CS240
Database Management Systems

What is a database ?

- A collection of inter-related data that models some aspect of the real world

What is an example of a database ? Is there a database that you use every day ?

Examples

- Data regarding academic aspects of IIITG students
  - Courses
  - Instructors
  - Registration information of students
  - Grades of students
- Web-based services (Amazon)
  - Online retailers: order tracking, customized recommendations
  - Online advertisements

Examples (contd.)

- Aadhaar
  - Biometric information
  - Phone numbers, address
- Banking and finance
  - customer information, accounts, loans, and banking transactions.
  - Credit card transactions
  - Finance: sales and purchases of financial instruments (e.g., stocks and bonds); storing real-time market data

Examples (contd.)

- Airlines:
  - reservations, schedules
- Telecommunication:
  - records of calls, texts and data usage
  - generating monthly bills
  - maintaining balances on prepaid calling cards

Databases are everywhere.

What are the common features of these databases ?

Common features

Contains data that is

- Highly valuable
- Relatively large
- Accessed by multiple users and applications, often at the same time.

But is it sufficient to collect all this data ?

But is a database sufficient ?

- How to read, write, modify data in a database ?
  - Buy from Amazon
  - Book a railway ticket
  - Withdraw money from a bank
  - Make a phone call !

How to do that efficiently and conveniently ?

Database Management System

- We need a collection of programs

- A Database Management System (DBMS) is a collection of interrelated data and a set of programs to access that data.

Can’t we just store all data in a set of files ?

Data redundancy and inconsistency

Major: CS		
Student roll no	**Address**	**Phone No**
2018-91	West Fort Road, Nagpur	9012345678

Minor: Mechanical		
Student roll no	**Address**	**Phone No**
2018-91	West Fort Road, Nagpur	9012345678

