

Database Systems

(CS 355 / CE 373)

Dr. Umer Tariq
Assistant Professor,
Dhanani School of Science & Engineering,
Habib University

Acknowledgements

- Many slides have been borrowed from the official lecture slides accompanying the textbook:

Database System Concepts, (2019), Seventh Edition,
Avi Silberschatz, Henry F. Korth, S. Sudarshan
McGraw-Hill, ISBN 9780078022159

The original lecture slides are available at:

<https://www.db-book.com/>

- Some of the slides have been borrowed from the lectures by Dr. Immanuel Trummer (Cornell University). Available at: (www.itrummer.org)

Outline: Week 1

- Introduction to Instructor
- Introduction to Class
- Syllabus
- Introduction to Course

>> ~~python~~ python code.py

>>

Let's Time Travel to a Hypothetical Past?

- It's 1976. A unique new institution has been setup in Karachi: "HABIB UNIVERSITY"
- It wants to have cutting-edge technology. Therefore, it has imported the latest IBM Computer (IBM Series/1) to be used by the Registrar Office.
- It supports most of the tools you have seen in Freshman programming courses
 - Operating System with a command prompt
 - Text editor (VS Code)
 - Python
- **Your Task**: Use this computer to manage student admissions, course enrollments, and grades.



code.py

• Human error for entry

Let's Time Travel to a Hypothetical Past?

>>

>> python admissions.py Annia 22 LHE CS

>> " enrollments.py " " " "

