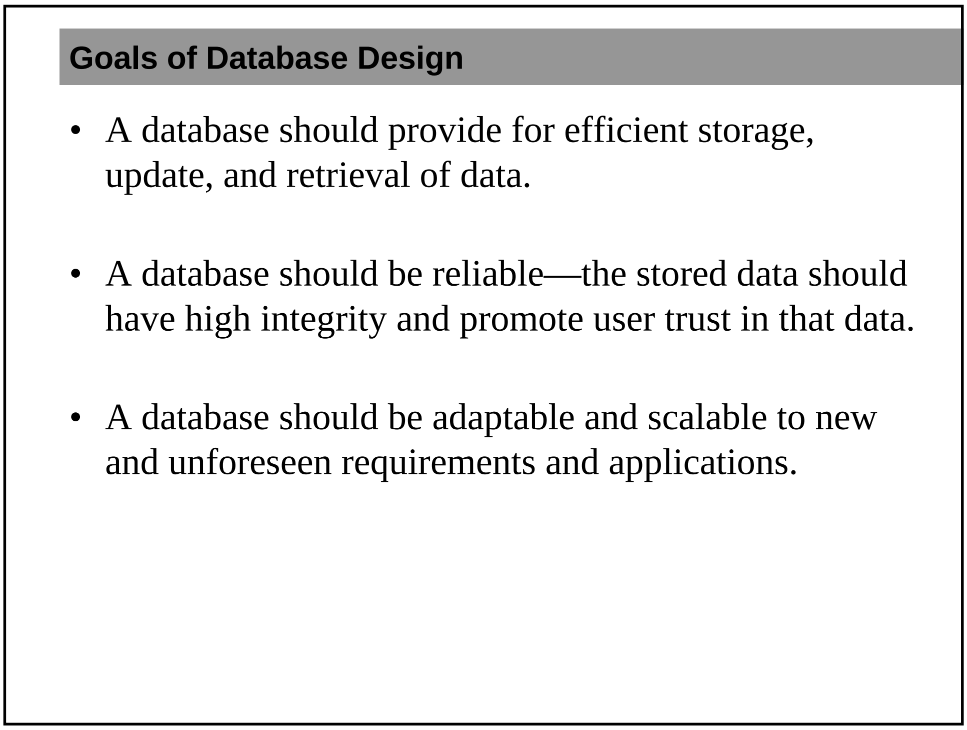
Data Normalization

- An logical entity (or physical table) is in **first normal form** if there are no attributes (fields) that can have more than one value for a single instance (record).
- An logical entity (or physical table) is in **second normal form** if it is already in first normal form and if the values of all nonprimary key attributes are dependent on the full primary key.
- An logical entity (or physical table) is in **third normal form** if it is already in second normal form and if the values of all nonprimary key attributes are not dependent on other nonprimary key attributes .

