Data Base System:

Chapter 1:

* Why we study data bases?

• Databases are useful

↳ Many computing applications deal with large amounts of information.

↳ Data base system give a tools for storing, searching and managing this information.

⇒Databases in CS:

→ Databases are a core topic in computer science

→ Basic concepts and skills with database systems are part of skill set you will be assumed to have as a CS graduate.

What is Database?

organized collection of data which is stored in a such way that different operations can be performed

→ A set of information held in a computer.

→ A collection of data arranged for ease and speed of search and retrieval

operations
→ search
→ retrieval
→ update
→ select

Applications of Databases

→ Library Catalogues
→ Bank accounts
→ Credit card detail
→ Medical records
→ Airline booking
→ Student record

→ Customer histories → Telephone directories

⇒ Data base system consists of:

→ Data (The database)
→ Software
→ hardware
→ Users

Data base Application

⇒ Data bases allow users to:

→ Store
→ update
→ Retrieve (recovery / back up)
→ Organize
→ protect

Users of Database

(1)- End users:

Use database system to achieve some goals

(2)- Application developer

Write software to allow end users to interface with the database system

(3)- Data base Administrator (DBA):

Design & manages the data base system.

(4). Data base systems programmer:

Writes the database software itself.

Data Base Managment System (DBMS)

A data base management system is the software that controls the information also manages and controls access to the database.

Example:

→ Oracle → DB2 (IBM) → MS SQL Server
→ MS Access → Ingres → PostgresQL
→ My SQL

What the DBMS does?

Provides users with

→ DDL (Data Defination language)
→ DML (Data manipulation language)
→ DCL (Data control language)

DBMS Provides
↳ persistence
↳ concurrency
↳ integrity
↳ security
↳ Data independence

Data Dictionary - Meta Data => The data about data.
↓
Describes the database itself

→The Dictionary holds:
- Descriptions of database objects (tables, users, rules, views, indexes, ...)

File Based Systems

(A collection of application programs that perform services for end user such as, production of reports)

In File Based systems, Data is stored in files. Each file has a specific format. Programs that use these files depend on Knowledge about that format.

Disadvantages:- (Limitations
↳ No standards
↳ Data Duplication
↳ Data Dependence (
↳ No way to generate queries
↳ No provision for security, recovery, concurrency etc

Parallel accesses

same data may be stored by different names in different formats.

→ Data Redundancy and inconsistency
duplication of information in different files
→ Difficulty in accessing data
→ Data Isolation
→ Integrity problem → Data Integrity
correction of data
→ Security problems

Why Database?

→ Redundancy can be reduced.
→ Inconsistency can be avoided.
→ The data can be shared.
→ Standards can be enforced.
→ Security restrictions can be applied.
→ Integrity can be maintained.
→ provision of Data independence.

File system

(1) Manual file system

→ to keep track of Data
→ used tagged file folders in a filing cabnet.
→ Organized according to expected use.
e.g: file per customer.

→ easy to create, but hard to
→ locate data
→ aggregate and summarize data.

(2)- Computerized file system:
→ to accomodate the data growth and information need. structure
→ manual file system/ were duplicated in the computer.
→ Data processing (DP) specialists wrote customized programs to
→ Write, delete, update data (i.e; Data Managment)
→ extracts and present data in various formats (i.e; report)

Charatterstic of good DataBase

Data integrity → ensuring data is valid.
Data independence → Data is seperated from software
Avoiding data redundancy → repitition of data is avoided.
Data security → Data is not accessible to unauthorized users.
Data Maintaine → set procedures for adding, deleting, records for the purpose of optimization

DataBase Languages

A Data sub-languages consists of two parts:

→ Data Defination Language DDL

→ Data Manupulation Language DML

(1)- DDL :

organization of Data

DDL used to define a schema & to modify an existing ones.

"A language that allows the DBA or user to describe and name the entities, attributes, and relationships required for the application together with any associated integrity and security constraints."