>> " grades.py

admissions.csv

Annia 22 LHE CS EE
Saved 11 KH1 CE

enrollments.csv

Annia 22 ^{CS} DB
Saved 11 CF DB

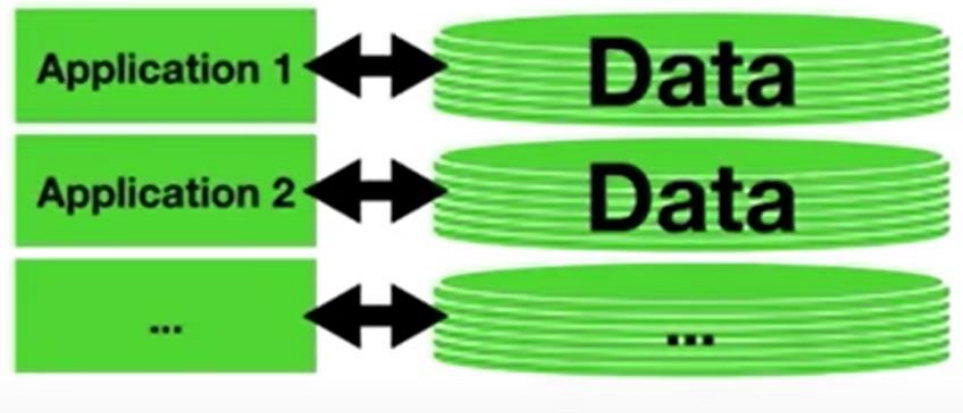
grades.csv

Annia 22 CS DB A
Saved 11 CE DB B



File-Based Approach

admission



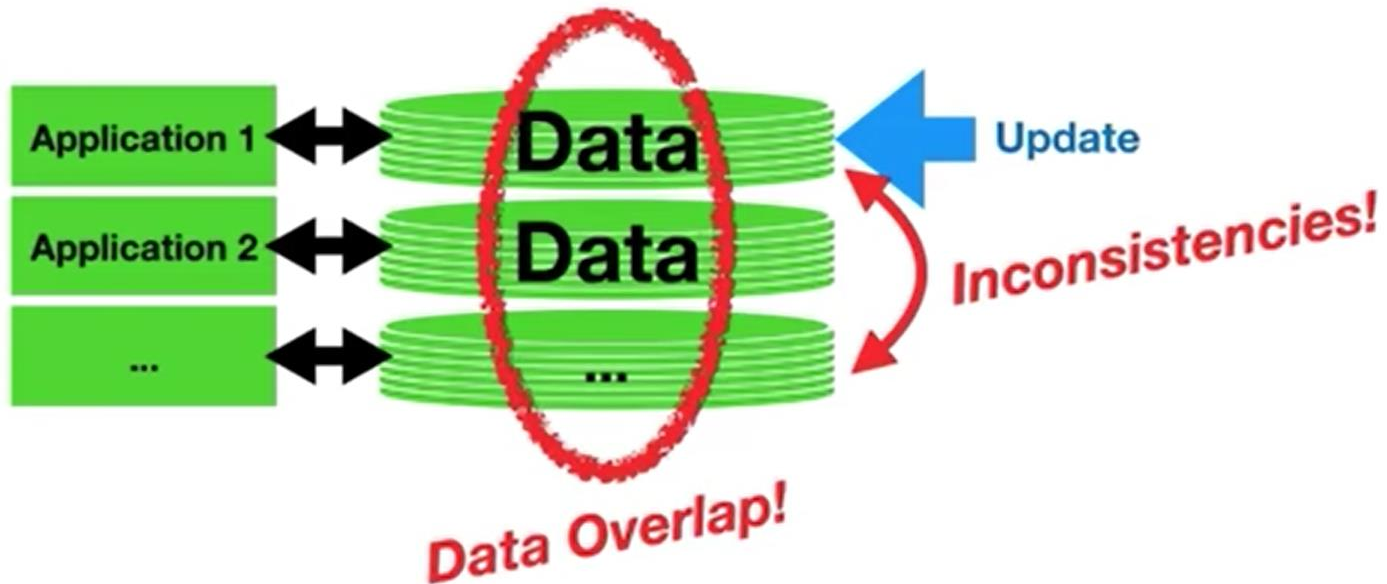
File-Based Approach: Limitations



File-Based Approach: Limitations



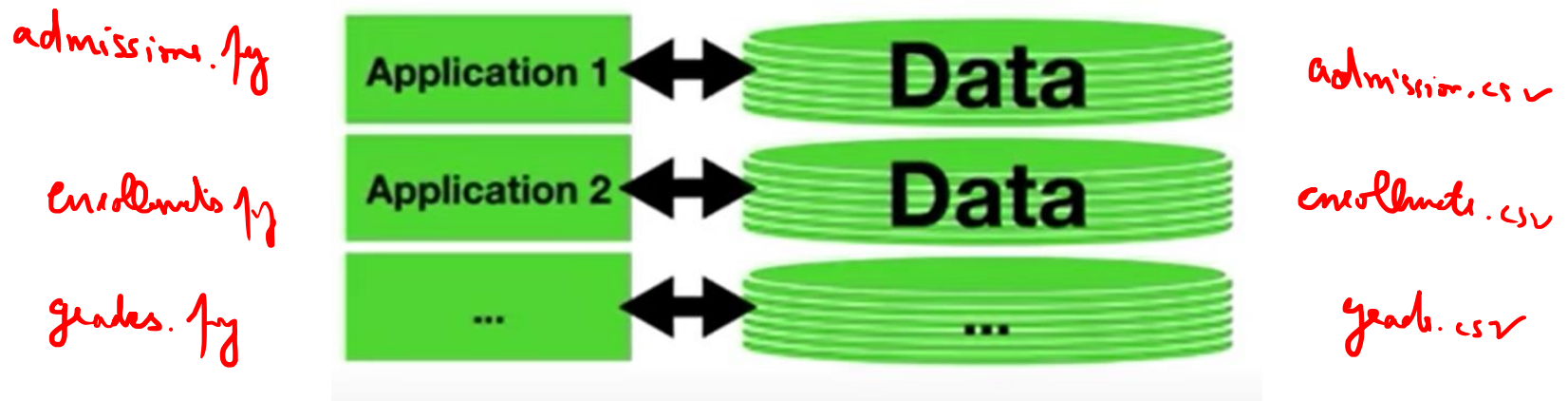
File-Based Approach: Limitations



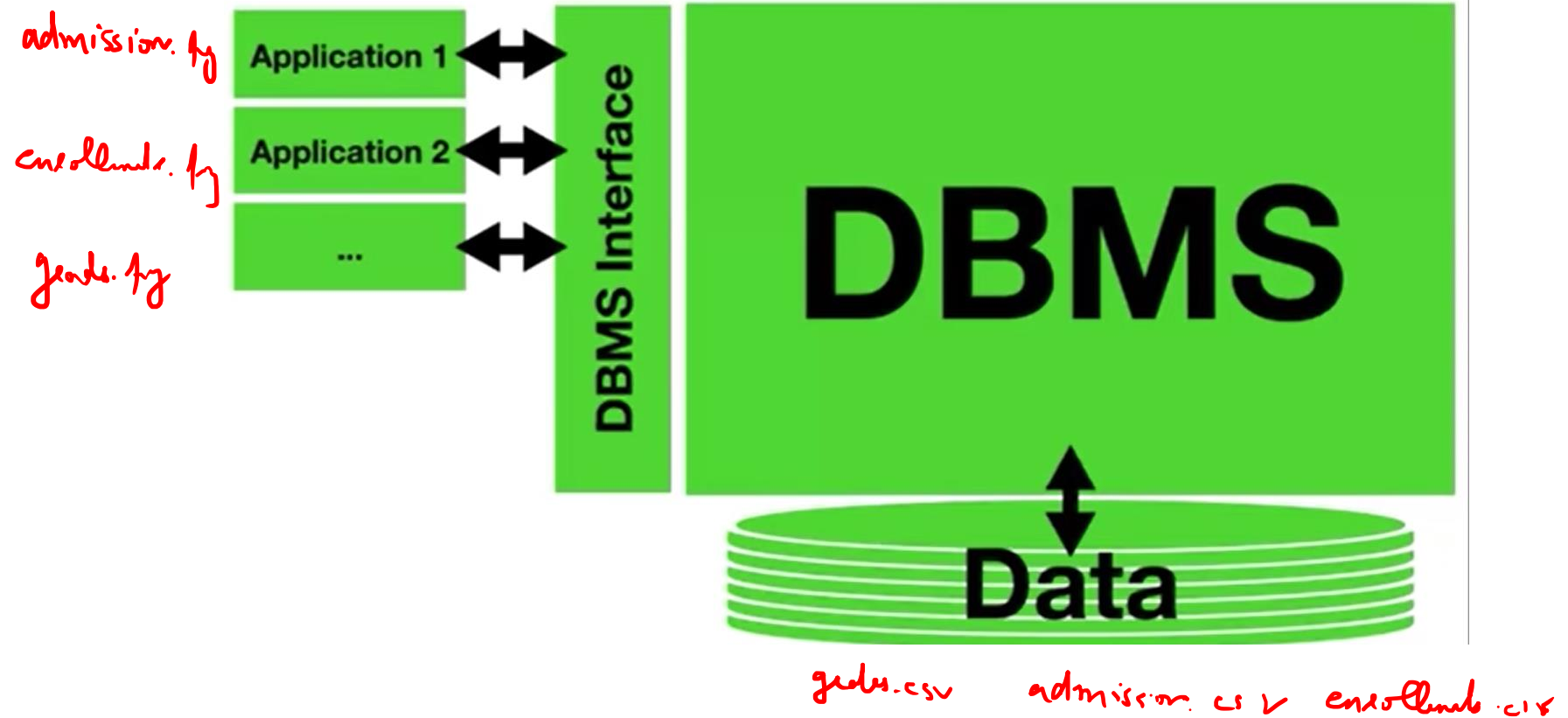
Limitations of File-Based Approach

- Data Redundancy
- Date Inconsistency
- Data Integrity Problems
 - Integrity constraints (e.g., $\text{account balance} > 0$) become “buried” in program code rather than being stated explicitly
 - Hard to add new constraints or change existing ones
- Atomicity of Updates
- Concurrent Access by Multiple Users
- Security Problems