Normalization Process

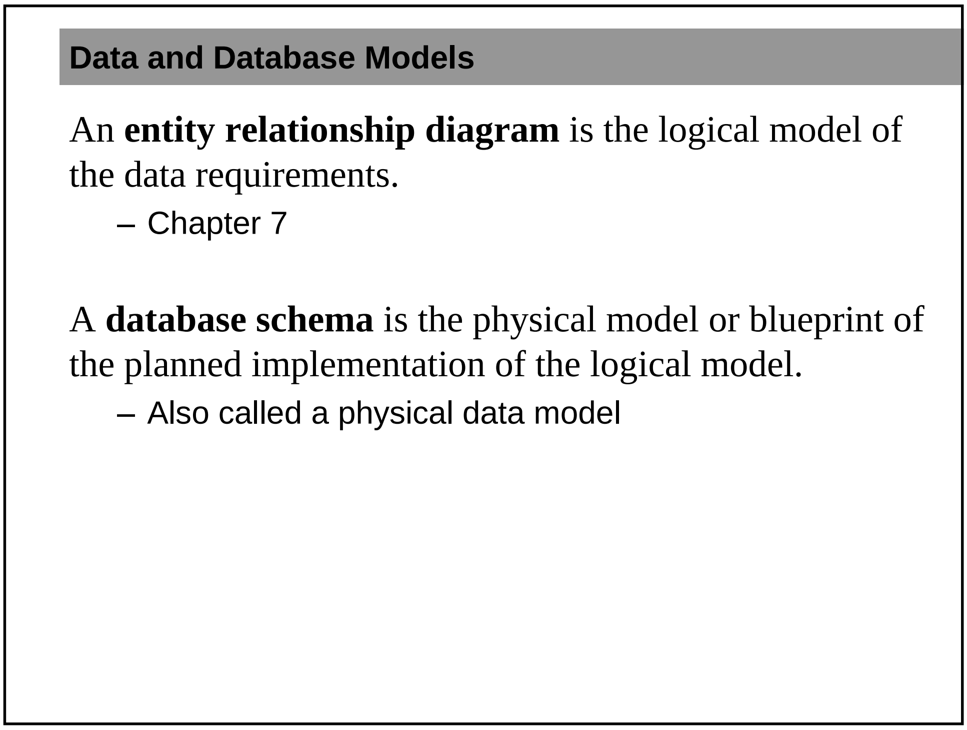
- Usually four steps giving rise to
  - First Normal Form (1NF)
  - Second Normal Form (2NF)
  - Third Normal Form (3NF)
  - Boyce-Codd Normal Form (BCNF)
  - Fourth Normal Form (4NF)
- At each step we consider relationships between the functional dependencies of a relation's attributes
- Normalisation is a:
  - framework
  - series of tests


UNNORMALISED ENTITY
remove repeating groups
step1
1st NORMAL FORM
remove partial dependencies
step2
2nd NORMAL FORM
remove transitive dependencies
step3
3rd NORMAL FORM / Boyce-Codd Normal Form
remove multi-dependencies
step4
4th NORMAL FORM



Goals of Database Design

- A database should provide for efficient storage, update, and retrieval of data.
- A database should be reliable—the stored data should have high integrity and promote user trust in that data.
- A database should be adaptable and scalable to new and unforeseen requirements and applications.