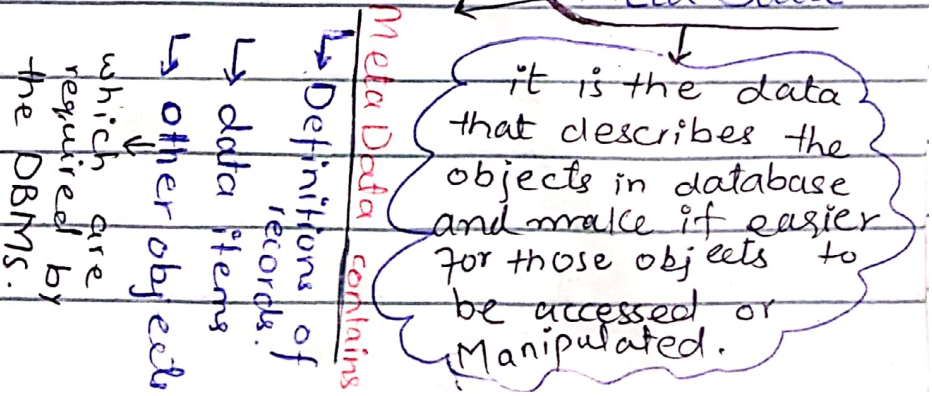
It cannot be used to manipulate the data.

The result of compilation of DDL statment is a set of "tables" stored in special files collectively called "System Catalog".

↓

integrates the

Meta Data



Two terms which is used to describe System catalog are :

- Data Dictionary → "collection of names, definations, attributes about data elements that are being used or captured in database"
- Data Directory → "The purpose of data directory is to identify the master source of specific data."

(2) DML : → procedural/programmable Logic → What to process? How to process?

what to process?

Non-procedural language

DML is used to both read and update the database.

"A language that provides a set of operations to support the basic Data manipulation operations on the data held in the database"

Operations of DML :

→ insertion of new data into database
→ modification of data store in database
→ selsieel of data
↓
selection
→ Deletion of Data

Query Language

The part of DML that involves data retrieval

Procedural DMLs:

"A language that allows the user to tell the system what data is needed and exactly how to retrieve the data."

Non-procedural DMLs: → Declarative languages.

"A language that allows the user to state what data is needed rather than how it is to be retrieved."

Fourth-Generation Languages (4GLs)

↳ It is shorthand programming Language.

→ Fourth-generation languages encompass: ↳ features

- presentation languages such as query language and report language.
- Specialty languages such as spreadsheets and database language.
- Application generators that define, insert, update and retrieve data from the data base to build applications.
- very high-level languages that are used to generate application code.

Examples of 4GLs:

→ SQL

→ QBE

Forms Generators

A forms generator is an interactive facility for rapidly creating data input and display layouts for screen forms. The form generator allows the user to define what the screen is to look like, what information is to be displayed and where on the screen it is to be displayed.

It may also allow the definition of colors for screen elements and other characteristics such as bold, underline, blinking, reverse video and so on.

Report Generators

A report generator is a facility for creating reports from data stored in the database. It is similar to query language in that it allows the user to ask questions of the database and retrieve data.

Two main types of report generators
↳ Language-oriented
↳ Visually oriented

Graphics Generators

It is a facility to retrieve data from the database and display the data as a graph showing trends and relationships in the data.
it allows the user to create bar charts, pie-charts, line charts, scatter charts and so, on.

Application Generators

↓
it is a facility for producing a program that interfaces with the database.

The use of an application generator can reduce the time it takes to design an online software application.

Functions of a DBMS

→ Data storage, retrieval and update.
→ A User-accessible catalog.
→ Transaction support
→ Concurrency control services
→ Recovery services
→ Authorization services
→ Integrity services
→ Services to promote data independence
→ Utility services

The Three-level ANSI-SPARC Architecture :

Objectives of Three-level architecture :

- All users should be able to access same data.
- A user's view is protected to changes made in other views.
- Users should not need to know physical database storage details.

- DBA should be able to change database storage structures without affecting the users views.
- Internal structure of database should be unaffected by changes to physical aspects of storage.
- DBA should be able to change conceptual structure of database without affecting all users.

Daigram 2.1 → from book
↓
pg # 85 => (Theory from book)

① External level

↳ user's view database
↳ describes the part of database that is relevant to a particular user.

② conceptual level

↳ community view of the database.
↳ Describes what data is stored in database and relationships among the data.

(3) Internal level

→ Physical representation of the database on the computer.

→ Describe how the data is stored in the database.

There are two kinds of inter-level mappings :

→ external/conceptual mapping

→ conceptual/internal mapping

Data Independence

→ ↓Logical data independence:

"Immunity of external schemas to change in conceptual schema"

→ conceptual schema changes

→ should not require changes to external schema or rewrites of application programs.