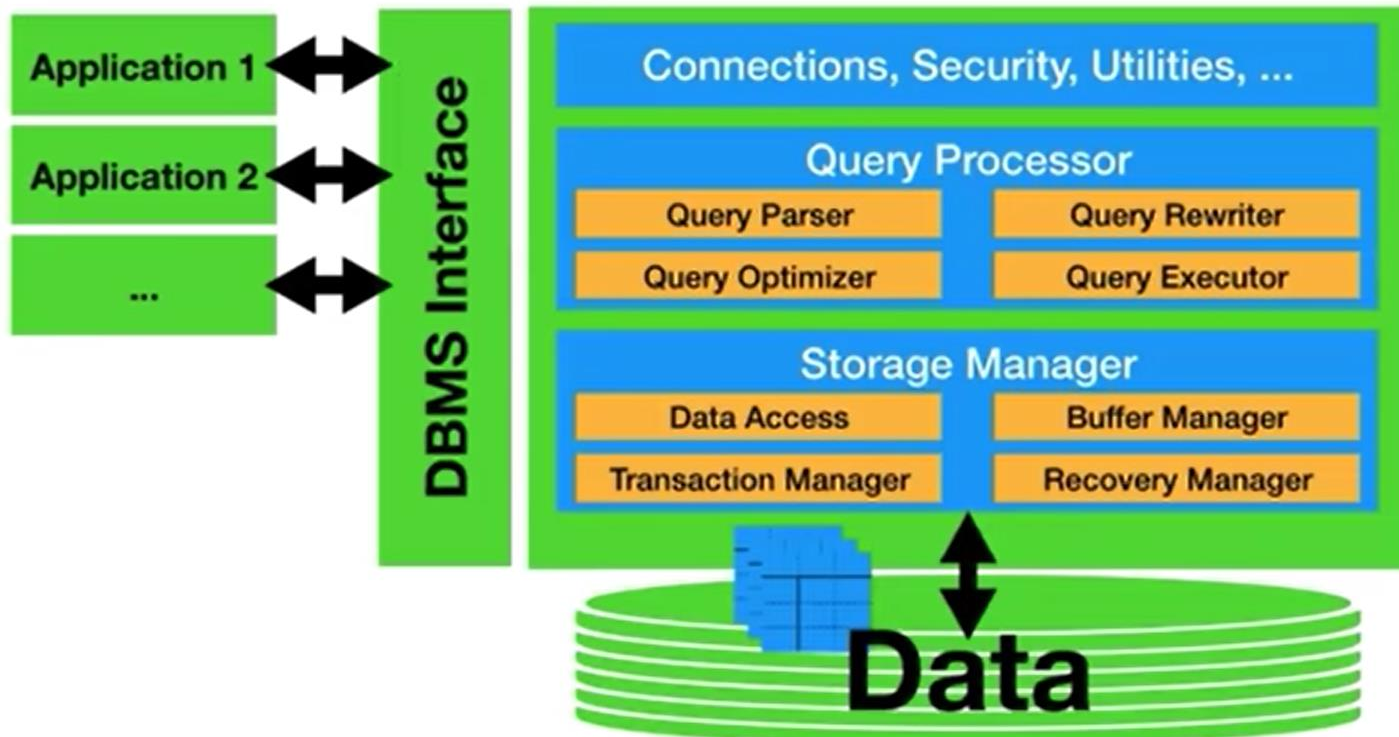
File-Based Approach



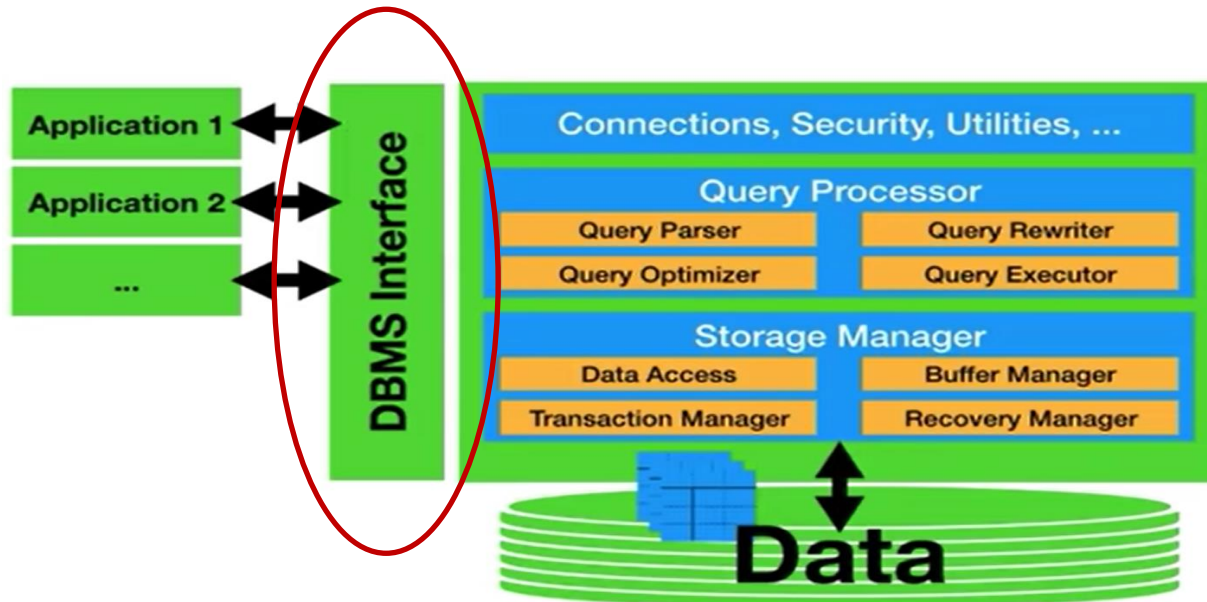
Database Management System (DBMS)



Database Management System (DBMS)



What should be the DBMS Interface?



What should be the DBMS Interface?

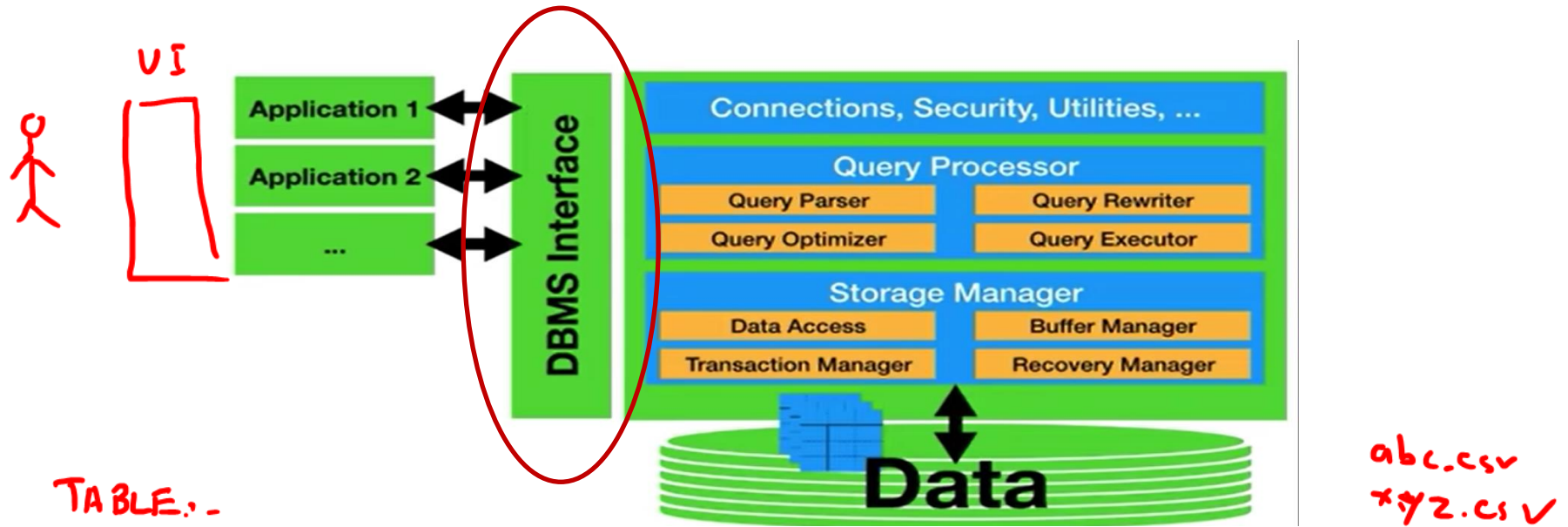


TABLE:-

DEPARTMENTS

name	building	budget
CS	C-100	1
CE	E-100	-
EE	N-100	-

FACULTY

ID	Name	Dept	Rate
121	Uma	CE	KH

STUDENT

ID	Name	Major
1	Amin	CS
2	Saad	CE

'GIVE ME LIST OF ALL CS STUDENTS?'

Database Management Systems (DBMS)

- Database Management System (DBMS) is a collection of
 - Interrelated data
 - A set of programs that allow users to access and modify those data
- The primary goal of a DBMS: provide a way to store and retrieve information that is both
 - Convenient
 - Efficient

Purpose of Database Systems

- Database systems are used to manage collections of data that are:
 - Highly valuable
 - Relatively large
 - Accessed by multiple users and applications, often at the same time.

Examples of Software Applications that Use Database Systems

- Enterprise Information
 - Sales: customers, products, purchases
 - Accounting: payments, receipts, assets
 - Human Resources: Information about employees, salaries, payroll taxes.
- Manufacturing: management of production, inventory, orders, supply chain.
- 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
- Universities: registration, grades

Examples of Software Applications that Use Database Systems

- Airlines: reservations, schedules
- Telecommunication: records of calls, texts, and data usage, generating monthly bills, maintaining balances on prepaid calling cards
- Web-based services
 - Online retailers: order tracking, customized recommendations
 - Online advertisements
- Navigation systems: For maintaining the locations of various places of interest along with the exact routes of roads, train systems, buses, etc.