- Change address of student 2018-91

-> Higher storage, access cost

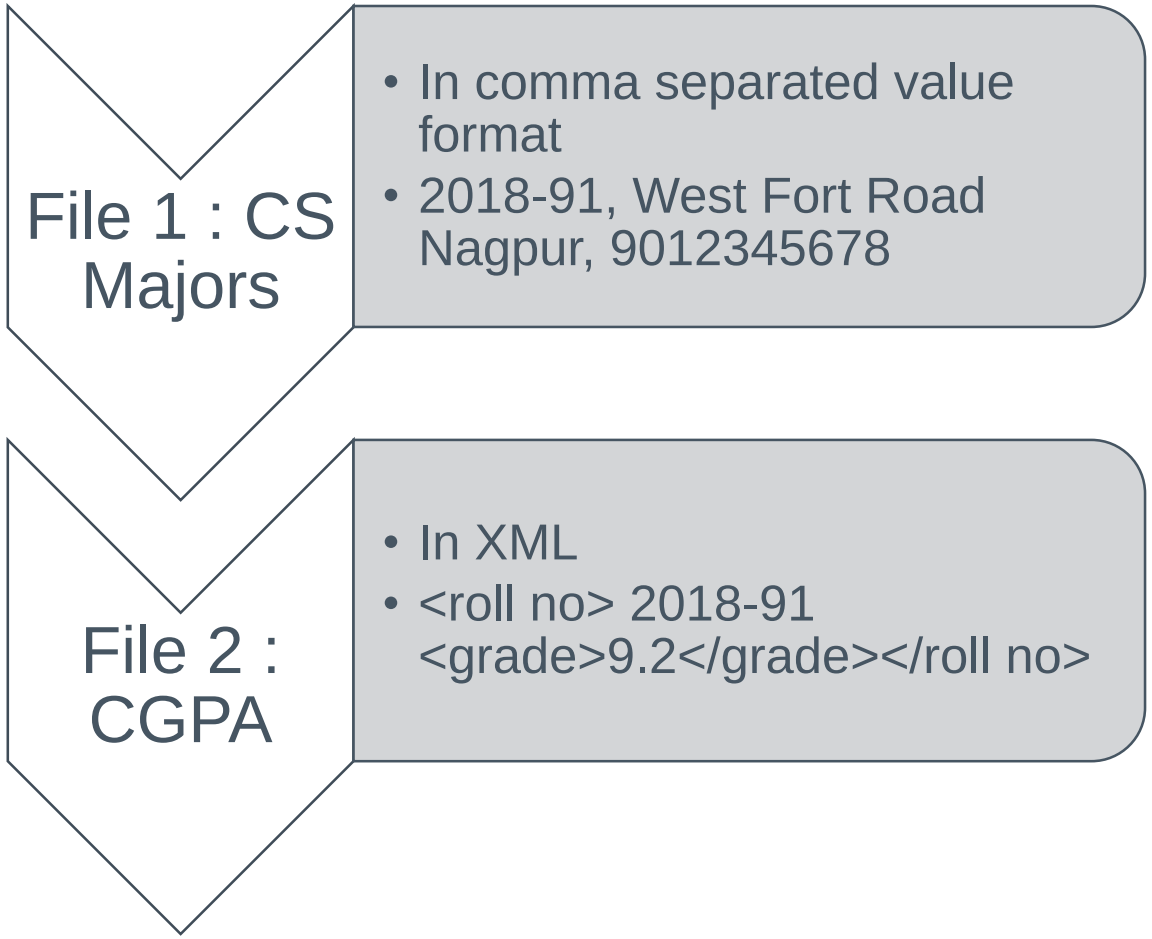
-> Data inconsistency

Difficulty in accessing data

- Find the phone numbers of all students who have CGPA > 8
- What if there is no application program that calculates this ?
- Suppose we additionally require that these students have A grades in their DBMS course ?

-> We need more responsive systems for general use

Data isolation



› Files in different formats

-> Writing new application programs is difficult

Integrity problems

› Every bank account must maintain a minimum balance - consistency constraint

-> Developers must add new code to maintain this

-> The constraint is buried in code, hard to add new constraints

› No customer must have more than one account and every account must have an Aadhaar number associated with it

Atomicity problem

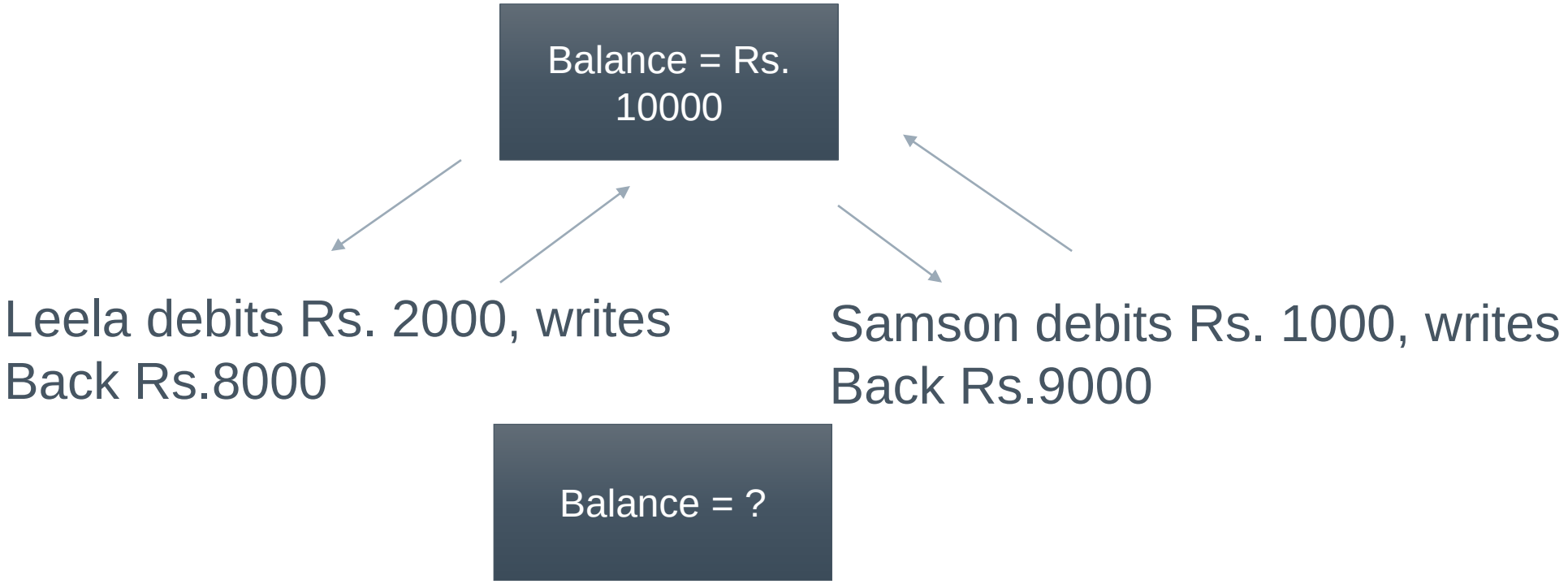
- Leela withdraws Rs.1000 from one bank account and deposits it into her mother's account
- The program fails during execution

