

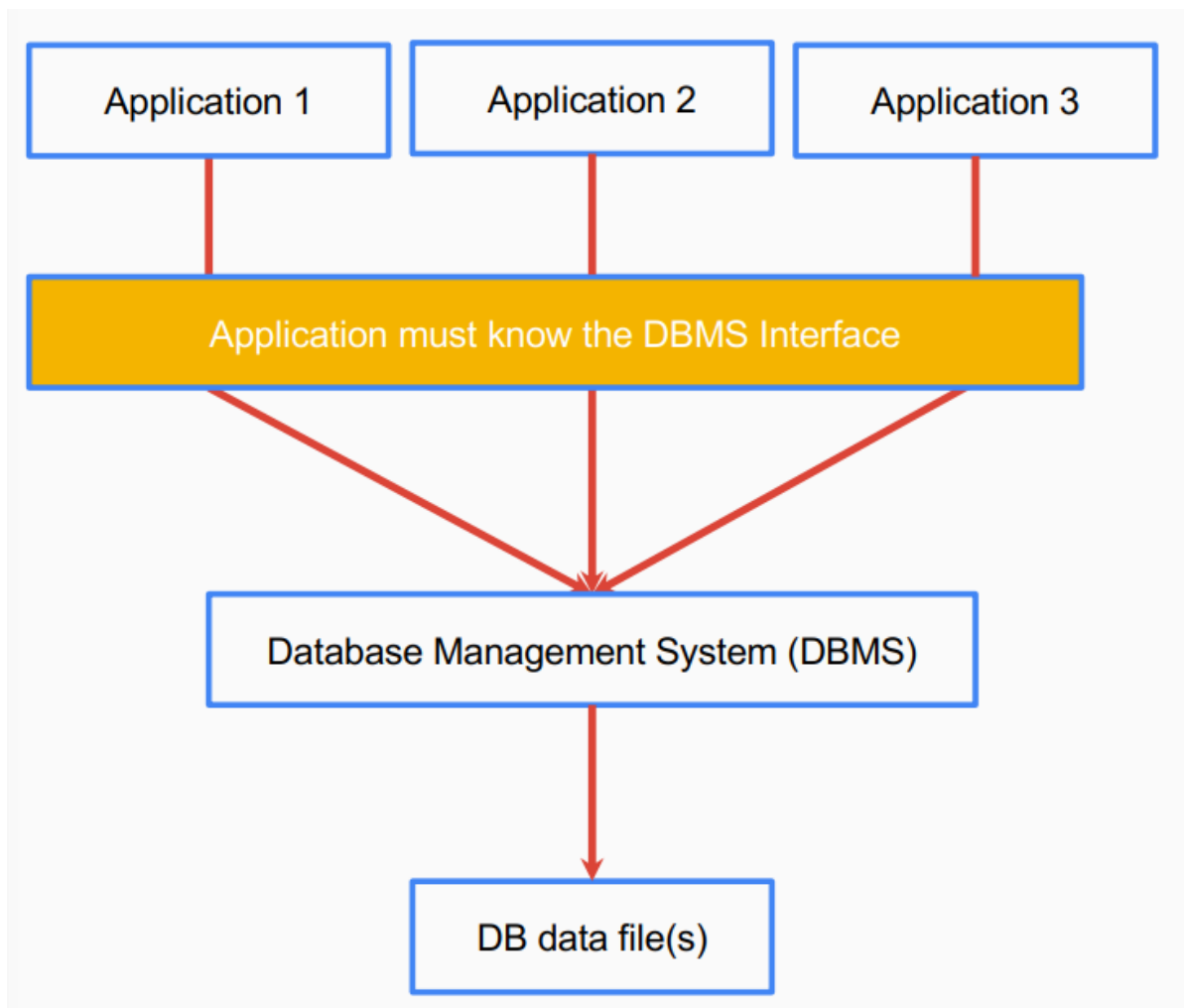
Week2 Lecture01 Introduction to Relational Databases

what is a database?