Requirements / Features

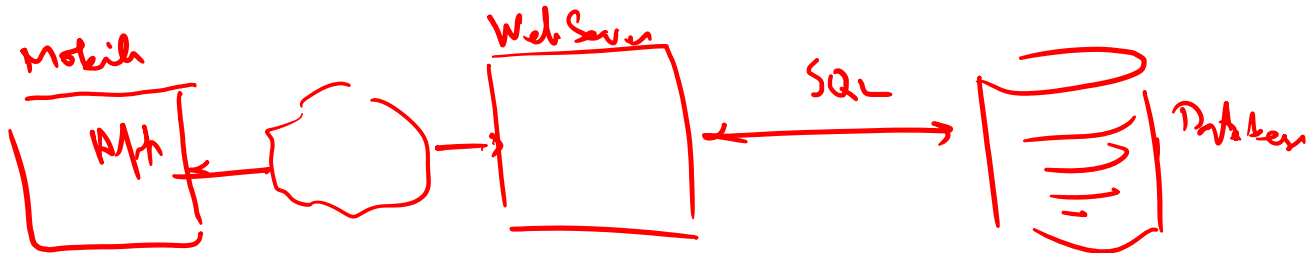
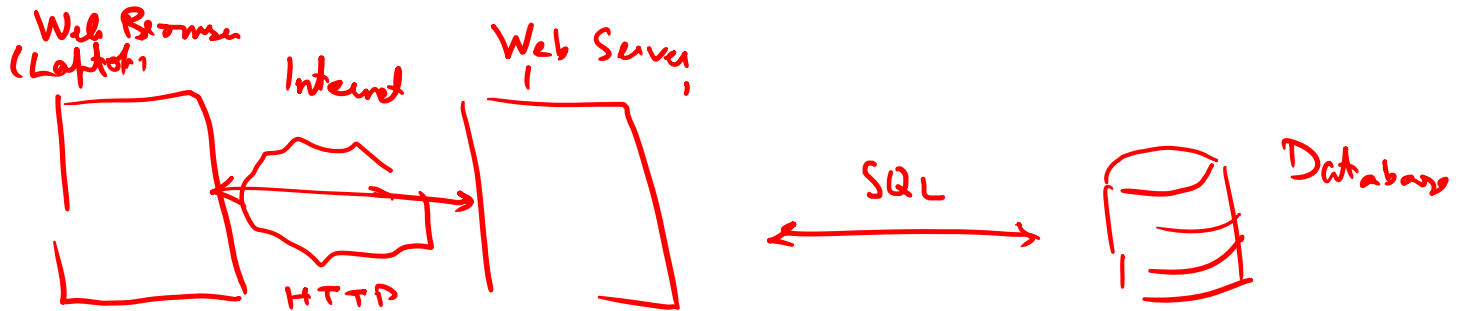
Application Architecture

DESIGN

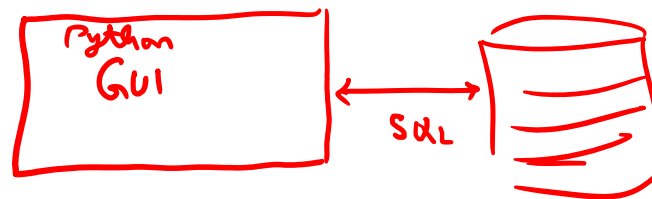
Code

Test

Deploy



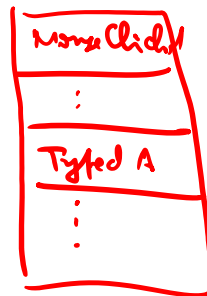
DESKTOP / LAPTOP



Application Architecture



Event List



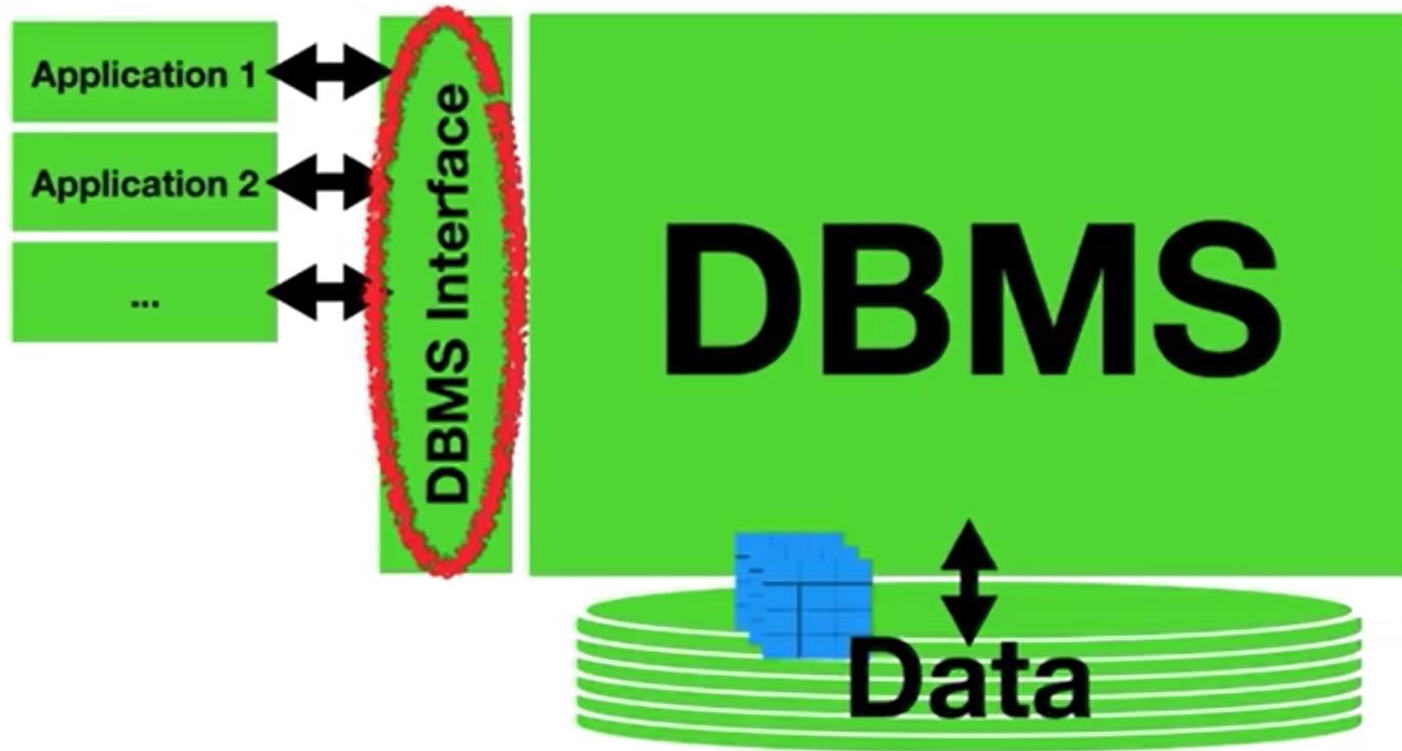
On Button Click 1

```
{
  ==
  ==
  ==
}
```