-> Money is withdrawn, but not deposited !

Funds transfer must be atomic – either it must happen in its entirety or not happen at all

Difficult to maintain atomicity in a conventional file system

Concurrent access anomalies



Difficult to provide supervision of programs to prevent this in a conventional file system

Security problems

- If Aadhaar number is shared with a mobile company, the mobile company must not be able to access personal information
- A user must not be able to view the accounts of any one else in a bank

-> Difficult to enforce security constraints in a conventional file system

That is why we need a Database Management System

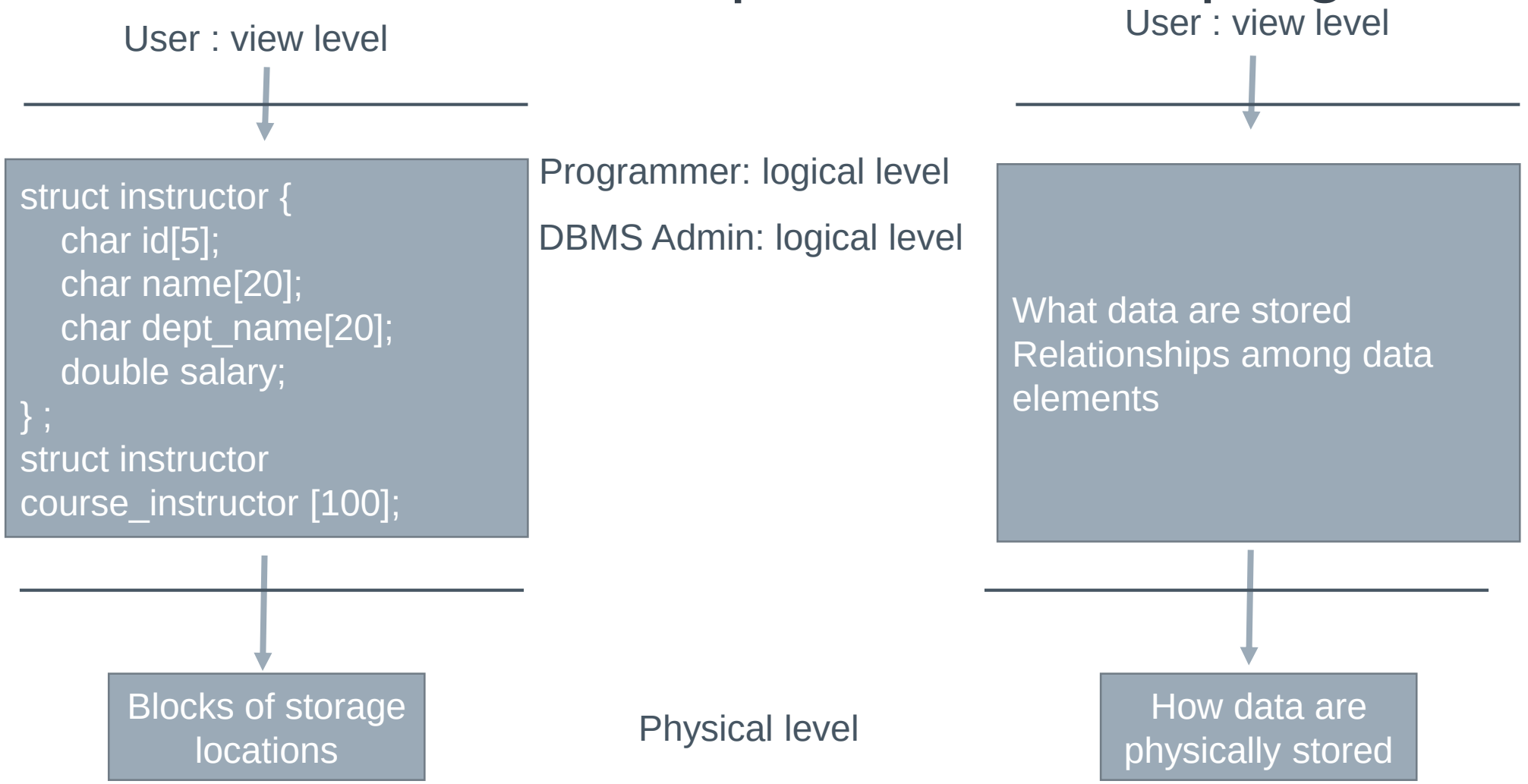
What are the common features of DBMSs?

