**+كان الرب مع يوسف فكان رجلا ناجحا+**

# What is Meant by Data and Information?

* Data

1. Raw facts
2. No context

* Information

1. Data with context
2. Processed data.

# Examples on DB applications

* Hospital system
* Business clients
* Car registration
* Airline reservation
* Supermarket
* Hotel reservation

# File Systems Problems

1. Duplication

* same data may be stored in multiple files.

1. Inconsistency

* same data may be stored by different names in different format.

1. Rigidity

* requires customized programming to implement any changes.
* cannot do ad-hoc queries.

1. Implications

* Waste of space
* Data inaccuracies
* High overhead of data manipulation and maintenance

# DB Systems

1. It answers queries fast.
2. Queries from multiple users can execute concurrently.
3. without affecting each other.
4. It recovers from crash.

* No corrupted data after restart

Instance: actual data.

A database schema **defines how data is organized within a relational database. (Description of database)**

Queries: **is the question send to Database.**

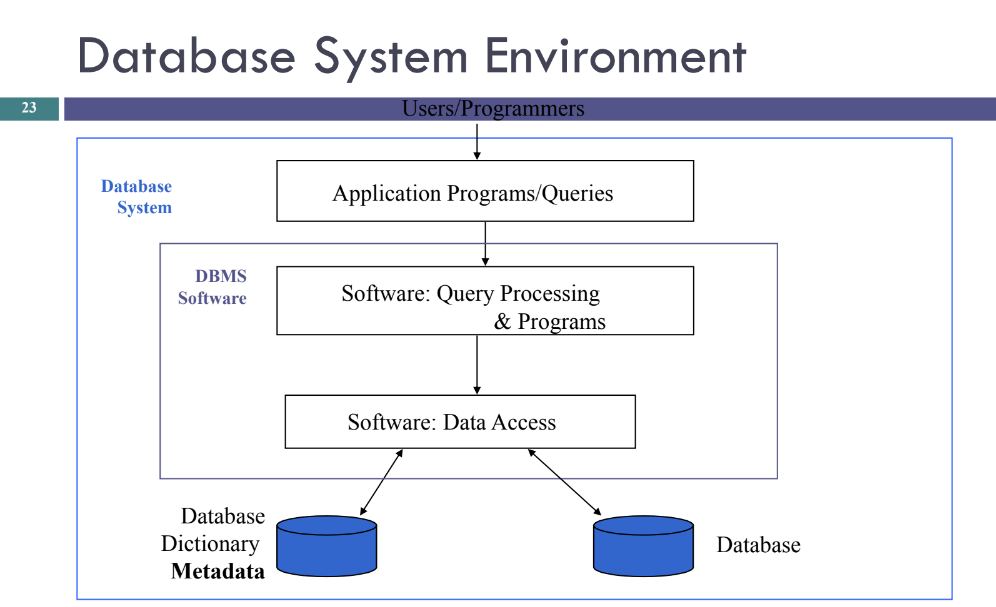
|  |  |
| --- | --- |
| File System | Databases |
| * Small Systems * Often single user * Simple structure * Isolated data * Redundant data * Relatively cheap * Less secure | * Large systems * Multiple users * Complex structure * Shared data * Reduced redundancy. * Relatively expensive * More secure using views |