while(TRUE)

```
{
  1. Pick the NEXT Event
  2. Call the EventHandler
}
```

Significance Of DBMS Interface



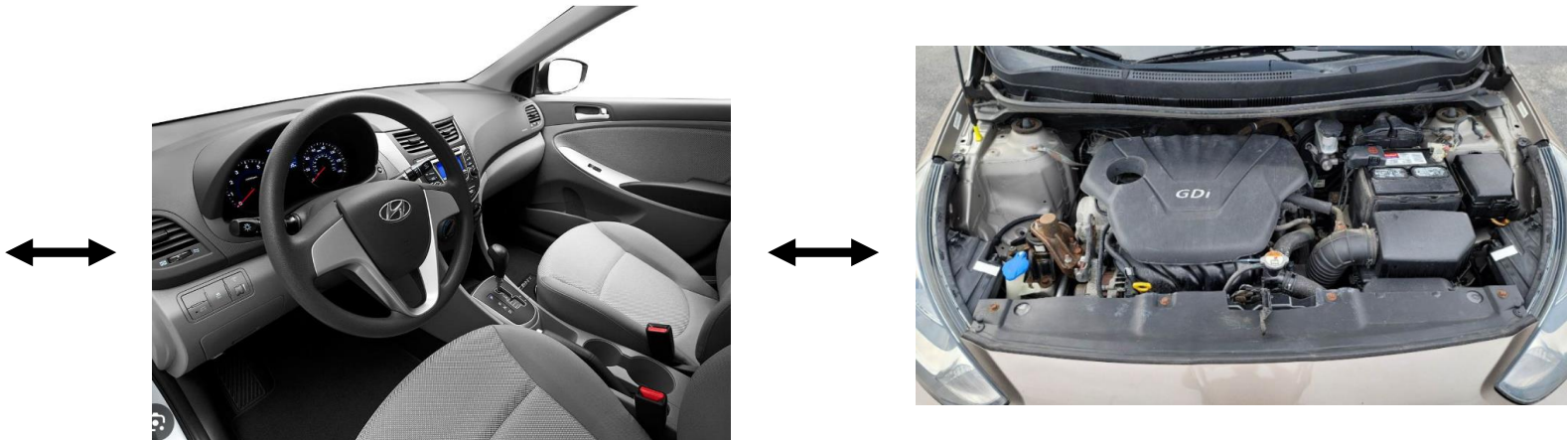
Why Focus on DBMS Interface?

- Managing complexity
 - Abstraction is your friend!



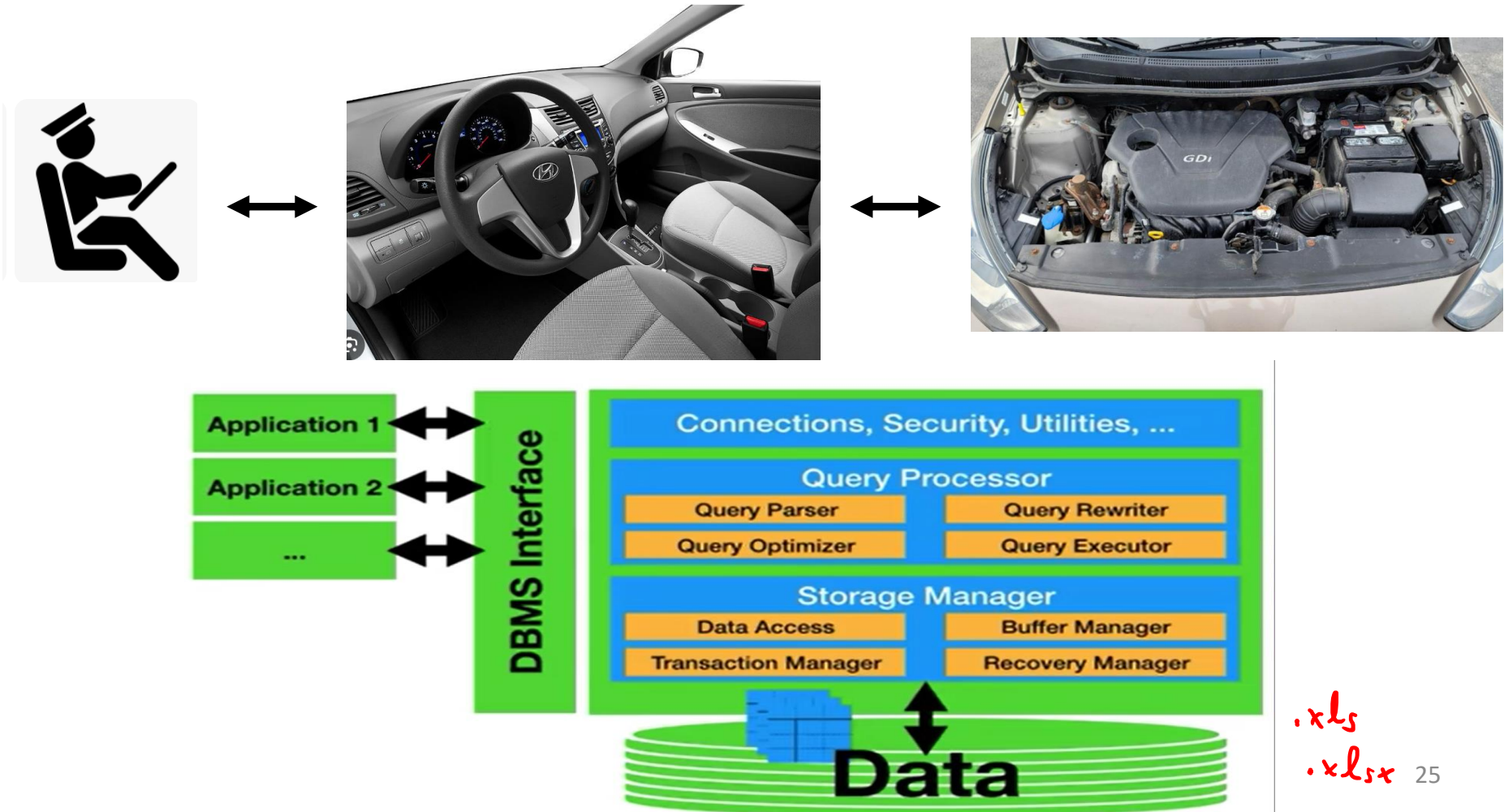
Why Focus on DBMS Interface?

- Managing complexity
 - Abstraction is your friend!

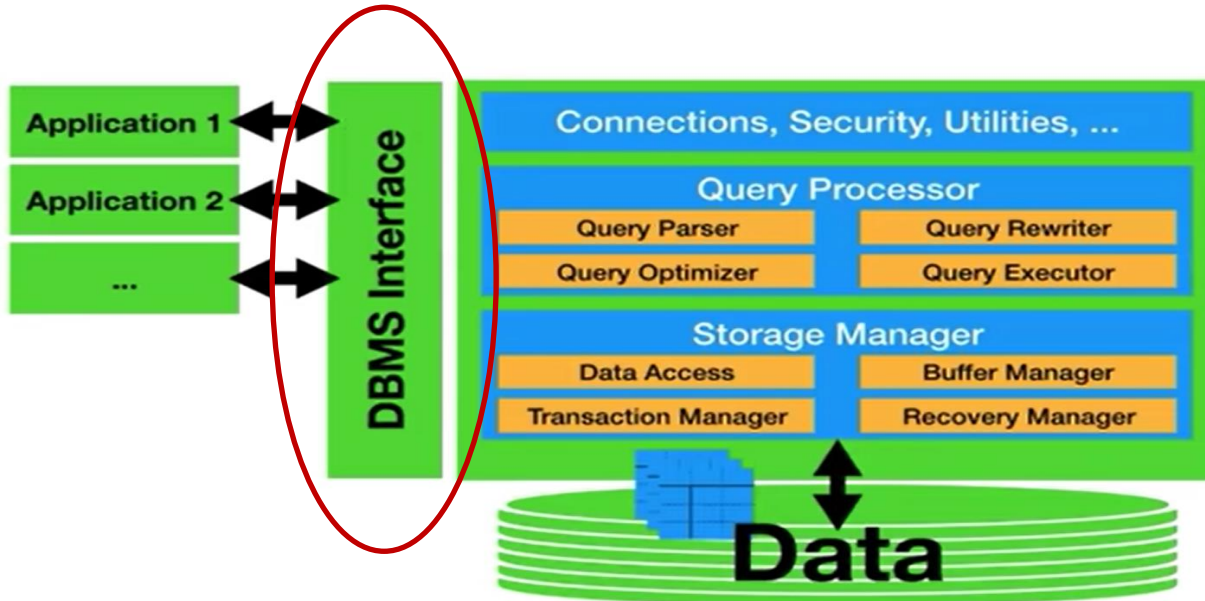


Why Focus on DBMS Interface?