Data Abstraction

- A DBMS provides users with an abstract view of its data – it hides details of how data is stored and maintained.

- Other examples of abstractions ?

Data Abstraction : comparison with a program



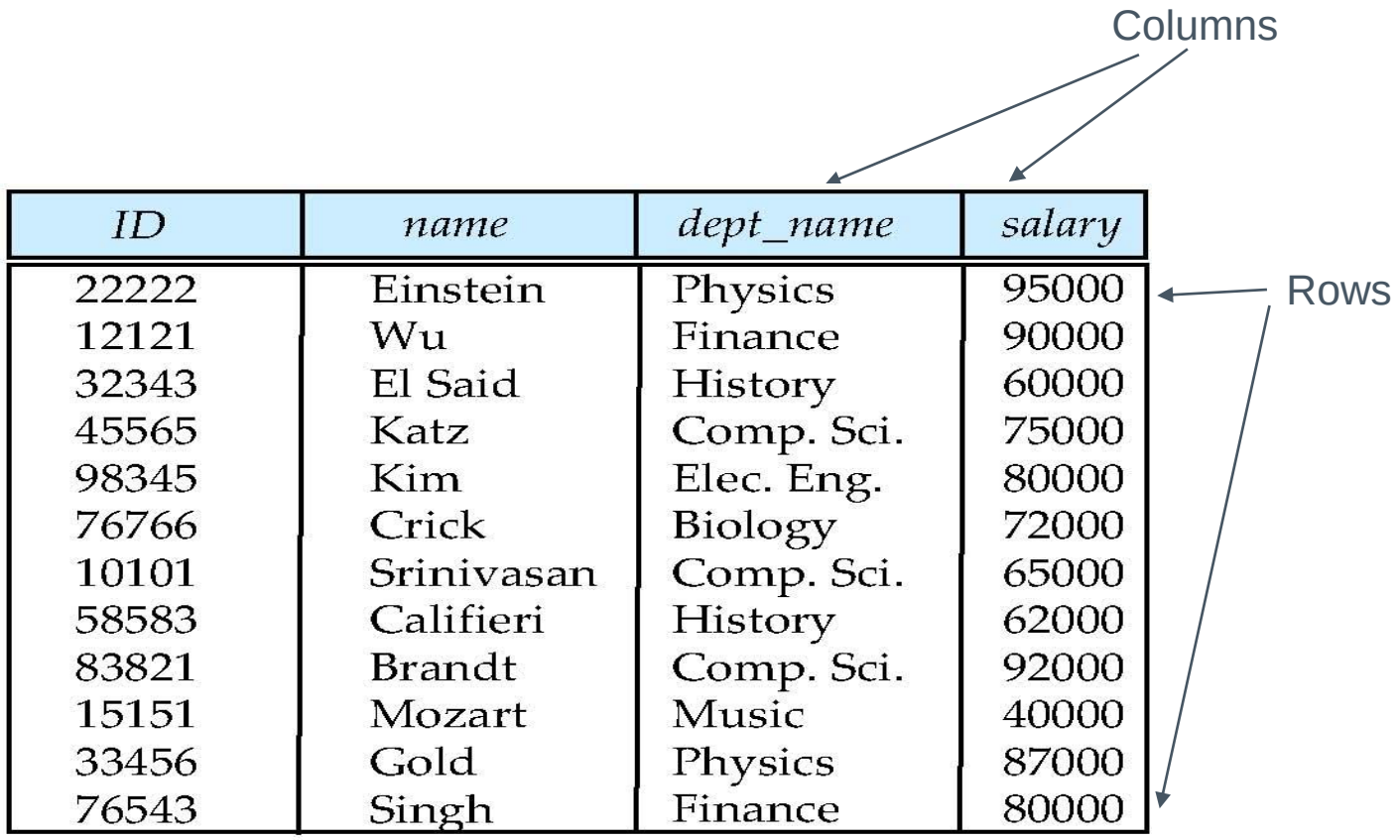
Each level hides details from the level above and provides an interface