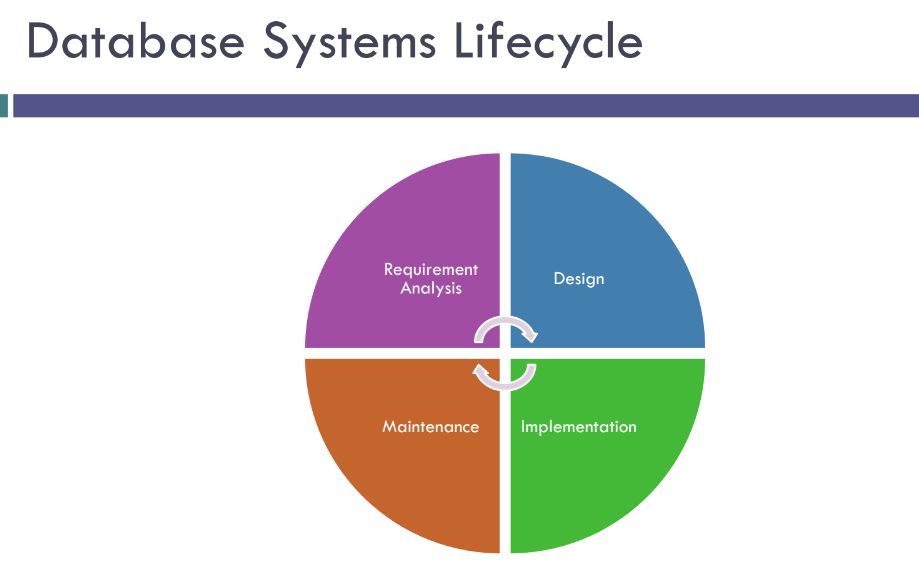
# Definitions

* Database: Collection of related data. A database contains a model of something!
* A Database Management System (DBMS): is a software system designed to store, manage, and facilitate access to the database.
* Database/Application Program: A computer program that interacts with the database through the DBMS.

# Components of a Database System

* Query Engine
* Query optimizer
* Storage management
* Transaction management





# Database Players

1. DB Administrator DBA

* Access authorization, coordination & monitoring database usage, problem determination performance tuning etc.

1. Designers

* choose the appropriate structures to represent & store the data.

1. System analysts & application programmers
2. Users

# Database States

Empty State: DB is empty when we first define the DB schema. (Schema مخطط)

Initial State: DB is first populated or loaded with data.

Current State: snapshot in time

# Example of Relational DB Management System Products

* Oracle
* Sybase
* Informix (Unix)
* DB2, SQL/DS (IBM)
* Access, SQL Server (Microsoft)

# Database Models

* A Database model is a collection of concepts that can be used to define the DB structure (data items, types, relationships, operations, behaviors and constrains)

# Database Models

 Relational model.

 Hierarchical & Network model.

 Object-oriented models.

 Distributed model.

 NoSQL models  Document model.

# Levels of Abstraction

1. Views

* define how users see the data (use of data)

1. Conceptual schema

* defines logical structure (meaning of data)

1. Physical schema

* describes the files and indexes used (storage of data)

# Three-Schema Architecture

## Defines DBMS schemas at three levels:

1. Internal schema is used to describe physical storage structures and access paths (e.g indexes).

◼Typically uses a physical data model.

1. Conceptual schema is used to describe the structure and constraints for the whole database for a community of users.
2. External schemas are used to describe the various user views.

Data Models: A collection of tools for describing 1- data 2-data relationships 3-data constraints.

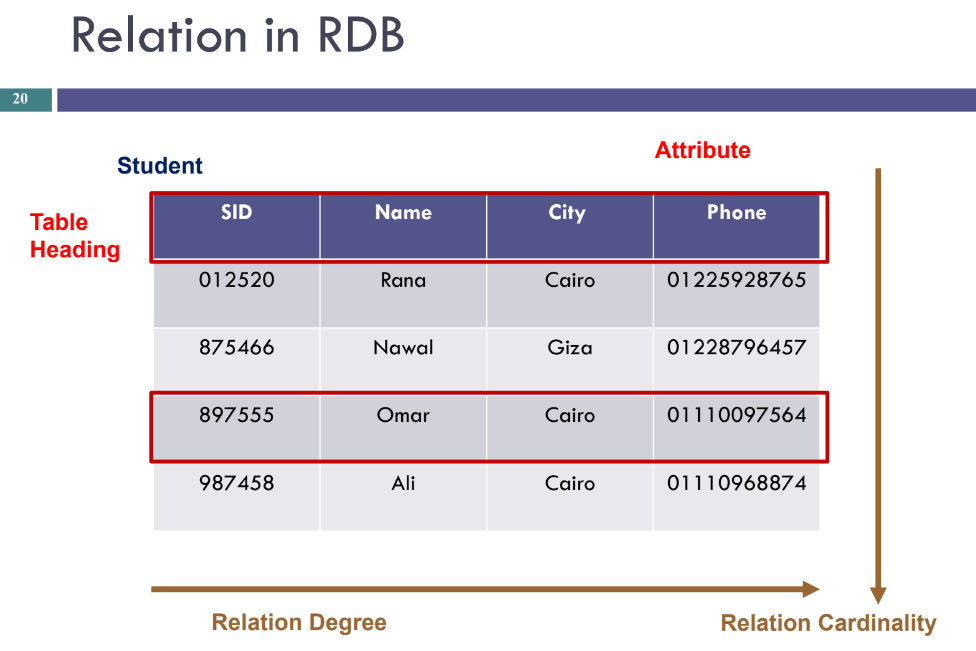
# Relational Model

## History of Relational Model

* Introduced by Ted Codd in 1970 in a classic paper. Ted Codd was an IBM Researcher Many database concepts & products based on this model.