- Managing complexity
 - Abstraction is your friend!



What should be the DBMS Interface?

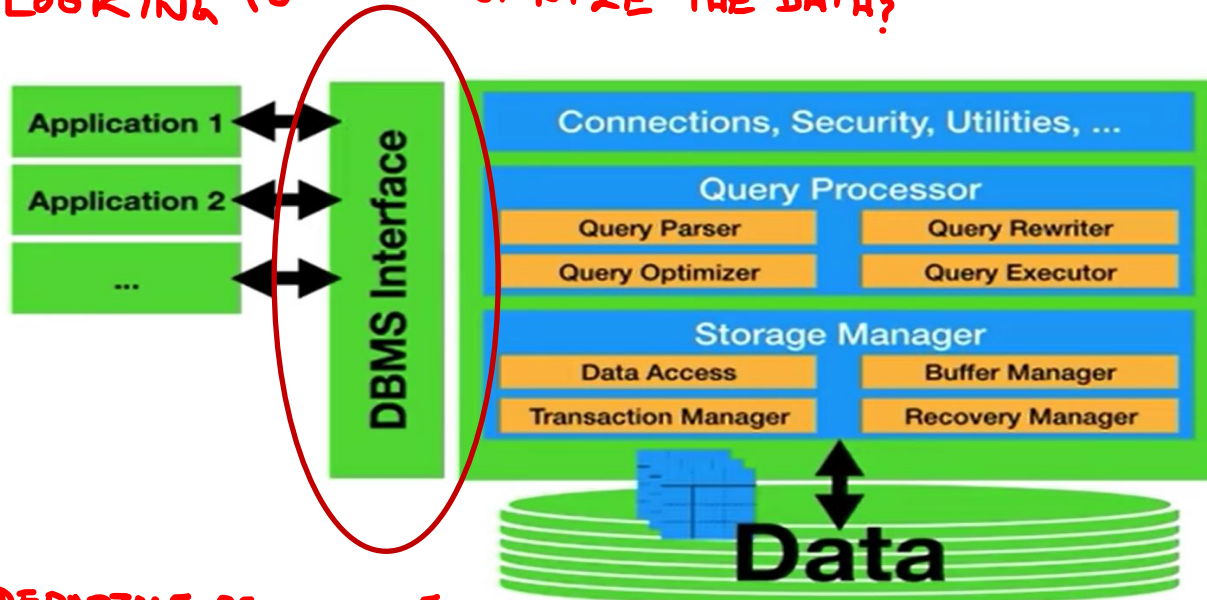


What Should be the DBMS Interface?



What should be the DBMS Interface?

"WE ARE LOOKING TO ORGANIZE THE DATA?"



TABLES:-

DEPARTMENTS

ID	Name	Building
1	CS	HV
2	CE	C-100
3	EE	C-101

FACULTY

ID	Name	Address
107	Umar	KHI

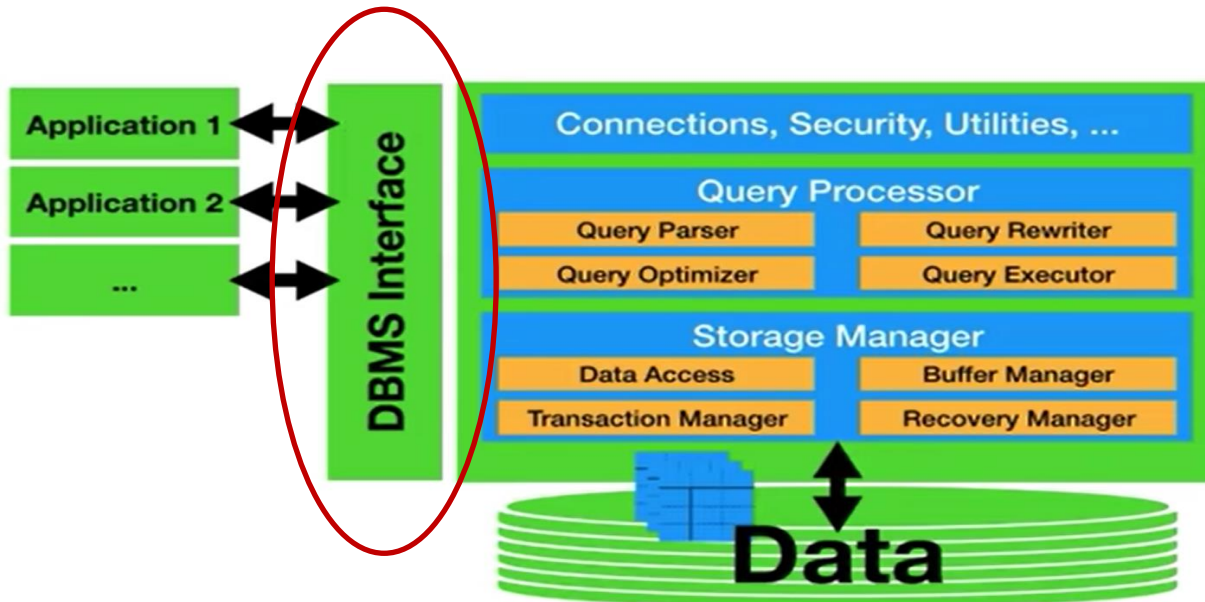
STUDENT

ID	NAME	ADDRESS	Major
1	AMIR	KHI	CS

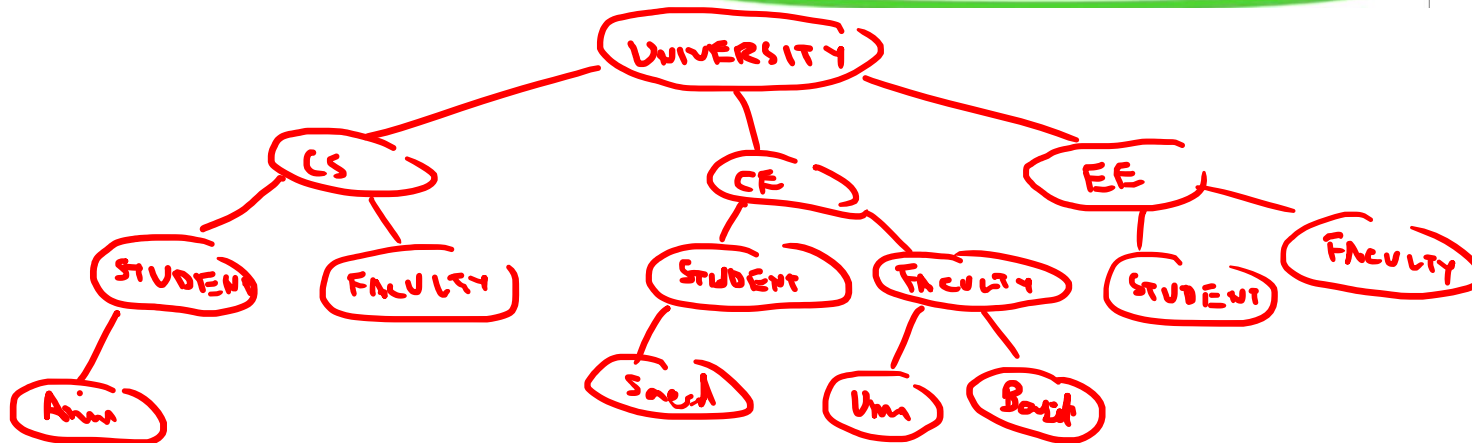
QUERY:-

Give me a list of all CS students?