Physical data independence

- Users at the logical level need not know the implementation of physical level structures
- Provides protection from changes in physical structures

Relational Model

› All data are stored in tables (also called relations)



ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table

Relational Model

- Database has several records of several types
- Each table (also called relation) has records (2222, Einstein, Physics, 95000) of a particular type (instructor)
- Each record type has a fixed number of fields, called attributes (ID, name, department name, salary)

The most widely used model !

How can we create a database ?

We need a language!

Instances and schemas

PROGRAMS

- Type definitions and variable declarations
- Values of variables at an instance

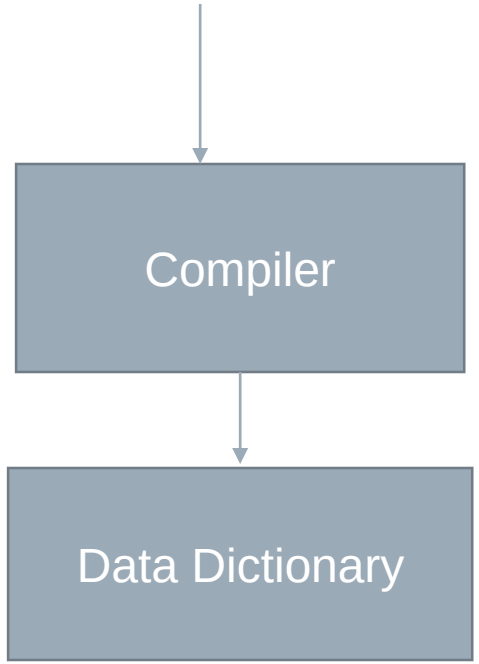
DBMS

- Database schema : overall design of the database
- Database instance: collection of information stored in a database at a particular moment

Database Languages

- Data definition language
  DDL
- To create schemas
- To specify integrity constraints, which are checked every time a DB is updated

create table *instructor* (
ID **char**(5),
name **varchar**(20)**,**
dept_name **varchar**(20),
salary **numeric**(8,2))



Is a DDL sufficient ?

How to retrieve, insert, delete or modify data ?

Query

- A statement requesting retrieval of information
- Eg: Get the names of all students who are enrolled in the Winter semester of 2019

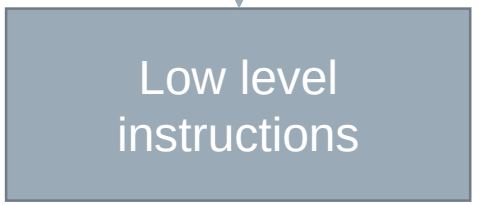
- Query language: The portion of DML that involves information retrieval

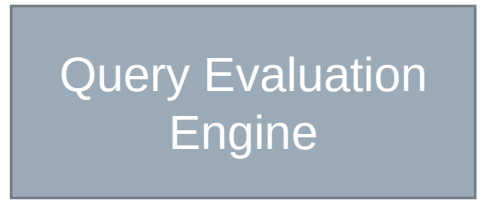
SQL

- A non-procedural language
- Consists of both DDL and DML

DML Query

```
select name
        from instructor
        where dept_name = ‘Comp. Sci.'
)
```





How can an end user access a database ?