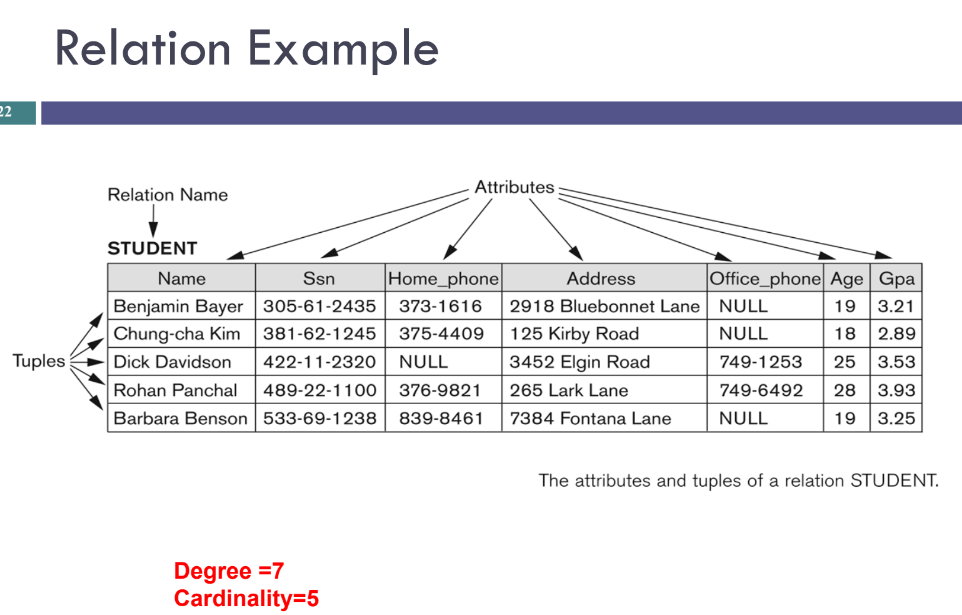
# Relations

* A relational database is a set of relations.
* Relations are basically tables of data.
* Each row represents a record in the relation.
* Each relation has a unique name in the database.
* Each row in the table specifies a relationship between the values in that row.



# Definition Summary

|  |  |
| --- | --- |
| Informal Terms | Formal Terms |
| Table | Relation |
| Column Header | Attribute |
| All possible Column Values | Domain |
| Row | Tuple |
| Table Definition | Schema of a Relation |
| Populated Table | State of the Relation |



# Relations and Attributes

* Each relation has some number of attributes Sometimes called “columns”.
* Each attribute has a domain specifies the set of valid values for the attribute.

A domain D is a set (pool) of values, from which one or more attributes takes their values.

# Tuples and Attributes

* Each row is called a tuple (A fixed-size, ordered set of name-value pairs)
* Each attribute in the tuple has a unique name.
* A relation is a set of tuples.
* Each tuple appears exactly once.
* The order of tuples in a relation is not relevant.

# Schema VS Instance

* the name of the relation and the set of attributes is called the schema.
* the current values contained in the relation represent an instance.

# Relation Schemas

## Every relation has a schema.

## A relation schema includes:

* + - an ordered set of attributes
    - the domain of each attribute

## Database schema is a collection of relation schemas.

# Data Model

A collection of concepts that can be used to define the DB structure.

◼Data items & types ◼Relationships ◼Constraints

# Characteristics of Relations

▪ Tuples have no particular order.

▪ Ordering of attributes not important

▪ All values belonging to a particular attribute are from the same domain

▪ Attributes are atomic. (Cannot put multiple values in the same cell)

▪ Attributes may have a null value (Null is not a zero and is not an empty string)

# Types of Constraints

▪ Domain Constraints

▪ Key Constraints

▪ Integrity Constraints

* Entity Integrity Constraint
* Referential Integrity Constraint
* Semantic Integrity Constraint

# Domain Constraints

The value of each attribute, A, must be an atomic value from the domain of A.

# Key Constraints

* Value of a key uniquely identifies a tuple in a relation.
* Super key (K): is a subset of attributes of R that can identify a tuple. It might contain more than one attribute.