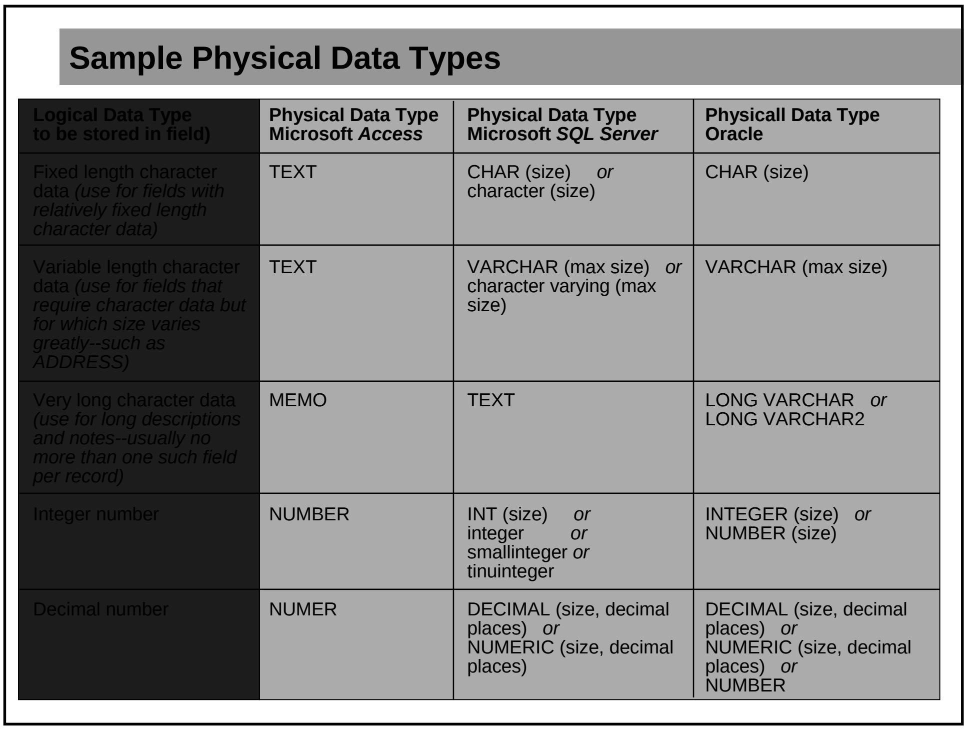
Data and Database Models

An **entity relationship diagram** is the logical model of the data requirements.

- Chapter 7

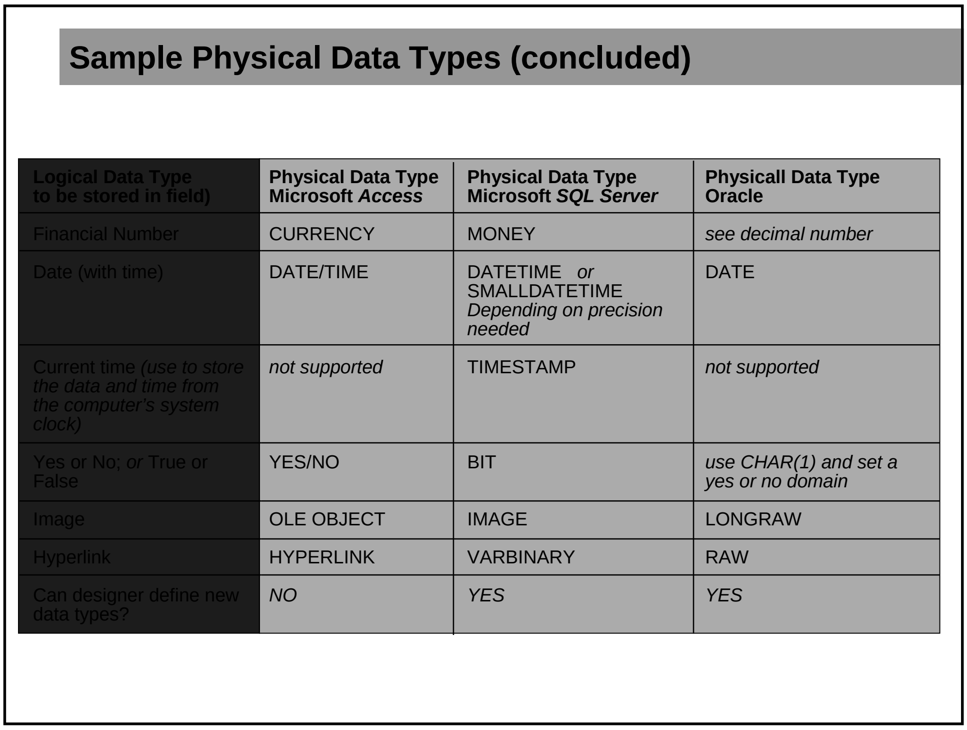
A **database schema** is the physical model or blueprint of the planned implementation of the logical model.