What should be the DBMS Interface?



GRAPHS:..



Relational Model

- Uses a collection of tables to represent both data and the relationship among those data
- Each table has multiple column, and each a unique name
- Tables are also known as **relations**

Columns

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
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

Rows



Ted Codd
Turing Award 1981

(a) The *instructor* table

A Sample Relational Model

<i>ID</i>	<i>name</i>	<i>dept_name</i>	<i>salary</i>
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

<i>dept_name</i>	<i>building</i>	<i>budget</i>
Comp. Sci.	Taylor	100000
Biology	Watson	90000
Elec. Eng.	Taylor	85000
Music	Packard	80000
Finance	Painter	120000
History	Painter	50000
Physics	Watson	70000

(b) The *department* table

SSS Vme ECE —

Schema vs Instance

- (Logical) Schema

- The overall logical structure of the database

*instructor (ID, name, dept name, salary)
department (dept-name, building, budget)*

- Instance

- The actual content of the database at a particular point in time

“Database Languages” Supported by DBMS Interface

- The DBMS interface supports the following types of languages:
- Data Definition Language (DDL)
 - Used to define the Database Schema
- Data Manipulation Language (DML)
 - Used to Retrieve / manipulate data

“Database Languages” Supported by DBMS Interface

- The DBMS interface supports the following types of languages:
- Data Definition Language (DDL)
 - Used to define the Database Schema
- Data Manipulation Language (DML)
 - Used to Retrieve / manipulate data
- In practice, the DDL and DML are not two separate languages: instead they simply form parts of a single database language.
- For relational databases, the most popular database language is **Structured Query Language (SQL)**

Structured Query Language (SQL)

- The standard to access/retrieve/manipulate data in a **relational database**
- Examples of a Data Definition Language (DDL) Component

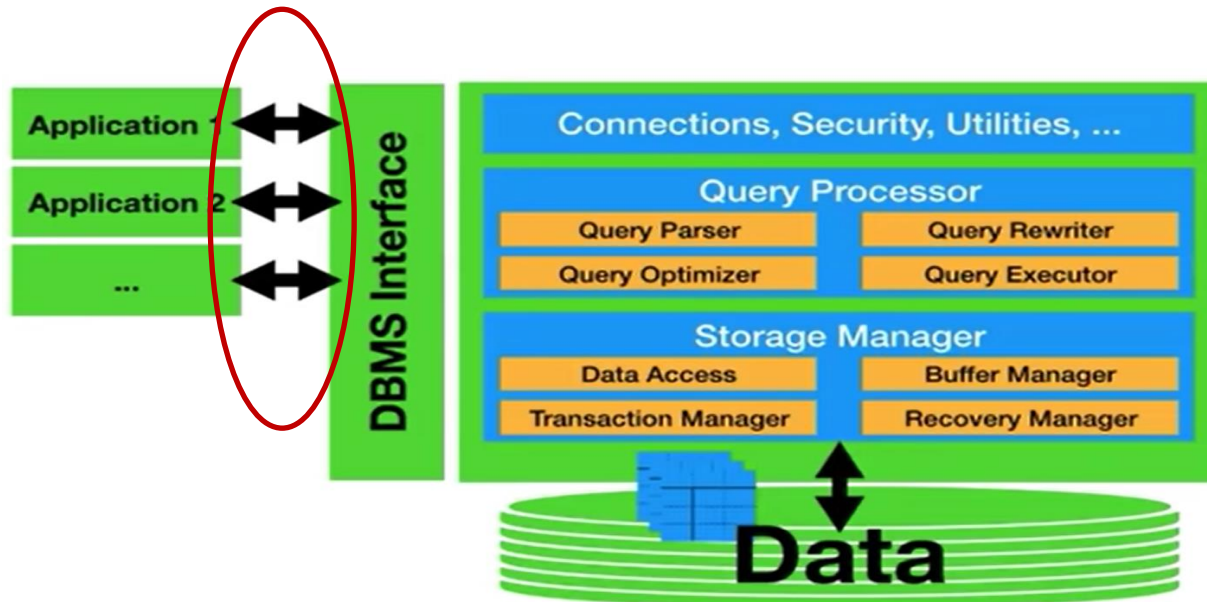
```
create table department  
  (dept_name  char (20),  
   building   char (15),  
   budget     numeric (12,2));
```

- Examples of a Data Manipulation Language (DML) Component

```
select instructor.name  
from instructor  
where instructor.dept_name = 'History';
```

```
select instructor.ID, department.dept_name  
from instructor, department  
where instructor.dept_name= department.dept_name and  
       department.budget > 95000;
```

Database Access from Application Program



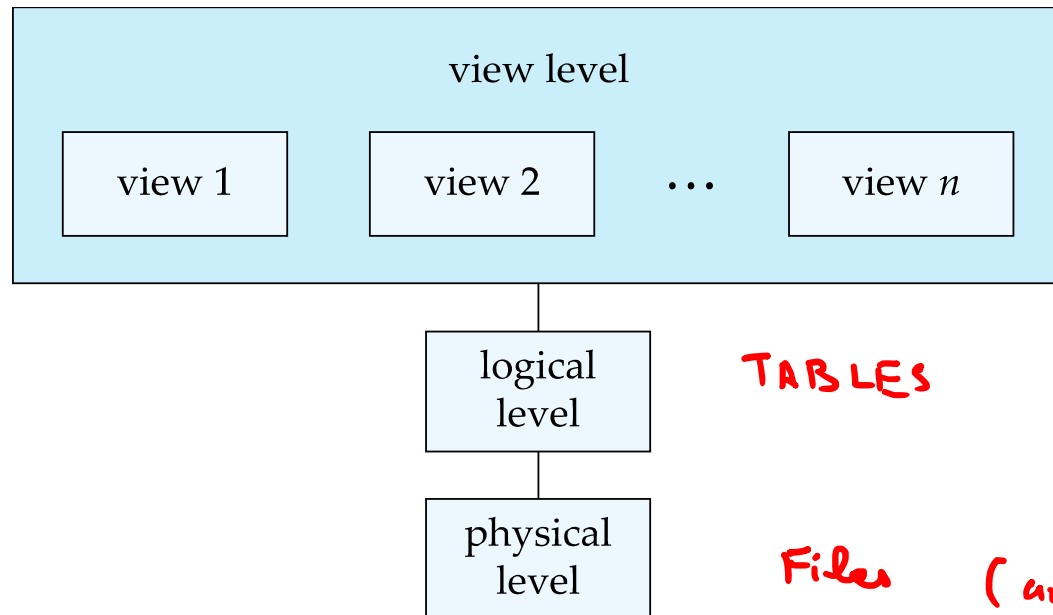
- Application programs generally access databases through one of the following:
 - Language extensions to allow embedded SQL
 - Application program interface (e.g., ODBC/JDBC) which allow SQL queries to be sent to a database

Database Design

- Database-based applications are developed to meet the needs of an enterprise/business
- The design of a complete database application (so that it meets the needs of the business) requires attention to a broad set of issues such as:
 - What data to store
 - What should be the user interface
 - What should be the architecture of the application
- “Database Design”
 - Limited to design of database schema (the set of tables that will be used to store the data).