External schema is independent to conceptual schema.

→ physical Data independence:-

↓

immunity of conceptual schema to changes in the internal schema.

→ Internal schema changes e.g: File organizations storage devices

→ Should not require change to conceptual or external schemas.

↳ external schema is independent to internal schema.

chapter no. 4:

The Relational Model

In relational model, all data is logically structured within relations (tables) and is made up of named attributes (column) of data. Each tuple (row) contains one value per attribute.

RELATION

A relation is a table with columns and rows

Attribute

An attribute is a named column of a relation

Attribute types

- Each attribute of a relation has a name.
- The set of allowed values for each attribute is called Domain of the attribute
- Attribute values are required to be atomic; that is, indivisible.

  e.g.: The value of an attribute can be account number, but cannot be a set of account number.
- Domain is said to be atomic if all its members are atomic → single unit
- The special value "Null" is a member of every domain.
- The null values causes complications in the definition of many operations

Tuple → A tuple is a row of relation.

Degree → The degree of a relation is the number of attribute it contains.

Cardinality → The cardinality of a relation is the number of tuples it contains.

Relational Schema

A named relation defined by a set of attributes and domain name pairs.

Formally, given domains $D_1, D_2 \ldots D_n$ a relation r is a subset of of

$D_1 \times D_2 \times \ldots \times D_n$

Thus, a relation is set of n-tuples $(a_1, a_2, \ldots, a_3)$ where each $a_i \in D_i$

- Schema of a relation consists of:
  - ⇒) attribute definitions
    - → name
    - → type / domain
  - Integrity Constraints

↓

"The set of pre-defined rules responsible for maintaing the quality and consistency of data in database"

Integrity constraints has 4 types:

- Domain Constraint
- Entity constraint
- Referential Integrity constraint
- Key constraint

Relation instance

→ The current values (relation instance) of a relation are specified by a table.

→ An element is a tuple, represented by a row in a table.

("tuple may be stored in an arbitary order.")

Relational data base
↓
Schema

→ set of relation Schemas, each with a distinct name.

(properties of relation same as properties of "attribute") ⇒ previous page

Relation Keys

We need to be able to identify one or more attributes called "Relational keys" that uniquely identifies each tuple in a relation.

Super Key

An attribute or set of attributes that uniquely identifies a tuple within a relation.

A Super key may contain additional attributes that are not necessary for unique identification, and we are interested in identifying super keys that contain only the minimum no. of attributes necessary for unique identificaton.

Candidate Key

↓

A Super key such that no proper subset is a super key within relation.

A candidate key K for a relation R has Two properties

- Uniqueness: In each tuple of R, The values of K uniquely identify that tuple
- Irreducibility: No proper subset of K has the uniqueness property.

There may be several candidate keys for a relation. When a key consist of two or more attribute. we call it, Composite key.

Primary Key:-

The candidate key that is selected to identify tuples uniquely within the relation.

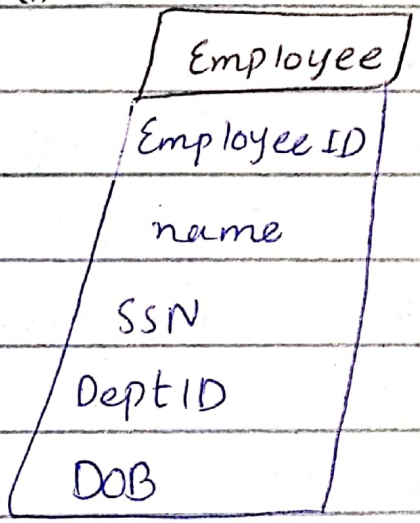
Because a relation has no duplicate tuples, it is always possible to identify each row uniquely.

"A relation always has a primary key"

The candidate keys that are not selected to be primary key are called Alternative key.

e.g For Branch relation, if we choose branch No. as primary key, postcode will be alternative key.

(1)



- No two rows has can have the same primary key value.
- Every row must have a primary key value.
- The primary key field cannot be null.
- Value in primary key column can never be modified or updated, if any foriegn key refers to that primary key.

Foriegn Key

An attribute or set of attributes within one relation that matches the candide key of some relation.

When an attribute appears in more than one relation, its appearance usually represents a relationship between tupks of two relations.

4-2-6

⇒ Representing Relational Schemas

book Pg # 161