- Also called a physical data model



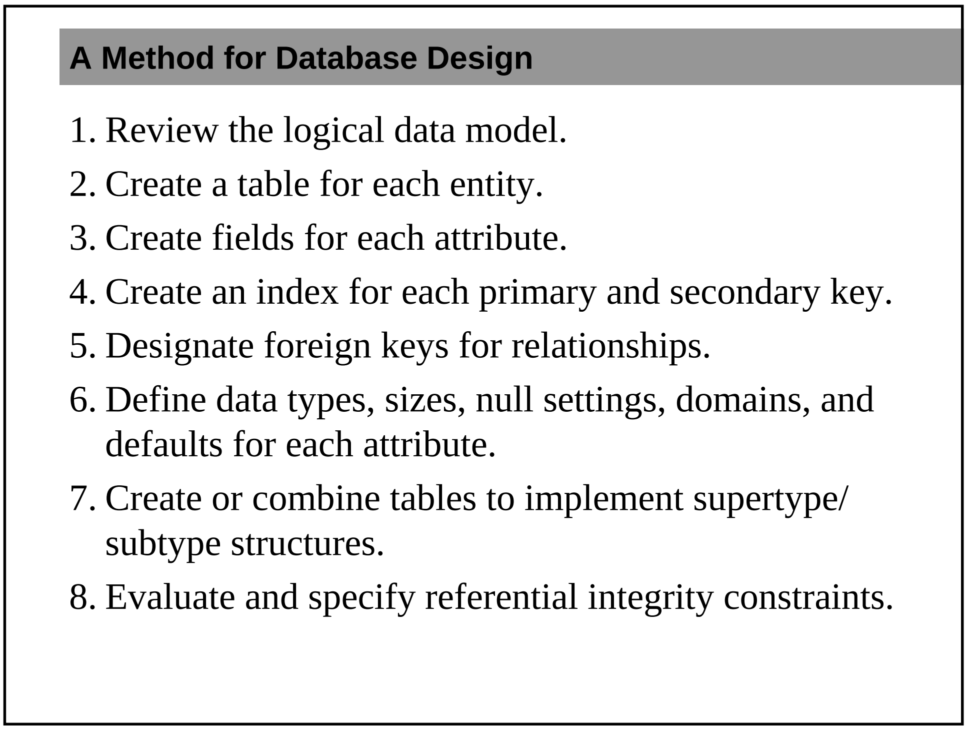
Sample Physical Data Types

Logical Data Type to be stored in field)	Physical Data Type Microsoft *Access*	Physical Data Type Microsoft *SQL Server*	Physicall Data Type Oracle
Fixed length character data *(use for fields with relatively fixed length character data)*	TEXT	CHAR (size) *or* character (size)	CHAR (size)
Variable length character data *(use for fields that require character data but for which size varies greatly--such as ADDRESS)*	TEXT	VARCHAR (max size) *or* character varying (max size)	VARCHAR (max size)
Very long character data *(use for long descriptions and notes--usually no more than one such field per record)*	MEMO	TEXT	LONG VARCHAR *or* LONG VARCHAR2
Integer number	NUMBER	INT (size) *or* integer *or* smallinteger *or* tinuinteger	INTEGER (size) *or* NUMBER (size)
Decimal number	NUMER	DECIMAL (size, decimal places) *or* NUMERIC (size, decimal places)	DECIMAL (size, decimal places) *or* NUMERIC (size, decimal places) *or* NUMBER



Sample Physical Data Types (concluded)

Logical Data Type to be stored in field)	Physical Data Type Microsoft *Access*	Physical Data Type Microsoft *SQL Server*	Physicall Data Type Oracle
Financial Number	CURRENCY	MONEY	*see decimal number*
Date (with time)	DATE/TIME	DATETIME *or* SMALLDATETIME *Depending on precision needed*	DATE
Current time *(use to store the data and time from the computer's system clock)*	*not supported*	TIMESTAMP	*not supported*
Yes or No; *or* True or False	YES/NO	BIT	*use CHAR(1) and set a yes or no domain*
Image	OLE OBJECT	IMAGE	LONGRAW
Hyperlink	HYPERLINK	VARBINARY	RAW
Can designer define new data types?	*NO*	*YES*	*YES*



A Method for Database Design

1. Review the logical data model.
2. Create a table for each entity.
3. Create fields for each attribute.
4. Create an index for each primary and secondary key.
5. Designate foreign keys for relationships.
6. Define data types, sizes, null settings, domains, and defaults for each attribute.
7. Create or combine tables to implement supertype/ subtype structures.
8. Evaluate and specify referential integrity constraints.