Levels of Abstraction

- Physical level: describes how data is stored.
- Logical level: what data is stored in database, and the relationships among the data.
- View level: describes part of the database

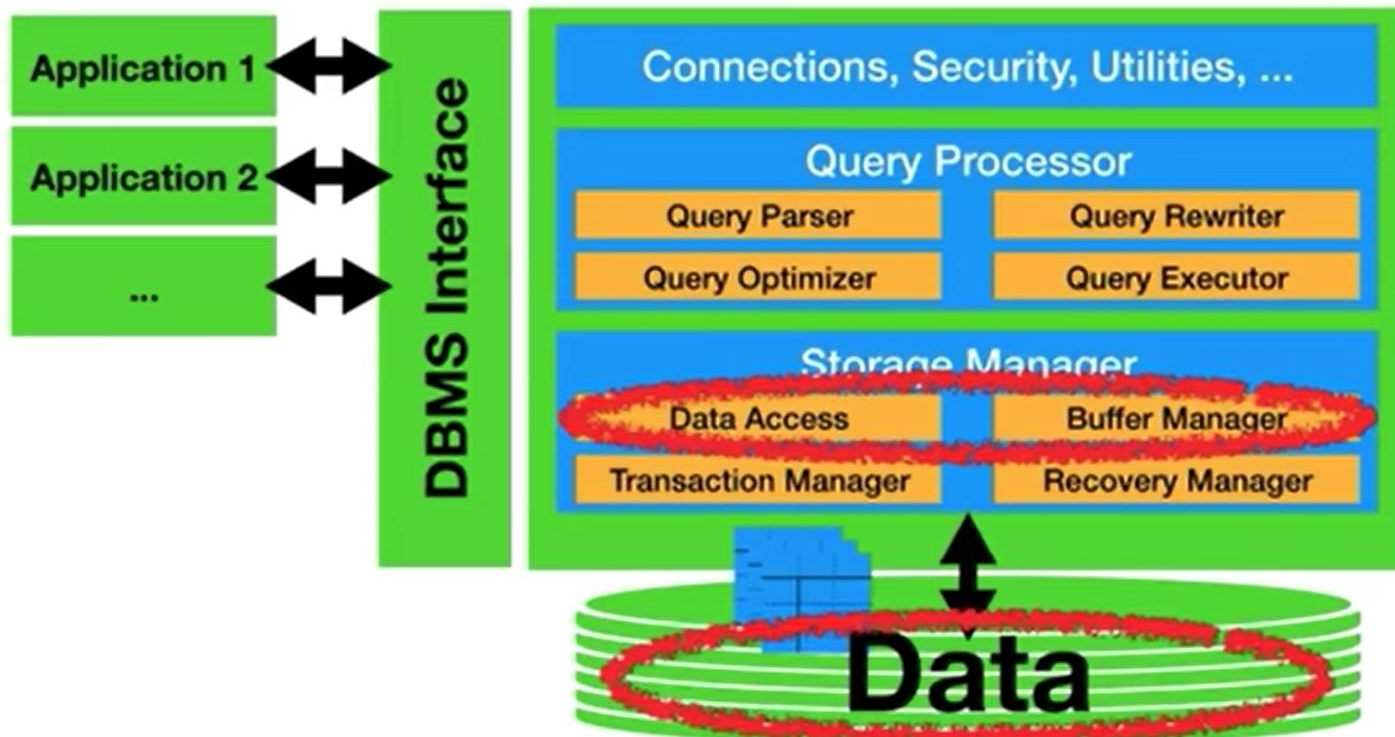


TABLES

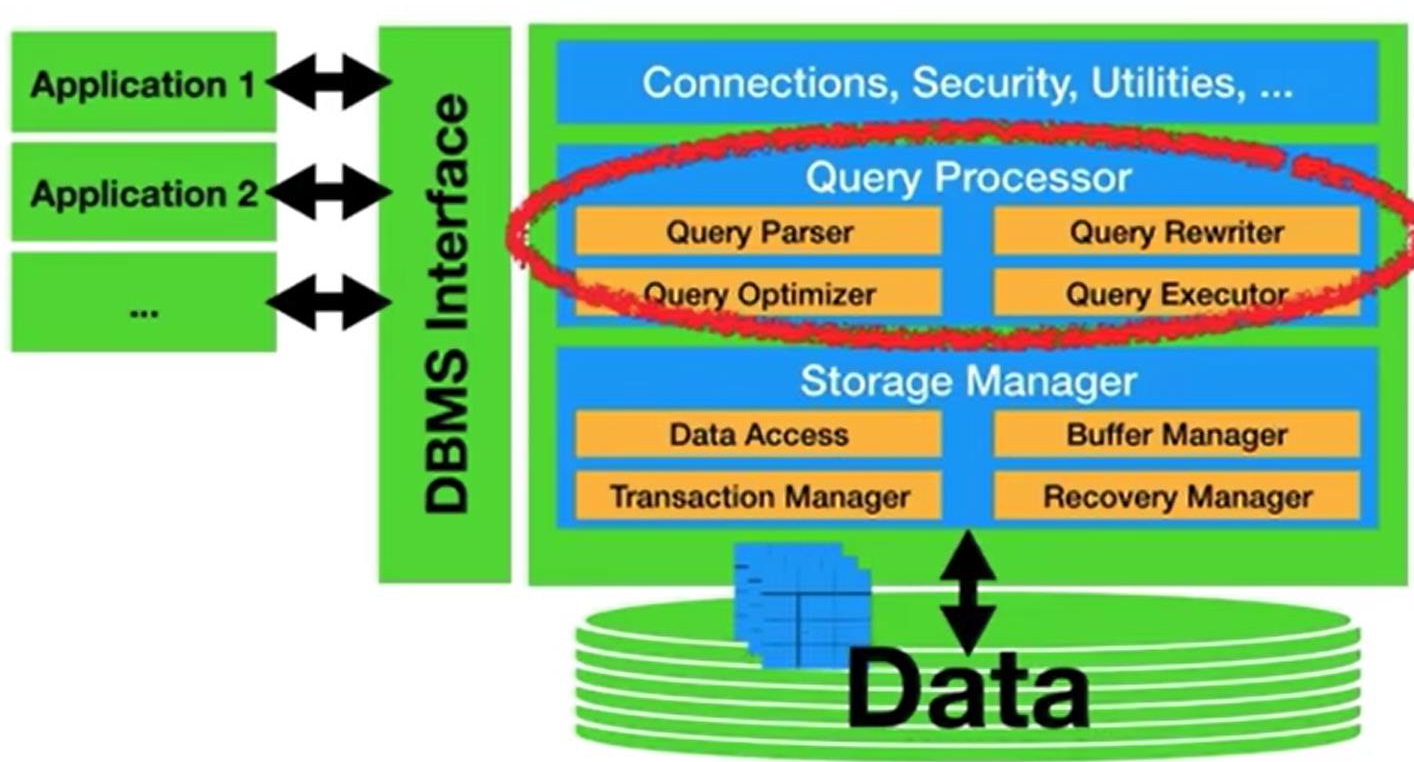
Files

(admissions.csv, grade.csv, master-data.csv)

Supporting the Abstraction Provided by DBMS Interface



Supporting the Abstraction Provided by DBMS Interface



Supporting the Abstraction Provided by DBMS Interface