- A collection of data
- A structured set of data
- using to update and select data

CRUD :)

structure



SQL(Structured query language) - > data definition, manipulation, query

DMBS

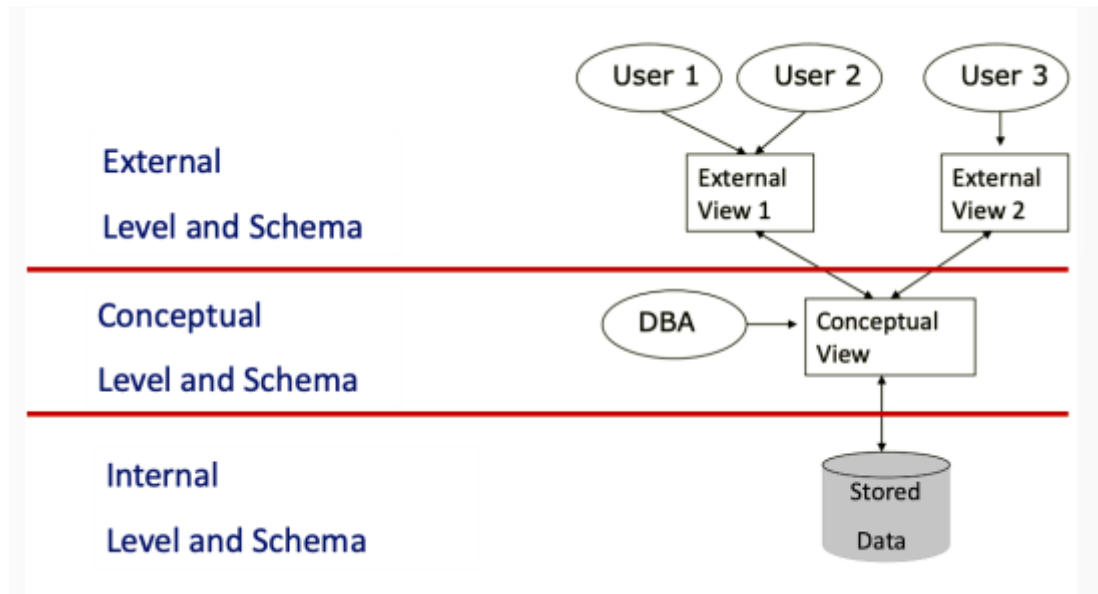
- sql
- sqlite
- DB2
- Oracle

ANSI/SPARC Model

American National Standards Institute

Standards Planning Requirements
Committee

three layers of Database System



Internal Level

- Defines how data is stored in the database (storage space allocation, data structures, indexing, data compression, encryption, etc.)
- Used by System designers

Conceptual (Middle) Level

- Defines what data is stored in the database and the relationships between data (e.g., table definitions, constraints on the data, security and integrity information)
- Deals with the organization of the entire database content
- Database Administrator

External

- Defines the user's view of the database (the part of the database that is relevant to each user)
- Based on the user's interests,
 - E.g., YY/MM/DD, DD/MM/YYYY
- Used by users and applications programmers

Relational Model

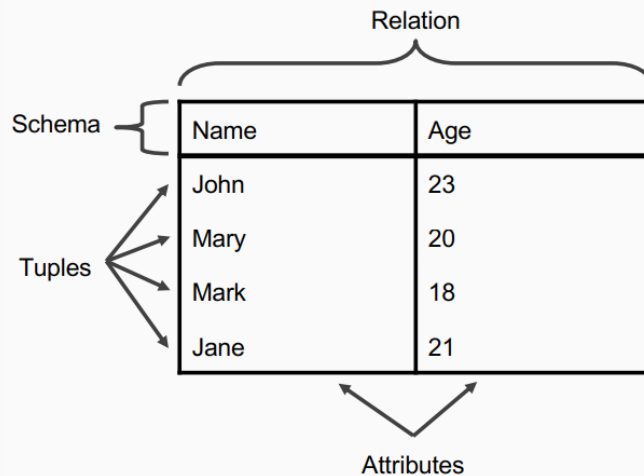
Relational Data Structure

Data is stored in relations which are tables with columns and rows

(a relation is a set of tuples)