▪ no 2 tuples have same values for K

* A key is a minimal superkey; a superkey from which we cannot remove any attributes and still be able to uniquely identify tuples in a relation.
* A relational schema may have more than one key:

▪ Each key is called a candidate key

▪ One designated as the primary key

Composite Primary key, that is, primary key that is made up of more than one field.

# Integrity Constraints

* Integrity constraints are specified on a schema and hold for every instance of the schema.
* Entity integrity constraint

◼ A primary key value cannot be null.

* Referential integrity constraint

# Foreign Key

▪ A Foreign key is how we relate relations to each other

▪ A Foreign key in R is a set of attributes FK in R such that FK is a primary key of some other relation R

▪ A foreign key is used to specify a referential integrity constraint.

# 3-Integrity Constraints

* Integrity constraints are specified on a schema and hold for every instance of the schema.
* **Entity integrity constraint**

Primary key:

* Unique
* Not null
* Stable and minimum # of attributes
* **Referential integrity constraint**

Foreign key:

◼ Match a primary key ◼ Null

# Operations on Relations

▪ Operations include insert, delete, modify (update), and retrieval.

▪ Some operations can violate database constraints.

# SQL

* Structured Query language SQL is pronounced as “S-Q-L” or sometimes as “See-Quel” which is the standard language for dealing with Relational Databases.
* It is effectively used to create, insert, search, update, delete, modify database records.
* SQL commands are case insensitive:
  + Same: SELECT, Select, select
  + Same: Product, product
* Use single quotes for constants:
  + ‘abc’ – yes
  + “abc” – no

# SQL is a...

Data Definition Language (DDL)

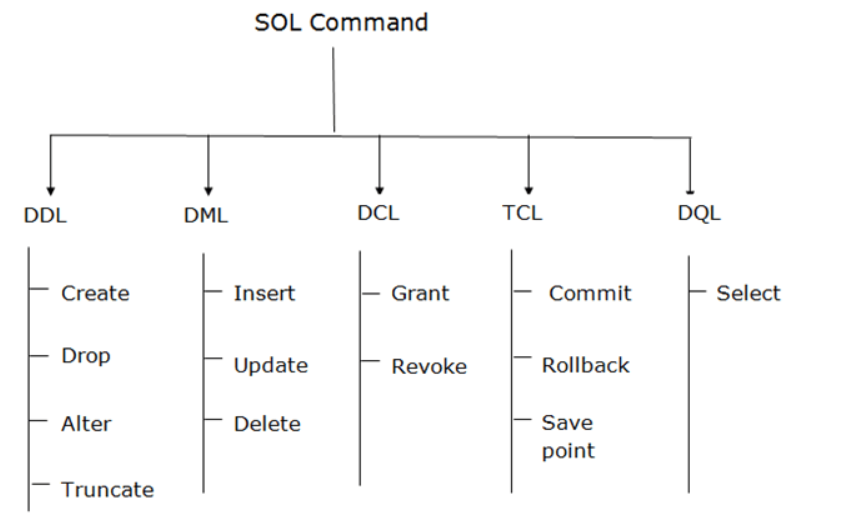
* + Define relational schemata.
  + Create/alter/delete tables and their attributes.

Data Manipulation Language (DML)

* + Insert/delete/modify tuples in tables.
  + Query one or more table

Data Control Language (DCL)

* + Specify user permissions.
  + Grant/revoke



# DDL: Data Definition Language:

It is used to create and modify the structure of database objects in database. Includes statements like:

▪ **CREATE:** Creates a new table or any other database object.

▪ **ALTER:** Modifies an existing database object, such as a table

▪ **DROP:** Removes an entire table or any other object in database

## Creating a new database:

CREATE DATABASE databasename; (Example: create database customers;)

# Data Types for Attributes

1**- Numeric:** integer number (INTEGER, INT, AND SMALLINT), and floating number (FLOAT, REAL, and DOUBLE).

2- **Character:** data types are either fixed length (CHAR (n), where n is the number of character) or variable length (VARCHAR(n)).

3- **Boolean**: TRUE or FALSE.

4- **Timestamp**

Create Table Syntax

CREATE TABLE <Table Name> (

<column1 datatype constraint\_1 constraint\_2 >,

<column2 datatype>,

.....

constraint\_3,

Constraint\_4);

# Constraints

• **NOT NULL:** A Constraint that ensures that a column cannot have NULL value.