Relational Data Structure

- Each relation has a **schema**
 - sometimes called
 - scheme
 - heading
- The **schema** defines the relation's attributes (columns).
- Each column has a **domain** - a set from which all possible values can come.



domain of name->varchar(20)

Named and Unnamed Tuples

named:

{ (Name, John), (Age, 23) }

unnamed:

(John, 23) (equivalent to the above)

There is no real difference between named and unnamed tuples, but be careful with the ordering of unnamed tuples

Tuples of the graph (named schema)

{ { (Name, John), (Age, 23) },
 { (Name, Mary), (Age, 20) },
 { (Name, Mark), (Age, 18) },
 { (Name, Jane), (Age, 21) } }

Degree and Cardinality

Degree of a relation: how long is the tuple

Cardinality of a relation: how many tuples exists

example:

Attributes are: ID, Name, Salary & Department

The degree of the relation is 4

ID	Name	Salary	Department
M139	John Smith	18000	Marketing
M140	Mary Jones	22000	Marketing
A368	Jane Brown	22000	Accounts
P222	Mark Brown	24000	Personnel
A367	David Jones	20000	Accounts

Schema is { ID, Name, Salary, Department }

Tuples, e.g.
 { (ID, A368),
 (Name, Jane Brown),
 (Salary, 22,000),
 (Department, Accounts) }

The cardinality of the relation is 5

Properties of Relations

- Each cell contains exactly one single value.
- Each attribute has a distinct name.
- The values of an attribute are all from the same domain.
- Each tuple is distinct, i.e., no duplicate tuples.
- The order of attributes has no significance.
- The order of tuples has no significance either

Candidate Keys

A candidate key is a single field or the least combination of fields that uniquely identifies each record in the table

- Every tuple has a unique value for that set of attributes: **uniqueness**
- No proper subset of the set has the uniqueness property: **minimality (Superkey)**

超键：在关系中能唯一标识元组的属性集称为关系模式的超键。注定义中的“属性集”，超键可以是一个很大的集合，只要他能确定是哪一行就行，因此'id','user','pwd','section','name'都可以是超键的集。

候选键：不含有多余属性的超键，比如在上面的超键中，'id'自己就可以独自确定是哪一行，所以他自己可以是一个候选键，除去它以外的另外四个也可以是候选键，但是这五个放在一起因为有了多余的列，他们就不是候选键。（另外四个可以是候选键的原因是每一列都有可能有重复的内容）

主键：在所有的候选键里面找一个作为主键供使用，也就是说可以是id，也可以是另外四个的合体，也有可能是其他的选择，只要能保证选择的集合能唯一确定即可。

Primary Keys

One candidate key is usually chosen to identify tuples in a relation, which is primary key

NULLs

- Missing information can be represented using NULLs
- primary key can't have NULL

Foreign Keys

Foreign Keys are used to link data in two relations.

examples

Foreign Key Example

Department

DID	DName
13	Marketing
14	Accounts
15	Personnel

{*DID*} is a Candidate Key for **Department** – Each entry has a unique value for DID

Employee

EID	EName	DID
15	John Smith	13
16	Mary Brown	14
17	Mark Jones	13
18	Jane Smith	NULL

{*DID*} is a Foreign Key in **Employee** – each employee's *DID* value is either NULL, or matches an entry in the **Department** relation. This links each **Employee** to at most one **Department**

Referential Integrity (foreign key constrain)

RESTRICT

RESTRICT stops any action that violates integrity

CASCADE

CASCADE allows the changes made to flow through

SET NULL

SET NULL allows the changes to happen but every attribute effected will become null (only foreign key)

级别	关系维护
CASCADE	父表delete、update的时候，子表会delete、update掉关联记录；
NO ACTION	如果子表中有匹配的记录,则不允许对父表中的那个键进行update/delete操作
RESTRICT	如果想要删除父表的记录时，而在子表中有关联该父表的记录，则不允许delete记录；
SET NULL	在父表上update/delete记录时，将子表上匹配记录的列设为null要注意子表的外键列不能为not null
SET DEFAULT (Navicat中的空白项)	父表有变更时,子表将外键列设置成一个默认的值 但InnoDB不能识别