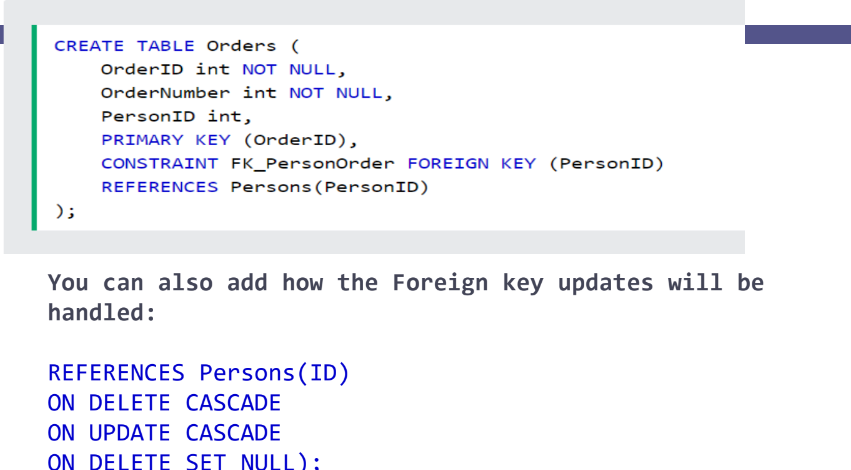
• **DEFAULT:** A Constraint that provides a default value for a column when none is specified.

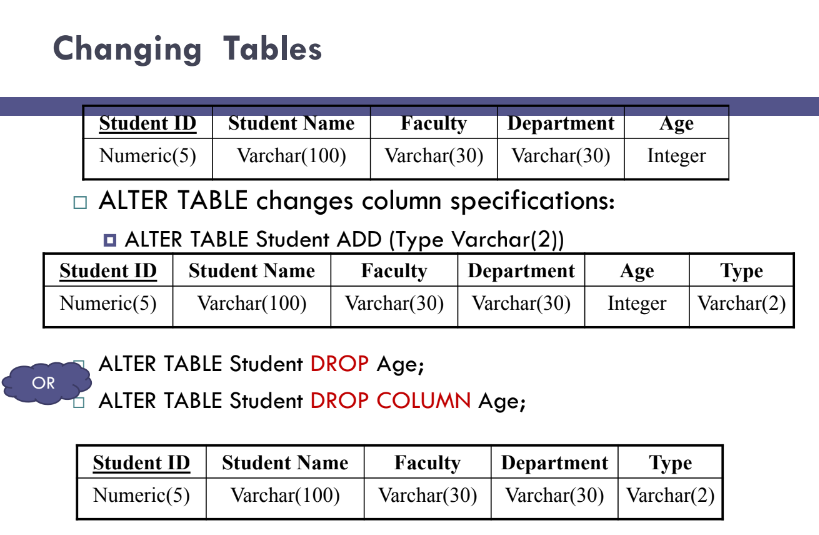
• **UNIQUE:** A Constraint that ensures that all values in a column are different.

• **PRIMARY Key:** A Constraint that uniquely identify each row/record in a database table (NOT NULL + UNIQUE)

• **FOREIGN KEY (FK):** A Constraint that ensures referential integrity. A foreign key from 1 table to another is used link a tuple in the 1st table to a unique tuple in the 2nd table.

• **CHECK:** A constraint that ensures that all values in a column satisfy a certain condition.





# ALTER TABLE: Constraints

Adding and dropping Constraints

* ALTER TABLE<table name>
  + ADD CONSTRAINT < CONSTRAINT Syntax>
* ALTER TABLE<table name>
  + ADD CONSTRAINT < CONSTRAINT Name > < CONSTRAINT Syntax>
* ALTER TABLE<table name>
  + Drop CONSTRAINT < CONSTRAINT Name >

## Examples:

ALTER TABLE Customers

ADD PRIMARY KEY (SSN, Telephone);

Or

ALTER TABLE Registered

ADD CONSTRAINT PK1 PRIMARY KEY (SSN, Telephone);

ALTER TABLE Persons

ALTER COLUMN City DROP DEFAULT;

# SQL DROP: For deleting Tables

* DROP TABLE statement allows you to remove tables from your schema:
  + DROP TABLE TableName;
* DROP Database statement allows you to delete all the database:
  + DROP database <database name>;